SESSION

NOVEL APPLICATIONS AND ALGORITHMS + PDE

Chair(s)

TBA

GPGPU and Multi-Core Architectures for Computing Clustering Coefficients of Irregular Graphs

A. Leist, K.A. Hawick and D.P. Playne
[1]Computer Science, Massey University
[2]Albany, North Shore 102-904, Auckland, New Zealand

Abstract— *Network science makes heavy use of simulation models and calculations based upon graph-oriented data structures that are intrinsically highly irregular in nature. The key to efficient use of data-parallel and multi-core parallelism on graphical processing units (GPUs) and CPUs is often to optimise the data layout and to exploit distributed memory locality with processing elements. We describe work using hybrid multi-core and many-core devices and architectures for implementing and optimising applications based upon irregular graph and network algorithms.*

Keywords: multi-core; GPU; CUDA; Cell; data parallelism

1. Introduction

The recent introduction of the Graph 500 benchmarks [1] highlights the increasing importance of data intensive, graph-based algorithms in high performance computing. While traditional supercomputers have to yield more and more of the top spots in the TOP 500 [2] to hybrid systems featuring graphics processing units (GPUs) as compute accelerators, which provide the bulk of the processing power in those systems, irregular graph structures pose a challenge for general purpose computation on GPUs (GPGPU) [3], [4].

We implement the clustering coefficient as defined by Newman et. al. [5], a graph metric that is commonly used when analysing social networks, on a number of multi-core CPU and GPU architectures. We use this metric to compare the performance and scaling behaviour of a graph-based, bandwidth limited algorithm on these heterogeneous devices. Code fragments and pseudo-code are used to show how the implementations were optimised.

The clustering coefficient is a graph metric that is based on the concept of clustering in social networks, sometimes also called network transitivity, introduced by Watts and Strogatz [6]. It is often used when analysing networks with small-world characteristics [7], [6], [8]. Newman et. al. define the clustering coefficient C as follows:

$$C = \frac{3 \times (\text{number of triangles on the graph})}{\text{number of connected triples of vertices}}$$

Here, triangles are elementary circuits of length three, that is, three distinct vertices connected by three arcs creating a cycle. A connected triple is a path of length two that connects three distinct vertices.

In Section 2 we describe in detail how the clustering coefficient is computed for an arbitrary graph. We give algorithm fragments showing how we implement this on: single- and multi-core CPUs using POSIX threads and threading building blocks (TBB) multi-threading libraries (Section 2.1), NVIDIA's compute unified device architecture (CUDA) for both single-GPU (Section 2.2) and multi-GPU (Section 2.3) systems, as well as on the Cell Broadband Engine (CellBE) (Section 2.4). For details on these parallel hardware architectures, see the unpublished technical note [9].

We use two commonly found graph structures, small-world and scale-free, to compare the performance of the clustering coefficient algorithm on these platforms in Section 3. We discuss the outcomes and draw some conclusions in Section 4.

2. The Clustering Coefficient

Each one of the different hardware architectures described in this paper—x86 multi-core CPU, GPU and CellBE—uses a very different approach to parallelism and thus requires an algorithm that is specifically tailored for its architecture to achieve peak performance. This section describes the implementations of the clustering coefficient algorithm along with architecture specific performance optimisations.

2.1 CPU - Sequential, PThreads & TBB

For reference purposes and to better explain the algorithm we use, we give a serial CPU code implementation of the clustering coefficient calculation in Algorithm 1.

The outermost loop of the sequential implementation executes once for every vertex $v_i \in V$. The iterations do not interfere with each other and can thus be executed in parallel. It is merely necessary to sum up the numbers of triangles and paths found in each of the parallel iterations to get the total counts for the graph before the clustering coefficient can be calculated. Algorithms 2 and 3 describe an implementation that uses POSIX Threads (PThreads) to achieve parallelism.

The TBB implementation, like the PThreads version, applies the parallelism to the outermost loop. TBB's `parallel_reduce` can be used to do this parallel reduction without having to explicitly specify the chunk size and number of threads or having to worry about keeping all threads busy.

Algorithm 1 Pseudo-code for the sequential CPU implementation of the clustering coefficient.

```
function CLUSTERING(G)
  Input parameters: The graph G := (V, A) is an array of adjacency-lists,
  one for every vertex v ∈ V. The arc set A_i ⊆ A of a vertex v_i is stored in
  position V[i]. |V| is the number of vertices in G and |A_i| is the number
  of neighbours of v_i (i.e. its degree).
  R ← determine reverse adjacency-list lengths
  declare t //triangle counter
  declare p //paths counter
  for all v_i ∈ V do
    for all v_j ∈ A_i do
      if v_j = v_i then
        p ← p − R[v_i] //correct for self-arcs
        continue with next neighbour v_{j+1}
      p ← p + |A_j|
      if v_i > v_j then
        for all v_k ∈ A_j do
          if v_k = v_i then
            p ← p − 2 //correct for cycles of length 2 for v_i and v_j
            continue with next neighbour v_{k+1}
          if v_k ≠ v_j AND v_i > v_k then
            for all v_l ∈ A_k do
              if v_l = v_i then
                t ← t + 1
  return (3t)/p //the clustering coefficient
```

Algorithm 2 Pseudo-code for the multi-core CPU implementation of the clustering coefficient using PThreads. See Alg. 1 for a description of input parameter G and the triangle and paths counting algorithm.

```
declare v_curr //the current vertex
declare mutex m //mutual exclusion for v_curr
function CLUSTERING(G)
  R ← determine reverse adjacency-list lengths
  v_curr ← 0 //initialise current vertex v_curr
  n ← number of CPU cores
  do in parallel using n threads: call PROCESS(G, R)
  wait for all threads to finish processing
  declare t ← sum of triangles found by threads
  declare p ← sum of paths found by threads
  return (3t)/p //the clustering coefficient
```

Algorithm 3 Algorithm 2 cont. `PROCESS` runs in parallel.

```
function PROCESS(G, R)
  declare t //local triangle counter
  declare p //local paths counter
  declare v_s //start vertex
  declare v_e //end vertex
  repeat
    acquire lock on mutex m
    v_s ← v_curr
    v_e ← v_s+ work block size //v_e must not exceed |V|
    v_curr ← v_e
    release lock on mutex m
    for all v_i ∈ V_i ≡ {v_s, ..., v_e} ⊆ V do
      count triangles and paths as described in the sequential CPU alg.
  until v_s ≥ |V|
  return {t, p}
```

Algorithm 4 shows how the full iteration range is defined and passed to `parallel_reduce`. TBB recursively splits the iteration range into sub-ranges until a certain threshold is reached. Then TBB uses available worker threads to execute `PROCESS_TASK` (Algorithm 5) in parallel. When the two halves of a range have been processed, then TBB invokes function `JOIN` (Algorithm 6) to combine the results. Eventually, all sub-ranges have been processed and the results have been joined into the root of the task tree. TBB returns and the results can be extracted from this root object.

2.2 GPU - CUDA

The CUDA implementation is much more complex due to the different hardware architecture and lower level performance tuning necessary to achieve high performance on the GPU.

Arbitrary graphs, like small-world networks, where the structure is not known beforehand, can be represented in different ways in memory. For CUDA applications, the data

Algorithm 4 Pseudo-code for the multi-core CPU implementation of the clustering coefficient using TBB. See Algorithm 1 for a description of the input parameter G and the triangle and paths counting algorithm.

```
function CLUSTERING(G)
  R ← determine reverse adjacency-list lengths
  declare blocked_range br(0, |V|)
  call parallel_reduce(br, PROCESS_TASK)
  retrieve results and calculate clustering coefficient
```

Algorithm 5 TBB executes `PROCESS_TASK` in parallel.

```
declare t //task local triangle counter
declare p //task local paths counter
function PROCESS_TASK(br, G, R)
  declare v_s ← br.begin() //start vertex
  declare v_e ← br.end() //end vertex
  for all v_i ∈ V_i ≡ {v_s, ..., v_e} ⊆ V do
    count triangles and paths as described in the sequential CPU impl.
```

Algorithm 6 TBB calls `JOIN` to combine the results of the two halves of a range.

```
function JOIN(x, y)
  Input parameters: x and y are task objects.
  x.t ← x.t + y.t
  x.p ← x.p + y.p
```

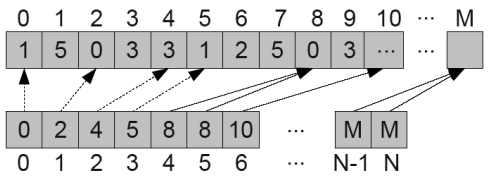


Fig. 1: The data structure used to represent the graph in graphics card device memory. It shows the vertex set V (bottom) and the arc set A (top). Every vertex $v_i \in V$ stores the start index of its adjacency-list A_i at index i of the vertex array. The adjacency-list length $|A_i|$ can be calculated by looking at the adjacency-list start index of v_{i+1} ($V[i+1] - V[i]$). The vertex array contains $|V|+1$ elements so that this works for the last vertex too.

representation and resulting data accesses often have a major impact on the performance and have to be chosen carefully. Figure 1 illustrates the data structure used to represent a graph in device memory.

Another issue when processing arbitrary graphs with CUDA is that the adjacency-lists differ in length, and often it is necessary to iterate over such a list of neighbours. But since in CUDA's single-instruction, multiple-thread (SIMT) architecture all 32 threads of warp are issued the same instruction, iterating over the neighbours-lists of 32 vertices can cause warp divergence if these lists are not all of the same length. In the case of warp divergence, all threads of the warp have to execute all execution paths, which in this case means they all have to do x iterations, where x is the longest of the 32 adjacency-lists. And as described in the CPU implementation of the clustering coefficient algorithm, this particular case requires nested loops, which make the problem even worse.

However, the outermost loop can be avoided when the implementation iterates over the arc set A instead of the vertex set V. This improves the performance of the CUDA

Algorithm 7 Pseudo-code for the CUDA implementation of the clustering coefficient. It operates on the arc set A, executing one thread for every arc $a_i \in A$ for a total of $|A|$ threads. Self-arcs are filtered out by the host as they never contribute to a valid triangle or path. The host program prepares and manages the device kernel execution.

```
function CLUSTERING(V, A, S)
  Input parameters: The vertex set V and the arc set A describe the structure
  of a graph G := (V, A). Every vertex v_i ∈ V stores the index into the arc
  set at which its adjacency-list A_i begins in V[i]. The vertex degree |A_i| is
  calculated from the adjacency-list start index of vertex v_{i+1} (V[i+1] −
  V[i]). In order for this to work for the last vertex v_N ∈ V, the vertex
  array contains one additional element V[N+1]. |A| is the number of arcs
  in G. S[i] stores the source vertex of arc a_i.
  declare V_d[|V|+1], A_d[|A|], S_d[|A|] in device memory
  copy V_d ← V
  copy A_d ← A
  copy S_d ← S
  declare t_d, p_d ← 0 in device memory //triangle and path counters
  do in parallel on the device using |A| threads:
        call KERNEL(V_d, A_d, S_d, t_d, p_d)
  declare t, p
  copy t ← t_d
  copy p ← p_d
  return (3t)/p //the clustering coefficient
```

Algorithm 8 Algorithm 7 continued. The device kernel is the piece of code that executes on the GPU.

function KERNEL(V, A, S, t, p)
  declare $i \leftarrow$ thread ID queried from CUDA runtime
  declare $v_i \leftarrow S[i]$ //arc source
  declare $v_j \leftarrow A[i]$ //arc end
  $p \leftarrow p + |A_j|$
  if $v_i > v_j$ **then**
    for all $v_k \in A_j$ **do**
      if $v_k = v_i$ **then**
        $p \leftarrow p - 2$ //correct for cycles of length 2 for both v_i and v_j
        continue with next neighbour v_{k+1}
      if $v_i > v_k$ **then**
        for all $v_l \in A_k$ **do**
          if $v_l = v_i$ **then**
            $t \leftarrow t + 1$

kernel considerably and also changes the total number of threads from $|V|$ to $|A|$. $|A|$ is usually much larger than $|V|$, giving CUDA more threads to work with, which it can use to hide memory latencies and which also means that the implementation should scale better to future graphics cards with more processing units. The implementation is described in Algorithm 7.

As the performance results section shows, this implementation performs well for graphs with only slightly varying vertex degrees, like Watts-Strogatz small-world networks [6]. If the vertex degrees of the input graph vary considerably, as it is typical for scale-free graphs with power-law degree distributions [10], [11], [12], a small variation of this implementation performs considerably better. In this second approach, the input array S not only references the source vertex of an arc, but also uses a second integer to store the end vertex. Furthermore, the host sorts this array by the degree of the arc end vertices before passing it to the CUDA kernel. Even though this means that the end vertex of each arc is stored twice, once in S and once in A, it makes it possible to process the arcs based on the vertex degrees of their end vertices, which determine the number of iterations done by the outer one of the two loops in the CUDA kernel. This means that threads of the same warp can process arcs with similar end vertex degrees, thus reducing warp divergence considerably. The sorting is done by the host using TBB's `parallel_sort`.

Further CUDA specific optimisations applied to both versions of the clustering kernel are shown in Algorithm 9. They include counting the triangles and paths found by each individual thread in its registers, before writing them to shared memory, where the total counts for a thread block are accumulated, which are eventually written to global memory with a single atomic transaction per counter and thread block. Furthermore, texture fetches are used when iterating over the adjacency-lists of vertices, taking advantage of data locality. And because the caching done when fetching the neighbours v_k of vertex v_j may be overwritten by the inner loop, a constant number of arc end vertices are pre-fetched and written to shared memory. This pre-fetching is only done for older devices like the GTX295, as the latest generation of Fermi-based NVIDIA GPUs, which includes the GTX480, provides automatic caching in L2 (shared by all multiprocessors) and L1 (on each multiprocessor) caches. The overhead of manually caching the data in shared memory decreases the performance on these devices.

2.3 Multi-GPU - CUDA & POSIX Threads

When multiple GPUs are available in the same host system, then it may be desirable to utilise all of them to further reduce the execution time of the algorithm. And because the iterations

Algorithm 9 CUDA performance optimisations.

```
// shared memory counters
__shared__ unsigned int nTrianglesShared;
__shared__ unsigned int nPaths2Shared;
if (threadIdx.x == 0) {
  nTrianglesShared = 0;
  nPaths2Shared = 0;
}
__syncthreads();

// each thread uses registers to count
unsigned int nTriangles = 0;
int nPaths2 = 0;
...
// NOTE: this explicit caching can be
// counter-productive on Fermi devices!
const int prefetchCount = 7;
__shared__ int nbr2Prefetch[prefetchCount*
                            BLOCK_SIZE];
if (srcVertex > nbr1) {
  for (int nbr2Idx = 0; nbr2Idx < nArcsNbr1;
       ++nbr2Idx) {
    //pre-fetch nbr2 to shared mem. to take
    //advantage of the locality in texture fetches
    int nbr2;
    int prefetchIdx=nbr2Idx%(prefetchCount+1);
    if (prefetchIdx == 0) {//global mem. read
      nbr2 = tex1Dfetch(arcsTexRef,
                        nbr1ArcsBegin+nbr2Idx);
      for (int i=0; i < prefetchCount; ++i) {
        nbr2Prefetch[i*blockDim.x+threadIdx.x]=
          tex1Dfetch(arcsTexRef,
                     nbr1ArcsBegin+nbr2Idx+i+1);
      }
    } else { // read from shared memory
      nbr2 = nbr2Prefetch[(prefetchIdx-1) *
                          blockDim.x+threadIdx.x];
    }
    ...
    for (int nbr3Idx=0; nbr3Idx < nArcsNbr2;
         ++nbr3Idx) {
      nTriangles += tex1Dfetch(arcsTexRef,
                      nbr2ArcsBegin+nbr3Idx)
                    == srcVertex ? 1 : 0;
    }
  }

  // write local counters to shared memory
  atomicAdd(&nTrianglesShared,nTriangles);
  atomicAdd(&nPaths2Shared, (unsigned int)nPaths2);
}
// write to global mem (once per thread block)
__syncthreads();
if (threadIdx.x == 0) {
  atomicAdd(nTotalTriangles,
    (unsigned long long int)nTrianglesShared);
  atomicAdd(nTotalPaths2,
    (unsigned long long int)nPaths2Shared);
}
```

of the outermost loop are independent from each other with no need for synchronisation, the work can be distributed over the available GPUs in the same way as multiple CPU cores are utilised by threads (See Section 2.1). One PThread is created for every GPU and controls the execution of all CUDA related functions on this particular GPU. The data structure of the graph is replicated on all graphics devices and instead of executing $|A|$ CUDA threads to count all triangles and paths with just one kernel call, a work block of N arcs $\{a_i, \ldots, a_{i+N-1}\} \subseteq A$ is processed during each kernel call. A new work block is determined in the same way as it is done when using PThreads to execute on multiple CPU cores. The work block size depends on the available graphics hardware and the size of the thread blocks in the CUDA execution grid: $N =$ (number of threads per block)$\times$(blocks per streaming multiprocessor)$\times$(number of streaming multiprocessors). The goal is to make it large enough to allow CUDA to fully utilise the hardware and small enough to keep all available GPUs

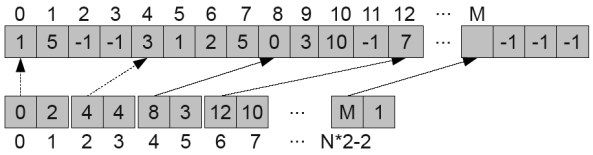


Fig. 2: The data structure used to represent the graph in system memory of the Cell BE. It shows the vertex set V (bottom) and the arc set A (top). Every vertex $v_i \in V$ stores the start index of its adjacency-list A_i at index $i \times 2$ of the vertex array. The adjacency-list length $|A_i|$ is stored at the next index. The vertex array contains $|V| \times 2$ elements. Every adjacency-list in the arcs array is padded to the next multiple of 16-bytes (4-bytes per value) in order to conform with the memory alignment requirements. The padding elements have the value -1, which is an invalid vertex ID.

busy for roughly the same amount of time.

2.4 Cell Processor - PS3

Like the CUDA implementation, the implementation for the Cell Broadband Engine (BE) requires a lot of architecture specific tuning to achieve good performance. The memory layout used is similar to the one used by the CUDA kernels, using one array for the vertices and one array for the arcs. However, the requirement that the direct memory accesses (DMA) used to transfer data from main memory to the local store of a Synergistic Processor Element (SPE) are aligned on 16-byte boundaries makes some changes necessary. See Figure 2 for an illustration and description of the memory layout.

The main task of the Cell's PowerPC Processor Element (PPE) is to manage the Synergistic Processor Elements (SPEs) as illustrated in Algorithm 10. It is used to load the graph and store it in system memory using the memory layout described before. Then it initialises the SPEs, which do most of the actual computation (See Algorithms 11 and 12). However, the PPE would not be fully utilised if providing the SPEs with further work was all it did. Therefore, it performs some of the same computational tasks in its spare time, further improving the overall performance. The implementation of the triangle and paths counting algorithm on the PPE is basically the same as the single-threaded CPU implementation described in Algorithm 1, except that it uses the PPE's vector unit in the same way as the SPE implementation does in its innermost loop. These vector operations are described in Algorithm 13.

Traversing an arbitrary graph as it is done by the triangle and path counting algorithms requires many reads from unpredictable memory addresses. And since the local store of the SPEs with its 256KB capacity is relatively small, much too small to hold the entire graph structure of anything but very small graphs, it is necessary to load the required parts of the graph from system memory into local memory when needed. For example, when processing a certain vertex, then its adjacency-list has to be copied into the local store. This is done by issuing a DMA request from the Synergistic Processor Unit (SPU) to its Memory Flow Controller (MFC) (every SPE has one SPU and one MFC). However, the performance of the implementation would not be good if the SPU stalled until the requested data becomes available. Instead, the implementation for the SPE is split into phases (See Figure 3 and Algorithm 12). A phase ends after a DMA request has been issued and the following phase, which uses the requested data, is not executed until the data is available. This implementation of multi-buffering uses 16 independent

Algorithm 10 Pseudo-code for the Cell BE implementation of the clustering coefficient. This algorithm describes the tasks of the PowerPC Processor Element. It operates on the vertex set V, issuing blocks of vertices to the the Synergistic Processor Elements for processing, as well as processing small work chunks itself when it has nothing else to do. Self-arcs are filtered out beforehand, as they never contribute to a valid triangle or path.

function CLUSTERING(V, A)
Input parameters: The vertex set V and the arc set A describe the structure of a graph $G := (V, A)$. Every vertex $v_i \in V$ stores the index into the arc set at which its adjacency-list A_i begins in $V[i \times 2]$ and its degree in $V[i \times 2 + 1]$. $|V|$ is the number of vertices in G. $SPE = \{spe_0, spe_1, \ldots, spe_5\}$ is the set of SPEs.
for all $spe_i \in SPE$ **do**
initialise spe_i and start processing a block of vertices
while more vertices to process **do**
for all $spe_i \in SPE$ **do**
if inbound mailbox of spe_i is empty **then**
write the start and end vertices of the next work block to the mailbox
process a small work block on the PPE
for all $spe_i \in SPE$ **do**
send interrupt signal and wait until spe_i finishes processing
aggregate results and calculate clustering coefficient

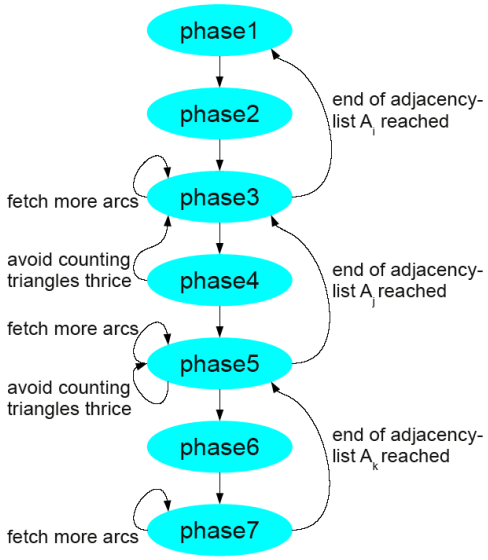


Fig. 3: The phases of the SPE implementation and how they are connected to each other. The progression from phase x to phase $x + 1$ is always due to phase x issuing a DMA request to copy data from system memory into local memory, which is needed for phase $x + 1$ to execute. Phases with an odd number end after they issue a request to fetch the start index and length information about a particular adjacency-list, whereas phases with an even number end after they issue a request to fetch the actual adjacency-list data for a particular vertex. The figure illustrates under which conditions a phase is repeated or the execution path goes back up towards *phase1*. See Algorithm 12 for the pseudo-code of the phases implementation.

buffers to process the work block issued to the SPE. Whenever a buffer is waiting for data, the implementation switches to another buffer that is ready to continue with the next phase.

The Cell PPE and SPE units all have their own vector units and 128-bit wide vector registers. This allows them to load four 32-bit words into a single register and, for example, add them to four other words stored in a different register in a single operation. A program for the Cell BE should

Algorithm 11 The pseudo-code for the SPE implementation of the clustering coefficient on the Cell BE. See Algorithm 10 for the PPE implementation and Algorithm 12 for the different execution phases.

```
function CLUSTERING(v_s, v_e)
  Input parameters: Each SPE receives an initial work block [v_s, ..., v_e) ⊆ V of source vertices to process.
  copy init. data from system mem. to the local store
  initialise buffers B = {b_0, b_1, ..., b_15}
  repeat
    v_curr ← v_s //initialise current vertex v_curr
    for all b_i ∈ B do
      b_i.phase ← phase1 //set the next phase of b_i
      mark buffer as "ready"
    //process the current work block
    set all buffers as active
    while at least one buffer is active do
      b ← any "ready" buffer
      call b.phase //execute the next phase of b
    //check if there is more work to do
    v_s ← read next value from inbound mailbox
    if no interrupt signal recieved (v_s ≠ −1) then
      v_e ← read next value from inbound mailbox
  until interrupt signal received
  copy the results back to system memory
```

be vectorised where possible to fully utilise the available processing power. Algorithm 13 describes how the innermost loop of the PPE and SPE implementations use of the vector units.

It turns out that the performance gain from using both the PPE and the SPEs to process the data is smaller than expected compared to using either only the PPE or only the SPEs to do the actual data crunching. It appears that the memory system is the bottleneck when using all of the available processing units on the Cell processor on a data-intensive problem like the one at hand.

3. Performance Results

This section compares the performance of the different clustering coefficient implementations. Table 1 lists the platforms used for the performance experiments. System (a) was used for the single-core and multi-core CPU measurements, both systems (a) and (b) for the GPU results and system (c) for the CellBE implementation.

Two different graphs models are used as input to the algorithms. The Watts-Strogatz network model [6] generates small-world graphs, where every vertex is initially connected to its k nearest neighbours. These edges are then randomly rewired with a probability p. The graphs generated for the performance measurements ($k = 50$ and $p = 0.1$, see Figure 4) have a high clustering coefficient of ≈ 0.53. The vertex degrees do not deviate much from k.

The Barabási-Albert scale-free network model [11] generates graphs with a power-law degree distribution for which the probability of a node having k links follows $P(k) \sim k^{-\gamma}$. Typically, the exponent γ lies between 2 and 3 [10], [12]. The vertex degrees in the resulting graph vary considerably. The graphs generated for the performance measurements ($k \approx 50$, see Figure 5) have a clustering coefficient of ~ 0.01.

The timing results show that the type of graph used as input to the algorithms has a big effect on the execution times. The scale-free graphs take much longer to process than the small-world graphs, because even though only few vertices have a degree that is much higher than the average, most vertices are connected to one of these hub nodes and the algorithms therefore often have to iterate over the large adjacency-lists of these few vertices.

Table 2 gives an overview of the performance measurements and compares the results with each other. It shows

Algorithm 12 Algorithm 11 continued. The phases of the SPE implementation execute on a buffer b. Each phase models a step in the process of counting the triangles t and paths p. A phase ends after a DMA request to load data into local storage has been issued or when the end of a loop is reached.

```
function phase1(b)
  b.v_i ← v_curr //set the source vertex for this buffer
  v_curr ← v_curr + 1
  if b.v_i ≥ v_e then
    set buffer as inactive //end of work block reached
  else
    copy_async b.v_i.dat ← load adjacency-list info
    b.phase ← phase2
function phase2(b)
  copy_async b.A_i ← use b.v_i.dat to load A_i ⊂ A
  b.phase ← phase3
function phase3(b)
  if end of adjacency-list b.A_i reached then
    b.phase ← phase1 //loop condition not fulfilled
  else
    b.v_j ← next value in b.A_i
    copy_async b.v_j.dat ← load adjacency-list info
    b.phase ← phase4
function phase4(b)
  b.p ← b.p + |A_j|
  if b.v_j > b.v_i then
    b.phase ← phase3 //do not count triangle thrice
  else
    copy_async b.A_j ← use b.v_j.dat to load A_j ⊂ A
    b.phase ← phase5
function phase5(b)
  if end of adjacency-list b.A_j reached then
    b.phase ← phase3 //loop condition not fulfilled
  else
    b.v_k ← next value in b.A_j
    if b.v_k = b.v_i then
      b.p ← b.p − 2 //correct for cycles of length 2
      b.phase ← phase5
    else if v_k > v_i then
      b.phase ← phase5 //don't count triangle thrice
    else
      copy_async b.v_k.dat ← load adj.-list info
      b.phase ← phase6
function phase6(b)
  copy_async b.A_k ← use b.v_k.dat to load A_k ⊂ A
  b.phase ← phase7
function phase7(b)
  for all b.v_l ∈ b.A_k do
    if b.v_l = b.v_i then
      b.t ← b.t + 1 //triangle found
  b.phase ← phase5
```

that the multi-threading implementations using PThreads (12 threads) and TBB (automatic task management) are both $\approx 7\times$ faster than the sequential implementation for the largest measured instances of the small-world and scale-free graphs. Even though the processor only has 6 physical cores, Intel's hyper-threading technology effectively doubles this to 12 logical cores, which enables it to better utilise the physical cores. This makes it possible to scale beyond the actual number of cores. The TBB implementation performs almost exactly the same as the PThreads implementation. Its ease of development and automatic scaling to different system configurations thus makes it a powerful alternative to the more low-level multi-threading with PThreads.

As mentioned in Section 2.2, we have two CUDA kernels that differ in one aspect. The CUDA threads in kernel 1 access the array of arcs A in the given order, whereas kernel 2 uses a second array of arcs which is sorted by the degrees of the arc end vertices to determine which arc is processed by each thread. This second kernel uses $|A|\times$ (size of integer) more space and introduces some processing overhead, which shows in the lower performance when processing the small-world graphs. However, the reduced warp divergence gained

Table 1: The platforms used for the performance measurements. Note that only 6 of the total 8 SPEs on the CellBE are available to the developer (one is disabled and one reserved by the operating system).

ID	CPU	GPU	RAM	Operating System
(a)	Intel Core i7 970 @3.2 GHz (6 cores)	4× NVIDIA GTX480 (4 GPUs)	12 GB	Ubuntu 10.10 64-bit
(b)	Intel Core 2 Quad @2.66 GHz (4 cores)	NVIDIA GTX295 (2 GPUs)	4 GB	Ubuntu 10.10 64-bit
(c)	PS 3 CellBE @3.2 GHz (1 PPE & 6 SPEs)	NVIDIA RSX	256 MB	Yellow Dog Linux 6.1

Algorithm 13 Vector operations are used to speed-up the execution of the innermost loop (phase7) of the Cell BE PPE and SPE implementations. The comparison of vertex ID v_i with $v_l, v_{l+1}, v_{l+2}, v_{l+3}$ is done concurrently using the 128-bit vector unit. As the vector unit executes instructions in SIMD fashion, it is necessary to eliminate the branch. Several intrinsic instructions can be used to get the same effect as the if-condition: *spu_cmpeq* compares two vectors for equality and returns a bit-mask which represents true and false results; *spu_sel* selects one of two values (0 if the vertex IDs are not equal and 1 if a triangle has been found) based on this bit-mask; and *spu_add* adds the selected values to a vector that is used to count the number of triangles.

```
vec_uint4 case0 = spu_splats((uint32)0);
vec_uint4 case1 = spu_splats((uint32)1);
for (int nbr3Idx=0; <loop condition>;
     nbr3Idx+=4) {
  buf.nTrianglesVec =
    spu_add(buf.nTrianglesVec,
            spu_sel(case0,
                    case1,
                    spu_cmpeq(
   *((vec_int4*)&buf.arcsBuf3[nbr3Idx]),
                              buf.vertexId
                              )
                    )
            );
}
```

through this overhead pays off when processing scale-free graphs. Here the scenario is reversed and kernel 2 clearly outperforms kernel 1 by a considerable margin. This shows once again [4], [13] that the performance of graph algorithms running on the GPU in many cases depends on the graph structure and that there is not one best implementation for all cases. If the graph structure is not known beforehand, then it may be worthwhile to attempt to automatically determine the type of graph in order to be able to choose the optimal CUDA implementation.

The multi-GPU implementations using both GPUs of the GeForce GTX295 or up to four GeForce GTX480s perform best when the graph size is large enough to keep all processing units of the devices busy. Even the largest measured graph instances are not large enough to allow the GTX480s to scale particularly well. The slopes given in the table highlight this especially for the small-world graphs. The single-GPU implementation even outperforms the multi-GPU implementation for the smallest measured graph instances due to the overhead introduced by using multiple GPUs. The first three data points were filtered out when the fitted slopes were calculated so that these small graphs do not distort the scaling of the multi-GPU measurements.

The Cell implementation positions itself between the single- and multi-threaded CPU implementations when processing the scale-free graphs or the smaller instances of the small-world graphs. The timing results of the small-world graphs suddenly increase at the $V = 400,000$ mark and even more considerably at the $V = 800,000$ mark. This is caused

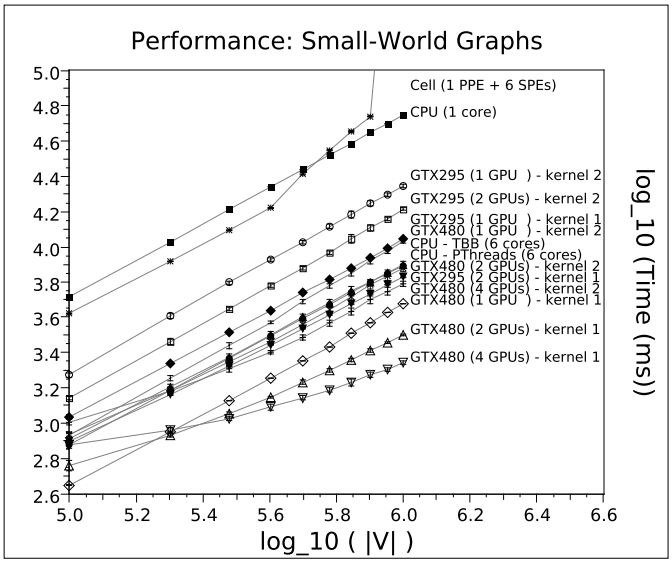


Fig. 4: The timing results in milliseconds for Watts-Strogatz small-world graphs with rewiring probability $p = 0.1$ and degree $k = 50$. The number of vertices $|V|$ ranges from $100,000 - 1,000,000$. All data points are the mean values of 20 measurements. Error bars show the standard deviations.

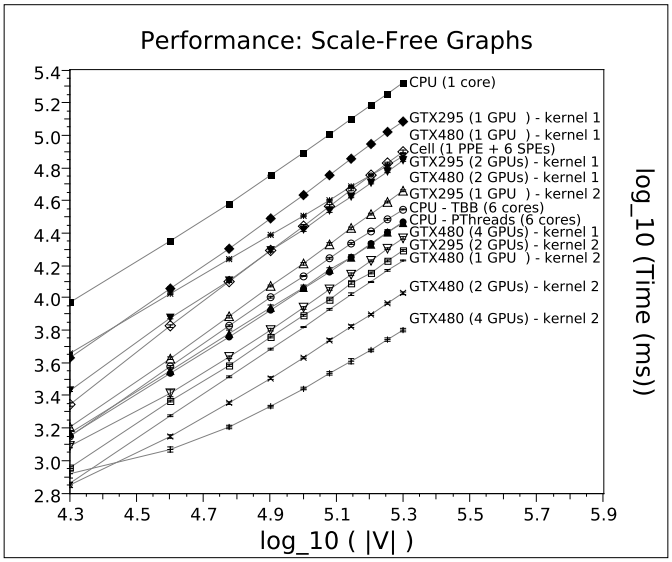


Fig. 5: The timing results in milliseconds for Barabási scale-free graphs with a degree $k \approx 50$. The number of vertices $|V|$ ranges from $20,000 - 200,000$. All data points are the mean values of 20 measurements. Error bars show the standard deviations.

Table 2: Performance comparison. The speed-up values are for the largest measured graph instance. An exception are the small-world network measurements on the Cell processor as mentioned in the main text. Speed-up **S1** is relative to the single-core CPU implementation, whereas speed-up **S2** compares the multi-core/GPU implementations to the respective single-core/GPU implementations. The slopes of the least square linear fits show how well the algorithms scale with increasing graph size. Values are rounded to 3 significant digits.

Compute Device	S1	S2	Slope
Small-world			
Core i7 970 (1 core)	1.00	1.00	1.02
Core i7 970: PThreads (6 cores)	7.06	7.06	1.01
Core i7 970: TBB (6 cores)	7.02	7.02	1.00
CellBE (1 PPE & 6 SPEs)	1.31	N/A	1.00
GTX295: kernel 1 (1 GPU)	5.02	1.00	1.02
GTX295: kernel 1 (2 GPUs)	8.24	1.64	0.98
GTX480: kernel 1 (1 GPU)	11.8	1.00	1.07
GTX480: kernel 1 (2 GPUs)	17.6	1.50	0.88
GTX480: kernel 1 (4 GPUs)	25.3	2.15	0.65
Scale-free			
Core i7 970 (1 core)	1.00	1.00	1.42
Core i7 970: PThreads (6 cores)	7.14	7.14	1.37
Core i7 970: TBB (6 cores)	7.23	7.23	1.32
CellBE (1 PPE & 6 SPEs)	2.80	N/A	1.21
GTX295: kernel 2 (1 GPU)	5.96	1.00	1.36
GTX295: kernel 2 (2 GPUs)	10.7	1.79	1.33
GTX480: kernel 2 (1 GPU)	12.2	1.00	1.38
GTX480: kernel 2 (2 GPUs)	19.7	1.61	1.30
GTX480: kernel 2 (4 GPUs)	33.1	2.71	1.17

by the minimalistic 256 MB of main memory available in the PlayStation 3, which forces the system to start paging memory to the hard drive. We therefore filter the results above $V = 400,000$ out when calculating the slope for these graphs to report the true scaling of the CellBE and compare the performance of the CellBE using graph instances of size $V = 400,000$.

4. Discussion & Conclusions

Power consumption and physical size constraints have led to a "slowing down of Moore's Law" [14], [15] for processing devices at least in terms of conventional approaches using uniform and monolithic core designs. The consequence is that parallel computing techniques—such as incorporating multiple processing cores and other acceleration technologies—have become increasingly important [16].

Following Moore's Law, the number of transistors on a GPU roughly doubled from the GT200 to the GT400 series graphics cards, which have become available within just under 2 years from each other. But more importantly, the effective performance achieved in our experiments has roughly doubled too. The many-core architecture of today's GPUs has been shown [17] to significantly outperform traditional multi-core CPU architectures for algorithms that can be adapted to the specific requirements of the CUDA programming model.

Considering its age at the time of writing, the results of the Cell Broadband Engine are still quite impressive and show the potential of this hybrid CPU architecture compared to an architecture with fewer full-fledged cores. However, it requires considerably more effort to achieve good results when developing for the Cell than it does for an x86-based multi-core CPU.

In summary, we have implemented the clustering coefficient on a number of popular parallel architectures and discussed the design decisions necessary to achieve good performance and scaling on these platforms. We have used code fragments to highlight the differences in the implementations and explained architecture specific optimisation strategies. We have compared the runtime performance and scalability using both small-world and scale-free graphs.

We found that developing for multi-core CPUs using PThreads or TBB is much easier than developing for GPUs or the CellBE. This is partly due to the fact that compilers for x86-based processors have been around for much longer and are expected to perform most of the necessary low-level optimisations automatically, and partly due to the processors themselves having very sophisticated caching, pre-fetching and branch-prediction logic. The developer has to tackle a lot of these challenges explicitly when programming for the GPU or CellBE.

These efforts can yield very worthwhile performance enhancements. The GPUs dominated the benchmarks even though this algorithm is not particularly SIMD friendly, showing that even some graph-based, bandwidth limited algorithms can be implemented efficiently on their many-core architecture.

Recent GPU developments, like the automatic caching on NVIDIA's Fermi devices, have somewhat relaxed the demands placed on the developer and this trend is likely to continue with ever improving hardware and software. A feature recently introduced in CUDA toolkit 3.2 [18] even enables the allocation of dynamic memory in kernel code, which we intend to exploit in future work to generate graphs directly in graphics device memory.

References

[1] Graph500.org, "The Graph 500 List," http://www.graph500.org/, last accessed November 2010.

[2] H. Meuer, E. Strohmaier, H. Simon, and J. Dongarra, "36th list of top 500 supercomputer sites," www.top500.org/lists/2010/11/press-release, November 2010.

[3] N. Bell and M. Garland, "Implementing Sparse Matrix-Vector Multiplication on Throughput-Oriented Processors," in *Proc. Supercomputing*, 2009.

[4] A. Leist, D. Playne, and K. Hawick, "Exploiting Graphical Processing Units for Data-Parallel Scientific Applications," *Concurrency and Computation: Practice and Experience*, vol. 21, pp. 2400–2437, December 2009, CSTN-065.

[5] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, no. 2, p. 026118, July 2001.

[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, June 1998.

[7] S. Milgram, "The Small-World Problem," *Psychology Today*, vol. 1, pp. 61–67, 1967.

[8] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, June 2003.

[9] K. Hawick, A.Leist, and D.P.Playne, "Mixing Multi-Core CPUs and GPUs for Scientific Simulation Software," *Res. Lett. Inf. Math. Sci.*, vol. 14, no. ISSN 1175-2777, pp. 25–77, 2010. [Online]. Available: http://www.massey.ac.nz/massey/learning/departments/iims/research/research-letters/

[10] D. J. d. Price, "Networks of Scientific Papers," *Science*, vol. 149, no. 3683, pp. 510–515, July 1965.

[11] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, October 1999.

[12] R. Albert, H. Jeong, and A.-L. Barabasi, "The diameter of the world wide web," *Nature*, vol. 401, pp. 130–131, 1999.

[13] K. A. Hawick, A. Leist, and D. P. Playne, "Parallel Graph Component Labelling with GPUs and CUDA," *Parallel Computing*, vol. 36, pp. 655–678, 2010. [Online]. Available: www.elsevier.com/locate/parco

[14] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, vol. April, p. 4, 1965.

[15] S. K. Moore, "Multicore is bad news for supercomputers," *IEEE Spectrum*, vol. 45, no. 11, p. 11, 2008.

[16] G. Goth, "Entering a parallel universe," *Communications of the ACM*, vol. 52, no. 9, pp. 15–17, September 2009.

[17] K. Hawick, A. Leist, and D. Playne, "Regular Lattice and Small-World Spin Model Simulations using CUDA and GPUs," Computer Science, Massey University, Tech. Rep. CSTN-093, 2009, to appear in Int. J. Parallel Programming (2010).

[18] *NVIDIA CUDA™ C Programming Guide Version 3.2*, NVIDIA® Corporation, 2010, last accessed December 2010. [Online]. Available: http://www.nvidia.com/

Application of Quaternion Interpolation (SLERP) to the Orientation Control of 6-Axis Articulated Robot using LabVIEW® and RecurDyn®

Jin Su Ahn[1], **Won Jee Chung**[1], **Su Seong Park**[1]
[1]School of Mechatronics, Changwon National University, South Korea

Abstract - *In general, the orientation interpolation of industrial robots has been done based on Euler angle system which can result in singular point (so-called Gimbal Lock). However, quaternion interpolation has the advantage of natural (specifically smooth) orientation interpolation without Gimbal Lock. This paper presents the application of quaternion interpolation, specifically Spherical Linear IntERPolation (in short, SLERP), to the orientation control of the 6-Axis articulated robot (RS2) using LabVIEW® and RecurDyn®. For the comparison of SLERP with linear Euler interpolation in view of smooth movement (profile) of joint angles (toqrues), the two methods are dynamically simulated on RS2 by using both LabVIEW® and RecurDyn®. Finally our original work, specifically the implementation of SLERP and linear Euler interpolation on the actual robot, i.e. RS2, is done using LabVIEW® motion control tool kit. The SLERP orientation control is shown to be effective in terms of smooth joint motion and torque when compared to a conventional (linear) Euler interpolation.*

Keywords: *Quaternion, Spherical Linear interpolation (SLERP), Euler Angle, Linear Euler Interpolation, 6-Axis Articulated Robot (RS2), LabVIEW®, RecurDyn®.*

1 Introduction

Nowadays, the performance of robot has been improved according to the development of robot control techniques. In some applications of robot, its performance is superior to human being's one. Even robots can be applied to the fields to which workers cannot be committed. For example, some welding robots can perform excellent welding better than human workers. These robots need accurate orientation interpolation with sooth movement. Besides welding, various tasks such as spray painting, sealing and handling require smooth orientation control.

In general, the orientation interpolation of industrial robots has been done based on Euler angle system [1]. However, the orientation interpolation using Euler angles can result in singular point (so-called Gimbal Lock [2]), which can cause the malfunction of robots with systematic errors [3]. In addition, it can lead to undesirable results because it ignores interrelation between joint axes even in simple linear interpolation.

However, quaternion interpolation has the advantage of natural (specifically smooth) orientation interpolation without singular point such as Gimbal Lock. Since Quaternion interpolation has been mostly used in 3-dimensional computer graphics, it has been applied to robot simulation instead of real robot control (or implementation) as shown in refs. [4-6].

In this paper, we will investigate on orientation control using quaternion interpolation for 6-Axis articulated robot (we will call it as RS2 hereinafter) which has been developed at our lab for research purpose. The robot control based on LabVIEW® is briefly explained for the RS2 model. In addition, we will show our programming methods regarding both forward kinematics and inverse kinematics for RS2, which are needed for quaternion interpolation. In Section 3, Quaternion interpolation, specifically Spherical Linear IntERPolation (in short, SLERP) [7], is explained with linear Euler interpolation (which has been widely used for orientation control of industrial robots). For the comparison of SLERP with linear Euler interpolation in view of smooth movement of joint angles, the two methods are dynamically simulated on RS2 by using both LabVIEW® and RecurDyn®. In Section 4, our original work, specifically the implementation of SLERP on the actual robot, i.e. RS2, is done using LabVIEW® motion control tool kit. Especially the linear Euler interpolation is also implemented on RS2, which is also compared with the SLERP in terms of torque. Finally Fig. 5 summarizes shows the structure of this paper. Concluding remarks will be made in Section 5.

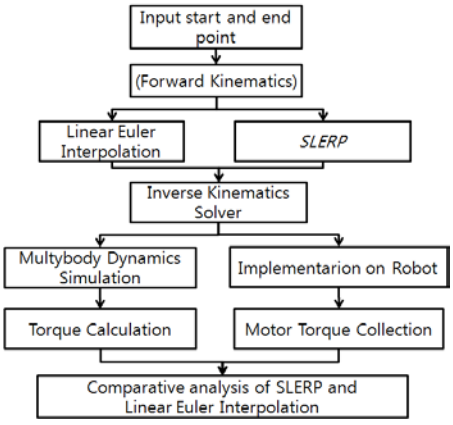


Fig. 1 Structure of this paper

2 Robot System (RS2) based on LabVIEW®

2.1 Introduction of RS2

Usually 6-axes manipulators which are widely used for welding, spray painting and so on, have payloads from 10 kg to 300kg. The payload over 500 kg belongs to Heavy Duty Handling Articulated Manipulator (abbreviated as HDHAM). In order to enhance both the control accuracy and the reliability of HDHAM, the synthetic technology including design, prototyping and control should be accompanied. For this purpose, in this paper, one fourth (1/4) model of HDHAM with 6 DOF (Degrees Of Freedom) (named as RS2) has been used as a preliminary step to manufacture the original model of HDHAM. The original HDHAM (2.4 m in height and 3.6 m in length) will be destined for handling payload of 600 kg. The RS2 shown in Fig. 2 is used for investigating the orientation control technology of original HDHAM in a laboratory. [8]

For the control of RS2 system, LabVIEW® is adopted as a graphical programming language that uses icons instead of lines of text to create applications. LabVIEW® programs are called Virtual Instruments (VIs), because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. LabVIEW® contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help us troubleshoot code we write. Especially the LabVIEW® hardware used in this paper is NI PXI-7350 Motion Controller, which sends commands to the servo drivers of Mitsubishi® J2-Super series [9] for the motion control of RS2.

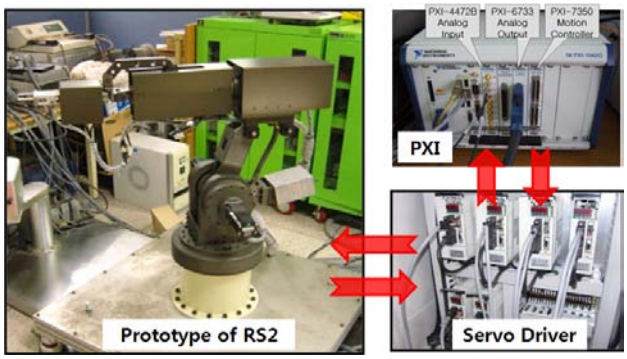


Fig. 2 RS2 System

2.2 Forward and Inverse kinematics of RS2 based on LabVIEW®

In forward kinematics, the length of each link and the angle of each joint are given and we have to calculate the position of any point in the robot. Specifically, forward kinematics is computation of the position(X, Y, Z) and orientation(α, β, γ) of robot's end-effector. It is widely used in robotics. The orientation (α, β, γ) of robot means Euler angles [10]. In inverse kinematics, the length of each link and position of the point are given and we have to calculate the angle of each joint.

In our previous paper [11], we had solved forward and inverse kinematics solution for RS2. In this paper, we just show the LabVIEW® graphical program, which has developed in our lab, based on forward and inverse kinematics solutions for RS2. Forward kinematics program calculates the position and orientation of end-effector corresponding to input angle of each joint through the homogeneous transformation matrix ${}^{0}_{6}T$ as shown in Fig 3. The advantage of developed program is that the homogeneous transformation matrix has been easily calculated only by modifying input angles. This forward kinematics routine of LabVIEW® is often called in the interpolation programs for RS2 which will be explained in the later section.

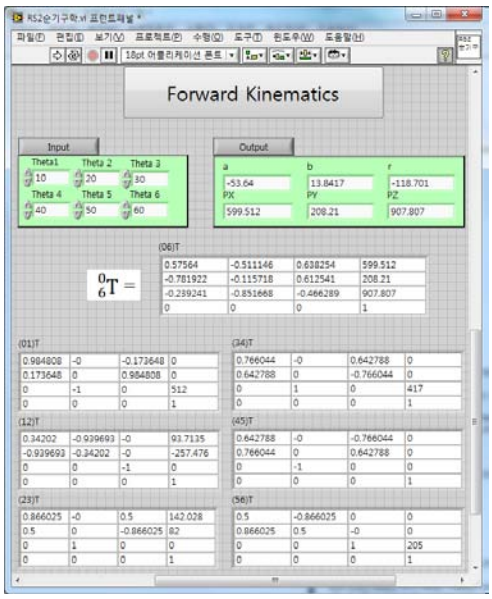


Fig. 3 Forward Kinematics Program

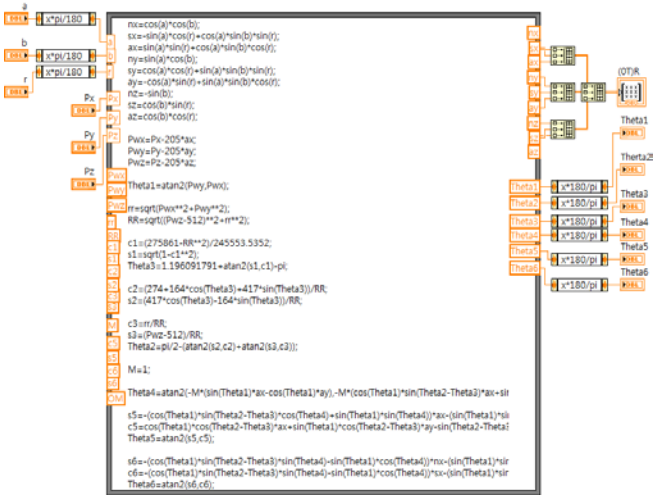


Fig. 4 Inverse Kinematics Program

In the meanwhile, inverse kinematics program calculates joint angles corresponding to input values of 6 DOF(X, Y, Z, α, β, γ). Fig. 4 shows a part of source routine of inverse kinematics for RS2, written in LabVIEW® graphical program. The inverse kinematics program is linked to interpolation programs as Sub VI type (in a format of subprogram LabVIEW®) for both dynamic simulation of interpolation and real implementation of interpolation on RS2. In the interpolation program, the inverse kinematics program calculates the angle of each joint every sampling time (a few milliseconds). Especially the results of inverse kinematics program play an important role in generating command values of joint angles for NI PXI motion controller of LabVIEW®.

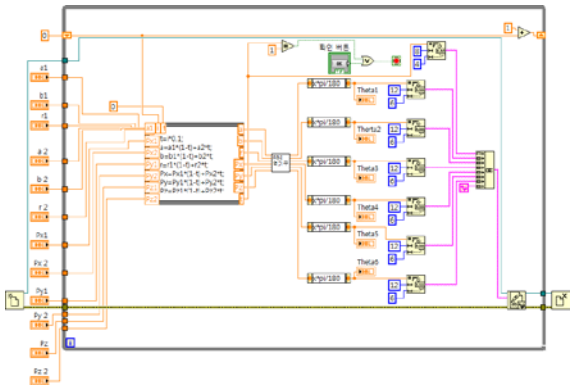


Fig. 5 LabVIEW® source program of linear Euler Interpolation

3 Dynamic Simulation using Linear Euler Interpolation and SLERP

3.1 Linear Euler interpolation

The space of orientations can be parameterized by Euler angles. When Euler angles are used, a general orientation is written as a series of rotations about three mutually orthogonal axes in space. In general, Euler angles are widely used for orientation of robot. Using the equivalence between Euler angles and rotation composition, it is possible to change to and from matrix convention.

In this paper, we used Z-Y-X Euler angle [12], where the rotation matrix R has been obtained from the homogeneous transformation matrix of forward kinematics program stated in section 2.2. In addition, the rotation matrix can be equivalently interchanged with Euler angles (α, β, γ) as follows:

$$\boldsymbol{R} = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \tag{1}$$

$$\triangleq \begin{bmatrix} \cos\beta\cos\alpha & -\cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha & \sin\gamma\sin\alpha + \cos\gamma\sin\beta\cos\alpha \\ \cos\beta\sin\alpha & \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha & -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha \\ -\sin\beta & \sin\gamma\cos\beta & \cos\gamma\cos\beta \end{bmatrix} \tag{2}$$

where

$$\begin{aligned} \alpha &= \arccos(a_z), \\ \beta &= -\arctan2(a_x, a_y) \\ \gamma &= \arctan2(n_z, s_z) \end{aligned} \tag{3}$$

wherein nx, ny, …. and az are asuumed to be given.

The simple linear interpolation between two Euler angles is most obvious method. To develop interpolation program using Euler angles, linear Euler interpolation has been used in the LabVIEW® graphical program of Fig. 5 as follows [7] :

$$\begin{aligned} \boldsymbol{r}_0 = (\alpha_0, \beta_0, \gamma_0), \quad \boldsymbol{r}_1 = (\alpha_1, \beta_1, \gamma_1) \\ LinEuler(\boldsymbol{r}_0, \boldsymbol{r}_1, t) = \boldsymbol{r}_0(1-t) + \boldsymbol{r}_1 t \\ (0 < t < 1) \end{aligned} \tag{4}$$

Equation 4 shows that linear Euler interpolation program performs interpolation based on Euler angles of both start point $\boldsymbol{r}_0$ and end point $\boldsymbol{r}_1$.This program calculates the angle of each joint every sampling time by using the inverse kinematics program developed in section 2.2.

3.2 *SLERP*

A quaternion has been introduced for the notation of orientation since it has simple notation of rotation as well as being convenient for the interpolation for orientation [7]. The quaternion can express itself into a rotational axis and rotational angle about the axis. The quaternion can be defined by Eq. (5) :

$$\boldsymbol{q} = w + (x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}) \tag{5}$$

Here x, y, z, w are real numbers while $x\boldsymbol{i}, y\boldsymbol{j}, z\boldsymbol{k}$ denote complex numbers. Due to the characteristics of complex numbers, it follows that

$$\begin{aligned} \boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = -1 \\ \boldsymbol{ij} = \boldsymbol{k}, \boldsymbol{jk} = \boldsymbol{i}, \boldsymbol{ki} = \boldsymbol{j}, \boldsymbol{ijk} = -1 \end{aligned}$$

Here it can be said that x, y and z denote the axis of rotation while w indicates the angle of rotation. Besides, any rotation matrix can be converted into a quaternion as follows [13] :

$$\begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{bmatrix}$$

$$= \boldsymbol{q}(w, (x, y, z)) \tag{6}$$

Here

$$w = \frac{\sqrt{n_x + s_y + a_z + 1}}{2} \tag{7}$$

$$x = \frac{s_z - a_y}{4w}, y = \frac{a_x - n_z}{4w}, z = \frac{n_y - s_x}{4w} \quad (8)$$

In this paper, to develop orientation interpolation program, we have used Spherical Linear intERPtion, i.e., *SLERP* proposed by Shoemaker [14], one of quaternion-based interpolation methods. *SLERP* has a geometric formula independent of quaternions, and independent of the dimension of the space in which the arc is embedded.

Let $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ be the first and last points (specifically quternions) of the arc, and let t be the parameter, $0 \le t \le 1$. Compute θ as the angle subtended by the arc, so that $\cos\theta = \boldsymbol{q_1} \cdot \boldsymbol{q_2}$, the 4-dimensional dot product of the unit quaternions from the start point to the end point. Then *SLERP* can be expressed by equations (9) and (10):

$$\boldsymbol{q_1} = w + (x_1\boldsymbol{i} + y_1\boldsymbol{j} + z_1\boldsymbol{k})$$
$$\boldsymbol{q_2} = w + (x_2\boldsymbol{i} + y_2\boldsymbol{j} + z_2\boldsymbol{k})$$

$$slerp(t; \boldsymbol{q_1}, \boldsymbol{q_2}) = \frac{\boldsymbol{q_1}\sin((1-t)\theta) + \boldsymbol{q_2}\sin(t\theta)}{\sin\theta} \quad (9)$$

$$\theta = cos^{-1}(\boldsymbol{q_1} \cdot \boldsymbol{q_2}) \quad (10)$$

Figure 6 shows the LabVIEW® graphical source program of *SLERP*, which is more complicated than that of linear Euler interpolation program. The inputs of linear Euler interpolation program are Euler angles, while the input to *SLERP* are quaternions. For the comparative analysis of linear Euler interpolation with *SLERP*, the *SLERP* program needs the same conditions as linear Euler interpolation as follows. First, the *SLERP* program converts input quaternions into their corresponding input Euler angles by using equations (2), (3) and (6). Then this program performs *SLERP* interpolation based on equation (10) which results in quaternion outputs. Then the quaternion outputs are converted into Euler angles in the similar manner to the input Euler angles. Finally the *SLERP* program shown in Fig. 6 calls the inverse kinematics routine (shown in Fig. 4) to obtain joint angles from the Euler angles, under the assumption that the positions of end-effector trajectory are given between start and end points.

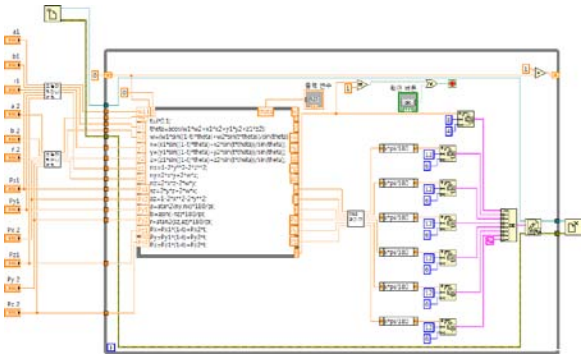


Fig. 6 Spherical Linear Interpolation Program Source

3.3 Dynamic simulation of linear Euler interpolation and *SLERP* for RS2

In order to compare linear Euler interpolation with *SLERP* for RS2 by using RecurDyn® (multi-body dynamic simulation software), first of all, the two interpolation LabVIEW® programs developed in subsections 3.1 and 3.2 calculate joint angles according to every sampling time under the same conditions of start and end points. The simulation results are shown in Fig. 7 where the blue and red colors denote linear Euler interpolation and *SLERP*, respectively.

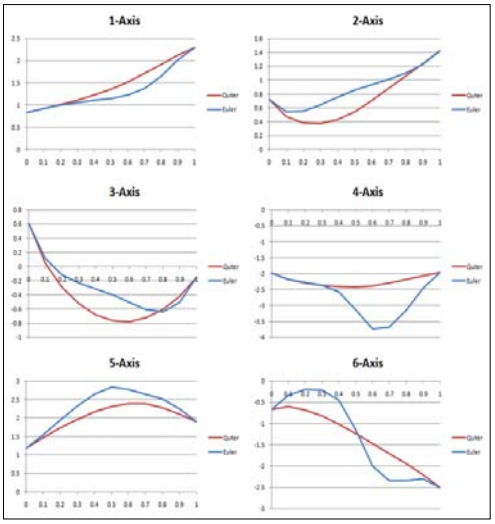


Fig. 7 Simulated angle of each Joint based on linear Euler interpolation and SLERP

Then the joint angles are applied to the RS2 model of RecurDyn® so that linear Euler interpolation can be compared with *SLERP* from the viewpoint of smoothness of both joint torques and end-effector velocity. The dynamic simulation of RecurDyn® aims at testing the virtual performance of the two interpolation methods before their real implementation on RS2. The 3-dimenional (3-D) model of RS2 has been constructed in RecurDyn® by importing the 3-D RS2 model of Solid Works, as shown in the upper right of Fig. 8. Table 1 shows the maximum torque of each joint for linear Euler interpolation and *SLERP*. It can be noticed that the magnitudes of the maximum torques for joint 1, 5 and 6 of *SLERP* are much smaller than those of linear Euler interpolation. Moreover *SLERP* can result in smooth joint torque profiles in comparison with linear Euler interpolation, as shown in Fig. 9. In addition, Fig. 10 shows that the end-effector velocity profile of *SLERP* is more smooth than that of linear Euler interpolation. Consequently it can be stated that *SLERP* has the advantage of natural (specifically smooth joint profile with less torque) orientation interpolation without singular point, compared with linear Euler interpolation.

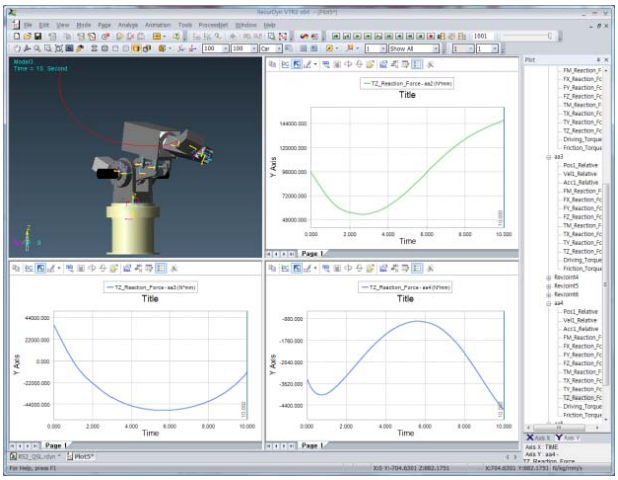


Fig. 8 Dynamic Simulation using RecurDyn

Table 1 Maximum Torque of Each Joint

(Unit: N·mm)

Joint	Euler	*SLERP*
1	2,209	520
2	147,578	146,293
3	-42,895	-50,393
4	-4,723	-4,577
5	2,025	871
6	-4,591	-978

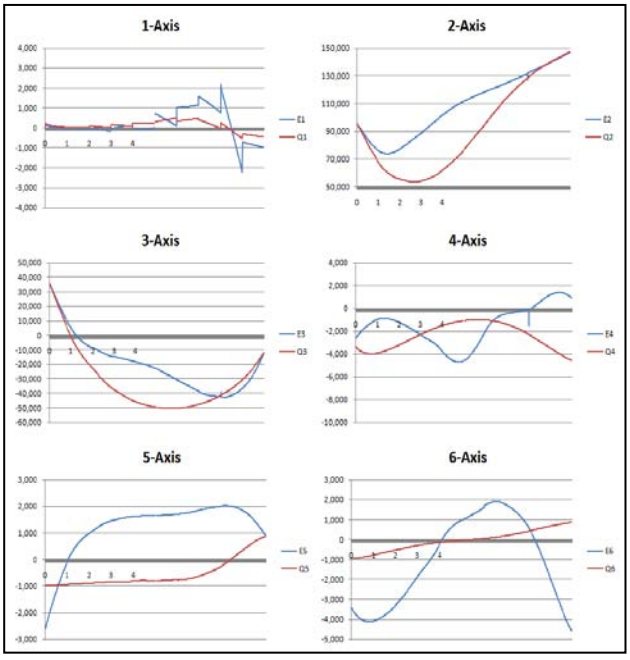


Fig. 9 Calculated torque profiles of each joint using RecurDyn

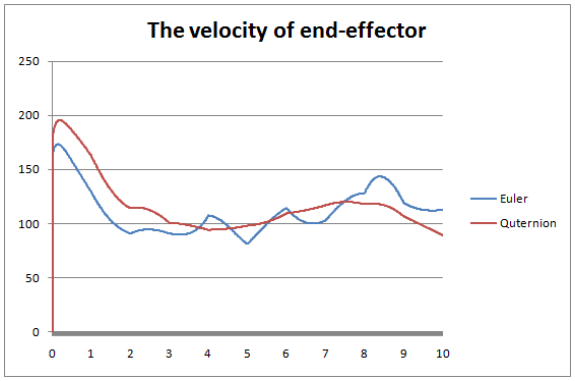


Fig. 10 The velocity of RS2 end effector

4 Implementation of *SLERP* on RS2 using LabVIEW® Motion Control

In order to implement the orientation control of both linear Euler interpolation and *SLERP* on RS2, we have developed their orientation interpolation programs based on LabVIEW® graphical program as shown in Fig. 11. This figure shows that the implementation of two orientation control flows on RS2 is organized in three parts. The first part is the orientation interpolation routines of both linear Euler interpolation and *SLERP* based on LabVIEW® graphical programs developed in previous section.

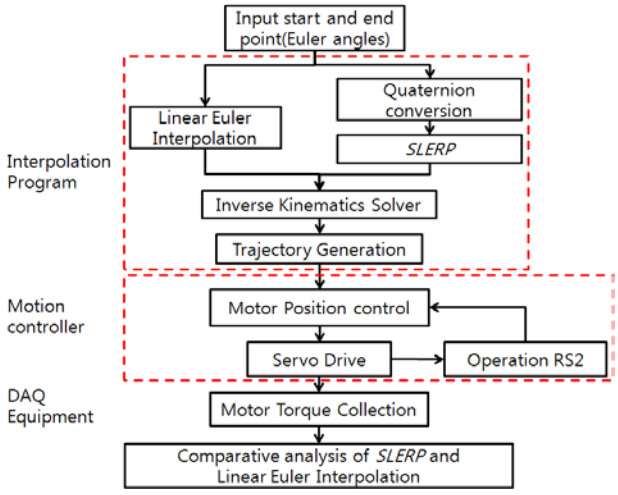


Fig. 11 Flow chart of implementation of two orientation controls on RS2

In the meanwhile, the second part is the position control routine through NI PXI-7350 Motion Controller with Mitsubishi J2 series servo drives and HC-MFC servo motors. Specifically RS2 can be controlled according to pulses sent by NI PXI-7350 Motion Controller. Finally, the third part is for collecting torque voltage of each servo motor (specifically servo drive) using NI PXI-6133 DAQ equipment. As shown

in Table 2, the maximum joint torques of SLERP are smaller than those of linear Euler interpolation, as expected.

Table 2 Maximum Joint Torque Voltage

(Unit: mV)

Joint	Euler	*SLERP*
1	170	150
2	590	500
3	640	630
4	800	750
5	160	150
6	350	340

5 Conclusion

In general, the orientation interpolation of industrial robots has been done based on Euler angle system which can result in singular point (so-called Gimbal Lock). However, it is well known that quaternion interpolation has the advantage of smooth orientation interpolation without Gimbal Lock. This paper presented the real application of quaternion interpolation, specifically Spherical Linear IntERPolation (in short, *SLERP*), to the orientation control of the 6-Axis articulated robot (RS2) using LabVIEW® and RecurDyn®. For the comparison of SLERP with linear Euler interpolation in view of smooth profiles of joint angles and torques, the two methods have been dynamically simulated on RS2 by using both LabVIEW® and RecurDyn®. Finally our original work, specifically the implementation of *SLERP* and linear Euler interpolation on the actual robot, *i.e.* RS2, has been done using LabVIEW® motion control tool kit. The *SLERP* orientation control was shown to be effective in terms of smooth joint motion and torque when compared to a conventional (linear) Euler interpolation.

6 Acknowledgement

The authors of this paper were partly supported by the Second Stage of Brain Korea21 Projects

7 References

[1] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, Robotics: Control, Sensing, Vision and Intelligence. , McGraw-Hill (1987), pp 22

[2] Hoag, D. Apollo Guidance and Navigation: Considerations of Apollo IMU Gimbal Lock. Tech. Rep. E-1344, MIT Instrumentation Laboratory, April 1963.

[3] Jones, E. M., and Fjeld, P. Gimbal Angles, Gimbal Lock, and a Fourth Gimbal for Christmas, Nov 2002. http://www.hq.nasa.gov/office/pao/History/alsj/gimbals.html

[4] Purwar A. Jin Z. Ge Q. J."Computer Aided Synthesis of Piecewise Rational Motions for Spherical 2R and 3R Robot Arms," Annual mechanisms and robotics conference, DETC2006 2006, pp.1209 - 1222 , 2006

[5] Ahlers, S.G., and McCarthy, J.M., 2000, ‘‘The Clifford Algebra of Double Quaternions and the Optimization of TS Robot Design,’’ Applications of Clifford Algebras in Computer Science and Engineering, E. Bayro and G. Sobczyk,eds., Birkhauser.

[6] Chung W.j, Kim K.J. Kim S.H."Steering Control Algorithm of an Inter-Block Locomotion Robot Using a Quaternion with Spherical Cubic Interpolation," Systemics cybernetics and informatics 2005, pp.374 - 379 , 2005

[7] Dam B. Erik, Koch Martin, Lillholm Martin: Quaternions, interpolation and animation, Technical report DIKU-TR9815, Department of Computer Science, University of Copenhagen, 1998

[8] J. S. Ahn, W. J. Chung, “OnDesign Prototype and Gain Optimization for Heavy Duty Handling Articulated Manipulator (HDHAM) with 6 DOF,” The 14th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2010, Volume 2 pp 174~179

[9] MITSUBISHI, General-Purpose Interface MR-J2S- A Servo Amplifier Instruction Manual.

[10] Herbert Goldstein, Classical Mechanics (2nd ed.), Reading, MA: Addison-Wesley, ISBN 978-0-201-02918-5

[11] J. S. Ahn, W. J. Chung, "A Study on 6-Axis Articulated Robot Using a Quaternion Interpolation," KSMTE of Spring Conference 2010, pp 294~300, 2010

[12] Bonev I.A, Zlatanov D, Gosselin C.M, "Advantages of the Modified Euler Angles in the Design and Control of PKMs,"Parallel kinematics seminar, Development methods and application experience of parallel kinematics 2002 , pp.171 - 188 , 2002

[13] Mebius J.E, “Derivation of the Euler-Rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations,” arXiv General Mathematics 2007.

[14] Ken Shoemake, "Animating Rotation with Quaternion CurvesK," In Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85), pp. 245-254.

A Recursive Dual Minimum Algorithm

Qi Zhu[1], **and Shaohua Tan**[2]
[1]Department of Computer Science, University of Houston - Victoria, Victoria, Texas, US 77901
[2]Department of Electrical Engineering, National University of Singapore, Singapore, Singapore 119260

Abstract— *The various RDM learning algorithms are developed by choosing different $\Lambda(\theta)$ for LIP models, including Projection, Recursive Dual Minimum, Recursive Dual Mean Minimum, λ-weighted Dual M<inimum, Instantaneous RDM and Batch RDM algorithms. An exmaple is shown the applications in adaptive identification and control fields.*

Keywords: Dual Minimum, High-order, Parameter Estimation, System Identification

1. Introduction

Least square algorithms are widely used in finding or estimating the numerical values of the parameters to fit a function to a set of data optimally and to characterize the statistical properties of estimates [2]. Traditionally least square algorithms are used to establish the mathematical framework of digital signal processing applications, such as communications, control, radar, and seismology [4], [9]. Recursive least squares or online procedures are more useful when parameters are identified from recurring in time, which can be used in a wide variety of real world problems when the model structure is well understood and input data becomes available at regular intervals of time, such as speech [7], vehicle mass estimation [5], structural damage assessment [3], etc.

In [11], we have developed a High-order Mixed L2-Linfty Estimation for LIP models under noiseless and noisy data. In this paper, we propose various RDM (*Recursive Dual Minimum*) learning algorithms by choosing different kinds of time-variant symmetric positive semi-definite forgetting factor matrix function $\Lambda(\theta)$. This paper is organized as follows. In Section 2, we first provide the general form of recursive DM (RDM) learning algorithm. Then, in Section 3, we further derive the various specialised RDM learning algorithms that include several very useful and interesting results, such as the projection learning algorithm, the recursive dual minimum learning algorithm, the recursive dual mean minimum learning algorithm, the λ-weighted recursive dual minimum learning algorithm, the instantaneous k-order RDM learning algorithm and the batch k-order RDM learning algorithm. In Section 4, we supply simulations of RDM learning algorithm in realistic industry applications. Finally, in Section 5, we conclude this paper and point out the future research directions.

2. General Form of RDM Learning Algorithm

2.1 Preliminaries

First we define an important integral function $\rho(t)$ as follows.

Definition $\rho(t)$ is defined to be a function from the set of non-negative integers, it satisfies the following two conditions:

(i)

$$0 = \rho(0) \leq \rho(1) \leq \rho(2) \leq \cdots \leq \rho(t) \leq \cdots \quad (1)$$

(ii) The number of elements of the set $t \triangleq \rho^{-1}(\rho(t))$ is uniformly bounded for all t. That is, there exists a positive integer N_ρ which

$$card(\rho^{-1}(\rho(t))) \leq N_\rho \quad \text{for all } t \quad (2)$$

where the function $card()$ represents the *cardinality* of the set.

(iii) The pseudometric $\rho(t_1, t_2)$ is defined as

$$\rho(t_1, t_2) \triangleq |\rho(t_1) - \rho(t_2)| \quad (3)$$

When $\rho(t)$ as defined in the above definition, then when $t \to \infty$, $\rho(t) \to \infty$. This can be very easily proved.

2.2 RDM Learning Algorithm

Using the function $\rho(t)$ defined in section 2.1, and the assumption that the number of data samples up to time t is $\rho(t)$. we have

$$\Lambda_t = \Lambda_{\rho(t)} = \begin{bmatrix} \lambda_{t1} & & & \\ & \lambda_{t2} & & \\ & & \ddots & \\ & & & \lambda_{t\rho(t)} \end{bmatrix} \quad (4)$$

where $\lambda_{ti} \geq 0$ for $i = 1, 2, \cdots, \rho(t)$.

$$Y_{\rho(t)} = [y_1, y_2, \cdots, y_{\rho(t)}]^T \quad (5)$$

$$\Phi_{\rho(t)} = [\varphi_1, \varphi_2, \cdots, \varphi_{\rho(t)}]^T \quad (6)$$

$$E_{\rho(t)} = [e_1, e_2, \cdots, e_{\rho(t)}]^T \quad (7)$$

$$P_t = \Phi^T_{\rho(t)} \Lambda_t \Phi_{\rho(t)} = \sum_{i=1}^{\rho(t)} \lambda_{ti} \varphi_i \varphi_i^T \quad (8)$$

$$Q_t = \Phi^T_{\rho(t)} \Lambda_t Y_{\rho(t)} = \sum_{i=1}^{\rho(t)} \lambda_{ti} y_i \varphi_i \quad (9)$$

$$R_t = Y^T_{\rho(t)} \Lambda_t Y_{\rho(t)} = \sum_{i=1}^{\rho(t)} \lambda_{ti} y_i^2 \quad (10)$$

$$J_t = \frac{1}{2} \theta_{t-1}^T P_t \theta_{t-1} - \theta_{t-1}^T Q_t + \frac{1}{2} R_t \quad (11)$$

Furthermore, we have the **General Form of RDM learning algorithm** as follow.

$$\theta_t = \theta_{t-1} + \frac{\alpha J_t (Q_t - P_t \theta_{t-1})}{\beta + (Q_t - P_t \theta_{t-1})^T (Q_t - P_t \theta_{t-1})} \quad (12)$$

where $\beta > 0, 0 < \alpha < 4, t = 1, 2, ..., \rho(t)$.

2.3 Properties

For RDM learning algorithm, Λ_t in (4) satisfies the condition that

$$\sum_{i=1}^{\rho(t)} \lambda_{ti} \leq M_\lambda \quad (13)$$

holds for all t where M_λ is a positive constant number.

Theorem 1 For any given initial value θ_0, the vector sequence θ_t generated in (12) has the following properties:

(i)

$$\|\theta_t - \theta^\star\| \leq \|\theta_0 - \theta^\star\| \quad (14)$$

(ii)

$$\lim_{t \to \infty} \frac{\alpha J_t}{\sqrt{\beta + (Q_t - P_t \theta_{t-1})^T (Q_t - P_t \theta_{t-1})}} = 0 \quad (15)$$

(iii)

$$\lim_{t \to \infty} \|\theta_t - \theta_{t-s}\| = 0 \quad \text{for any finite s.} \quad (16)$$

Proof. If we introduce a Lyapunov function V_t as

$$V_t = \tilde{\theta}_t^T \tilde{\theta}_t = (\theta_t - \theta^\star)^T (\theta_t - \theta^\star)$$

Then Theorem 1 can be carried out with the similar way in [10]. □

Theorem 2 For any set of noiseless data samples, RDM learning algorithm (12) globally converges decreasingly in the sense that

$$\lim_{t \to \infty} J_t = 0 \quad (17)$$

Proof. A similar proof in [10] can carry out this theorem. □

From (4) - (12), we can see that when Λ_t is computed recursively. We can also obtain some specialised RDM learning algorithms by various choices of λ_t.

3. Various Specialized RDM Learning Algorithms

In this section we will give several important RDM learning algorithms by choosing different kinds of time-variant symmetric non-negative definite matrix function Λ_t.

3.1 Projection Learning Algorithm

If we choose $\rho(t) = t$ and

$$\Lambda_t = \begin{bmatrix} 0_{t-1} & 0 \\ 0 & 1 \end{bmatrix} \quad (18)$$

where 0_{t-1} is the $(t-1)$th-order square zero matrix. In this case, RDM learning algorithm in (12) can reduce to the 1st-order learning algorithm as.

$$\theta_t = \theta_{t-1} + \frac{\alpha' e_t \varphi_t}{\beta + e_t^2 \varphi_t^T \varphi_t} (y_t - \varphi_t^T \theta_{t-1}) \quad (19)$$

where $t = 1, 2, \cdots, \beta > 0, 0 < \alpha' = \frac{1}{2} < 2$, and $e_t = y_t - \varphi_t^T \theta_{t-1}$.

Delete e_t^2 from both numerator and denominator from (19) and β is a any positive number, so we can obtain the Projection Learning Algorithm [1], [6] in adaptive control.

$$\theta_t = \theta_{t-1} + \frac{\alpha' \varphi_t}{\beta + \varphi_t^T \varphi_t} (y_t - \varphi_t^T \theta_{t-1}) \quad (20)$$

where $t = 1, 2, \cdots, \beta > 0, 0 < \alpha' = \frac{1}{2} < 2$.

Thus Projection learning algorithm is a special case of RDM algorithm when we choose a set of specific parameters. This Projection learning algorithm minimizes the cost function $J_t = \frac{1}{2} e_t^2$.

3.2 Recursive Dual Minimum Learning Algorithm

The conventional recursive least squares (RLS) algorithm is a powerful learning algorithm in adaptive control. But the algorithm is applied with the condition that Φ is of full rank, the convergence is very slow, and the computation is quite intensive. In this subsection, we propose the recursive Dual Minimum learning algorithm, which is free of full rank and has much less computation.

Choose $\rho(t) = t$ and $\Lambda_t = I_t$ where I_t is the tth-order identity matrix, then the Eqs. of θ_t and J_t as.

$$\theta_t = \theta_{t-1} + \frac{\alpha J_t T (Q_t - P_t \theta_{t-1})}{\beta + (Q_t - P_t \theta_{t-1})^T T (Q_t - P_t \theta_{t-1})} \quad (21)$$

$$J_t = \frac{1}{2} \theta_{t-1}^T P_t \theta_{t-1} - \theta_{t-1}^T Q_t + \frac{1}{2} R_t \quad (22)$$

where $\beta > 0, 0 < \alpha < 4, t = 1, 2, ...,$, and T is chosen to be a symmetric positive definite matrix.

P_t, Q_t and R_t defined in (8) - (10) have the recursive computation formulas as

$$\begin{array}{ll} P_t = P_{t-1} + \varphi_t\varphi_t^T & \text{with } P_0 = 0 \\ Q_t = Q_{t-1} + y_t\varphi_t & \text{with } Q_0 = 0 \\ R_t = R_{t-1} + y_t^2 & \text{with } R_0 = 0 \end{array} \quad (23)$$

This new Recursive Dual Minimum learning algorithm does not need to assume that Φ is of full rank and minimizes the cost function $J_t = \frac{1}{2}\sum_{i=1}^{t} e_i^2$, which is the same as the cost function of the conventional recursive least squares [11]. Note that the matrix $\Lambda_t = I_t$ does not satisfy the condition in (13), so we must choose a small symmetric positive definite matrix T to avoid the burst phenomenon. This technique for the Recursive Dual Minimum is effective in practice of identification and control.

3.3 Recursive Dual Mean Minimum Learning Algorithm

In the preceding subsection, we have discussed the Recursive Dual Minimum learning algorithm, which is a powerful learning algorithm in identification. However, since the trace of $\Lambda_t = I_t$ is not uniformly bounded, P_t, Q_t and R_t may become very large when t increases. In this subsection, we will propose a Recursive Minimum Mean Squares learning algorithm.

Choosing $\rho(t) = t$ and $\Lambda_t = \frac{1}{t}I_t$, the Recursive Minimum Mean Squares learning algorithm is:

$$\theta_t = \theta_{t-1} + \frac{\alpha J_t(Q_t - P_t\theta_{t-1})}{\beta + (Q_t - P_t\theta_{t-1})^T(Q_t - P_t\theta_{t-1})} \quad (24)$$

$$J_t = \frac{1}{2}\theta_{t-1}^T P_t\theta_{t-1} - \theta_{t-1}^T Q_t + \frac{1}{2}R_t \quad (25)$$

where $\beta > 0, 0 < \alpha < 4, t = 1, 2, ...$

P_t, Q_t and R_t defined in (8) - (10) have the recursive computation formulas as

$$\begin{array}{ll} P_t = \frac{t-1}{t}P_{t-1} + \frac{1}{t}\varphi_t\varphi_t^T & \text{with } P_0 = 0 \\ Q_t = \frac{t-1}{t}Q_{t-1} + \frac{1}{t}y_t\varphi_t & \text{with } Q_0 = 0 \\ R_t = \frac{t-1}{t}R_{t-1} + \frac{1}{t}y_t^2 & \text{with } R_0 = 0 \end{array} \quad (26)$$

The Recursive Dual Mean Minimum learning algorithm in $(25, 26)$ minimizes the cost function of mean of squares of errors $J_t = \frac{1}{2t}\sum_{i=1}^{t} e_i^2 \to 0$.

3.4 Recursive λ-weighted Dual Minimum Learning Algorithm

In adaptive control [1], [8], the conventional recursive least squares algorithm with forgetting index λ (named as λ-RLS algorithm) is another powerful learning algorithm. However, the algorithm is applied with the condition that Φ is of full rank and the computation is quite intensive. In this subsection, we propose a new Recursive λ-weighted Dual Minimum learning algorithm which is free of the full rank of Φ and has less computation.

Choose $\rho(t) = t$ and Λ_t to be

$$\Lambda_t = \begin{bmatrix} \lambda^{t-1} & & & \\ & \lambda^{t-2} & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \quad (27)$$

where $0 < \lambda < 1$. Then the Recursive λ-weighted Minimum Squares learning algorithm includes $(11, 12)$ and P_t, Q_t and R_t defined in (8) - (10) have the recursive computation formulas as

$$\begin{array}{ll} P_t = \lambda P_{t-1} + \varphi_t\varphi_t^T & \text{with } P_0 = 0 \\ Q_t = \lambda Q_{t-1} + y_t\varphi_t & \text{with } Q_0 = 0 \\ R_t = \lambda R_{t-1} + y_t^2 & \text{with } R_0 = 0 \end{array} \quad (28)$$

It is easy to prove that the matrix Λ_t defined in Eq. (27) satisfies the condition in (13) because

$$tr(\Lambda_t) = \sum_{i=1}^{t}\lambda^{t-i} = \frac{1-\lambda^t}{1-\lambda} \le \frac{1}{1-\lambda} \quad (29)$$

is uniformly bounded for all t. Thus, the Recursive λ-weighted Dual Minimum learning algorithm globally minimizes the cost function $J_t = \frac{1}{2}\sum_{i=1}^{t}\lambda^{t-i}e_i^2$.

3.5 Instantaneous k-order RDM Learning Algorithm

In this subsection, we derive a power instantaneous k-order dynamic RDM learning algorithm from Eq. (12) for LIP models when choosing specific Λ_t. This learning algorithm updates at every step when the system has one more new data sample.

Choosing $\rho(t) = t$ and

$$\Lambda_t = \begin{bmatrix} 0_{t-k} & 0 \\ 0 & \Lambda(t,k) \end{bmatrix} \quad (30)$$

where 0_{t-k} is the (t-k)th-order zero matrix. And $\Lambda(t,k)$ is an k-order symmetric non-negative matrix satisfying

$$tr(\Lambda(t,k)) \le M_\lambda \quad (31)$$

for all t for a positive constant number M_λ. One important matrix for Λ_t is when $\Lambda(t,k) = diag\{\lambda_{t-k+1}, \lambda_{t-k+2}, \cdots, \lambda_t\}$, then the Λ_t is

$$\Lambda_t = \begin{bmatrix} 0_{t-k} & 0 & \cdots & 0 \\ 0 & \lambda_{t-k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_t \end{bmatrix} \quad (32)$$

Our input and output matrices are with k rows from time $t-k+1$ to t.

$$\Phi_t = \Phi(t,k) = [\varphi_{t-k+1}, \varphi_{t-k+2}, \cdots, \varphi_t]^T \quad (33)$$

$$Y_t = Y(t,k) = [y_{t-k+1}, y_{t-k+2}, \cdots, y_t]^T \quad (34)$$

When $t < k$, λ_{t-k+1} to λ_0, φ_{t-k+1} to φ_0, and y_{t-k+1} to y_0 are arbitrarily set as the initial values.

P_t, Q_t and R_t defined in (8) - (10) have the recursive computation formulas as

$$\begin{aligned} P_t &= P(t,k) = \Phi_t^T \Lambda(t,k)\Phi_t = \Phi^T(t,k)\Lambda(t,k)\Phi(t,k) && (35)\\ Q_t &= Q(t,k) = \Phi_t^T \Lambda(t,k)Y_t = \Phi^T(t,k)\Lambda(t,k)Y(t,k) && (36)\\ R_t &= R(t,k) = Y_t^T \Lambda(t,k)Y_t = Y^T(t,k)\Lambda(t,k)Y(t,k) && (37)\\ J_t &= J(t,k) = \frac{1}{2}R_t - \theta_{t-1}^T Q_t + \frac{1}{2}\theta_{t-1}^T P_t \theta_{t-1} \\ &= \frac{1}{2}R(t,k) - \theta_{t-1}^T Q(t,k) + \frac{1}{2}\theta_{t-1}^T P(t,k)\theta_{t-1} && (38) \end{aligned}$$

Then, the **instantaneous k-order Recursive Dual Minimum** learning algorithm

$$\theta_t = \theta_{t-1} + \frac{\alpha J_t (Q_t - P_t\theta_{t-1})}{\beta + (Q_t - P_t\theta_{t-1})^T (Q_t - P_t\theta_{t-1})} \quad (39)$$

where $\beta > 0, 0 < \alpha < 4, t = 1, 2, ...,$ and it minimizes the cost function $J_t = J(t,k) = \frac{1}{2}\sum_{i=t-k+1}^{t} \lambda_i e_i^2$ to its global minimum.

3.6 Batch k-order RDM Learning Algorithm

The instantaneous k-order RDM learning algorithm introduced in the preceding subsection updates the parameter vector at every step based on the current input and output data and the last $k-1$ data. However, the batch k-order RDM learning algorithm developed in this subsection updates parameters at every k-step based on the last k data samples.

Choose $\rho(t) = kt$ and

$$\Lambda_t = \begin{bmatrix} 0_{kt-k} & 0 \\ 0 & \Lambda(kt,k) \end{bmatrix} \quad (40)$$

where 0_{kt-k} is the (kt-k)th-order zero matrix. And $\Lambda(kt,k)$ is an k-order symmetric non-negative matrix satisfying

$$tr(\Lambda(kt,k)) \le M_\lambda \quad (41)$$

M_λ is a positive constant and $\Lambda_t = diag\{\lambda_{kt-k+1}\lambda_{kt-k+2}, \cdots, \lambda_{kt}\}$, then the Λ_t is

$$\Lambda_t = \begin{bmatrix} 0_{kt-k} & 0 & \cdots & 0 \\ 0 & \lambda_{kt-k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{kt} \end{bmatrix} \quad (42)$$

We further introduce the following notations:

$$\begin{aligned} \Phi_t &= \Phi(kt,k) = [\varphi_{kt-k+1}, \varphi_{kt-k+2}, \cdots, \varphi_{kt}]^T && (43)\\ Y_t &= Y(kt,k) = [y_{kt-k+1}, y_{kt-k+2}, \cdots, y_{kt}]^T && (44)\\ P_t &= P(kt,k) = \Phi_t^T \Lambda(kt,k)\Phi_t \\ &= \Phi^T(kt,k)\Lambda(kt,k)\Phi(kt,k) && (45)\\ Q_t &= Q(kt,k) = \Phi_t^T \Lambda(kt,k)Y_t \\ &= \Phi^T(kt,k)\Lambda(kt,k)Y(kt,k) && (46)\\ R_t &= R(kt,k) = Y_t^T \Lambda(kt,k)Y_t \\ &= Y^T(kt,k)\Lambda(kt,k)Y(kt,k) && (47)\\ J_t &= J(kt,k) = \frac{1}{2}R_t - \theta_{t-1}^T Q_t + \frac{1}{2}\theta_{t-1}^T P_t \theta_{t-1} \\ &= \frac{1}{2}R(kt,k) - \theta_{t-1}^T Q(kt,k) + \frac{1}{2}\theta_{t-1}^T P(kt,k)\theta_{t-1} && (48) \end{aligned}$$

Then, **Batch k-order Recursive Dual Minimum** learning algorithm

$$\theta_t = \theta_{t-1} + \frac{\alpha J_t (Q_t - P_t\theta_{t-1})}{\beta + (Q_t - P_t\theta_{t-1})^T (Q_t - P_t\theta_{t-1})} \quad (49)$$

where $\beta > 0, 0 < \alpha < 4, t = 1, 2, ...,$ and it minimizes the cost function $J_t = J(kt,k) = \frac{1}{2}\sum_{i=kt-k+1}^{kt} \lambda_i e_i^2$ to its global minimum.

4. Case Study

In this section, we will present an example of industrial applications of various specialized RDM learning algorithms to show that the various RDM learning algorithms deduced are effective.

Example Suppose that we are analysts in the management services division of an accounting firm. One of the firm's clients is American Manufacturing Company, a major manufacturer of a wide variety of commercial and industrial products. American Manufacturing owns a large nine-building complex in Central City and heats this complex by using a modern coal-fueled heating system. In the past, American Manufacturing has encountered problems in determining the proper amount of coal to order each week to heat the complex adequately. Because of this, the firm has requested that the firm develop an accurate way to predict the amount of fuel (in tons of coal) that will be used to heat the nine-building complex in future weeks. The experience indicates that (1) weekly fuel consumption substantially depends on the average hourly temperature (in degrees Fahrenheit) during the week and (2) weekly fuel consumption also depends on factors other than average hourly temperature that contribute to an overall "chill factor". Some of these factors are:

1) Wind velocity (in miles per hour) during the week
2) "Cloud cover" during the week
3) Variations in temperature, wind velocity, and cloud cover during the week (perhaps caused by the movement of weather fronts).

In this example we use regression analysis to predict the *dependent variable* weekly fuel consumption y, on the basis of the *independent variable* average hourly temperature x. Then we will use additional independent variables, which measure the effects of factors such as wind velocity and cloud cover, to help us predict weekly fuel consumption. Suppose that we have gathered data concerning y and x for the $n = 8$ weeks prior to the current week. This data is given in Table 1. Here the letter i denotes the time order of a previously observed week, where x_i denotes the average hourly temperature and y_i denotes the fuel consumption that has been observed in week i. It should be noted that it would, of course, be better to have more than eight weeks of data. However, sometimes data availability is initially limited. Furthermore, we have purposely limited the amount of data to simplify subsequent discussions in this example.

Week, i	hourly temperature, x_i	fuel consumption, y_i
1	x_1 = 28.0	y_1 = 12.4
2	x_2 = 28.0	y_2 = 11.7
3	x_3 = 32.5	y_3 = 12.4
4	x_4 = 39.0	y_4 = 10.8
5	x_5 = 45.9	y_5 = 9.4
6	x_6 = 57.8	y_6 = 9.5
7	x_7 = 58.1	y_7 = 8.0
8	x_8 = 62.5	y_8 = 7.5

Table 1

FUEL CONSUMPTION DATA OF EXAMPLE 1

.

To develop a regression model describing the fuel consumption data, we first consider the fifth week in Table 1 (for the purposes of our discussion we could consider any particular week). In the fifth week the average hourly temperature was $x_5 = 45.9$, and the fuel consumption was $y_5 = 9.4$. If we were to observe another week having the same average hourly temperature of 45.9, we might well observe a fuel consumption that is different from 9.4. This is because factors other than average hourly temperature - factors such as average hourly wind velocity and average hourly thermostat setting - affect weekly fuel consumption. Therefore although two weeks might have the same average hourly temperature of $x_5 = 45.9$, there could be a lower average hourly wind velocity and thus a smaller fuel consumption in one such week than in the other week. It follows that there is an infinite population of potential weekly fuel consumptions that could be observed when the average hourly temperature is $x_5 = 45.9$.

To generalize the proceeding discussion, consider all eight fuel consumptions in Table 1. For $i = 1, 2, ..., 8$ we may express y_i in the form

$$y_i = \theta_0 + \theta_1 x_i + e_i \tag{50}$$

Here, e_i is the error term describes the effect on y_i of all factors that have occurred in the ith week other than the average hourly temperatures x_i.

When we plot the eight fuel consumptions against the eight average hourly temperatures. Note that the fuel consumptions tend to decrease in a straight-line fashion as the temperatures increase. The θ_0 is the *y-intercept* of the straight line, and θ_1 is the *slope* of the straight line. To interpret the meaning of the y-intercept, θ_0, assume that $x_i = 0$. Then

$$\theta_0 + \theta_1 x_i = \theta_0 + \theta_1 \times 0 = \theta_0$$

So the θ_0 is the mean weekly fuel consumption for all potential weeks having an average hourly temperature of 0^oF. To interpret the meaning of the slope, θ_1, consider two different weeks. Suppose that for the first week average hourly temperature is c. The mean weekly fuel consumption for all such potential weeks is

$$\theta_0 + \theta_1 \times c$$

For the second week, suppose that the average hourly temperature is $c + 1$. The mean weekly fuel consumption for all such potential weeks is

$$\theta_0 + \theta_1 \times (c + 1)$$

The difference between these mean weekly fuel consumptions is

$$[\theta_0 + \theta_1 \times (c + 1)] - [\theta_0 + \theta_1 \times c] = \theta_1$$

Thus the slope θ_1 is the change in the mean weekly fuel consumption that is associated with a 1-degree increase in average hourly temperature.

Then we refer to the equation (50) as the *simple linear* (or *straight-line*) *regression model* relating y_i to x_i in this example, and θ_0, θ_1 is the *parameters* of the model. Note that y_i is assumed to be randomly selected from the infinite population of potential values of the dependent variable that could be observed when the value of the independent variable x is x_i.

Let's use RDM learning algorithm, due to the finite data, we use them repeatedly in our recursive algorithm. When we choose $k = 4$ and set the initial parameter values $[\theta_0 \quad \theta_1] = [1 \quad 1]$ and the error bound 0.3, then we can get that $\theta_0 = 15.6631$ and $\theta_1 = -0.1243$. Fig. 1 - Fig. 3 show the results of the simulations using RDM learning algorithm. Then using the values of θ_0 and θ_1, we can predict the amount of fuel for the nine-building complex of American Manufacturing Company in future weeks.

5. Conclusion

In this paper, we have developed the various RDM learning algorithms by choosing different $\Lambda(\theta)$, and obtained several useful recursive learning algorithms for LIP models, such as Projection, Recursive Dual Minimum, Recursive Dual Mean Minimum, λ-weighted Dual Minimum, Instantaneous RDM and Batch RDM, etc. We have shown the

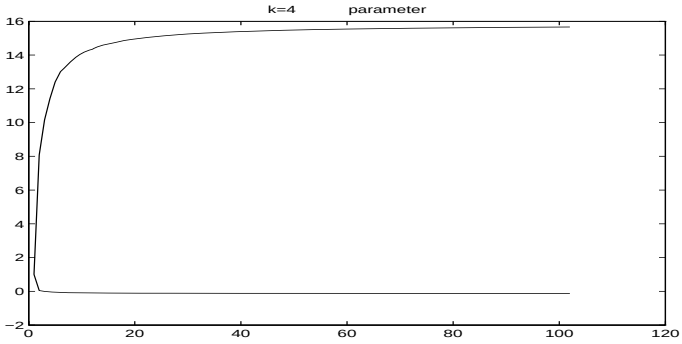


Fig. 1

PARAMETERS CONVERGENCE OF EXAMPLE 1 USING RECURSIVE DUAL MINIMUM LEARNING ALGORITHM WHEN $k = 4$.

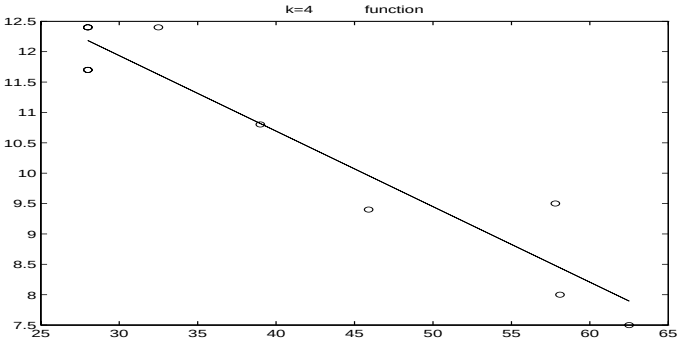


Fig. 3

FITTING CURVE OF EXAMPLE 1 USING RECURSIVE DUAL MINIMUM LEARNING ALGORITHM.

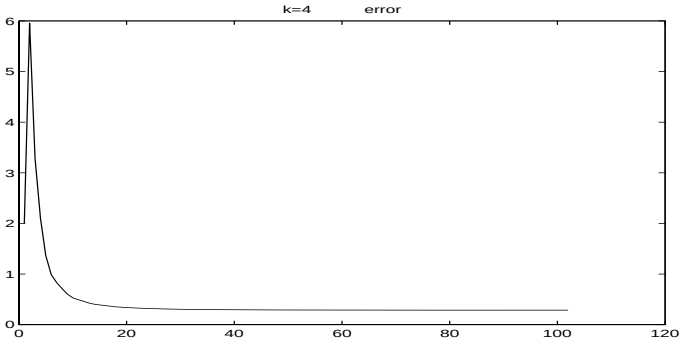


Fig. 2

ERROR CONVERGENCE OF EXAMPLE 1 USING RECURSIVE DUAL MINIMUM LEARNING ALGORITHM. ERROR WILL DROP BELOW 0.3 AFTER 101 STEPS.

effectiveness of them by a simulation example in adaptive identification and control fields.

Compared with other training methods, RDM learning method has several distinct features. It can avoid the *windup* and *burst* phenomena which are the crucial drawbacks for the correspondent conventional learning algorithms. And the k-order RDM has a faster training speed than the conventional ones when the k is chosen appropriately, the various new RDM algorithms can be successfully applied in adaptive controller design.

References

[1] K.J. Astrom and B. Wittenmark, "Adaptive Control." Addison-Wesley, Mass., 1989.

[2] A. Bjorck, "Numerical Method for Least Squares Problems." Society for Industrial and Applied Mathematics (SIAM), 1996.

[3] S. Chu, S. Lo, "Application of the On-line Recursive Least-squares Method to Perform Structural Damage Assessment." *Journal of Structural Control Health Monitoring.* doi: 10.1002/stc.362, 2009.

[4] A.A. Giordano, F.M. Hsu, "Least Square Estimation with Applications to Digital Signal Processing." John Wiley & Sons, 1985.

[5] H.K. Fathy; K. Dongsoo; J.L. Stein, "Online vehicle mass estimation using recursive least squares and supervisory data extraction," *American Control conference*, pp. 1842-1848, 2008.

[6] G. C. Goodwin and K. S. Sin, "Adaptive Filtering Prediction and Control." Prentice-Hall, Englewood Cliffs, N.J., 1984.

[7] A. Maddi, A. Guessoum, D. Berkani, "Application the recursive extended least squares method for modelling a speech signal." *Proc. of the 2nd ISCCP*, 2006.

[8] K. S. Narendra and A. M. Annaswamy, "Stable Adaptive Systems." Prentice Haall, 1989.

[9] Y. Nievergelt, "A Tutorial History of Least Squares with Applications to Astronomy and Geodesy." *Journal of Computational and Applied Mathematics*, 121(1), pp. 37-72, 2000.

[10] Q. Zhu, S. Tan, and Y. Qiao, "A High-order Recursive Quadratic Algorithm of Linear-in-the-Parameter Models," *Journal of Circuit, Systems, and Computers*, Vol. 17, No. 2, pp. 1-28, 2008.

[11] Q. Zhu, Ying Qiao, and S. Tan, "A Robust High-order Mixed L2-Linfty Estimation for Linear-in-the-Parameters Models," *Journal of Scientific Computing*, Vol 38, No 2, pp. 185-206, 2009.

B-spline solution of linear hyperbolic partial differential equations

Nazan Caglar[1], Hikmet Caglar[2], and Durmus Dundar[1]
[1]Faculty of Economic and Administrative Science, Istanbul Kultur University, 34156 Atakoy Istanbul, Turkey
[2]Department of Mathematics - Computer, Istanbul Kultur University, 34156 Atakoy Istanbul, Turkey

Abstract— *Second-order linear hyperbolic equations are solved by using B-spline method . The numerical solution of the equations are discussed and illustrated with an example. Numerical results reveal that B-spline method is implemented and effective.*

Keywords: Second-order linear hyperbolic equations;Finite difference;B-spline functions;Boundary conditions.

1. Introduction

We consider the second-order linear hyperbolic equation:

$$\begin{aligned} &u_{tt}(x,t) + 2\alpha u_t(x,t) + \beta^2 u(x,t) = \\ &u_{xx}(x,t) + f(x,t), \quad x \in (a,b), \quad t > 0 \end{aligned} \tag{1}$$

with initial conditions $u(x,0) = \Phi(x), \quad u_t(x,0) = \Psi(x)$ and boundary conditions $u(a,t) = g_1(t), \quad u(b,0) = g_2(t),$ where α and β are constants.

The equation above represents a damped wave equation and a telegraph equation, the existence and approximations of the solutions investigated in literature. In recent years, many research has been done in developing and implementing modern high resolutions methods for the numerical solution of the second-order linear hyperbolic equation(1).In recent years, many research has been done in developing and implementing modern high resolutions methods for the numerical solution of the second-order linear hyperbolic equation(1), see$[8-15]$. Recently, Gao and Chi$[8]$ proposed two semi-discretion methods to solve the one-space dimensional linear hyperbolic equation(1). Also, Huan-Wen Liu and Li-Bin Liu solved$[8]$ linear hyperbolic equation. In this paper, we propose a B-spline difference scheme to solve the linear hyperbolic equation(1).

The present paper will focus on a new method of solution of the linear hyperbolic equation by using third degree B-spline functions. The theory of spline functions is a very active field of approximation theory and boundary value problems (BVPs), when numerical aspects are considered. In a series of paper by Caglar et al. [2-7] BVPs of order two, third, fourth and fifth were solved using third, fourth and sixth-degree splines.

We propose B- spline difference scheme to solve the linear hyperbolic equation(1). The numerical results obtained by using the method described in this study give acceptable results. We have concluded that numerical results converge to the exact solution when k goes to zero and for smaller h the maximum absolute error decreased. In this paper , we have derived a new method based on B- splines for solution (1). In Section 2 , we give a brief derivation of B-spline function. In Section 3, the method are used to analysis to solution of problem (1). In Section 4, some numerical result, that are illustrated using MATLAB 6.5, are given to clarify the method. Finally, in Section 5 ends this paper with a brief conclusion.

2. The third-degree B-splines

In this section, third-degree B-splines are used to construct numerical solutions to the hyperbolic equations discussed in sections 3 and 4. A detailed description of B-spline functions generated by subdivision can be found in [1]. Consider equally-spaced knots of a partition $\pi : a = x_0 < x_1 < ... < x_n = b$ on [a,b]. Let $S_3[\pi]$ be the space of continuously-differentiable, piecewise, third-degree polynomials on π. That is, $S_3[\pi]$ is the space of third-degree splines on π. Consider the B-splines basis in $S_3[\pi]$. The third-degree B-splines are defined as

$$B_0(x) = \frac{1}{6h^3} \begin{cases} x^3 & 0 \le x < h \\ -3x^3 + 12hx^2 - 12h^2x + 4h^3 & h \le x < 2h \\ 3x^3 - 24hx^2 + 60h^2x - 44h^3 & 2h \le x < 3h \\ -x^3 + 12hx^2 - 48h^2x + 64h^3 & 3h \le x < 4h \end{cases}$$

$$B_{i-1}(x) = B_0(x - (i-1)h), \ i = 2, 3, ...,$$

To solve hyperbolic equation, B_i , B_i' *and* B_i'' evaluated at the nodal points are needed. Their coefficients are summarized in Table 1.

Table 1
VALUES OF B_i , B_i' *and* B_i''

	x_i	x_{i+1}	x_{i+2}	x_{i+3}	x_{i+4}
B_i	0	1/6	4/6	1/6	0
B_i'	0	$3/6h$	$0/6h$	$-3/6h$	0
B_i''	0	$6/6h^2$	$-12/6h^2$	$6/6h^2$	0

3. B-spline solutions for hyperbolic equation

In this section the B-spline method for solving hyperbolic equation is outlined, which is based on the collocation

approach[8]. We seek a function $S(x)$ that approximates the solution of hyperbolic equation(1), may be represented as

$$S(x) = \sum_{j=-3}^{n-1} C_j B_j(x), \tag{3}$$

where C_i are unknown real coefficients and $B_j(x)$ are third-degree B-spline functions. Let $x_0, x_1, ..., x_n$ be $n+1$ grid points in the interval $[a, b]$, so that

$x_i = a + ih, i = 0, 1, ..., n; x_{0=}a, x_n = b, h = (b-a)/n.$

We consider the equation (1),

difference schemes for this problem considered as following:

$$\frac{u_{i+1}-2u_i+u_{i-1}}{\Delta t^2} + 2\alpha\frac{u_i-u_{i-1}}{\Delta t} + \beta^2 u = \frac{\partial^2 u}{\partial x^2} + f(x,t), \tag{4}$$

where $\Delta t = k$

$$-u_{i+1}^{''} + (\tfrac{1}{k^2}+\beta^2)u_{i+1} = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_i + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_{i-1} + f(x,t), \tag{5}$$

and the initial conditions are given in (8)-(9)

$$u(x,0) = \phi(x) = u_0, u(k,x) = u_1, \tag{6}$$

$$u_t(x,0) = \psi(x) = (u_1 - u_0)/k, \tag{7}$$

$$u_1 = u_0 + k\psi(x). \tag{8}$$

Subsituting (6-8) in (5) then is obtained as follows

$$t = 2k \quad -u_2^{''} + (\tfrac{1}{k^2}+\beta^2)u_2 = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_1 + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_0 + f(x,t), \tag{9}$$

$$t = 3k \quad -u_3^{''} + (\tfrac{1}{k^2}+\beta^2)u_3 = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_2 + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_1 + f(x,t), \tag{10}$$

$$\begin{array}{cc} . & . \\ . & . \\ . & . \end{array}$$

$$t = nk \quad -u_n^{''} + (\tfrac{1}{k^2}+\beta^2)u_n = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_{n-1} + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_{n-2} + f(x,t), \tag{11}$$

The approximate solution of the equation (9)-(11) are sought in the form of the B-spline functions $S(x)$, it follows that

$$t = 2k \quad -S_2^{''} + (\tfrac{1}{k^2}+\beta^2)S_2 = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_1 + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_0 + f(x,t), \tag{12}$$

$$t = 3k \quad -S_3^{''} + (\tfrac{1}{k^2}+\beta^2)S_3 = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_2 + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_1 + f(x,t), \tag{13}$$

$$\begin{array}{cc} . & . \\ . & . \\ . & . \end{array}$$

$$t = nk \quad -S_n^{''} + (\tfrac{1}{k^2}+\beta^2)S_n = (\tfrac{2}{k^2}-\tfrac{2\alpha}{k})u_{n-1} + (\tfrac{2\alpha}{k}-\tfrac{1}{k^2})u_{n-2} + f(x,t), \tag{14}$$

and boundary conditions

$$\sum_{j=-3}^{n-1} C_j B_j(x) = g_1(t) \text{ for } x = 0, \tag{15}$$

$$\sum_{j=-3}^{n-1} C_j B_j(x) = g_2(t) \text{ for } x = 1, \tag{16}$$

The spline solution of eq.(12) with the boundary conditions are obtained by solving to the following matrix equation. The value of spline functions at the knots $\{x_i\}_{i=0}^{n}$ are determined using Table 1. Then the B-spline method in matrix form can be written as follows

$$AC = F$$

where

$$C = [\ \mathrm{C}_{-3}\ ,\ C_{-2}\ ,\ C_{-1}\ ,\ \ldots\ ,\ C_{n-3}\ ,\ C_{n-2}\ ,C_{n-1}\ ]^T,$$

$$F = \begin{bmatrix} g_1(2k) \\ (\frac{2}{k^2}-\frac{2\alpha}{k})u_1(x_0) + (\frac{2\alpha}{k}-\frac{1}{k^2})u_0(x_0) + f(2k,x_0) \\ (\frac{2}{k^2}-\frac{2\alpha}{k})u_1(x_1) + (\frac{2\alpha}{k}-\frac{1}{k^2})u_0(x_1) + f(2k,x_1) \\ . \\ . \\ . \\ (\frac{2}{k^2}-\frac{2\alpha}{k})u_1(x_{n-1}) + (\frac{2\alpha}{k}-\frac{1}{k^2})u_0(x_{n-1}) + f(2k,x_{n-1}) \\ (\frac{2}{k^2}-\frac{2\alpha}{k})u_1(x_n) + (\frac{2\alpha}{k}-\frac{1}{k^2})u_0(x_n) + f(2k,x_n) \\ g_2(2k) \end{bmatrix}$$

Also the matrix A can be writen as

$$A = \begin{bmatrix} \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & 0 & 0 & ... & 0 \\ \varphi_1 & \varphi_2 & \varphi_3 & 0 & 0 & ... & 0 \\ 0 & \varphi_1 & \varphi_2 & \varphi_3 & 0 & ... & 0 \\ . & . & . & & & . & . \\ . & . & . & & & . & . \\ . & . & . & & & . & . \\ 0 & 0 & ... & \varphi_1 & \varphi_2 & \varphi_3 & 0 \\ 0 & 0 & 0 & ... & \varphi_1 & \varphi_2 & \varphi_3 \\ 0 & 0 & 0 & ... & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{bmatrix},$$

where

$\varphi_1 = -\frac{1}{h^2} + (\frac{1}{k^2} + \beta^2)\frac{1}{6},$

$\varphi_2 = \frac{12}{6h^2} + (\frac{1}{k^2} + \beta^2)\frac{4}{6},$

$\varphi_3 = -\frac{6}{6h^2} + (\frac{1}{k^2} + \beta^2)\frac{1}{6}.$

It is easy to see that, the same approximation can be applied the other equations (13)-(14).

4. Numerical results

In this section, the method discussed in section 2 and 3 is tested on the following problem from the literature[9], and the maximum absolute errors in the analytical solutions are calculated. Also we compare our results with Liu et all[4] and Mahonty[14] in Table 3-4. Our methods has its own advantages, once the solution has been simple algorithm and computational. All computations were carried out using MATLAB 6.5.

Example: We consider the following equation[9]

$u_{tt}(x,t) + 2u_t(x,t) + \beta^2 u(x,t) = u_{xx}(x,t) + (4 - 4\alpha + \beta^2 + h^2)e^{-2t}sinhx,$

$\alpha > \beta \geq 0$, $x \in (a,b), \quad t > 0$

with initial conditions

$u(x,0) = sinhx, \quad u_t(x,0) = -2sinhx$

and boundary conditions

$u(0,t) = 0, \quad u(1,0) = e^{-2t}sinh$

Table 2

ABSOLUTE ERRORS OF B-SPLINE SOLUTION

h	$k = 0.01$	$k = 0.001$	$k = 0.0001$
$\frac{1}{16}$	9.6419e-04	9.8278e-05	9.8466e-06
$\frac{1}{32}$	4.8150e-04	4.9062e-05	4.9155e-06
$\frac{1}{64}$	2.4035e-04	2.4490e-05	2.4536e-06
$\frac{1}{121}$	1.2710e-04	1.2951e-05	1.2976e-06
$\frac{1}{521}$	2.9515e-05	3.0074e-06	3.0131e-07

Table 3

ABSOLUTE ERRORS OF B-SPLINE SOLUTION AND COMPARE WITH THE FINITE DIFFERENCE SCHEME

h	$t = 1$	$t = 2$	$t = 1$	t=2
	finite difference		B-spline	
$\frac{1}{16}$	0.6386e-02	0.5937e-02	0.95892e-03	0.75376e-03
$\frac{1}{32}$	0.2229e-02	0.1800e-02	0.48085e-03	0.37565e-03
$\frac{1}{64}$	0.6002e-03	0.4826e-03	0.24004e-03	0.18798e-03

The exact solution of the above problem is $u(x,t) = e^{-2t}sinhx$. The observed maximum absolute errors for different values of step size h and k are given in Table 2 . for $= 50, \beta = 5$. Also numerical results are shown in Fig. 1. The maximum absolute errors at t=1,2 for h=1/16,1/32,1/64 are tabulated in tables 3-4.

5. Conclusions

In this paper, a family of B-spline methods has been considered for the numerical solution of the hyperbolic equation. The third-degree B-spline was tested on hyperbolic equation and the maximum absolute errors have tabulated. The results showed that the present method is an applicable technique and approximates the exact solution very well.

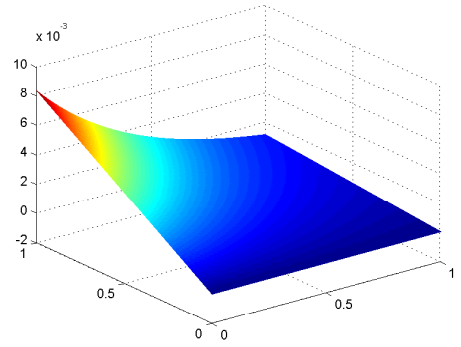


Fig. 1

RESULTS FOR $n = 121, k = 0.0001$.

References

[1] C. de Boor, A Practical Guide to Splines, 108,Springer-Verlag, New York, 1978.

[2] H. Caglar, M. Ozer, N.Caglar, The numerical solution of the one-dimensional heat equation by using third degree B-spline functions, Chaos Solitons Fract., 38, 1197Ű1201(2008).

[3] Caglar HN, Caglar SH, Twizell EH. The numerical solution of third-order boundary-value problems with fourth-degree B-spline functions. Int J Comput Math ,71,373Ű81(1999).

[4] Caglar HN, Caglar SH, Twizell EH. The numerical solution of fifth-order boundary-value problems with sixth-degree B-spline functions. Appl Math Lett 12,25Ű30(1999).

[5] Caglar N, Caglar H, Cagal B. Spline solution of nonlinear beam problems. J Concrete Appl Math 1(3),253Ű9(2003).

[6] Caglar H, Caglar N, Elfaituri K. B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to two-point boundary value problems. Appl Math Comput, 175,72Ű9(2006).

[7] Caglar N, Caglar H. B-spline solution of singular boundary value problems. Appl Math Comput 182,1509Ű13(2006).

[8] F.Gao, C.M.Chi, Unconditionally stable difference schemes for a one-space-dimensional linear hyperbolic equation, Applied Mathematics and Computation ,187(2),1272-1276(2007).

[9] H-W Liu,L-B Liu , An unconditionally stable spline difference schema of $O(k^2 + h^4)$ for solving the second-order 1D Linear hyperpolic equation, Mathematical and Computer Modelling 49, 1985-1993(2009).

[10] R.K.Mohanty,M.K.Jain, K.George, On the use of high order difference methods for the system of one space second order non-linear hyperbolic equations with variable coefficients, Journal of Computational and Applied Mathematics 72(2),421-431(1996).

[11] R.K.Mahonty, M.K.Jain, An unconditionally stable alternating direction implicit scheme for the two space dimensional linear hyperbolic equation, Numerical Methods for Partial Differntial Equations 17(6),684-688(2001).

[12] R.K.Mahonty, M.K.Jain, U Arora, An unconditionally stable ADI method for the linear hyperbolic equation in three space dimensional, International Journal of Computer Mathematics 79(1),133-142(2002).

[13] R.K.Mahonty, U Arora, A new discretization method oforder-four for the numerical solution of one-space dimensional second-order quasi-linear hyperbolic equation, International Journal of Mathematical Education in Science and Technology 33(6),829-838(2002).

[14] R.K.Mahonty, An unconditionally stable difference scheme for the one-space dimensional linear hyperbolic equation, Applied Mathematics Letters 17(1),101-105(2004).

[15] E.H.Twizel, An explicit difference method for the wave equation with extended stability range, BIT Numerical Mathematics 19(3),378-383(1979).

Third-Degree B-spline solution for a nonlinear diffusion Fisher's equation

Nazan Caglar[1], Hikmet Caglar[2], and Muge Iseri[1]
[1]Faculty of Economic and Administrative Science, Istanbul Kultur University, 34156 Atakoy Istanbul, Turkey
[2]Department of Mathematics - Computer, Istanbul Kultur University, 34156 Atakoy Istanbul, Turkey

Abstract—*Non-linear Fisher's equations are solved by using a spline method based on B-spline in the space direction and finite difference schema in the time direction. Numerical results reveal that spline method based on B-spline is implemented and effective.*

Keywords: Non-linear Fisher's equations;Finite difference;B-spline functions;Boundary conditions.

1. Introduction

We consider the generalized Fisher's equation:

$$u_t(x,t) = u_{xx}(x,t) + \alpha u(x,t)(1-u^{\beta}(x,t))$$
$$a < x < b, \qquad t > 0 \tag{1}$$

with initial condition

$$u(x,0) = \Phi(x),$$

and boundary conditions

$$u(a,t) = g_1(t), \quad u(b,t) = g_2(t)$$

where α and β are constants.

The classic and simplest case of the nonlinear reaction-diffusion equation is when β=1. It was suggested by Fisher as a deterministic version of a stochastic model for the spatial spread of a favored gene in a population [8].

$$u_t(x,t) = u_{xx}(x,t) + \alpha u(x,t)(1-u(x,t))$$
$$a < x < b, \qquad t > 0 \tag{2}$$

This equation is referred to as the Fisher equation, the discovery, investigation and analysis of traveling waves in chemical reactions was first presented by Luther [9]. In the last century, the Fisher's equation has became the basis for a variety of models for spatial spread, for example, in logistic population growth models $[10-11]$, flame propagation $[12-13]$, neurophysiology [14], autocatalytic chemical reactions $[15-17]$, branching Brownian motion processes [18], gene-culture waves of advance [19], the spread of early farming in Europe $[20-21]$, and nuclear reactor theory [22]. It is incorporated as an important constituent of nonscalar models describing excitable media, e.g., the Belousov-Zhabotinsky reaction [23]. In chemical media, the function $u(x,t)$ is the concentration of the reactant and the positive constant α represents the rate of the chemical reaction. In media of other natures, u might be temperature or electric potential.

The mathematical properties of equation (1) have been studied extensively and there have been numerous discussions in the literature. The most remarkable summaries have been provided by Brazhnik and Tyson [24]. One of the first numerical solutions was presented in literature with a pseudo-spectral approach. Implicit and explicit finite differences algorithms have been reported by different authors such as Parekh and Puri and Twizell et al. A Galerkin finite element method was used by Tang and Weber whereas Carey and Shen [25] employed a least-squares finite element method. A collocation approach based on WhittakerŠs sinc interpolation function [26] was also considered in [27]. Our solution based on B-spline method. In this paper, we propose a spline difference scheme to solve eq. (2).

The theory of spline functions is a very active field of approximation theory and boundary value problems (BVPs), when numerical aspects are considered. In a series of paper by Caglar et al. [2-7] BVPs of order two, third, fourth and fifth were solved using third, fourth and sixth-degree splines.

The paper is organized as follows: B-spline function is described in Section 2 briefly. In sections 3 the methods of solution of equation (2) is presented. In section 4 some numerical result, that are illustrated using MATLAB 7.0, are given to clarify the method. Concluding remarks are given in Section 5.

2. The third-degree B-splines

In this section, third-degree B-splines are used to construct numerical solutions to the Fisher equations discussed in sections 3 and 4. A detailed description of B-spline functions generated by subdivision can be found in [1].

Consider equally-spaced knots of a partition $\pi : a = x_0 < x_1 < ... < x_n = b$ on [a,b]. Let $S_3[\pi]$ be the space of continuously-differentiable, piecewise, third-degree

polynomials on π. That is, $S_3[\pi]$ is the space of third-degree splines on π. Consider the B-splines basis in $S_3[\pi]$. The third-degree B-splines are defined as

$$B_0(x) = \frac{1}{6h^3}\begin{cases} x^3 & 0 \le x < h \\ -3x^3 + 12hx^2 - 12h^2x + 4h^3 & h \le x < 2h \\ 3x^3 - 24hx^2 + 60h^2x - 44h^3 & 2h \le x < 3h \\ -x^3 + 12hx^2 - 48h^2x + 64h^3 & 3h \le x < 4h \end{cases} \quad (5)$$

$$B_{i-1}(x) = B_0(x-(i-1)h),\ i = 2, 3, ...$$

To solve hyperbolic equation, B_i , B_i' *and* B_i'' evaluated at the nodal points are needed. Their coefficients are summarized in Table 1.

Table 1
VALUES OF B_i , B_i' *and* B_i''

	$\mathbf{x}_i$	$\mathbf{x}_{i+1}$	$\mathbf{x}_{i+2}$	$\mathbf{x}_{i+3}$	$\mathbf{x}_{i+4}$
B_i	0	1/6	4/6	1/6	0
B_i'	0	-3/6h	0/6h	3/6h	0
B_i''	0	$6/6h^2$	$-12/6h^2$	$6/6h^2$	0

3. B-spline solutions for the nonlinear diffusion Fisher's equation

In this section a spline method for solving the Fisher equation is outlined, which is based on the collocation approach [5]. Let

$$S(x) = \sum_{j=-3}^{n-1} C_j B_j(x) \quad (6)$$

be an approximate solution of Eq.(1), where C_i are unknown real coefficients and $B_j(x)$ are third-degree B-spline functions. Let $x_0, x_1, ..., x_n$ be $n+1$ grid points in the interval [a,b], so that

$x_i = a + ih$, i $= 0, 1, ..., n$; $x_{0=}a, x_n = b, h = (b-a)/n$.

We consider the convection-diffusion equation (1),

difference schemes for this problem considered as following:

$$\frac{u_{i+1}-u_i}{\Delta t} - \frac{\partial^2 u}{\partial x^2} = u(1-u) + f(x,t) \quad (7)$$

where $\Delta t = k$

$$-ku_{i+1}^{''} + u_{i+1} - ku_{i+1}(1-u_{i+1}) = u_i + kf(x,t) \quad (8)$$

and the initial condition is given in (2)

$$u(x,0) = f(x) = u_0, \quad (9)$$

Subsituting (9) in (8) then is obtained as follows

$$t = 0+k$$
$$-ku_1^{''} + u_1 - ku_1(1-u_1) = u_0 + kf(x,k) \quad (10)$$
$$t = 0+2k$$
$$-ku_2^{''} + u_2 - ku_2(1-u_2) = u_1 + kf(x,k) \quad (11)$$
$$\vdots \qquad \vdots$$
$$t = 0+nk$$
$$-ku_n^{''} + u_n - ku_n(1-u_n) = u_{n-1} + kf(x,k) \quad (12)$$

The approximate solution of the equation (10)-(12) are sought in the form of the B-spline functions $S(x)$, it follows that

$$t = 0+k$$
$$-kS_1^{''} + S_1 - kS_1(1-S_1) = u_0 + kf(x,k) \quad (13)$$
$$t = 0+2k$$
$$-kS_2^{''} + S_2 - kS_2(1-S_2) = u_1 + kf(x,k) \quad (14)$$
$$\vdots \qquad \vdots$$
$$t = 0+nk$$
$$-kS_n^{''} + S_n - kS_n(1-S_n) = u_{n-1} + kf(x,k) \quad (15)$$

and boundary conditions (3)-(4) can be written as follows

$$\sum_{j=-3}^{n-1} C_j B_j(0) = 0 \text{ for } x = 0, \quad (16)$$

$$\sum_{j=-3}^{n-1} C_j B_j(1) = 0 \text{ for } x = 1\ , \quad (17)$$

The spline solution of eq.(13) with the boundary conditions is obtained by solving to the following matrix equation[see 2,4]. The value of spline functions at the knots $\{x_i\}_{i=0}^{n}$ are determined using table1. Then we can write in matrix-vector form as follows

This leads to the non-linear system of order n + 2 given by

$$\sum_{j=-3}^{n-1} C_j B_j(0) = 0 \text{ for } x = 0, \quad (18)$$

$$\sum_{j=-3}^{n-1} C_j B_j(1) = 0 \text{ for } x = 1\ , \quad (19)$$

$$-k\sum_{j=-3}^{n-1} C_j B_j^{''}(x) + \sum_{j=-3}^{n-1} C_j B_j(x) - k\sum_{j=-3}^{n-1} C_j B_j(x)(1 - \sum_{j=-3}^{n-1} C_j B_j(x)) = u_0 + kf(x,k) \qquad \text{for } x = 0, h, 2h, ..., 1 \quad (20)$$

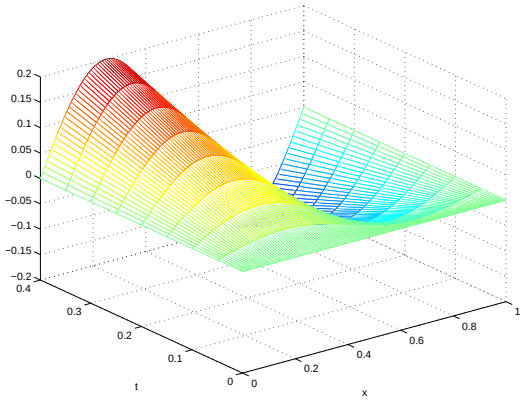


Fig. 1
RESULTS FOR $n = 121, k = 0.05$

The approximate solution (8) is obtained by solving non-linear system using Levenberg-Marquardt optimization method [29] and MATLAB 6.5.

It is easy to see that, the same approximation is applied the other equations (14)-(15).

4. Numerical results

In this section, the method discussed in section 2 and 3 is tested on the following problems from the literature[7], and absolute error in the analytical solutions are calculated. All computations were carried out using MATLAB 6.5.

Problem 1.

We consider a 1-D Fisher's diffusion partial differential equation

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = u(1-u) + f(x,t)\ , \qquad 0 \le x \le 1\ , \qquad t > 0\ , \tag{23}$$

with the initial conditions,

$$u(x,0) = 0\ , \qquad 0 \le x \le 1\ , \tag{24}$$

and the boundary conditions at $x = 0$ and $x = 1$ of the form

$$u(0,t) = u(1,t) = 0, \qquad t \ge 0\ . \tag{25}$$

The exact solution of this problem is $u(t,x) = 0.5tsin2\pi x$. The observed maximum absolute errors for various values of n and for a fixed value of k=0.05 are given in Table 1. The numerical results are illustrated in Figure 1.

Table 1. Comparison of the Numerical solution with the exact solution at different n and k=0.05

n	$k = 0.05$
21	$1.5944141\text{x}10^{-3}$
41	$4.0001700\text{x}10^{-4}$
61	$1.7790142\text{x}10^{-4}$
121	$4.4492962\text{x}10^{-5}$
191	1.7748496416^{-5}

5. Conclusions

A family of B-spline methods has been considered for the the numerical solution of the Fisher equations . The third-degree B-spline has been tested on the Fisher problem, and have tabulated the maximum absolute errors for different values of n . As is evident from the numerical results, the present method approximate the exact solution very well. Also the numerical results are illustrated in figures. The implementation of the present method is more computational than other numerical techniques.

References

[1] C. de Boor, A Practical Guide to Splines, 108,Springer-Verlag, New York, 1978.

[2] H. Caglar, M. Ozer, N.Caglar, The numerical solution of the one-dimensional heat equation by using third degree B-spline functions, Chaos Solitons Fract., 38, 1197Ű1201(2008).

[3] Caglar HN, Caglar SH, Twizell EH. The numerical solution of third-order boundary-value problems with fourth-degree B-spline functions. Int J Comput Math ,71,373Ű81(1999).

[4] Caglar HN, Caglar SH, Twizell EH. The numerical solution of fifth-order boundary-value problems with sixth-degree B-spline functions. Appl Math Lett 12,25Ű30(1999).

[5] Caglar N, Caglar H, Cagal B. Spline solution of nonlinear beam problems. J Concrete Appl Math 1(3),253Ű9(2003).

[6] Caglar H, Caglar N, Elfaituri K. B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to two-point boundary value problems. Appl Math Comput, 175,72Ű9(2006).

[7] Caglar N, Caglar H. B-spline solution of singular boundary value problems. Appl Math Comput 182,1509Ű13(2006).

[8] R.A. Fisher, The wave of advance of advantageous genes, Ann. Eugen. 7 (1937) 353-369.

[9] R.L. Luther, Raĺ umliche Fortpflanzung Chemischer Reaktinen, Z. fuĺ r Elektrochemie und angew, Phys. Chem. 12 (1906) 506-600.

[10] J.D. Murray, Lectures on Non-linear-Differential-Equations Models in Biology, Oxford University Press, London, 1977.

[11] N.F. Britton, Reaction-Diffusion Equations and Their Applications to Biology, Academic Press, New York, 1986.

[12] D.A. Frank-Kamenetskii, Diffusion and Heat Exchange in Chemical Kinetics, Princeton University Press, Princeton, NJ, 1955.

[13] F.A. Williams, Combustion Theory, Addison-Wesley, Reading, MA, 1965.

[14] H.C. Tuckwell, Introduction to Theoretical Neurobiology, Cambridge Studies in Mathematical Biology, vol. 8, Cambridge University Press, Cambridge, UK, 1988.

[15] H. Cohen, Nonlinear diffusion problems, in: A.H. Taut (Ed.), Studies in Applied Mathematics, MAA Studies in Mathematics, vol. 7, Mathematical Association of America, 1971, pp. 27-64 (distributed by Prentice-Hall, Englewood Cliffs, NJ).

[16] P.C. Fife, J.B. McLeod, The approach of solutions of nonlinear diffusion equations to travelling front solutions, Arch. Ration. Mech. Anal. 65 (1977) 335-361.

[17] D.G. Aronson, H.F. Weinberger, Nonlinear diffusion in population genetics, combustion, and nerve pulse propagation in: J.A.Goldstein (Ed.), Partial Differential Equations and Related Topics, Lecture Notes in Mathematics, vol. 446, Springer, Berlin, 1975,pp. 5-49.

[18] M.D. Bramson, Maximal displacement of branching Brownian motion, Comm. Pure Appl. Math. 31 (1978) 531-581.

[19] K. Aoki, Gene-culture waves of advance, J. Math. Biol. 25 (1987) 453-464.

[20] A.J. Ammerman, L.L. Cavalli-Sforza, Measuring the rate of spread of early farming, Man 6 (1971) 674-688.

[21] A.J. Ammerman, L.L. Cavalli-Sforza, The Neolithic Transition and the Genetics of Populations in Europe, Princeton University Press, Princeton, 1983.

[22] J. Canosa, Diffusion in nonlinear multiplicative media, J. Math. Phys. 10 (1969) 1862-1868.

[23] J.J. Tyson, P.C. Fife, Target patterns in a realistic model of the BelousovŰZhabotinskii reaction, J. Chem. Phys. 73 (1980) 2224-2257.

[24] P. Brazhnik, J. Tyson, On traveling wave solutions of FisherŠs equation in two spatial dimensions, SIAM J. Appl. Math. 60 (2) (1999) 371-391.

[25] G.F. Carey, Y. Shen, Least-squares finite element approximation of FisherŠs reactionŰdiffusion equation, Numer. Methods Part. Differ. Eq. 11 (1995) 175-186.

[26] J.P. Boyd, Chebyshev and Fourier Spectral Methods, Dover, New York, 2000.

[27] K. Al-Khaled, Numerical study of FisherŠs reactionŰdiffusion equation by the Sinc collocation method, J. Comput. Appl. Math. 137 (2001) 245-255.

[28] J. Rashidinia, R. Mohammadi, Non-polynomial cubic spline methods for the solution of parabolic equations, International Journal of Comp. Math. 85(5)(2008) 843-850.

[29] Fletcher R. Practical methods of optimization. A Wiley-Interscience Publication;1987.

Heat conduction in a solid substrate with a spatially-variable solar radiation input: Carslaw-Jaeger solution revisited

R.G. Kasimova[1], Yu.V.Obnosov[2]

[1]German University of Technology in Oman, Muscat, Sultanate of Oman

[2]Institute of Mathematics and Mechanics, Kazan Federal University, Kazan, Russia

Abstract

Temperature distributions recorded by thermocouples in a solid body (slab) subject to surface heating are used in a mathematical model of 2-D heat conduction. The corresponding Dirichlet problem for a holomorphic function (complex potential), involving temperature and heat stream function, is solved in a strip. The Zhukovskii function is reconstructed through singular integrals, involving an auxiliary complex variable. The complex potential is mapped by the Schwartz-Christoffel formula onto an auxiliary half-plane. The final heat conduction flow net (orthogonal isotherms and heat lines) is compared with the known Carslaw-Jaeger solution and shows a puzzling topology of energy fluxes for simple temperature-boundary conditions.

Key words: *Laplace's equation, topology of heat lines, complex potential, conformal mappings.*

1. Introduction

Analytical solutions to potential field problems, where the intricate topology of 2-D flow nets (stream lines and constant potential lines) was controlled by heterogeneity of the flow domain, but the boundary conditions were uniform (constant potentials on the inlet and outlet of a standard flow tube) were presented in [1], [2]. In this paper we study the effect of non-uniform boundary conditions, although assume that the medium, through which flow takes place, is homogeneous. Analytical solutions for steady 2-D heat conduction in solid bodies are needed in different engineering designs involving heat transfer [3]. A powerful technique to solve these problems is based on the theory of boundary-value problems for holomorphic functions (e.g., [4], [5]). In this paper we average the diurnal temperature swings, recorded by thermocouples on the surface of a concrete slab, and show that the corresponding explicit analytical solution gives a computer-algebra-visualized topology of heat lines, which is counterintuitive and puzzling.

2. Mathematical Model

We consider a vertical cross-section of the slab of a thickness b and thermal conductivity k, and a thermal barrier E_1OE_2 (practically, strip-type shading against solar radiation). Fig.1a depicts a vertical cross-section and Cartesian coordinates. Far from the barrier (the rays AE_1 and E_2B), the slab temperature is the same as the ambient air temperature, T_0 =constant. Along AOB, we have experimental data of temperature obtained by thermocouples and we take the daily averages of these values. The x-distribution of this average temperature is a single-minimum function f(x). We assume that this function is symmetric f(-x)=f(x) and $f(x) \to T_0$ at $x \to \pm\infty$ (this is confirmed by experiments). We introduce F(x) as:

$$f(x)=T_0 - F(x), \quad \text{at } y=0 \qquad (1)$$

where F(x) is a single-maximum ($T_M = T_0 - T_m$) function shown in Fig.1b. We assume that along the internal surface (DC in Fig.1a) temperature is constant, T_c:

$$T=T_c, \quad \text{at } y=-b \qquad (2)$$

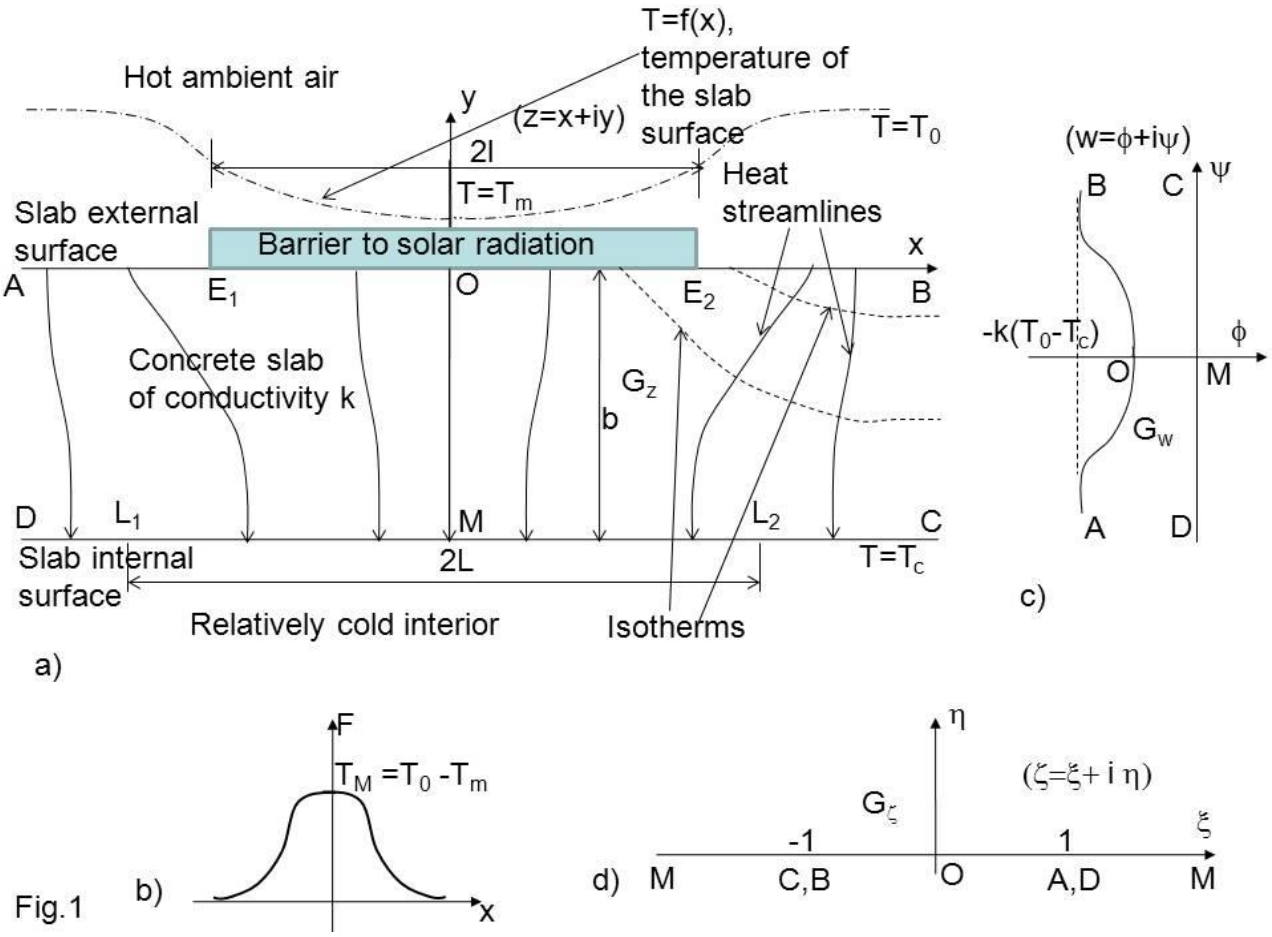


Fig.1

Fig.1. Vertical cross-section of a slab with a thermal barrier (a), temperature boundary condition on the exterior surface – the kernel of the Cauchy integral (b), complex potential domain for small T_M (c), auxiliary plain where the Dirichlet problem is solved (d).

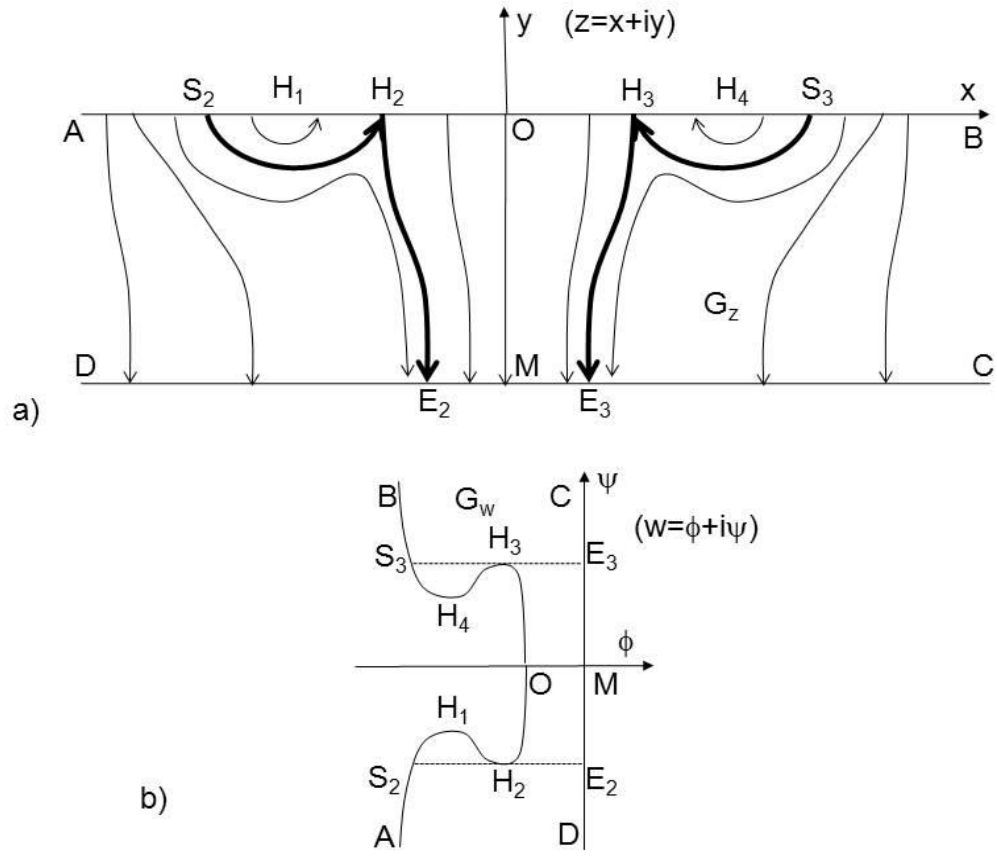


Fig.2

Fig.2. Heat line topology with four hinge points (a), the corresponding knob-shape bounded complex potential domain (b).

In case of no thermal barrier in Fig.1a, heat conduction in the slab of Fig.1a is 1-D. With the barrier, the so-called "thermal resistor" models (see, e.g., [6]) have been used. The resistor approximation assumes the AOB boundary condition to be a step-function (reflecting the barrier width) and the heat streamlines are postulated to be straight and perpendicular to the both slab boundaries, i.e. heat flow is again 1-D. Our objective is to assess analytically the spatial nonuniformity of temperature and heat lines caused by the boundary condition (1).

According to the Fourier law, heat conduction in the strip AOBCMD (we denote it G_z) of Fig.1a is governed by

$$\vec{J} = -k\nabla T(x,y) \qquad (3)$$

where $\vec{J}(x,y)$ is the heat flux vector, which has a vertical component v and horizontal component u.

We introduce a complex physical coordinate z=x+*i*y and a complex potential w=ϕ +i ψ where *i* is an imaginary unit, ϕ=-k(T-T_c) is the potential and ψ is a stream function, which is related to ϕ through the Cauchy-Riemann conditions:

$$\frac{\partial\phi}{\partial x} = \frac{\partial\psi}{\partial y} = u, \qquad \frac{\partial\phi}{\partial y} = -\frac{\partial\psi}{\partial x} = v \qquad (4)$$

Heat lines ψ=const allow a better visualization of heat transfer and an assessment of thermophysical efficiency [7].

Both ϕ and ψ are harmonic:

$$\Delta\phi(x,y) = 0, \qquad \Delta\psi(x,y) = 0 \qquad (5)$$

and w(z) is a holomorphic function.

An integral solution of the boundary-value problem (1),(2) and (5) is given in [3] (Chapter V, Section 3, eqn.2.19). Here we derive an alternative solution. Carslaw and Jaeger derived their solution by the Fourier transform method. The Laplace equation was analytically solved in [8] in a stream tube G_z by separation of variables and Fourier's series expansions. The Fourier methods are limited to the domains G_z that are homeomorphic to a simple streamtube (two constant temperature - two adiabatic segments as boundaries coinciding with the level lines of a Cartesian, cylindrical, spherical, etc. coordinate system, where the Laplace equation separates). Our method does not have this limitation and is applicable to any G_z-polygon with arbitrary mixed (Dirichlet-Newman) boundary.

Without any loss of generality we assume that along OM ψ=0 that follows from the symmetry of f(x). The isotherms (equipotential lines ϕ=const) are dashed in Fig.1a and heat streamlines (ψ =const) are shown in bold with arrows indicating the direction of heat transfer. The domain G_z is fixed but the domain corresponding to G_z in the w-plane, G_w, depends on f(x) and is surprisingly complex even for simple functions f(x).

If T_m is close to T_0 and the slope of f(x) is small, then G_w is a strip with a slightly bulged side AOB (Fig.1c). The streamlines in G_z (Fig.1a) are somewhat curved, most of all in the slab zones where the imposed f(x) has a relatively high slope magnitude |df/dx| (see Fig.1a).

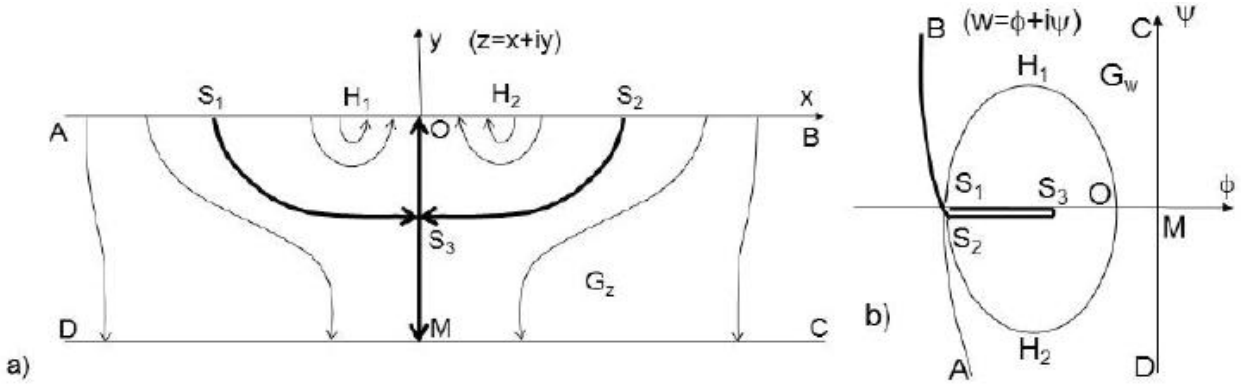


Fig.3. Heat line topology with two hinge points (a), the corresponding double-sheet Riemann surface as the complex potential domain (b).

For a smaller T_m (fixed T_0 but higher T_M) and/or stronger variation of the slope of f(x), the heat flow topology is shown in Fig.2a. On AOB (we recall, f(x) is a single-minimum function) at four points H_1, H_2, H_3, and H_4, the direction of the v-component of the thermal gradient changes from inside the slab to the exterior. Indeed, along AS_2H_1 and BS_3H_4 heat is conducted from the exterior surface into the slab. Along H_1H_2 and H_4H_3 heat is discharged back and along H_2H_3 heat moves from the exterior surface to the interior. There are two separatrices (dividing streamlines shown in bold), $S_2H_2E_2$ and $S_3H_3E_3$, which demarcate five different topological zones in G_z. The corresponding domain G_w is shown in Fig.2b where the image of AOB is a knob-shaped curve.

For even smaller T_m and/or stronger slopes of f(x) we may arrive at topology depicted in Fig.3a. Here we have two points H_1 and H_2 where flow changes its orientation from the interior to the exterior of the slab. The only separatrix (bold-styled in Fig.3a) has a saddle point S_3. Above $S_1S_3S_2$ heat is circulated from the air

into concrete and back, without entering the interior. The domain G_w shown in Fig.3b is a double-sheet Riemann surface. The second sheet $S_1H_1OH_2S_2$ is stitched to the first (main) sheet through the cut $S_1S_3S_2$, which images the separatrix in G_z. In Fig.3b we purposely distorted the branch AS_1 (of course, this branch in G_w is symmetrical to S_2B with respect to the ϕ axes) in order to illustrate the stitching of the second sheet. Points S_1 and S_2 are located on the opposite sides of the cut in G_w. If $T_m<T_c<T_0$, then still another heat conduction regime is realized, with heat flux from the interior (this regime may occur in cold countries and has not been experimentally observed in Oman).

We implement a mathematical technique, which can readily tackle any heat flow regime in Figs.1a-3a. The method is based on a conformal mapping of one domain (G_z in our case) onto an auxiliary domain (circle, half-plane) and, next, solving there a Dirichlet, mixed, Newton (Robin) or refraction problem (with the first to fourth boundary conditions, correspondingly) and further reconstruction of the second holomorphic function in the auxiliary domain ([1], [2]).

So, first, we map conformally G_z onto the upper half-plane Im ζ >0 of an auxiliary plane $\zeta=\xi+i\,\eta$ shown in Fig.1d by the Schwartz-Christoffel formula:

$$z = -\frac{b}{\pi}\log\frac{1+\varsigma}{1-\varsigma} \qquad (6)$$

In this plane points A and D, as well as C and B coincide.

Next, we introduce the Zhukovskii function (see [5]) as $Zh=w-i(T_0-T_c)kz/b+k(T_0-T_c)=R+i\,I$. The real part of this function is $R=Re[Zh]=\phi+k(T_0-T_c)y/b+k(T_0-T_c)$ and the imaginary part $I=Im[Zh]=\psi-k(T_0-T_c)x/b$. Obviously, Zh(z) is also holomorphic. In the half-plane Im ζ >0 the following boundary conditions hold for Zh(ζ):

$$R=0 \text{ at } |\xi|>1, \quad R=kF[x(\xi)] \text{ at } |\xi|<1 \qquad (7)$$

where eqn.(6) gives x(ξ) as :

$$x = -\frac{b}{\pi}\log\frac{1+\xi}{1-\xi}, \qquad |\xi|<1,$$
$$x = -\frac{b}{\pi}\log\frac{\xi+1}{\xi-1}, \qquad \xi>1 \qquad (8)$$

Obviously (see Fig.1b), $F(\xi)\to 0$ at $\xi\to\pm 1$. The function $F(\xi)$ can be easily interpolated from experimental (thermocouple) daily-averaged point-wise collected values. We used $F[x]=T_M\exp[-ax^2]$, where *a* is a fitting parameter, as an approximation for experimentally-measured temperature values. Any other function, e.g., $F[x]=T_M/[1+(b_cx)^2]$ (where b_c is another fitting parameter), can be used in eqn.(7) as a boundary condition. Eqn.(8), obtained from the conformal mapping, is fixed and does not depend on interpolation of experimental data and the choice of F[x].

Along with the boundary conditions (7) for the real part of the Zhukovskii function R(ζ), we note that at point M (where $\zeta\to\infty$) the imaginary part of this complex function, I(ζ)=0. Then an integral solution to the stated Dirichlet boundary-value problem is (see [5]):

$$Zh(\zeta) = -\frac{i}{\pi}\int_{-1}^{1}\frac{kF(\tau)d\tau}{\tau-\zeta} \qquad (9)$$

Passing to the Sokhotsky-Plemelj limit $\zeta\to\xi, \quad |\xi|<1$ from eqn.(9) we obtain the stream function along AOB:

$$\psi = \frac{k(T_0-T_c)}{b}x(\xi) - \frac{1}{\pi}\int_{-1}^{1}\frac{kF(\tau)d\tau}{\tau-\xi} \qquad (10)$$

We note that the integral in eqn.(10) is singular at $|\xi|<1$ (that corresponds to the line AOB in Fig.1a) and should be calculated in the sense of *v.p.* (principal value). Wolfram's *Mathematica* [9] has a routine *CauchyPrincipalValue* for this purpose, which we used in numerical integration. At $|\xi|>1$ (line DMC in Fig.1a) the integral in eqn.(10) is regular and we used the routine *NIntegrate* from [9].

It is convenient to expand the kernel in eqn.(10) in a series of Chebyshev's polynomials of the second kind as:

$$F(\tau) = T_M\sum_{n=1}^{\infty}b_nU_n(\tau), \quad |\tau|<1, \quad b_n = \frac{2}{\pi}\int_{-1}^{1}\frac{f(\tau)U_n(\tau)}{\sqrt{1-\tau^2}}d\tau$$

where $U_n(\tau)=\sin[\,n\arccos\tau\,]$. For any smooth (e.g., belonging to the Holder class is sufficient) function F(τ) this series is uniformly converging on the interval (-1,1). Then for the roof surface AB eqn. (10) is reduced to

$$\psi = \frac{k(T_0-T_c)}{b}x(\xi) - \mathrm{k}$$
$$T_M\sum_{n=1}^{\infty}b_nT_n(\xi), \quad |\xi|<1 \qquad (11)$$

and for the ray MD we have

$$\psi = \frac{k(T_0-T_c)}{b}x(\xi) + \mathrm{k}\,T_M$$
$$\sum_{n=1}^{\infty}b_n(\xi-\sqrt{\xi^2-1})^n, \quad \xi>1 \qquad (12)$$

where $T_n(\tau)=\cos[n\arccos\tau]$ are the Chebyshev polynomials of the first kind. For the ray MC $(\xi<-1)$ we have $\psi(-\xi)=\psi(\xi)$, i.e. eqn.(12) can be used.

The vertical component of the thermal gradient `

$$v = -\frac{\partial\psi}{\partial x} = -\frac{\partial\psi}{\partial\xi}\left(\frac{\partial x}{\partial\xi}\right)^{-1} \qquad (13)$$

Far from the insulation zone (large values of |x|) the horizontal component u of the gradient vanishes and $v \to v_\infty = -\dfrac{k(T_0 - T_c)}{b}$. We introduce a dimensionless vertical component $v^d = v / v_\infty$. On AOB, differentiation of eqns. (8) and (11) yields

$$v^d(\xi) = 1 - \frac{\pi r}{2}\sqrt{1-\xi^2}\sum_{n=1}^{\infty} n b_n U_n(\xi), \quad |\xi| < 1, \quad r = \frac{T_M}{T_0 - T_c} \qquad (14)$$

Then the hinge points (if they exist) in Figs.2a-3a are found from eqn.(14) as the roots of the equation $v^d(\xi) = 0, \quad |\xi| < 1$. We solved this equation using the *FindRoot* routine [9].

How much in terms of total energy saving we gain from thermal insulation? In order to answer this question we select two symmetrical points L_1 and L_2 on the interior surface, distance 2L apart. Without the barrier in Fig.1a, the total heat entering the interior (per unit length in the direction perpendicular to the plane in Fig.1a) through a strip of a width 2L is $Q_0 = 2L\, k(T_0 - T_c)/b$. From the definition of the stream function the total heat flowing through the same area but in 2-D conduction with insulation is $Q = -2\psi_{L1}$, where ψ_{L1} is directly expressed from eqns.(8) and (12) as:

$$\psi_{L1} = \frac{k(T_c - T_0)L}{b} + kT_M \sum_{n=1}^{\infty} b_n \tanh^n\left(\frac{\pi L}{4b}\right) \qquad (15)$$

We introduce a dimensionless energy saving through L_1L_2 as $S(L) = \delta Q / Q_0$, where $\delta Q = Q_0 + 2a\psi_{L1}$ and, with ψ_{L1} taken from eqn.(15), we have:

$$\delta Q = r\frac{b}{L}\sum_{n=1}^{\infty} b_n \tanh^n\left(\frac{\pi L}{4b}\right) \qquad (16)$$

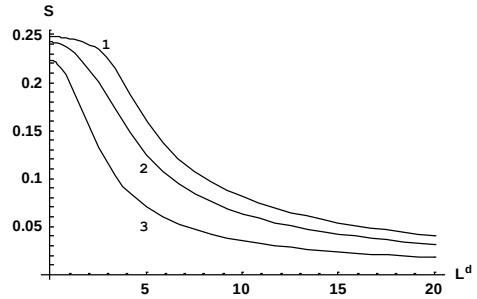


Fig.4. Energy saving factor S as a function of L^d for r=0.25 and a^d=0.025, 0.1 and 0.4 (curves 1-3, correspondingly).

As we have pointed out, the selected F(x) is $T_M \exp[-a x^2]$. Fig.4 shows S as a function of a dimensionless width $L^d = L/b$ for r=0.25 and a^d=0.025, 0.1 and 0.4 (curves 1-3, correspondingly, where $a^d = a\, b^2$), calculated by eqn.(16). In Fig.5 v^d is shown as a function of dimensionless abscissa $x^d = x/b$ along AOD for r=0.5 and a^d=1, 2 and 4 (curves 1-3, correspondingly), calculated by eqn.(14). As we can see from Fig.5, for the selected F(x) we have the flow topology of Fig.1a (no hinge points) for the first two curves and the two-hinge-points regime for the third curve. All three curves have two blips (maxima) which indicate that in the near-blip zone of the exterior plane the intensity of conduction into the slab is even higher than in the case of no thermal insulation, i.e. near the edges E_1 and E_2 in Fig.1a the barrier "sucks" energy.

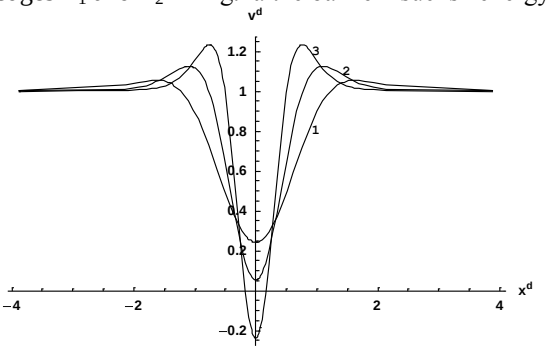


Fig.5. Vertical component of thermal gradient v^d as a function of x^d along AOD for r=0.5 and a^d=1, 2 and 4 (curves 1-3, correspondingly).

Without any series expansions we can bluntly use eqns. (6) and (9) in the following form:

$$w^d = iz^d - 1 - \frac{ir}{\pi}\int_{-1}^{1} \frac{F(\tau)\,d\tau}{\tau - \left(\dfrac{\exp[-\pi z^d] - 1}{\exp[-\pi z^d] + 1}\right)} \qquad (17)$$

where dimensionless variables are introduced as $w^d = w/(k(T_0 - T_c)), \quad z^d = z/b$.

By the help of the routines *Re* and *Im* [9] we separated the real and imaginary parts in eqn.(17). Then we used the *ContourPlot* routine [9] to plot the flow nets. Fig.6 shows the flow net for $F = \exp[-a^d (x^d)^2]$ with r=0.9, and a^d=15 (two hinge points regime of Fig.3a). In Fig.6, in order to avoid cluttering, only three equipotential contours are presented: ϕ^d=-0.1 (curve 1, single branch, see the Riemann surface in Fig.3b), ϕ^d =-0.3 (two branches, labeled 2) and ϕ^d =-0.4 (two branches, labeled 3). For the sake of comparisons we also plotted the equipotentials according to the mentioned solution [3], denoted here as (CJ-2.19), which in our notations and dimensionless variables reads:

$$\phi^d(x^d, y^d) = \frac{1}{2}\sin \pi y^d \int_{-\infty}^{\infty} \frac{1 - r\exp[-a^d \tau^2]}{\cosh \pi(1 - y^d) + \cosh \pi(x^d - \tau)}\, d\tau$$

(CJ-2.19)

Our eqn.(17) and eqn. (CJ-2.19) give identical contours.

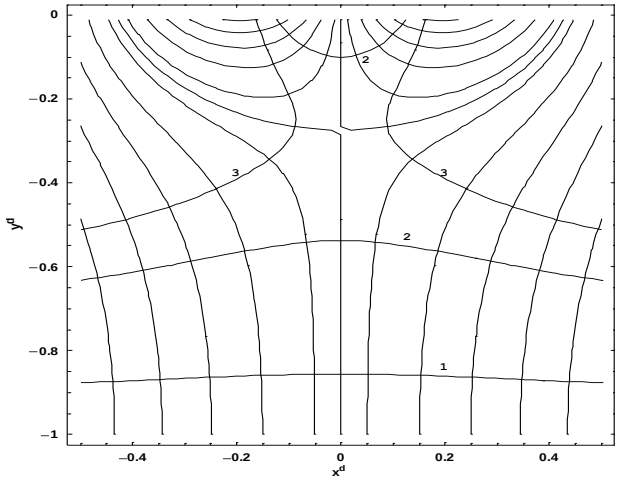


Fig.6. Flow net (isotherms and heat lines) for $F=\exp[-a^d (x^d)^2]$, r=0.9, a^d=15

It is clear that S_3 in Fig.3a is indeed a saddle point, i.e. if we approach this point from the left and right, then temperature decreases towards this point, but if we move from S_3 upward and downward, then temperature decreases[1]. S_3 (in Fig.6 corresponds to the contour-plotting lacuna) is a genuine critical point because the thermal gradient there is zero while H_1 and H_2 are not really critical points (only v there vanishes but the horizontal component u of $\vec{J}$ does not). *Mathematica* contour-plotting computations confirmed what we conceptualized as flow topologies in Figs.1a, 2a, 3a.

3. Conclusions

Our mathematical model and the final solution, eqn.(17), gives temperature and heat flux field in the slab as an output of the *ContourPlot* routine of a standard computer algebra package (*Mathematica*). The solution is simple, versatile and provides analytical expressions for isotherms, heat lines, and thermal gradient (magnitudes and directions). Our solution gives the same results as the known solution from [3] (obtained by a different method and not analyzed by them). The flow topology in Figs.1-3 is indeed counterintuitive and, to the best of our knowledge, has never been reported before. Our mathematical approach to solve the corresponding boundary-value problem of heat conduction can be easily extended to more complex geometries of conducting elements than in the presented case (strip), e.g. a rectangle or other polygons can be studied (this would require a more general Schwartz-Christoffel mapping than eqn.(6)).

[1] We recall that the maximum principle (valid for any elliptic equation and the Laplace equation used in this paper, in particular) prohibits global maxima and minima of temperature inside G_z.

3.1 Acknowledgments

This work was supported by the German University of Technology and the Russian Foundation for Basic Research grants No 08-01-00163, No 09-01-97008-r_povolgh'e_a, and Russian Federal Agency of Education (contract No P 944).

4. References

[1] Yu.V.Obnosov, R.G.Kasimova, A.Al-Maktoumi, A.R.Kacimov. Can heterogeneity of the near-wellbore rock cause extrema of the Darcian fluid inflow rate from the formation (the Polubarinova-Kochina problem revisited)? *Computers & Geosciences*, 36, 1252–1260. 2010. doi:10.1016/j.cageo.2010.01.014.

[2] Yu.V.Obnosov, R.G. Kasimova, A.R.Kacimov. A well in a 'target' stratum of a two-layered formation: the Muskat–Riesenkampf solution revisited. *Transport in Porous Media*, 2011, DOI 10.1007/s11242-010-9693-6

[3] H.S.Carslaw, J.C.Jaeger, *Conduction of Heat in Solids.* 2nd edition. Clarenden Press, Oxford, 1959.

[4] F.D.Gakhov, *Boundary Value Problems*, Pergamon Press, New York, 1966.

[5] P.Ya.Polubarinova-Kochina, *Theory of Ground-water Movement.* Princeton Univ. Press, Pinston, 1962.

[6] D.J. Sailor, D.Hutchinson, L.Bokovoy, Thermal property measurements for ecoroof soils common in the western U.S. *Energy and Buildings*, 40, 1246–1251, 2008.

[7] A.Bejan, *Convection Heat Transfer.* 3rd edition. Wiley, N.Y, 2004.

[8] J. A. Kolodziej, T.Strek. Analytical approximations of the shape factors for conductive heat flow in circular and regular polygonal cross-sections. *International Journal of Heat and Mass Transfer*, 44(5), 999-1012, 2001.

[9] S.Wolfram. *Mathematica. A System for Doing Mathematics by Computer*. Addison-Wesley, Redwood City, 1991.

Indirect Vector Control of Stand-Alone Self-Excited Induction Generator

S. N. Mahato[1], S. P. Singh[2], and M. P. Sharma[3]
[1]Department of Electrical Engineering, National Institute of Technology, Durgapur, India
[2]Department of Electrical Engineering, Indian Institute of Technology, Roorkee, India
[3]Alternate Hydro Energy Centre, Indian Institute of Technology, Roorkee, India

Abstract - *This paper presents the voltage build-up process and the terminal voltage control of a stand-alone self-excited induction generator (SEIG) using indirect vector control (IVC) technique under variable speeds and different types of load. Here, the three-phase SEIG is excited by a pulse-width modulated voltage source inverter (PWM-VSI) connected to a single-capacitor on the DC side with a start-up battery. The limitation of having stand-alone SEIG is poor voltage regulation, which occurs with change in speed and load condition. Hence, there should be a control system that keeps the terminal voltage of the SEIG and the DC bus voltage constant when the speed of the rotor and also, the load on the SEIG are varied. The indirect vector control scheme has been presented to maintain the terminal voltage of the generator and the DC bus voltage constant for variable rotor speed and load. The space-phasor model of the induction machine has been used in simulation. To predict the performance of the proposed system, a MATLAB/SIMULINK based study has been carried out for both AC and DC loads. The proposed control scheme has shown very good voltage regulation and phase balance even with unbalanced three-phase load.*

Keywords: AC and DC load, Indirect vector control, PWM inverter, Self-excited induction generator.

1 Introduction

The electrification of remote, rural areas are important for the sustainable development of a country. To provide power through grid extension becomes very difficult and expensive in such hard to access and remote areas. In such areas, plenty of renewable energy sources such as small hydro, wind, bio-mass etc. are available. A practical solution therefore is to develop isolated, small-scale power generation schemes that utilize the renewable energy resources locally available to supply the consumers. Due to the research of clean power (or renewable energy resources), and small-scale autonomous power generation systems, the SEIG has become very popular for generating power from renewable energy sources, such as wind and small hydro. The SEIG has distinct advantages like simplicity, low cost, ruggedness, little maintenance, absence of DC, brushless etc. as compared to the conventional synchronous generator. However, its major disadvantage is the inability to control the voltage and frequency under change in load and speed in stand-alone system.

A number of schemes have been suggested for regulating the terminal voltage. The scheme based on switched capacitors [1] finds limited application because it regulates the terminal voltage in discrete steps. A saturable reactor scheme of voltage regulator [2] involves a potentially large size and weight, due to the necessity of a large saturating inductor. In the short/long shunt configuration [3], the series capacitor used causes the problem of resonance while supplying power to an inductive load.

Unlike conventional excitation system that normally consists of variable impedance scheme [4], the PWM compensator permits the vector control implementation and presents precise and continuous reactive power control with fast response times, over a wide variation in speed. A wide variety of VAR generators with some control strategies using power electronic technology have been developed for stand-alone SEIG [5-7]. Lyra et al. [5] have analyzed a high performance variable speed energy generation system based on an isolated induction generator using a PWM VAR compensator to control the flux in the induction generator and the reactive power balance by implementing flux vector control methods. Seyoum et al. [6] have presented the stator flux oriented vector control for wind turbine driven isolated induction generator. However, an additional decoupling compensation should be applied for vector control in the stator flux orientation. In [7], a field-oriented controller has been used to excite the stand-alone induction machine efficiently, minimizing copper and iron losses, and to regulate the generated voltage for variable speed and load. An advanced solution for voltage control of the induction generator using rotor field-oriented control for small-scale AC and DC power applications has been given by Ahmed et al. [8]. Cardenas and Pena [9] have discussed a sensorless control structure based on a direct rotor flux oriented vector control system for variable speed wind energy applications. The modeling, control system design and simulation results for a stand-alone induction generator system with static reactive power compensator of current controlled PWM VSI using rotor flux oriented control has been presented by Liao and Levi [10]. Ahmed et al. [11] have used a hybrid excitation unit consisting of a capacitor bank and an active power filter to regulate the output voltage of stand-alone SEIG and proposed the advanced deadbeat current control strategy that works with variable speed to reduce the system cost. Pucci

and Cirrincione [12] have presented a maximum power point tracking for high performance wind generators. The field-oriented control of the machine has been further integrated with an intelligent sensorless technique.

Since, only few papers on application of vector control techniques for control of SEIG are available in the literature, further investigation on vector control of isolated induction generators needs to be carried out. Accordingly, indirect vector control strategy with rotor flux orientation with high dynamic performance has been used in this paper for voltage control of an isolated SEIG for both DC and AC power applications. The single DC side capacitor provides all the reactive current or the VAR required by the generator and the load. The space-phasor model of the induction generator has been used. The proposed scheme has been simulated in MATLAB/SIMULINK environment. The simulated studies for different transient conditions such as self-excitation, sudden application and removal of both AC and DC loads have been carried out to demonstrate the effectiveness of the scheme.

2 System Description and Control Scheme

Fig. 1 shows the general configuration of the system, where the DC and AC loads can be supplied by the generator. The basic system consists of a PWM VAR compensator connected to an induction generator. A battery on the DC side of the inverter is provided for initial excitation. When the flux reaches the desired level, the battery is disconnected and the generator supplies itself the necessary energy to control the voltage across the compensator DC capacitor. The reactive power required by the SEIG and load is provided by the VSI.

During start-up, the power produced by SEIG is used to charge the capacitor connected across the DC link to a set reference value. In this study, the DC voltage is maintained at 600 V. The DC bus voltage is measured to feedback the DC voltage controller. This controller provides the q-axis component of compensator reference AC currents that represents the flow of active power necessary to keep the DC voltage constant. The field weakening is done above base speed operation and the flux command is generated. The flux error is fed to the PI controller and the output of this PI controller gives the d-axis component of compensator reference AC current. The principal vector control parameters I_{ds}^{e*} and I_{qs}^{e*} , which are DC values in synchronously rotating frame, are converted to stationary frame with the help of unit vectors ($\cos\theta_e$ and $\sin\theta_e$). By using the transformation matrix, the resulting stationary frame two axes current commands are converted into three-phase current commands. These three-phase current commands are compared with the three-phase stator currents. The errors are amplified and compared with the triangular carrier signal to generate the switching pulses for the inverter.

Any variation in the output power of the SEIG is directly indicated by the variation in the terminal voltage of the generator. A reduction of the DC link voltage indicates that the active power drawn by the load is more than the generated power of the SEIG and the difference in power is supplied by the VSI and hence, the DC link voltage falls. An increase in the DC link voltage indicates that the active power drawn by the load is less than the generated power of the SEIG. In both these two cases, the controller varies the ON-OFF period of the IGBTs of the inverter and the terminal voltage of the generator is maintained constant.

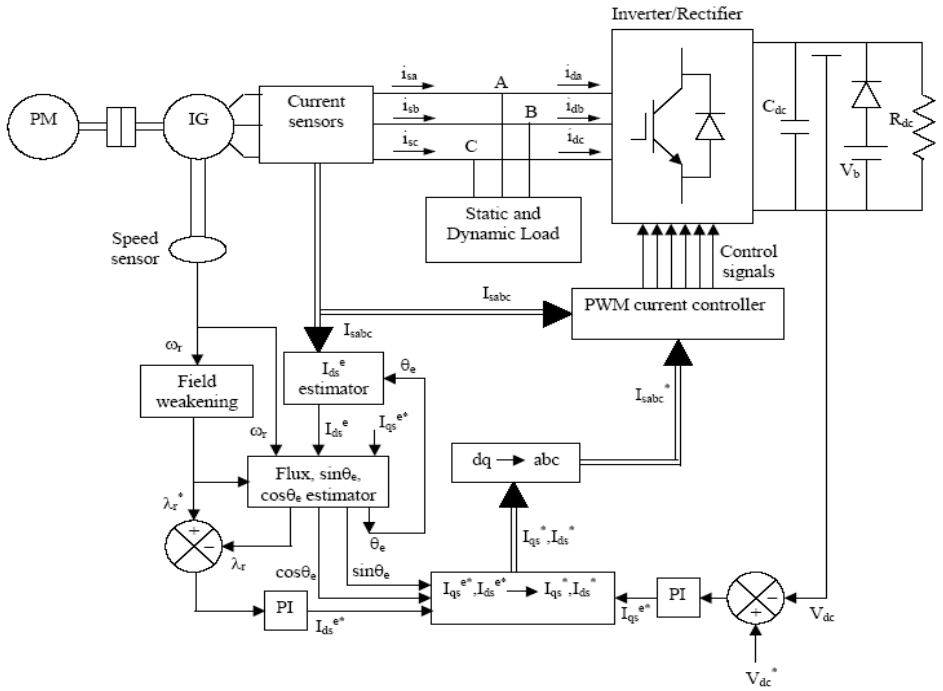


Fig. 1. Block diagram of the proposed system.

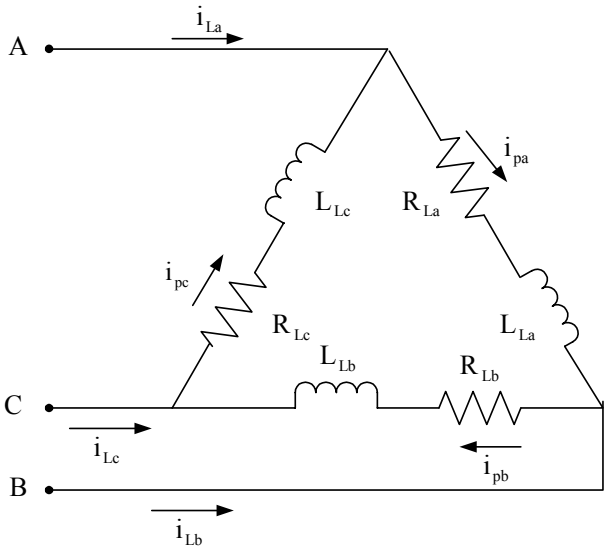


Fig. 2. Circuit diagram of three-phase inductive load

3 Mathematical Modeling

3.1 Modeling of the SEIG

The induction machine has been represented by the space-phasor model. Instead of using two axes such as the d and q for a balanced polyphase machine, the flux linkage phasors can be thought of as being produced by equivalent single-phase stator and rotor windings.

The four dq equations of equation (A1) in Appendix-A can be reduced to two space-phasor equations as:

$$v_s = v_{qs} - jv_{ds} = (R_s + L_s p)i_s + L_m p i_r \quad (1)$$

Similarly, the rotor equations can be written as:

$$v_r = v_{qr} - jv_{dr} = (L_m p - j\omega_r L_m)i_s + (R_r + L_r p - j\omega_r L_r)i_r \quad (2)$$

From equations (A2) and (A3) in Appendix-A, the voltage equations can be written as:

$$V_s = R_s i_s + \frac{d\lambda_s}{dt} \quad (3)$$

$$V_r = R_r i_r + \frac{d\lambda_r}{dt} - j\omega_r [B]\lambda_r \quad (4)$$

where, $\boldsymbol{V_s} = [v_{ds} \;\; v_{qs}]^T$, $V_r = [0\ 0]$, $\boldsymbol{i_s} = [i_{ds} \;\; i_{qs}]^T$, $\boldsymbol{i_r} = [i_{dr} \;\; i_{qr}]^T$, $\lambda_s = [\lambda_{ds} \;\; \lambda_{qs}]^T$, $\lambda_r = [\lambda_{dr} \;\; \lambda_{qr}]^T$,

$\boldsymbol{R_s} = \text{diag}[R_s \; R_s]$, $\boldsymbol{R_r} = \text{diag}[R_r \; R_r]$ and $[B] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

The electromagnetic torque developed is given by,

$$T_e = \frac{3}{2}\frac{P}{2} L_m (i_{qs} i_{dr} - i_{ds} i_{qr}) \quad (5)$$

The torque balance equation of SEIG is defined as:

$$T_{\text{shaft}} = T_e + J\left(\frac{2}{P}\right) p\omega_r \quad (6)$$

The shaft torque, T_{shaft} of the prime-mover and speed is represented by a linear curve given as:

$T_{\text{shaft}} = k_1 - k_2\omega_r$

where, T_{shaft} is the shaft torque which shows the drooping characteristic of prime-mover and k_1 and k_2 are constants. J is the moment of inertia of the induction machine including the machine (prime-mover) coupled on its shaft

3.2 Modeling of the control scheme

3.2.1 Rotor flux vector estimation

The derivations of the control equations of indirect vector control can be done from the d-q equivalent circuit of the induction machine in synchronously rotating reference. The rotor circuit equations are written as:

$$\frac{d\lambda_{dr}^e}{dt} + R_r i_{dr}^e - (\omega_e - \omega_r)\lambda_{qr}^e = 0 \quad (7)$$

$$\frac{d\lambda_{qr}^e}{dt} + R_r i_{qr}^e + (\omega_e - \omega_r)\lambda_{dr}^e = 0 \quad (8)$$

where, $\lambda_{dr}^e = L_r i_{dr}^e + L_m i_{ds}^e$ (9)

$$\lambda_{qr}^e = L_r i_{qr}^e + L_m i_{qs}^e \quad (10)$$

From (9) and (10), i_{dr}^e and i_{qr}^e can be written as:

$$i_{dr}^e = (\lambda_{dr}^e - L_m i_{ds}^e)/L_r \quad (11)$$

$$i_{qr}^e = (\lambda_{qr}^e - L_m i_{qs}^e)/L_r \quad (12)$$

Eliminating i_{dr}^e and i_{qr}^e from (7) and (8) using the equations (11) and (12),

$$\frac{d\lambda_{dr}^e}{dt} + \frac{R_r}{L_r}\lambda_{dr}^e - \frac{L_m}{L_r} R_r i_{ds}^e - \omega_{sl}\lambda_{qr}^e = 0 \quad (13)$$

$$\frac{d\lambda_{qr}^e}{dt} + \frac{R_r}{L_r}\lambda_{qr}^e - \frac{L_m}{L_r} R_r i_{qs}^e + \omega_{sl}\lambda_{dr}^e = 0 \quad (14)$$

where $\omega_{sl} = \omega_e - \omega_r$.

For decoupling control, $\lambda_{qr}^e = 0$, i.e., $\frac{d\lambda_{qr}^e}{dt} = 0$

so that total rotor flux λ_r is directed along the d^e axis and $\lambda_r = \lambda_{dr}^e$.

From (13) and (14), we get

$$\frac{L_r}{R_r}\frac{d\lambda_r}{dt} + \lambda_r = L_m i_{ds}^e \quad (15)$$

and $\omega_{sl} = \frac{L_m R_r}{\lambda_r L_r} i_{qs}^e$ (16)

The derivative of rotor flux can be written as:

$$p\lambda_r = \frac{R_r}{L_r}(L_m i_{ds}^e - \lambda_r) \quad (17)$$

The rotor flux is calculated from the above equation.

The slip frequency ω_{sl}^* can be written as:

$$\omega_{sl}^* = \frac{L_m R_r}{L_r \lambda_r^*} i_{qs}^* \quad (18)$$

and $$\omega_e = \omega_r + \omega_{sl}^* \quad (19)$$

The field angle (i.e., the angle of the synchronously rotating frame) is calculated as:

$$\theta_e = \int \omega_e dt \quad (20)$$

3.2.2 Calculation of d and q axes components of compensator reference current

The field weakening is done above base speed operation and accordingly the flux command (λ_{rref}) is generated. The flux error at the n[th] sampling instant is expressed as :

$$\lambda_{rer(n)} = \lambda_{rref(n)} - \lambda_{r(n)}.$$

The flux error λ_{rer} is fed to the PI controller and the output of the PI controller at the n[th] sampling instant is expressed as:

$$I_{ds(n)}^{e*} = I_{ds(n-1)}^{e*} + K_{pd}\{\lambda_{rer(n)} - \lambda_{rer(n-1)}\} + K_{id}\lambda_{rer(n)} \quad (21)$$

where K_{pd} and K_{id} are the proportional and integral gain constants, respectively, of the PI controller.

The DC bus voltage error (V_{DCer}) at the n[th] sampling instantis: $V_{DCer(n)} = V_{DCref(n)} - V_{DC(n)}$. The error is fed to the PI controller and the output of the PI controller for maintaining DC bus voltage at the n[th] sampling instant is given by

$$I_{qs(n)}^{e*} = I_{qs(n-1)}^{e*} + K_{pq}\{V_{DCer(n)} - V_{DCer(n-1)}\} + K_{iq}V_{DCer(n)} \quad (22)$$

where K_{pq} and K_{iq} are the proportional and integral gain constants, respectively, of the DC bus PI controller.

3.2.3 Calculation of compensator reference currents in stationary reference frame

The q-axis and d-axis reference currents (I_{qs}^{e*} and I_{ds}^{e*}), which are DC values in synchronously rotating frame, are converted to stationary frame with the help of unit vectors as given below:

$$I_{qs}^* = I_{qs}^{e*}\cos\theta_e + I_{ds}^{e*}\sin\theta_e \quad (23)$$

$$I_{ds}^* = -I_{qs}^{e*}\sin\theta_e + I_{ds}^{e*}\cos\theta_e \quad (24)$$

These q and d axes stationary reference currents (I_{qs}^* and I_{ds}^*) are converted to three phase reference currents (I_{sa}^*, I_{sb}^* and I_{sc}^*).

3.2.4 PWM current controller

The reference currents (I_{sa}^*, I_{sb}^* and I_{sc}^*) are compared with the sensed currents (I_{sa}, I_{sb} and I_{sc}). The current errors are amplified and compared with the triangular carrier wave to generate the gate pulses. If the amplified error signal corresponding to phase 'a' (I_{saer}) is greater than the triangular carrier wave signal then the switch S_1 (upper device) of the phase 'a' leg of the VSI bridge is ON and the switch S_4 (lower device) is OFF, and the value of the switching function $SA = 1$. If the amplified error signal corresponding to phase 'a' (I_{saer}) is less than the triangular carrier wave signal then the switch S_1 (upper device) is OFF and the switch S_4 (lower device) is ON, and $SA = 0$. Similar logic applies to other phases.

3.2.5 Modeling of the VSI

The derivative of the DC bus voltage is defined as:
When there is no load on the DC side of the inverter,

$$pv_{dc} = (i_{da}\mathrm{SA} + i_{db}\mathrm{SB} + i_{dc}\mathrm{SC}) / \mathrm{C_{dc}} \quad (25)$$

and when DC load is present, the equation (27) is modified as:

$$pv_{dc} = [(i_{da}\mathrm{SA} + i_{db}\mathrm{SB} + i_{dc}\mathrm{SC}) - (\frac{v_{dc}}{R_d})] / \mathrm{C_{dc}} \quad (26)$$

where SA, SB and SC are the switching functions for the ON/OFF positions of the VSI bridge switches S1-S6 and R_d is the DC load resistance.

The DC bus voltage reflects at the output of the inverter in the form of the three-phase PWM AC voltages v_a, v_b and v_c. These voltages may be expressed as:

$$v_a = V_{dc}\frac{(+2SA - SB - SC)}{3}$$

$$v_b = V_{dc}\frac{(-SA + 2SB - SC)}{3}$$

$$v_c = V_{dc}\frac{(-SA - SB + 2SC)}{3}$$

The line voltages may be expressed as:

$$v_{ab} = v_a - v_b = v_{dc}(\mathrm{SA} - \mathrm{SB}) \quad (27)$$

$$v_{bc} = v_b - v_c = v_{dc}(\mathrm{SB} - \mathrm{SC}) \quad (28)$$

$$v_{ca} = v_c - v_a = v_{dc}(\mathrm{SC} - \mathrm{SA}) \quad (29)$$

3.3 Modeling of the load

3.3.1 DC load

The current through the DC load resistance connected across the DC capacitor of the inverter is given by:

$$i_{dc} = \frac{v_{dc}}{R_{dc}} \quad (30)$$

3.3.2 Three phase resistive load

The phase currents i_{pa}, i_{pb} and i_{pc} in the three-phase load circuit are modeled as:

$$i_{pa} = \frac{v_{ab}}{R_{La}} \quad (31)$$

$$i_{pb} = \frac{v_{bc}}{R_{Lb}} \quad (32)$$

$$i_{pc} = \frac{v_{ca}}{R_{Lc}} \quad (33)$$

3.3.3 Three phase inductive load

The derivatives of the phase currents of the delta connected inductive load shown in Fig. 2 are given as:

$$pi_{pa} = (v_{ab} - R_{La} i_{pa})/L_{La} \quad (34)$$

$$pi_{pb} = (v_{bc} - R_{Lb} i_{pb})/L_{Lb} \quad (35)$$

$$pi_{pc} = (v_{ca} - R_{Lc} i_{pc})/L_{Lc} \quad (36)$$

From these phase currents of the load, the line currents of this delta-connected load are given by:

$$i_{La} = i_{pa} - i_{pc} \quad (37)$$

$$i_{Lb} = i_{pa} - i_{pc} \quad (38)$$

$$i_{Lc} = i_{pa} - i_{pc} \quad (39)$$

4 Simulation results and discussion

The developed models have been simulated in MATLAB/SIMULINK environment. The simulated results have been presented for no-load excitation, sudden application and removal of both AC and DC loads on a 2.2 kW, 3-phase, star connected induction machine. The parameters of the induction machine obtained by conducting DC resistance test, synchronous speed test and blocked rotor test are given in Appendix B.

4.1 Self-excitation of the SEIG

The voltage build-up of the generator and the DC voltage across the capacitor at no-load condition are shown in Fig. 3. The reference DC voltage is set at 600 V. It is found that the generator voltage remains small until the air-gap flux linkage is at low level, and thereafter, there is a rapid growth of voltage, which settles down to a steady-state value due to magnetizing flux saturation. The DC voltage is settled to the reference value. Fig. 4 gives the DC voltage build-up at no-load with different values of capacitors. It is found that when the capacitance value is large, it takes longer time to reach to steady-state value.

4.2 Sudden application and removal of load

4.2.1 Load application and removal on DC side

To investigate the response of the system with sudden application and removal of load on the DC side, the generator is initially excited at no-load and suddenly a DC load of (R_{dc} = 5.6 p.u.) is applied at t = 1.2 sec. and this load is removed at t = 1.8 sec. as shown in Fig. 5. At the time of application of load, the voltages t end to decrease but quickly return to

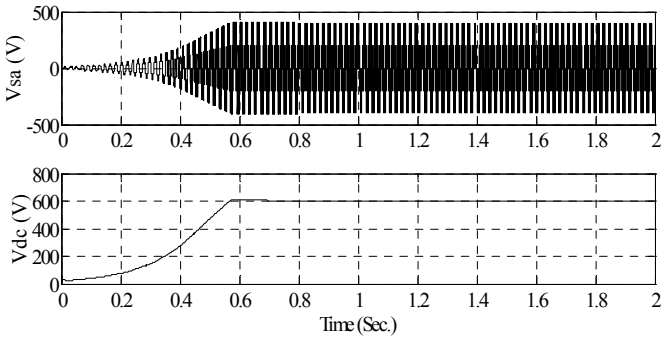


Fig. 3. Voltage build-up at no-load.

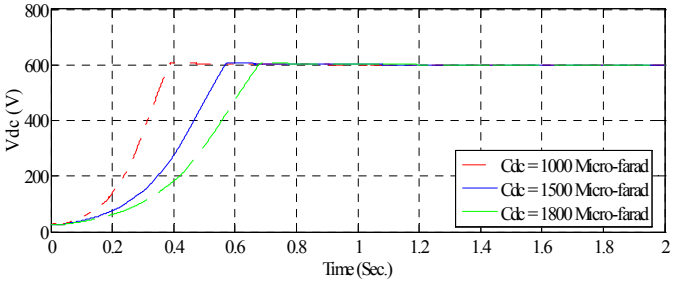


Fig. 4. DC voltage build-up at no-load for different values of capacitances.

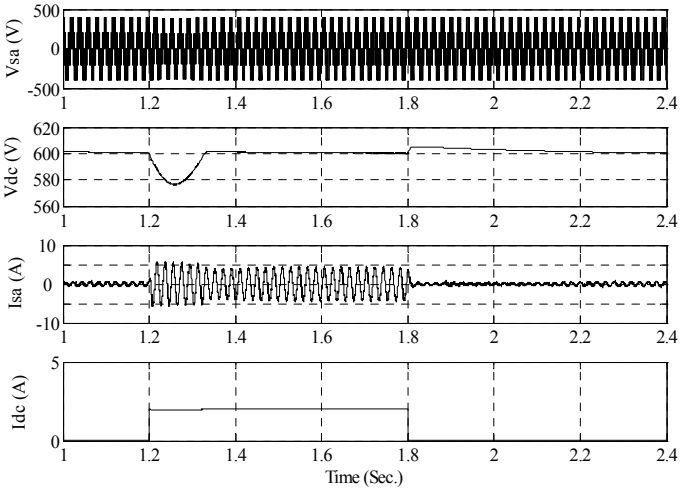


Fig. 5. Waveforms during application and removal of load on DC side.

reference values. When the load is removed, the phase voltage of the generator and the DC voltage increase due to mismatch in active power produced by the SEIG, which is more than the power consumed by the load. The duty cycle of the inverter is adjusted by the control action and the voltages are maintained at their reference values.

4.2.2 Load application and removal on AC side

To investigate the response of the system due to application and removal of balanced resistive load across the terminals of the SEIG, a resistive load (R_L = 6.6 p.u./phase) is applied at t = 1.4 sec. and removed at t = 2 sec. as shown in Fig. 6.

Similarly, the responses of the system due to load perturbation with balanced inductive load (R_L = 2.6 p.u./phase, X_L = 1.97 p.u./phase) of 0.8 pf are shown in Fig. 7. In both these cases, the voltage reduces during application and increases during removal of load due to mismatch of power generated and power consumed. The voltage comes to the reference value quickly due to control action. The phase current of the SEIG increases at the time of application of load.

Figs. 8 and 9 show the three-phase load currents, stator phase currents and stator phase voltages during sudden application of unbalanced three-phase resistive load (R_{La} = 5.6 p.u., R_{Lb} = 6.76 p.u. and R_{Lc} = 7.5 p.u.) and inductive load (R_{La} = 1.9 p.u., X_{La} = 1.4 p.u.; R_{Lb} = 2.8 p.u., X_{Lb} = 2.1 p.u. and R_{Lc} = 3.8 p.u., X_{Lc} = 2.8 p.u.) respectively. It is observed that stator phase voltages and currents are balanced even if the load current is unbalanced due to the PWM inverter action. Hence, the indirect vector control technique acts also as a good phase balancer.

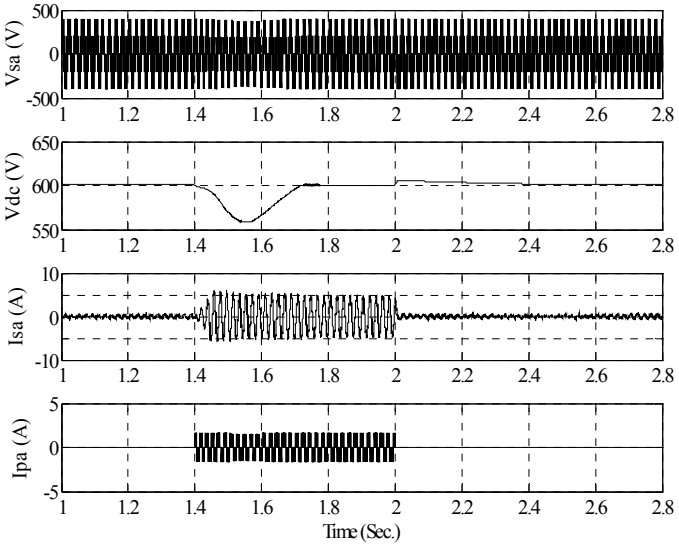


Fig. 6. Stator phase 'a' voltage, DC voltage, stator phase 'a' current and load current during application and removal of resistive load on SEIG terminals.

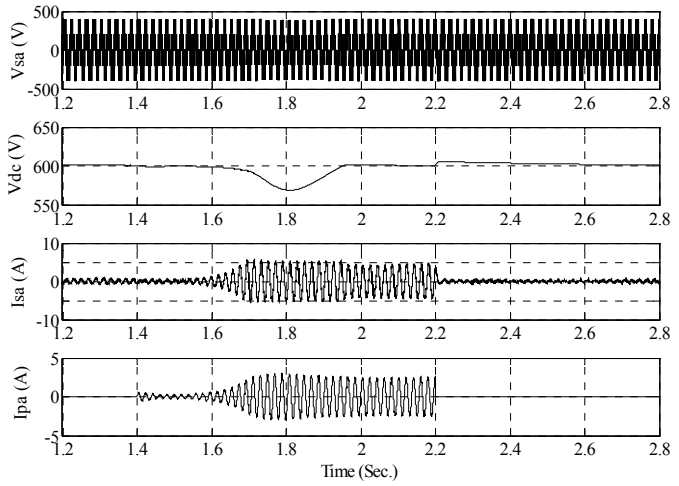


Fig. 7. Stator phase 'a' voltage, DC voltage, stator phase 'a' current and load current during application and removal of inductive load on SEIG terminals.

5 Conclusions

This paper provides a high performance variable speed induction generator system using indirect vector control technique with rotor flux orientation for small-scale AC and DC power applications. A PWM VAR compensator is used to control the flux in the generator and the reactive power balance. The excitation power is supplied from the capacitor connected on the DC side of the PWM inverter. The induction machine has been represented by space-phasor model. The developed models have been implemented using MATLAB/SIMULINK. The controller has been tested for different transient conditions such as voltage build-up, sudden application and removal of both the resistive and the inductive loads and also for unbalanced three-phase loads. It has a fast dynamic response, robust, reliable and very good phase balance even with unbalanced three-phase load.

6 Appendices

6.1 Appendix A

The dq stationary reference frame model of the induction machine is given by:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{qr} \\ v_{dr} \end{bmatrix} = \begin{bmatrix} R_s + L_s p & 0 & L_m p & 0 \\ 0 & R_s + L_s p & 0 & L_m p \\ L_m p & -\omega_r L_m & R_r + L_r p & -\omega_r L_r \\ \omega_r L_m & L_m p & \omega_r L_r & R_r + L_r p \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad \text{(A1)}$$

where $L_s = L_{ls} + L_m$ and $L_r = L_{lr} + L_m$.
Again, in dq reference frame, the stator and rotor flux linkages can be written as:

$$\begin{bmatrix} \lambda_{ds} \\ \lambda_{dr} \end{bmatrix} = \begin{bmatrix} L_{ls} + L_m & L_m \\ L_m & L_{lr} + L_m \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{dr} \end{bmatrix} \quad \text{(A2)}$$

$$\begin{bmatrix} \lambda_{qs} \\ \lambda_{qr} \end{bmatrix} = \begin{bmatrix} L_{ls} + L_m & L_m \\ L_m & L_{lr} + L_m \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{qr} \end{bmatrix} \quad \text{(A3)}$$

$$\text{Hence, } \begin{bmatrix} i_{ds} \\ i_{dr} \end{bmatrix} = \begin{bmatrix} L_{ls} + L_m & L_m \\ L_m & L_{lr} + L_m \end{bmatrix}^{-1} \begin{bmatrix} \lambda_{ds} \\ \lambda_{dr} \end{bmatrix} \quad \text{(A4)}$$

$$\text{Similarly, } \begin{bmatrix} \boldsymbol{i_{qs}} \\ \boldsymbol{i_{qr}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L_{ls} + L_m} & \boldsymbol{L_m} \\ \boldsymbol{L_m} & \boldsymbol{L_{lr} + L_m} \end{bmatrix}^{-1} \begin{bmatrix} \lambda_{qs} \\ \lambda_{qr} \end{bmatrix} \quad \text{(A5)}$$

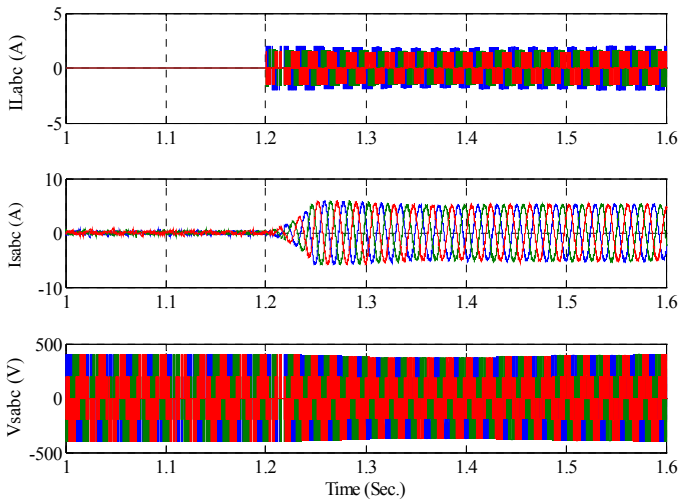


Fig. 8. Three-phase load currents, stator phase currents and stator phase voltages during sudden application of unbalanced resistive load on SEIG terminals.

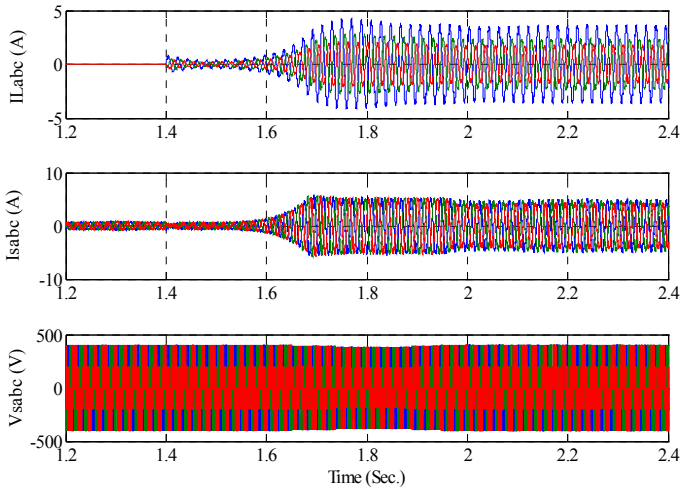


Fig. 9. Three-phase load currents, stator phase currents and stator phase voltages during sudden application of unbalanced inductive load on SEIG terminals.

6.2 Appendix B

The parameters of the induction machine are given as:
2.2 kW, 3-phase, 4-pole, 50 Hz, 415 V., 4.5 A., star connected, 1440 rpm, R_s = 3.84 Ω, R_r = 2.88 Ω, X_{ls} = 4.46 Ω, X_{lr} = 4.46 Ω, L_m = 0.2168 H, Base impedance = 53.24 Ω.

Coefficients of the Prime-mover Characteristics:
k_1 = 249.39, k_2 = 0.7875

7 References

[1] N. H. Malik and A. H. Al-Bahrani, "Influence of the terminal capacitor on the performance characteristics of a self-excited induction generator", IEE Proceedings, Part C, Vol. 137, No. 2, pp. 168-173, march 1990.

[2] S. M. Alghuwainem, "Steady-state analysis of a self-excited induction generator self-regulated by shunt saturable reactor", IEEE International Conference on Electrical Machines and Drives, pp. 101-103, 1997.

[3] S. S. Murthy, C. Prabhu, A. K. tandon and M. O. Vaishya, "Analysis of series compensated self-excited induction generators for autonomous power generation", IEEE Conference on Power Electronics, Drives and Energy Systems for Industrial Growth, pp. 687-693, 1996.

[4] M. B. Brennen and A. Abbondanti, "Static excitors for induction generators", IEEE Transactions on Industry Applications, Vol. 1A-13, pp. 422-428, September-October 1977.

[5] R. O. C. Lyra, S. R. Silva and P. C. Cortizo, "Direct and indirect flux control of an isolated induction generator", Proceedings of IEEE International Conference on Power Electronics and Drive Systems, Vol. 1, pp. 140-145, 1995.

[6] D. Seyoum, M. F. Rahman and C. Grantham, "Terminal voltage control of a wind turbine driven isolated induction generator using stator oriented field control", Proceedings of IEEE Power Electronics Conference and Exposition- APEC, Vol. 2, pp. 846-852, 2003.

[7] R. Leidhold, G. Garcia and M I. Valla, "Field-oriented controlled induction generator with loss minimization", IEEE Transactions on Industrial Electronics, Vol. 49, No. 1, pp. 147-156, February 2002.

[8] T. Ahmed, K. Nishida and M. Nakaoka, "Advanced voltage control of induction generator using rotor field-oriented control", Conference Record of IEEE Industry Applications Conference, Vol. 4, pp. 2835-2842, 2-6 October, 2005.

[9] Cardenas R. and Pena R., "Sensorless vector control of induction machines for variable speed wind energy applications", IEEE Trans. on Energy Conversion, Vol. 19, No. 1, pp. 196-205, 2004.

[10] Liao Y. W. and Levi E., **"**Modelling and simulation of a stand-alone induction generator with rotor flux oriented control", Electric Power Systems Research, Vol. 46, pp. 141-152, 1998.

[11] T. Ahmed, K. Nishida and M. Nakaoka, "A novel stand-alone induction generator system for AC and DC poer applications", IEEE Transactions on Industry Applications, Vol. 43, No. 6, pp. 1465-1474, November/December 2007.

[12] M. Pucci and M. Cirrincione, "Neural MPPT control of wind generators with induction machines without speed sensors", IEEE Transactions on Industrial Electronics, Vol. 58, No. 1, pp. 147-156, January 2011.

Computation as a Mesoscopic Phenomenon

M. J. George
Department of Physics, Southwestern College, Chula Vista, California, USA

Abstract – *This paper treats computation as a game process. We place an algorithm within a single computational domain. The domain structure is a graphical structure with the moves of the game representing vertices. The algorithmic process evolves through these moves. The noise of the physical computing device manifests a mesoscopic system. The algorithmic process is imbedded as part of this system. The edges of the graph have a time-based directionality. We treat the noise as time evolution of a quantum-like system along the edges. We identify both a quantum-like process and a thermal-like process for this evolution. We use a simple Ising model to provide the underlying structure. This allows the noise to be seen as an organized pattern evolution. Furthermore, there is a scale structure that one can relate to the renormalization group theory. This is relevant to modern computation, as it encroaches on the mesoscopic domain.*

Keywords: mesoscopic system, quantum Ising game

1 Introduction

Mesoscopic phenomena encompass the scientific domain between the strictly quantum realm and the complex classical world. Normally, computation is taken to reside purely in this latter realm, defined by elegant algorithms, and the purity of the switching apparatus of the digital computer. It seems absurd to consider computation as a mesoscopic phenomenon. And yet, we know that our computing devices are beginning to impinge on the strange quantum world more and more, and as each technological advance proceeds in processing, in nanotechnology, etc. we become less firmly entrenched in the secure classical world.

In the past, mesoscopic phenomena has for the most part not been regarded as anything other than noise, or "quantum mud" (see Ref. 12, for example). On the other hand, it is now recognized that ascribing much of this phenomena as noise was a result of a failure to address the nonlinear and nonequilibrium phenomena associated with the mesoscopic. For a review of mesoscopic phenomena, in general, see Ref. 1. For another useful review, see Ref. 2. This latter book focuses on applications of quantum mechanics to biology, but it also supplies a clear overview of many general areas on which quantum mechanics overlaps with phenomena close to the macroscopic scale (and hence intrudes in what we here refer to as the mesoscopic). As can be seen from Ref. 3, quantum field theory of phase transitions is definitely considering phenomena close to the mesoscopic.

Our work, while addressing the speculative domain of the mesoscopic world, is not intended as a speculative piece of work, but as a mathematical work, considering an approach to computation that utilizes physical ideas. Our main interest, looking ahead beyond what we have done here, is to provide a framework for extending the renormalization group theory [4]. This theory is perhaps the most advanced theory in thermodynamics. However, it is (despite the capability of incorporating time as a parameter) an equilibrium theory. The mesoscopic is very-well suited to discussing nonequilibrium, but scale-dependent phenomena. Therefore, we might expect that some progress might be made in extending the renormalization group theory slightly by consideration of mesoscopic phenomena.

We have previously worked at the type of mathematical framework to which we are referring above, in papers on non-equilibrium thermodynamics [5] and on Ising games [6]. The Ising model is discussed in many references, including Ref. 4. (In this reference, as well, certain aspects of renormalization of the Ising model are discussed.) The Ising model, as a simple but nontrivial model, and one that displays enormous versatility (by perhaps broadening our definition of Ising model to allow nonlocal interactions, vector fields, etc.) is well-suited to an attempt to express order in what has previously just been thought of as noise, i.e. mesoscopic phenomena. Game theory is also positive in this regard, as it reflects a sense of measurement, of control, of willful action that we can associate with our macroscopic experience, and yet can be developed in simple ways too that are nontrivial. Utilizing game theory outside the context of the social sciences (where equilibrium applications predominate) is discussed in Ref. 6. It should be born in mind that significant progress in the physical sciences is also likely to effect the equilibrium theories of the social sciences. Thus, the phenomena of social science can be thought of as mesoscopic to some extent. The quantum Ising game is a natural (albeit oversimplified) approach to discussing mesoscopic phenomena.

The paper is structured as follows: In Sec. 2, we present a nonmathematical perspective on our approach to mesoscopic computational systems. This is somewhat philosophical, and is intended to establish the intellectual framework in which to discuss computation as a natural phenomenon, and noise as expressive of (statistical) patterns akin to quantum mechanics. In the next section, we discuss the general mathematical background for using Ising games to study such phenomena. The idea that noise can be treated as having a phase structure, and therefore that a model like the Ising model might prove of interest, in a domain dominated by noise is merely an attempt to associate a mathematical structure, related to scaling and renormalization

group theory, with phenomena that are nonlinear and nonequilibrium. In Sec. 4, we present, in outline, a very simple example of the type of mathematical structure that we are referring to. No attempt is made in this section to display a complete theory. However, a concrete example helps to make our viewpoint more understandable. In Sec. 5, the conclusion, we make some general comments about our work, and give suggestions for future directions.

2 What is a mesoscopic computational system ?

We would like to answer the question of the title of this section fully. Unfortunately, the mere idea of computation in a mesoscopic regime is difficult to grasp. The "mesoscopic", as a concept, is one which, until recently, was simply interpreted as indicated part of the noise of the world. One notices, for example, in the scientific study of computation in supercomputers, such as done by the Borwein *et al.* [7], [8], that heavy selection procedures (to eliminate noise from consideration) by humans for directing computation cloud possible mesoscopic phenomena. When one focuses on energy, with large energy expenditure to maintain order, the quantum aspects of the computation may be hidden, simply because $h/\Delta E$ is so small that linear quantum effects are not observable [9]. (Here, h is Planck's constant, and ΔE is an energy fluctuation in the quantum system. The ratio $h/\Delta E$ is associated with the time over which this energy fluctuation would yield an observable effect.) Coupling a supercomputer with a mesoscopic computational system, such as the brain seems to be, may be a better way to observe mesoscopic effects [2]. If computers can be guided or subject to selection principles that force the appearance of classical behavior, what evidence do we have that computers are anything but classical systems? We claim that the selection process can be modeled as a game process. The imposition of game processing leads to the appearance of a classical system. We cannot prove this, but removal of the heavy game element to computing would result in the appearance of mesoscopic phenomena. In the mathematical sections following this section, we hope to address this to some extent. The easiest way to consider this is a "semi-classical" approach in which games at different scales overlap to some extent. The idea of scaling then arises, and it is possible to discuss renormalization. This goes beyond the scope of this paper, but is considered to some extent later.

The viewpoint of considering computation as a natural phenomenon is unusual: We humans are very attached to game behavior (and this can mask non-classical or non-equilibrium system characteristics). Forcing a system to behave linearly [2] by the imposition of games, for example, does not mean that one is dealing with a classical system. The needed experimental investigations and observations may be difficult to make, because they are at the edge of our computational capabilities. However, one feels some confidence that such data as results from these observations might produce some subtle signs that our computational systems display mesoscopic behavior. The idea that we can take computation as a mesoscopic phenomenon is merely speculative. However, we can certainly build a mathematical model expressive of this idea.

3 Mathematical discussion

We can use a quantum Ising game to represent a mesoscopic system. These types of models are very simple, and cannot be considered, in detail, to accurately represent actual mesoscopic systems except in certain limited, idealized respects. The perception of noise as complex phase phenomena is a mathematical viewpoint we are taking, in the absence of much actual scientific knowledge about mesoscopic systems (other than interpreting much of the phenomena of such systems as noise). Therefore, our viewpoint is merely mathematical. However, a simple model can often present us with clues to progress further.

The basic object of a quantum Ising game [6] can be chosen somewhat arbitrarily. However, we select it as a connection, somewhat like a quantum amplitude between two states, that can have an interpretation in terms of correlation. This allows us to develop a perspective in which graphs and phase interpretations can be naturally used, and in which one can easily interpolate between classical and quantum domains, representing the intermediate domain, the "mesoscopic" domain, as a substantial part of the overall picture. Although a quantum Ising game is not a quantum system (it is intrinsically nonlinear), we still refer to the system configurations as states. So, in quantum Ising games, the states, for computational systems, will be taken as matrices.

A computational system is viewed as prepared in a state $A_{x,t}$ where x specifies a node (which we think of as the insertion point of the computational system), that is a part of a graphical structure. This initial state poses the questions that we wish to be answered by the game. The entries of the matrix are polynomials, not necessarily numbers, and a polynomial, as a sequence of coefficients, can be regarded not only as an arithmetic object that can be computationally operated on, but also as a piece of information that has a certain meaning.

The graphical structure represents a "domain" of the computational system, which is to say, a unified processing structure. The state is inserted in the computational network at time *t*. The nodes are sites at which "moves" are made in the game. Thus, $A_{x,t}$ is a first move, into this domain, for a computing process. We represent the computing domain as the graph *G*, and for the purposes of our definition of a basic domain of the game, we think of *G* as a connected pseudo-multigraph. This accords with viewing the computational process as an integrated process, and is consistent with conventional views of computation.

We call *G* a pseudo-multigraph because, while it has edges that connect to the insertion node *x*, this node is not part of *G*. Thus, we are considering a situation in which vertices of edges (i.e. edges that are part of the graph) are not themselves considered to be part of the graph, or, using our

terminology, pseudo-graph. Also, we use the term multigraph to denote the possibility of multiple edges between vertices. One associates with a computational process the possibility of feedback, and this is why we allow multiple edges. Also, a computational system is going to have a certain "directionality" with respect to temporal flow. However, this does not need to be built into the computational system, i.e. we do not need to consider the graph to be directed.

For a quantum Ising game that represents a mesoscopic system, we can have several insertion points. This amounts to an obvious generalization we do not discuss. We may also include several domains, with domain walls. However, in this paper, we restrict our attention to one domain, only, and do not discuss these more complex issues. In a way, we are just using the idea of a bridge between classical and quantum phases, since there is little evidence that mesoscopic phenomena is anything more than "noise". Thus, since we are developing mathematics by physical analogy, there is little point in delving into complications, at least at this time.

There are nodes, outside the pseudo-multigraph, other than the insertion node, to which edges of the graph lead, that are "measurement" points, in traditional quantum mechanics, but which we think of as moves in a game. For simplicity, we merely think of a single such move, $B_{y,T}$. We represent the entire computational process as $B_{y,T}GA_{x,t}$. We think of y as the node corresponding to the move $B_{y,T}$, and T as the time at which the measurement is made. We refer to $B_{y,T}$ as the dual state, and its entries represent the result of measurement. As the state A and the dual state B represent, respectively, the inserted values of the computation and the output values, they need not represent the same type of quantities. The computation always terminates, despite the possibility of "infinite loops" in the domain processor: Such feedback, however, may strongly influence the entries of the dual state. (The computational process resolves itself into a finite pattern, including loops, that traces paths through the domain, and persists only from time t to time T, while looping behavior may persist beyond this time interval.) As well, the moves in the processor (as we refer to G) are each labeled with the times at which the moves are made: This "time-orders" the processing, and delineates the various paths of the computation.

The heart of a quantum Ising game is the structure along a single edge, joined by vertices. It is this structure that yields the mesoscopic nature of the game. For a real computational device, one must think in terms of changes of scale to understand how an actual computational system can be regarded as mesoscopic. When scale is changed, the structure of the game must be renormalized to that scale. While this does not alter the nature of the moves, the mathematical structure of the moves can change. This scale exists because there are structural elements of the computational process along the edges. These structural components would ordinarily have been relegated to noise that is not going to affect the outcome of the computation (usually), but which we are now interpreting as phase patterns in an Ising model.

Quantum mechanically, this, were it merely a quantum system, would correspond to a system of pseudo-particles, i.e. coherent aspects of an underlying quantum system. However, we treat a mesoscopic system as inherently an Ising game, and, as opposed to quantum mechanics, the phase structure of Ising games, quantum or not, will always be nonlinear. The underlying paradigm of Hilbert space for quantum mechanics, or the subtler paradigm of Banach space (think: Wavelets) for information theory are only first approximations in a game structure. In a very short paper such as this, one can hardly expect to address the issues involved in nonlinear phenomena, renormalization group theory, and even beyond this into systems theory. We will end our mathematical discussion instead with an extremely simple example to illustrate something of what is involved.

4 A simple example

For the purposes of illustration, we picture the state A as being formed from two numerical matrices, with arbitrarily chosen entries:

$$A = \begin{bmatrix} 1 & -2 \\ 0 & 4 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ -1 & 0 \end{bmatrix} \qquad (1)$$

This describes a "state" at a vertex, i.e. we picture that a "move" was made that resulted in this state, at time $t = 0$. One can think of this as some "intentional" move made by a human (or an "artificial" computing device) or as resulting from some "impersonal" move of nature (we do not think of games as necessarily related to "willful" intent). We have split the state into two "spin" states, where we picture that the underlying quantum spin system consists of just two spins (with spin operators acting on this state, spin 1 operator acting on the first matrix and spin 2 operator acting on the second). An Ising model with two spins (which we regard as "nearest-neighbors" for the example) isn't much of an Ising model, but, as we stated, we need to keep the example simple for this short paper. In addition, a better treatment would work with "renormalized" systems.

We will take one time step "to produce" the next "move". You can visualize this as the time evolution along a single edge. We are not going to get into complications about time ordering involving multiple edges or more complex time evolution along the edge, nor are we going to consider the resulting "move" as involved with quantum measurement, although it could be thought of as a quantum state for a determination of an amplitude from some "experimental" arrangement associated with the computation.

We can think of the time evolution between vertices, i.e. from one move to the next, as an incremental process, marked off by time increments and referring to progressive transformations along the edge joining the two vertices. The "arrow of time" supplies a natural direction for the edge. One vertex is associated with time $t = 0$ while we can suppose that the other vertex is associated with time $t = 1$. For simplicity, let us suppose that we need consider only two segments along this edge, each of "duration", $\Delta t = 1/2$: one

(interval 1) from time $t = 0$ to $t = 1/2$, and another (interval 2), from $t = 1/2$ to $t = 1$. We will treat the process in each time interval as being represented as a simple perturbation from the identity operator. We can take such operators as simple exponentials of matrices. Thus, the matrices can simply be built from the Ising Hamiltonian.

These Hamiltonian operators are called such, by virtue of using exponentiation of matrices, with an analogy to the manner in which the Ising model is constructed, and not because of any symmetry considerations. For example, they need not be Hermitian operators. One is dealing with the basic way in which the computational system operates between moves. These moves would be associated with our usual expectations on computational resolution of an algorithm, while the intermediate processing that occurs along an edge relates to nonlinear or nonequilibrium aspects of computation that under normal, macroscopic circumstances we would merely regard as noise.

What would these computational operators, relating to what we previously viewed as noise, perhaps, look like? We suggest a very simple Ising-like model be used for this, because of the versatility of such models as well as their simplicity. For the Hamiltonian operator, H_1, in the first time interval from $t = 0$ to $t = 1/2$, we will use a simple symmetrized operator for our example,

$$H_1 = S_1S_2 + S_2S_1 \qquad (2)$$

where S_1 and S_2 are 2 x 2 matrices, the matrix S_1 applying to the first "component" matrix of A in (1), and S_2 to the second. We can picture these matrices as having simple numerical components, as the matrices for A have. In general, we would want to think of the Hamiltonians as having polynomial components. The approach we can take to devising these operators can be based either on actual experimental data or numerical simulations of the mesoscopic domain, or can be thought about in terms of overlapping graphical structures.

It is worthwhile to make a few comments about this latter possibility. We have already mentioned, in a previous section, that we can view the computational structures as having domains, with an integrated level of algorithms associated in each domain, and there being a certain barrier between domains, i.e. domain walls analogous with ferromagnetism. On the other hand, we can also think of the integrated algorithmic structures as overlapping, the moves of one structure corresponding to certain vertices, and the moves of another structure to others. In a certain restricted area, the graph of one structure might be "fine-grained" relative to another structure. Then, between two consecutive moves of the coarse-grained algorithm, there may be several intermediate steps of the fine-grained algorithm. These intermediate steps could be viewed as the processes along an edge of the graph associated with the coarse-grained algorithm.

To persist with our picture of a quantum Ising game, we use a "Planck's constant", h, which is merely taken to be some scale factor. The time evolution, over the first time interval, $\Delta t = 1/2$, from $t = 0$ to $t = 1/2$, is given (just for the purpose of our example) by the "rotation",

$$\exp(iH_1\,\Delta t/h) \qquad (3)$$

We will not consider the effect of such an operator in "deflecting" the game process. The level of analysis goes beyond this paper. This is a "quantum-like" step. We can also consider "thermal-like" steps. For example, we can require the following time step, from $t = 1/2$ to $t = 1$, to be a thermal-like time-step, involving some real-number, γ, which we can think of as a reciprocal temperature (multiplied by "Boltzmann's constant"), with a reciprocal time dimension (i.e. γ is a rate). Suppose that the Hamiltonian operator that we have constructed for this second time step is H_2 (which need not be the same as H_1, either in form, or in detailed assignments of components). This thermal time step is associated with the "compression",

$$\exp(-\gamma H_2\Delta t) \qquad (4)$$

Since we have made allowance for only two processes (3) and (4) along the edge, if we call the move at time 1, B, we can describe this segment of the Ising game (such an impersonal game!) by

$$B\exp(-\gamma H_2\Delta t)\exp(iH_1\,\Delta t/h)A \qquad (5)$$

Here, B is some state, just as A. This takes us through a simple computation along an edge and completes our example. Note that because the Hamiltonians are just organized as part of a computation, the types of entries, e.g. real vs. complex, are unconstrained. Therefore, as well, the separation between quantum-like and thermal-like operators is merely artificial. We used specific numbers as components for the matrices, but one can leave the entire computation formal, using polynomials as components (or perhaps more general types of objects).

5 Conclusion

Besides the very limited nature of the example we gave in Sec. 4 (constrained due to the shortness of this paper), there is also the question of the computations that can be encompassed within the context of a quantum Ising game. It is necessary, first, to point out that such a game addresses essentially both nonlinear and nonequilibrium aspects of modeling. The game does not fall into the context of quantum field theory or quantum mechanics, although it does provide a context for discussion of renormalization, beyond the current theory of renormalization group.

In our example, we supplied just a very simple, Ising-like structure for the operators. Even this is not simple to investigate, as one is dealing with whole graphical structures related to a Hamiltonian, not to mention the graphical structure of the "moves". This supplies the possibility of realizing quite an intricate set of scales to consider in the context of renormalization, but one also wonders how to

develop this into a recognizable pattern of computations. When a mesoscopic level is contemplated, even in the very simple way we approach it here, the physical significance is not apparent. One suggestion would be to study this from the viewpoint of scalars, rather than 2 x 2 matrices as we have done here. The Hamiltonian structure can be related to patterns beyond the simple Ising-like patterns, using operators that have polynomial form, for example.

The two-level graphical structure involved with a quantum Ising game is important to emphasize. The game structure is associated with vertices, and the time evolution along edges, much like quantum field theory, we have associated with Ising-like Hamiltonians. The overlapping possibility for different scales, along the edges, is, we feel, essential to considering extending the renormalization group theory slightly, using the game structure. The unconventional physical context of the theory we have discussed may suggest that we are proposing some speculative physical theory. This is not the case. The discussion is oriented toward a mathematical framework only, and one that introduces (via the edge processes) a mesoscopic framework for thinking about computation.

Although this does not lead to the possibility of treating computation as a natural phenomenon, it certainly suggests this viewpoint. We can then ask ourselves what sort of physical systems might eventually provide an environment for scientific prediction, based on treating computation in this manner. Because human social organizations seem so obviously to allow treatment in a game framework, such structures might eventually fall in the category of mesoscopic computational systems. We can then think of the possibility of developing an associated scientific theory that might have predictive capabilities in the social sciences, not simply the physical sciences. At this time, available experimental work on mesoscopic systems (in the physical sciences) is not sufficiently advanced to encompass some extension of the renormalization group theory, using this approach, other than in a very speculative way.

In future work, we would like to discuss some of the major points that we have only been able to touch on above. Furthermore, although the current theory of quantum Ising games is purely mathematical, with progress in experimental work on mesoscopic phenomena, a physical theory of computation, i.e. as a natural phenomenon, may result from the Ising game framework. Often, such a simple framework can eventually lead to some significant insights. As mentioned in the introduction, computing devices are impinging more and more on the quantum realm, and such a theory may be quite relevant in the near future.

6 References

[1] Y. Imry. "Introduction to Mesoscopic Physics". Oxford, 1997.

[2] J. McFadden. "Quantum Evolution". Norton, 2001.

[3] S. Sachdev. "Quantum Phase Transitions". Cambridge, 1999.

[4] D. J. Amit, "Field Theory, Renormalization Group, and Critical Phenomena". McGraw-Hill, 1978.

[5] M. J. George. "A Nonequilibrium Model Based on Latin Squares"; Proceedings of the International Conference on Scientific Computing, 24 – 29, 2007.

[6] M. J. George. "Classical and Quantum Ising Games"; International Journal of Pure and Applied Mathematics, 42, 529 – 534, 2008.

[7] J. Borwein and D. Bailey. "Mathematics by Experiment", A. K. Peters, 2004.

[8] J. Borwein, D. Bailey and R. Girgensohn. "Experimentation in Mathematics", A. K. Peters, 2004.

[9] L. D. Landau and E. M. Lifshitz. "Quantum Mechanics", Pergamon, 1977.

[10] J. R. Flynn. "What Is Intelligence?", Cambridge, 2007.

[11] P. M. Chaikin and T. C. Lubensky. "Principles of Condensed Matter Physics", Cambridge, 1995.

[12] H. D. I. Abarbanel. "Physics of Chaotic Systems", in "The New Physics", ed. G. Fraser, Cambridge, 2006.

A Modified EMD Algorithm and its Applications

Mayer Humi[1]
[1]Department of Mathematical Sciences,
Worcester Polytechnic Institute,
Worcester, MA 01609, USA

Abstract— *The classical EMD algorithm has been used extensively in the literature to decompose signals that contain nonlinear waves. However when a signal contain two or more frequencies that are close to one another the decomposition might fail. In this paper we propose a new formulation of this algorithm which is based on the zero crossings of the signal and show that it performs well even when the classical algorithm fail. We address also the filtering properties and convergence rate of the new algorithm versus the classical EMD algorithm. These properties are compared then to those of the principal component algorithm (PCA). Finally we apply this algorithm to the detection of gravity waves in the atmosphere.*

Keywords: Filtering, EMD algorithm

1. Introduction

In scientific literature there exist many classical sets of functions which can decompose a signal in terms of "simple" functions. For example Taylor or Fourier expansions are used routinely in scientific and engineering applications.(and many other exist). However in all these expansions the underlying functions are not intrinsic to the signal itself and a precise approximation to the original signal might require a large number of terms. This problem become even more acute when the signal is non-stationary and the process it represents is nonlinear.

To overcome this problem many researchers used in the past the "principal component algorithm" (PCA) to come up with an "adaptive" set of functions which approximate a given signal. A new approach to this problem emerged in the late 1990's when a NASA team has developed the "Empirical Mode Decomposition" algorithm(EMD) which attempt to decompose a signal in terms of it "intrinsic mode functions"(IMF) through a "sifting algorithm". A patent for this algorithm has been issued [1].

The EMD algorithm is based on the following quote [2]: "According to Drazin the first step of data analysis is to examine the data by eye. From this examination, one can immediately identify the different scales directly in two ways: by the time lapse between successive alterations of local maxima and minima and by the time lapse between the successive zero crossings....We have decided to to adopt the time lapse between successive extrema as the definition of the time scale for the intrinsic oscillatory mode"

A step by step description of the EMD sifting algorithm is as follows:

1) Let be given a function $f(t)$ which is sampled at discrete times $\{t_k, k = 1, \dots n\}$.
2) let $h_0(k) = f(t_k)$.
3) Identify the max and min of $h_0(k)$.
4) Create the cubic spline curve M_x that connects the maxima points. Do the same for the minima M_n. This creates an envelope for $h_0(k)$.
5) At each time t_k evaluate the mean m_k of M_x and M_n (m_k is referred to as the sifting function).
6) Evaluate $h_1(k) = h_0(k) - m_k$.
7) If norm of $||h_0 - h_1|| < \epsilon$ for some predetermined ϵ set the first intrinsic function $IMF_1 = h_1$ (and stop).
8) if the criteria of (7) are not satisfied set $h_0(k) = h_1(k)$ and return to (3) ("Sifting process").

The algorithm has been applied successfully in various physical applications. However as has been observed by Flandrin [3] and others the EMD algorithm fails in many cases where the data contains two or more frequencies which are close to each other.

To overcome this difficulty we propose hereby a modification of the EMD algorithm by replacing steps 4 and 5 in the description above by the following:

4. find the midpoints between two consecutive maxima and minima and let N_k be the values of h_0 at these points.
5. Create the spline curve m_k that connects the points N_k.

The essence of this modification is the replacement of the mean which is evaluated by the EMD algorithm as the average of the max-min envelopes by the spline curve of the mid-points between the maxima and minima. This is

in line with the observation by Drazin (which was referred to above) that the scales inherent to the data can be educed either from the max-min or its zero crossing. In the algorithm we propose hereby we mimic the "zero-crossings" by the mid-points between the max-min.

It is our objective in this paper to justify this modification of the EMD algorithm through some examples and theoretical work. The plan of the paper is as follows: In Sec. 2 we provides examples of signals composed two or three close frequencies (with and without noise) where the classical EMD algorithm fails but the modified one yields satisfactory results. In Sec. 3 we carry out analytical analysis of the two algorithms which are applied to the same signal. In Sec. 4 we discuss the convergence rate, resolution and related issues concerning the classical and new "midpoint algorithm" . Sec. 5 address the application of this algorithm to atmospheric data and in Sec. 6 we compare the EMD and PCA algorithms

2. Examples and Comparisons

Extensive experimentations were made to test and verify the efficiency of the modified algorithm. We present here the results of one of these tests in which the signal contains three close frequencies. (In our tests we considered also the effects of noise and phase shifts among the different frequencies)

$$f(t) = \frac{1}{3}[\cos(\omega_1 t) + \cos(\omega_2 t) + \cos(\omega_3 t)] \qquad (1)$$

where

$$\omega_1 = 12\omega_0,\ \ \omega_2 = 10\omega_0,\ \ \omega_3 = 8\omega_0,\ \ \omega_0 = \frac{\pi}{256}.$$

To apply the EMD algorithm to this signal, we used a discrete representation of it over the interval $[-2048, 2048]$ by letting $t_{k+1} - t_k = 1$, $k = 1, \ldots, 4097$.

The results of the signal decompositions into IMFs and a comparison these IMFs with the frequencies present in the original signal are presented in figures $1 - 5$. In all these figures the red lines represent the frequencies in the original signal (or its power spectrum) and the blue lines the corresponding intrinsic mode functions or their power spectrum which were obtained by the midpoint algorithm.

Fig. 1 is a plot of the data for the signal described by (1). Fig. 2 represents the first IMF in the decomposition (versus the leading frequency in the data) while Figs. $3 - 5$ depict the spectral density distribution for the first three IMFs versus those related to the original frequencies in the data. It should be observed that although the amplitude of the spectral densities in these plots are different (especially for IMF 3) the maxima of the spectral density in each plot is very close to the original one.

The EMD algorithm is a high pass filter. For the $n - th$ iteration of the filter its efficiency is measured by the parameter α which is defined by

$$Y_n = \alpha_n Y_{n-1} + \alpha(X_n - X_{n-1})$$

where X_k and Y_k are the input and output of the $k - th$ iteration. Fig 6 present the value of the parameter α as a function of the iteration number for first IMF derived from the data of the signal in (1).

3. Some Analytical Insights

To obtain analytical insights about the performance of the EMD-midpoint algorithm we considered the following signal

$$f(t) = \frac{1}{2}[\cos(\omega_4 t) + \cos(\omega_5 t)],\ \ \omega_4 = \frac{3\pi}{64},\ \ \omega_5 = \frac{\pi}{32}. \quad (1)$$

Since the ratio of the frequencies in this signal is a rational number the signal is actually periodic with period $p = 128$ (See Fig. 7) and the behavior of the classical versus the midpoint algorithm can be delineated analytically (i.e without discretizations).

On the interval $[0, p]$ the extrema of the signal are given by $\frac{df}{dt} = 0$ and therefore it is easy to construct the spline approximation $S_{max}(t)$, $S_{min}(t)$ to the maximum and minimum points and compute their average. Similarly we can find the midpoints between the maxima and minima and evaluate the corresponding spline approximation $S_{mid}(t)$ to the signal at these points. after one iteration of the sifting process the "sifted signal" is given respectively by

$$h_{mn}(t) = f(t) - \frac{S_{max}(t) + S_{min}(t)}{2}, \qquad (2)$$

and

$$h_{mid}(t) = f(t) - S_{mid}(t). \qquad (3)$$

The efficiency of the two algorithm can be deduced by projecting these new signals on the Fourier components of the original signal. To this end we compute

$$a_{mn} = \int_0^p h_{mn}(t)\cos(\omega_4 t)dt,\ \ b_{mn} = \int_0^p h_{mn}(t)\sin(\omega_4 t)dt. \qquad (4)$$

$$c_{mn} = \int_0^p h_{mn}(t)\cos(\omega_5 t)dt,\ \ d_{mn} = \int_0^p h_{mn}(t)\sin(\omega_5 t)dt. \qquad (5)$$

and

$$a_{mid} = \int_0^p h_{mid}(t)\cos(\omega_4 t)dt,\ \ b_{mid} = \int_0^p h_{mid}(t)\sin(\omega_5 t)dt. \qquad (6)$$

$$c_{mid} = \int_0^p h_{mid}(t)\cos(\omega_4 t)dt, \quad d_{mid} = \int_0^p h_{mid}(t)\sin(\omega_5 t)dt. \tag{7}$$

The amplitude of the Fourier components of the two frequencies in the classical EMD algorithm is

$$A_{mn} = \sqrt{a_{mn}^2 + b_{mn}^2}, \quad B_{mn} = \sqrt{c_{mn}^2 + d_{mn}^2}. \tag{8}$$

Similarly for the mid-point algorithm we

$$A_{mid} = \sqrt{a_{mid}^2 + b_{mid}^2}, \quad B_{mid} = \sqrt{c_{mid}^2 + d_{mid}^2}. \tag{9}$$

The objective of the sifting process is to eliminate one of the Fourier components in favor of the other. As a result the first IMF will contains, upon convergence, only one of the Fourier components in the original signal. Therefore the efficiency of the two algorithm can be inferred by comparing A_{mn} versus B_{mn} and A_{mid} versus B_{mid}. Computing the integrals that appear in eqs.(4)-(7) we obtain

$$A_{mn} = 31.63346911, \quad B_{mn} = 29.70292046, \tag{10}$$

$$A_{mid} = 34.19647843, \quad B_{mid} = 20.81145369. \tag{11}$$

These results show that after one iteration the classical EMD did not separate the two frequencies effectively. On the other hand the mid-point algorithm performed well.

4. Convergence Rates

To compare the convergence rates of the classical versus the midpoint algorithm we considered three cases all of which were composed of two frequencies. In the first case the two frequencies were well separated. In the second case the two frequencies were close while in the third case they were almost "overlapping". In all cases the signal was given by

$$f(t) = \frac{1}{2}(\cos\omega_1 t + cos\omega_2 t)$$

This signal was discretized on the interval $[-2048, 2048]$ with $\Delta t = 1$.

For the first case the two frequencies were

$$\omega_1 = 12\omega, \quad \omega_2 = 8\omega, \quad \omega = \frac{\pi}{256}.$$

As can be expected both the classical and midpoint algorithm were able to discern the individual frequencies through the sifting algorithm. However it took the classical algorithm 59 iterations to converge to the first IMF. On the other hand the midpoint algorithm converged in only 7 iterations (using the same convergence criteria). We wish to point out also that the midpoint algorithm has a lower computational cost than the classical algorithm. It requires in each iteration the computation of only one spline interpolating polynomial. On the other hand the classical algorithm requires two such polynomials, one for the maximum points and one for the minimum points.

For the second test the frequencies were

$$\omega_1 = \frac{\pi}{24} + \frac{\pi}{288}, \quad \omega_2 = \frac{\pi}{24} - \frac{\pi}{288}$$

that is the difference between the two frequencies is $\frac{\pi}{144}$.

In this case the midpoint algorithm was able to separate the two frequencies. Fig 8 and Fig 9 compare the power spectrum of the original frequencies versus those of IMF_1 and IMF_2 which were obtained through this algorithm. Convergence to IMF_1 was obtained in 18 iterations and IMF_2 was obtained by 7 additional iterations.

The classical EMD algorithm did converge to IMF_1 in 45 iterations but the power spectrum of this IMF deviated significantly from the first frequency in the signal(See Fig 10). IMF_2 failed (completely) to detected correctly the second frequency.

In third case the frequencies were

$$\omega_1 = \frac{\pi}{24} + \frac{\pi}{1000}, \quad \omega_1 = \frac{\pi}{24} - \frac{\pi}{1000}.$$

In this case the classical algorithm was unable to separate the two frequencies i.e IMF_1 contained both frequencies (See Fig 11). The midpoint algorithm did somewhat better but the resolution was not complete (See Fig 12). Moreover the sifting process in both cases led to the creation of "ghost frequencies" which were not present in the original signal.

At this juncture one might wonder if a "hybrid algorithm" whereby the sifting function is the average (or some similar combination) of those obtained by the classical and midpoint algorithms might outperform the separate algorithms (in spite of the obvious additional computational cost). However our experimentations with such algorithm did not yield the desired results (i.e. the convergence rate and resolution did not improve).

5. Applications to Atmospheric Data

There have been recent interest in the observation and properties of gravity waves which are generated when wind is blowing over terrain. In part this interest stems from the fact that these waves carry energy and accurate measure of this data is needed to improve the performance of numerical weather prediction models.

As part of this scientific campaign the USAF flew several balloons that collected information about the pressure and temperature as a function of height. The temperature data

collected by one of these balloons is presented in Fig. 13 [6]. To analyze this signal we detrended first it by subtracting its mean from the data. When the mid-point EMD algorithm was applied to this detrended-signal the first IMF extracted the experimental noise from while the second and third IMFs educed clearly the gravity waves (the second IMF is depicted in Fig. 14). On the other hand the classical EMD algorithm failed to educe these waves from the detrended-signal.

Subtracting the gravity waves that were detected by the mid-point algorithm from the detrended-signal we obtain the "turbulent residuals" whose spectrum is shown in Fig 15. The slope of this signal in the "inertial frequency range" is -2.7 which corresponds well with the fact that the flow in stratosphere is "quasi two-dimensional" [7-9].

6. EMD or PCA- A Comparison

Before the emergence of the EMD algorithm an adaptive data analysis was provided by the "Principal Component Algorithm"(PCA) which is referred to also as the "Karahunan-Loeve (K-L) decomposition algorithm". (For a review see [10]) Here we shall give only a brief overview of this algorithm within in the geophysical context.

Let a signal be represented by a a time series X (of length N) of some variable.We first determine a time delay Δ for which the points in the series are decorrelated. Using Δ we create n copies of the original series

$$X(k), \;\; X(d+\Delta), \ldots, X(k+(n-1)\Delta).$$

(To create these one uses either periodicity or choose to consider shorter time-series). Then one computes the auto-covariance matrix $R = (R_{ij})$

$$R_{ij} = \sum_{k=1}^{N} X(k+i\Delta)X(k+j\Delta). \tag{1}$$

Let $\lambda_0 > \lambda_1, \ldots, > \lambda_{n-1}$ be the eigenvalues of R with their corresponding eigenvectors

$$\phi^i = (\phi_0^i, \ldots, \phi_{n-1}^i), \;\; i = 0, \ldots, n-1.$$

The original time series X can be reconstructed then as

$$X(j) = \sum_{k=0}^{n-1} a_k(j)\phi_0^k \tag{2}$$

where

$$a_k(j) = \frac{1}{n}\sum_{i=0}^{n-1} X(j+i\Delta)\phi_i^k. \tag{3}$$

The essence of the PCA is based on the recognition that if a large spectral gap exists after the first m_1 eigenvalues of R then one can reconstruct the mean flow (or the large component (of the data by using only the first m_1 eigenfunctions in (2). A recent refinement of this procedure due to Ghil et al ([10]) is that the data corresponding to eigenvalues between $m_1 + 1$ and up to the point m_2 where they start to form a "continuum" represent waves. The location of m_2 can be ascertained further by applying the tests devised by Axford [11] and Dewan [7].

Thus the original data can be decomposed into mean flow, waves and residuals (i.e. data corresponding to eigenvalues $m_2+1, \ldots, n-1$ which we wish to interpret at least partly as turbulent residuals).

The crucial step in this algorithm is the determination of the points m_1 and m_2 whose position has to ascertained by additional tests whose results might be equivocal.

We applied this algorithm to the geophysical data described in Sec. 5.1 with $\Delta = 96$ and computed the resulting spectrum of the correlation matrix R. This spectrum is depicted in Fig. 16 . Based on this spectrum we choose $m_1 = 3$ and $m_2 = 11$ we obtain the corresponding wave component of the signal that is shown in Fig. 17.

We conclude that while the PCA algorithm provides an alternative to the EMD algorithm the determination of the cutoff points is murky in many cases. However it will be advantageous if one apply the two algorithms in tandem in order to obtain a clear cut confirmation of the results.

References

1 N. E. Huang - USA Patent $\#6,311,130B1$, Date Oct 30,2001

2 N. E. Huang et all, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis", Proceedings of the Royal Society Vol. 454 pp.903-995 (1998)

3 Gabriel Rilling and Patrick Flandrin, "One or Two Frequencies? The Empirical Mode Decomposition Answers", IEEE Trans. Signal Analysis Vol. 56 pp.85-95 (2008).

4 Zhaohua Wu and Norden E. Huang, "On the Filtering Properties of the Empirical Mode Decomposition, Advances in Adaptive Data Analysis", Volume: 2, Issue: 4 pp. 397-414. (2010)

5 Albert Ayenu-Prah and Nii Attoh-Okine, "A Criterion for Selecting Relevant Intrinsic Mode Functions in Empirical Mode Decomposition", Advances in Adaptive Data Analysis, Vol. 2, Issue: 1(2010) pp. 1-24.

6 George Jumper, “Private communication" (2001)

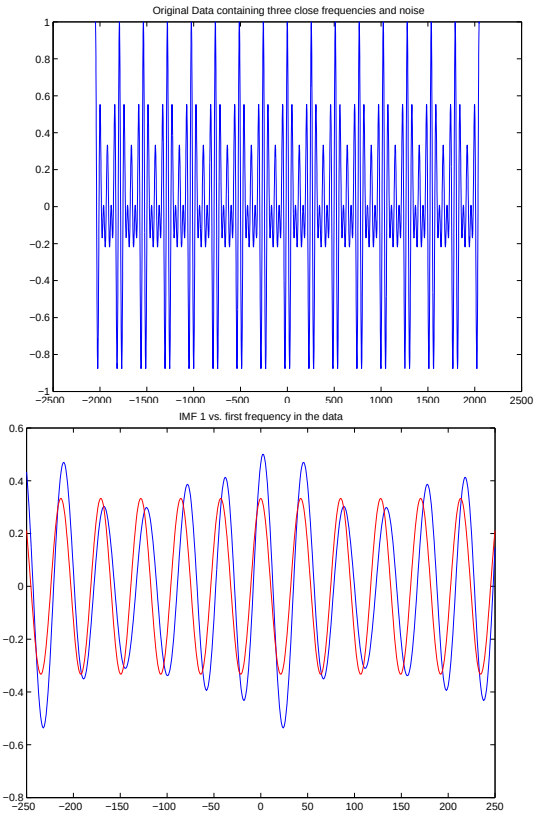
7 Dewan, E.M., “On the nature of atmospheric waves and turbulence, Radio Sci." 20, p. 1301-1307 (1985).

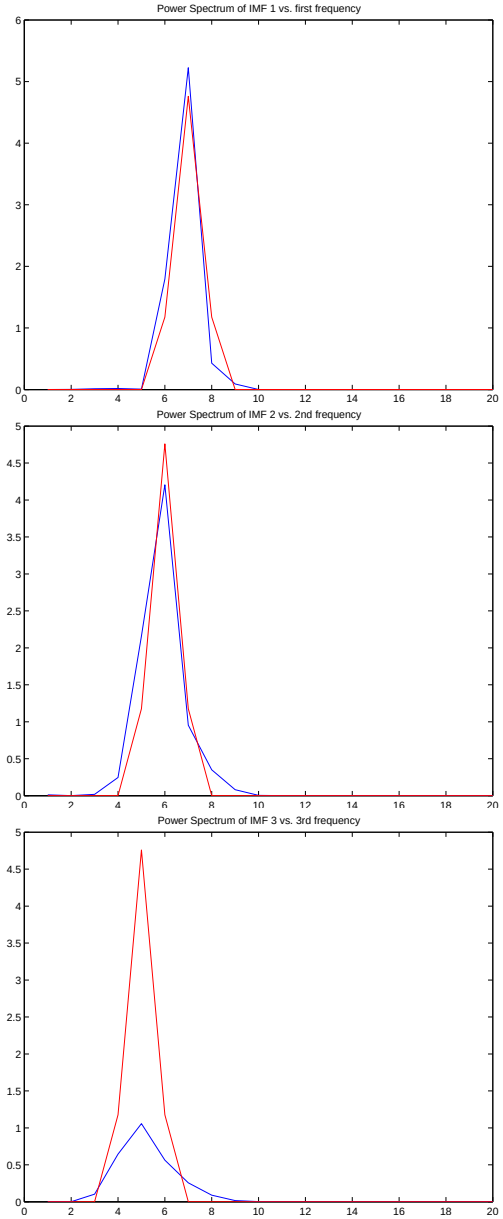
8 Kraichnan, R., “On Kolmogorov inertial-range theories", J. Fluid Mech., 62, p. 305-330 (1974).

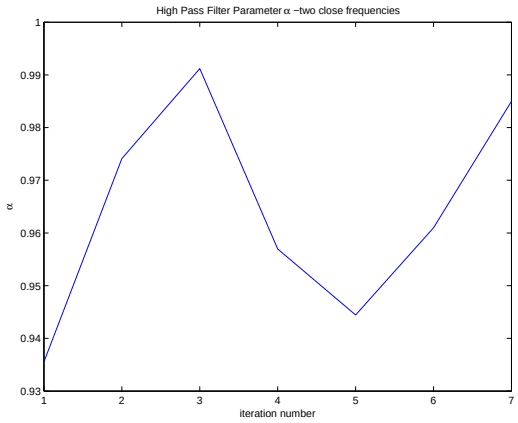
9 Lindborg, E., “Can the atmospheric kinetic energy spectrum be explained by two dimensional turbulence", J. Fluid Mech, 388, p. 259-288 (1999).

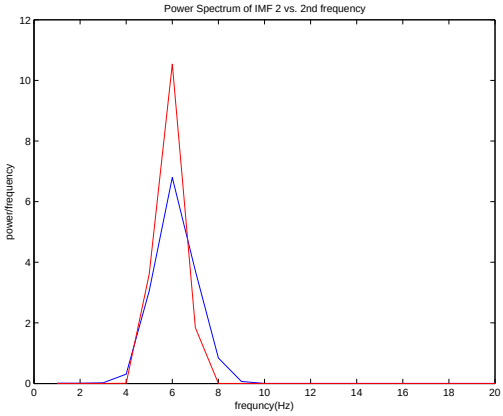
10 C. Penland, M. Ghil and K.M. Weickmann, “Adaptive filtering and maximum entropy spectra, with application to changes in atmospheric angular momentum”, J. Geophys. Res., 96, 22659-22671 (1991).

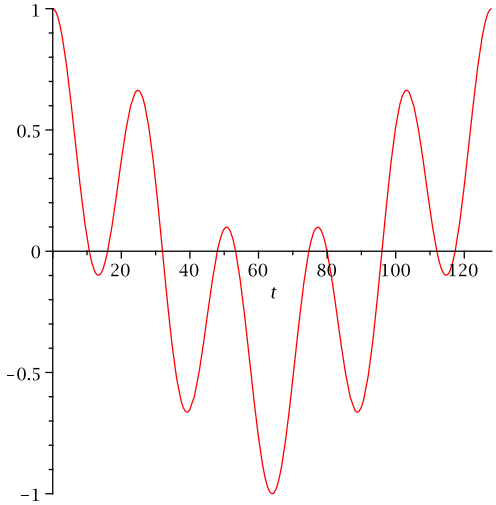
11 D. N. Axford, “Spectral analysis of aircraft observation of gravity waves”, Q.J. Royal Met. Soc., 97, 313-321 (1971).

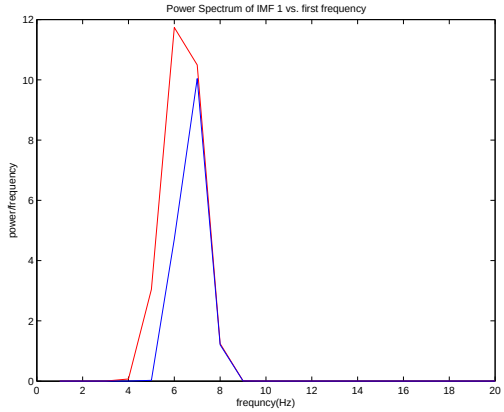


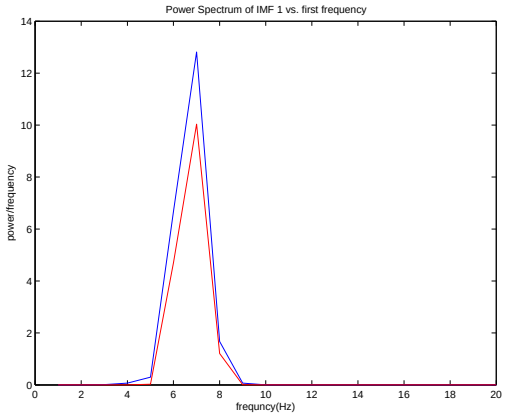


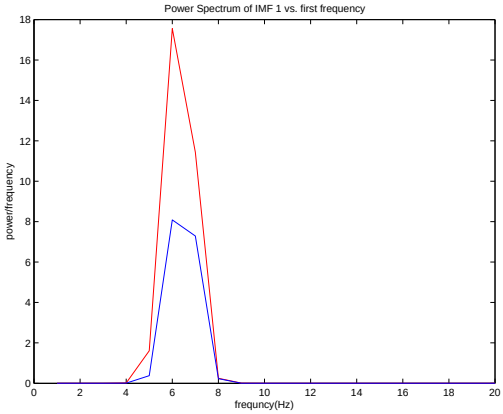

High Pass Filter Parameter α −two close frequencies
α
iteration number

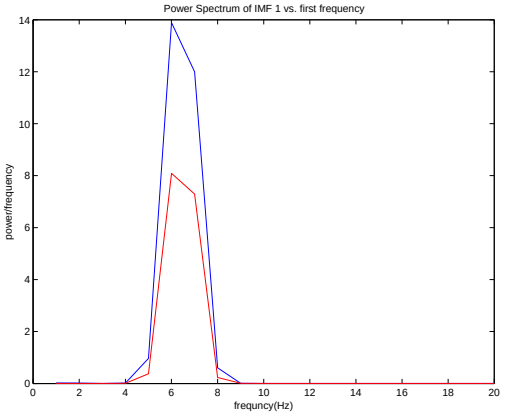

Power Spectrum of IMF 2 vs. 2nd frequency
power/frequency
frequncy(Hz)

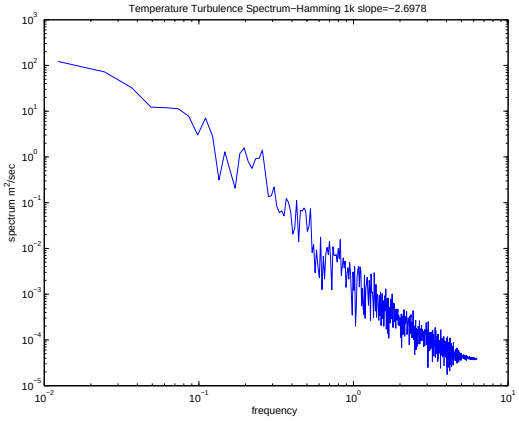

t

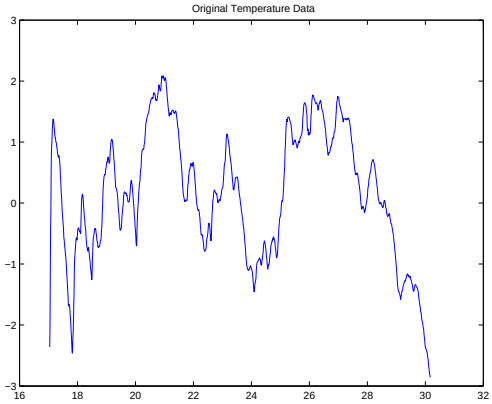

Power Spectrum of IMF 1 vs. first frequency
power/frequency
frequncy(Hz)

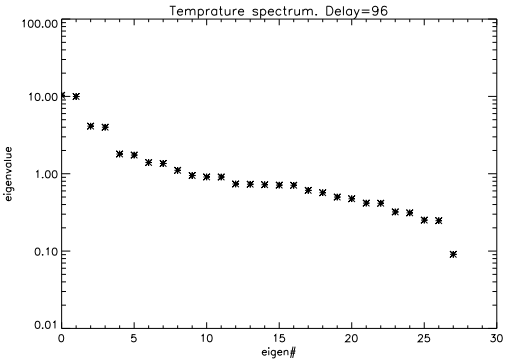

Power Spectrum of IMF 1 vs. first frequency
power/frequency
frequncy(Hz)

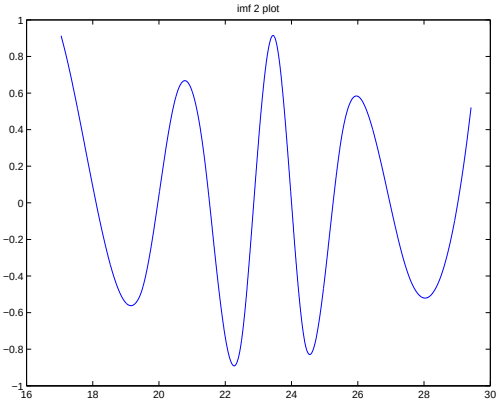

Power Spectrum of IMF 1 vs. first frequency
power/frequency
frequncy(Hz)

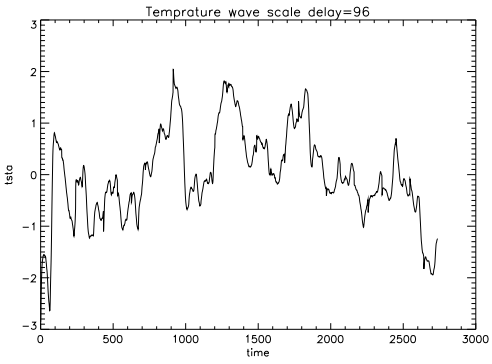

Power Spectrum of IMF 1 vs. first frequency
power/frequency
frequncy(Hz)


Temperature Turbulence Spectrum−Hamming 1k slope=−2.6978
spectrum m²/sec
frequency


Original Temperature Data


Temprature spectrum. Delay=96
eigenvalue
eigen#


imf 2 plot


Temprature wave scale delay=96
tsta
time

Thermal-Mechanical Vibration And Instability of A Fluid-Conveying Single-Walled Carbon Nanotube Based on Nonlocal Elasticity Theory

Tai-Ping Chang [1] **and Mei-Feng Liu** [2]
[1] Department of Construction Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, ROC
[2] Department of Applied Mathematics, I-Shou University, Kaohsiung, Taiwan, ROC

Abstract *- Based on the theories of thermal elasticity mechanics and nonlocal elasticity, an elastic Bernoulli–Euler beam model is developed for thermal-mechanical vibration and buckling instability of a single-walled carbon nanotube (SWCNT) conveying fluid and resting on an elastic medium. The finite element method is adopted to obtain the numerical solutions to the model. The effects of temperature change, nonlocal parameter and elastic medium constant on the vibration frequency and buckling instability of SWCNT conveying fluid are investigated. It can be concluded that at low or room temperature, the fundamental natural frequency and critical flow velocity for the SWCNT increase as the temperature change increases. The fundamental natural frequency for the SWCNT decreases as the nonlocal parameter increases, both the fundamental natural frequency and critical flow velocity increase with the increase of the elastic medium constant.*

Keywords: Nonlocal elasticity; Carbon nanotube conveying fluid; Temperature change; Vibration frequency; Instability.

1 Introduction

Carbon nanotubes (CNTs) discovered by Iijima [1] have attracted worldwide attention. Recently, the analysis of CNTs has been of great interest to many researchers because of their exceptional mechanical, electronic, electrochemical, physical and thermal properties [2-6]. The classic elastic continuum models have been widely used to study the vibration behavior of CNTs. Many studies related to the field are depicted in the references (Yoon et al. [3,7], Reddy et al. [8], Wang et al. [9] and Zhang et al. [10]). It is quite essential to perform the vibration and buckling analysis of carbon nanotubes by considering the thermal effects since the influence of temperature change on the instability of SWCNTs conveying fluid is significant. Among others, the following researchers have already contributed to the development of this field: Zhang et al. [11], Wang et al. [12], Ni et al. [13], Li and Kardomateas [14]. The nonlocal elasticity theory was first initiated by Eringen [15]. The importance of nonlocal elasticity theory stimulated the researchers to investigate the properties of the micro/nano structures more accurately and conveniently. Application of nonlocal continuum theory to nanotechnology was initially reported by Peddieson et al. [16]. Many studies related to nonlocal elasticity theory are depicted in the references (Zhang et al. [17], Sudak [18] [26], Lu et al. [19], Zhang et al. [20], Wang et al. [21] and Murmu and Pradhan [22]).
In this paper, an elastic Bernoulli–Euler beam model is developed for the thermal-mechanical vibration and buckling instability of SWCNT conveying fluid based on the nonlocal elasticity theory. The effects of temperature change, nonlocal small scale and Winkler modulus parameter on the properties of vibration frequency and buckling instability are investigated.

2 Thermal-nonlocal beam model for SWCNTs conveying fluid

In the present study, CNT is assumed to be fixed at both ends as shown in Fig. 1, and embedded into an elastic medium such as polymer. The length of the nanotube is denoted as L. Here we shall neglect the gravity effect as usual. Thus, let the mean flow velocity and the mass per unit length of the fluid be U and M, respectively. The governing equation for thermal vibration and structural instability of a CNT conveying fluid based on nonlocal elasticity theory can be derived as

$$\left[EI+\left(e_0a\right)^2 N_t-\left(e_0a\right)^2 MU^2\right]\frac{\partial^4 w}{\partial x^4}-2\left(e_0a\right)^2 MU\frac{\partial^4 w}{\partial x^3\partial t}$$
$$-\left(e_0a\right)^2\left(M+m\right)\frac{\partial^4 w}{\partial x^2\partial t^2}+\left[MU^2-N_t-\left(e_0a\right)^2 K\right]\frac{\partial^2 w}{\partial x^2}$$
$$+2MU\frac{\partial^2 w}{\partial x\partial t}+(M+m)\frac{\partial^2 w}{\partial t^2}+Kw=0 \tag{1}$$

where x is an axial coordinate, t is time, $w(x,t)$ is deflection of the CNT, E and I are Young's modulus and the moment of inertia of the cross-section of the CNT, m is the mass of the CNT per unit length, K is the Winkler

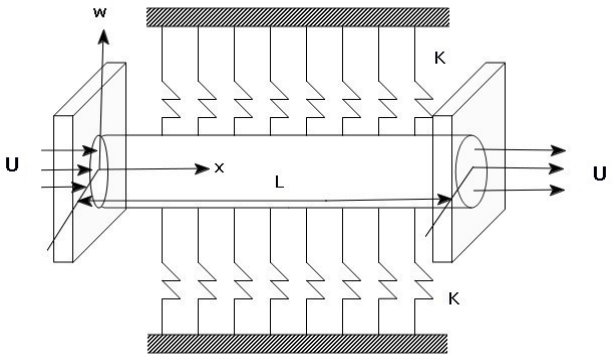


Fig. 1. A fluid-conveying single-walled carbon nanotube embedded in an elastic medium with two fixed ends.

constant of the surrounding elastic medium described as a Winkler-like elastic foundation [23]. The term $(e_0 a)$ denotes the small scale coefficient accounting for the small size effects. N_t is the constant axial force due to the thermal effects. It should be noted that Eq. (4) reduces to classical Euler-Bernoulli equation when $N_t = 0$ and $(e_0 a) = 0$.

On the basis of the theory of thermal elasticity mechanics, the axial force N_t can be written as [24]

$$N_t = -\frac{EA}{1-2\upsilon}\alpha_x T \tag{2}$$

where α_x denotes the coefficient of thermal expansion in the direction of the x-axis, υ is Poisson's ratio, and T is the temperature change.

In the present study, the CNT is considered as fixed at both ends, therefore, the boundary conditions at the ends of the CNT are

$$w(0,t) = \frac{\partial w(0,t)}{\partial x} = w(L,t) = \frac{\partial w(L,t)}{\partial x} = 0 \tag{3}$$

3 Solutions by finite element method

In the present study, the finite element method is adopted to determine the solutions to Eqs. (1-3). Using the finite element formulation, the assembled form of equation can be achieved as follows

$$[M]\{\ddot{W}\} + [C]\{\dot{W}\} + [K]\{W\} = 0 \tag{4}$$

where $\{W\}$ is a vector of the system global displacements of the structure and $[M]$,$[C]$ and $[K]$ are global mass, non-proportional damping and stiffness matrices of carbon nanotube conveying fluid, respectively. For a self-excited vibration, the solution of Eq. (4) can be written in the following form:

$$\{W\} = \{\bar{W}\}\exp(\lambda t) \tag{5}$$

Substituting Eq. (5) into Eq. (4), a generalized eigenvalue problem can be achieved as follows:

$$(\lambda^2[M] + \lambda[C] + [K])\{\bar{W}\} = 0 \tag{6}$$

where λ and $\{\bar{W}\}$ are the eigenvalue and eigenvector of the system and can be complex numbers in general. It should be noted that the real part of the eigenvalue is related to the system damping and the imaginary parts is related to the system natural frequencies. To determine a non-trivial solution to Eq. (6), the determinant of the coefficient matrix must vanish, that is,

$$\det(\lambda^2[M] + \lambda[C] + [K]) = 0 \tag{7}$$

Based on the above equation, the eigenvalues of the fluid-conveying CNTs can be computed for various parameter values.

4 Numerical results and discussions

In this paper, the equation of thermal-mechanical vibration and buckling instability of SWCNT conveying fluid has been derived based on the nonlocal elasticity theory. Here we discuss the vibration frequency and buckling instability of single-walled nanotubes. The outer radii and thicknesses of the nanotube are assumed to be $R_{\text{out}} = 3.5$ nm and $h = 0.34$ nm, respectively. The mass density of CNT is $2.3\ g/\text{cm}^3$ with Young's modulus E of 1 TPa, the mass density of water is 1 g/cm^3, aspect ratio $L/(2R_{\text{out}}) = 100$, nonlocal parameter $e_0 a / L = 0.05$ and Winkler constant $K = 0$ MPa. As stated by Jiang et al. [25], the coefficients of thermal expansion for CNTs are negative at lower temperature and become positive at higher temperature. In the present study, only the low temperature is considered. The Poisson's ratio is considered as $\upsilon = 0.3$ [24]. The coefficient of thermal expansion is considered as $\alpha_x = -1.6\times10^{-6}\times K^{-1}$ [26] for the case of low or room temperature. The natural frequency $\text{Im}(\lambda)$ is computed numerically from Eq. (7). Several results are presented on the variation of fundamental natural frequency of SWCNT with flow velocity for various parameter values. Fig. 2 depicts the

variation of fundamental frequency of SWCNT with flow velocity for different temperature changes in low or room temperature. As the flow velocity increases, the nanotube becomes more flexible and the natural frequencies get reduced. When the flow velocity exceeds a certain value, the fundamental natural frequency becomes zero and the nanotube becomes unstable, this corresponds to the inducing of instability of the SWCNT. The flow velocity producing the zero natural frequency is classified as the critical flow velocity of the system. It should be noted that the results presented in Fig. 2 show a similar tendency with those presented in Refs. [5] and [7]. As for the presence of temperature change in low or room temperature, the increase of the temperature change tends to increase the natural frequencies of the SWCNT as it can be detected from Fig. 2. Furthermore, it is noted that the critical flow velocity for the nanotube including the thermal effect is much larger than that without considering the change of temperature and increases with the increase of temperature change. Fig. 3 depicts the variation of fundamental frequency of SWCNT with flow velocity for different values of e_0a/L in low or room temperature. As it is found from Fig. 3 that the natural frequency is significantly influenced by the nonlocal parameter e_0a/L, while the critical flow velocity is much less influenced by the nonlocal parameter e_0a/L. The nonlocal parameter $e_0a/L=0$ denotes the result obtained by classical Euler beam model. As the nonlocal parameter increases, the fundamental natural frequency decreases. Fig. 4 presents the variation of fundamental frequency of SWCNT with flow velocity for different values of Winkler constant K in low or room temperature. It is found that both the natural frequency and critical flow velocity are significantly influenced by Winkler constant. As the elastic medium constant K increases, the fundamental frequency also increases, which is reasonable since increasing the elastic medium constant makes the SWCNT stronger. The critical flow velocity also increases with the increase of the elastic medium constant K. Based on the results in Figs. 2-4, the effects of temperature change, nonlocal parameter and elastic medium constant are very significant on the fundamental natural frequency and critical flow velocity of fluid-conveying SWCNT embedded in an elastic medium.

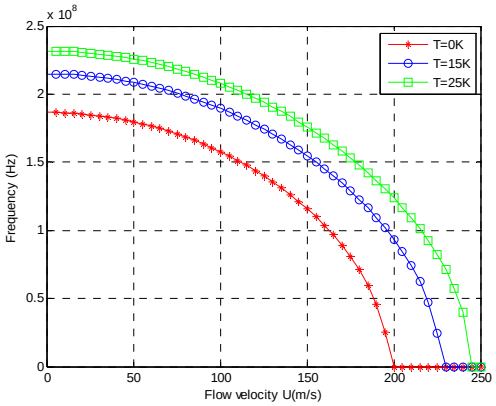


Fig. 2. Variation of fundamental frequency of SWCNT with flow velocity for different temperature changes in low or room temperature ($e_0a/L=0.05, K=0MPa$).

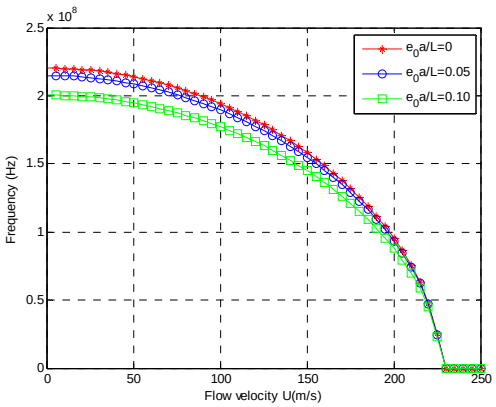


Fig. 3. Variation of fundamental frequency of SWCNT with flow velocity for different values of e_0a/L in low or room temperature ($T=15K, K=0MPa$).

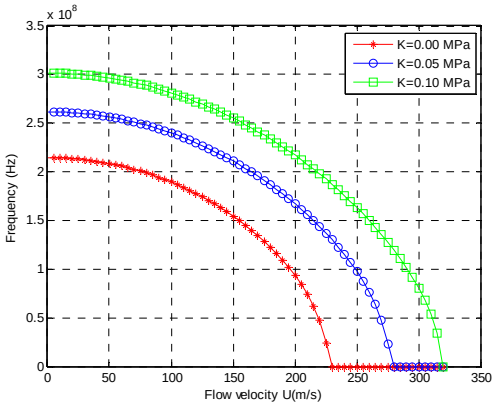


Fig. 4. Variation of fundamental frequency of SWCNT with flow velocity for different values of Winkler constant K in low or room temperature ($e_0a/L=0.05, T=15K$).

5 Conclusions

Based on the theories of thermal elasticity mechanics and nonlocal elasticity, an elastic Bernoulli–Euler beam model is developed for thermal-mechanical vibration and buckling instability of a single-walled carbon nanotube (SWCNT) conveying fluid and resting on an elastic medium. The finite element method is adopted to obtain the numerical solutions to the model. The effects of temperature change,

nonlocal parameter and elastic medium constant on the vibration frequency and buckling instability of SWCNT conveying fluid are investigated. Several results are presented on the variation of fundamental natural frequency of SWCNT with flow velocity for various parameter values. It can be detected that at low or room temperature, the fundamental natural frequency and critical flow velocity for the SWCNT increase as the temperature change increases. Besides, the natural frequency is significantly influenced by the nonlocal parameter, while the critical flow velocity is much less influenced by the nonlocal parameter. The fundamental natural frequency for the SWCNT decreases as the nonlocal parameter increases. Finally, both the fundamental natural frequency and critical flow velocity increase with the increase of the elastic medium constant. Therefore, it can be conclude that the effects of temperature change, nonlocal parameter and elastic medium constant are very significant on the fundamental natural frequency and critical flow velocity of fluid-conveying SWCNT embedded in an elastic medium.

6 References

[1] S. Ijjima, Helical microtubules of graphitic carbon, Nature 354 (1991) 56-58.

[2] J.Z. Liu, Q.S. Zheng, L.F. Wang, Q. Jiang, Mechanical properties of single-walled carbon nanotube bundles as bulk materials, J. Mech. Phys. Solids, 53 (2005) 123-142.

[3] J. Yoon, C.Q. Ru, A. Mioduchowski, Flow-induced flutter instability of cantilever carbon nanotubes, Int. J. Solids Struct., 43 (2006) 3337-3349.

[4] L. Wang, Q. Ni, On vibration and instability of carbon nanotubes conveying fluid, Comp. Mater., 43 (2008) 399-402.

[5] L. Wang, Q. Ni, M. Li, Q. Qian, The thermal effect on vibration and instability of carbon nanotubes converying fluid, Physica E 40 (2008) 3179-3182.

[6] H.L. Lee, W.J. Chang, Vibration analysis of a viscous-fluid-conveying single-walled carbon nanotube embedded in an elastic medium, Physica E 41 (2009) 529-532.

[7] J. Yoon, C.Q. Ru, A. Mioduchowski, Vibration and instability of carbon nanotubes conveying fluid, Compos. Sci. Technol., 65 (2005) 1326-1336.

[8] C.D. Reddy et al., Free vibration analysis of fluid-conveying single-walled carbon nanotubes, Appl. Phys. Lett., 90 (2007) 133122-133124.

[9] C.M. Wang, V.B.C. Tan, Y.Y. Zhang, Timoshenko beam model for vibration analysis of multi-walled carbon nanotubes, J. Sound Vib.,294 (2006) 1060–1072.

[10] Y. Zhang, G. Liu, X. Han, Transverse vibrations of double-walled carbon nanotubes under compressive axial load, Phys. Lett., A 340 (2005) 258–266.

[11] Y.Q. Zhang, X. Liu, G.R. Liu, Thermal effect on transverse vibrations of double-walled carbon nanotubes, Nanotechnology 18 (2007) 445701-445707.

[12] L. Wang, Q. Ni, M. Li, Q. Qian, The effect on vibration and instability of carbon nanotubes conveying fluid, Physica E 40 (2008) 3179–3182.

[13] B. Ni, S.B. Sinnott, P.T. Mikulski, J.A. Harrison, Compression of Carbon Nanotubes Filled with C_{60}, CH_4, or Ne: Predictions from Molecular Dynamics Simulations, Phys. Rev. Lett., 88 (2002) 205505-205508.

[14] R. Li, G.A. Kardomateas, Thermal Buckling of Multi-Walled Carbon Nanotubes by Nonlocal Elasticity, J. Appl. Mech.,74 (2007) 399–405.

[15] A.C. Eringen, On differential equations of nonlocal elasticity and solutions of screw dislocation and surface waves, J. Appl. Phys., 54 (1983) 4703–4710.

[16] J. Peddieson, G.R. Buchanan, R.P. McNitt, Application of nonlocal continuum models to nanotechnology, Int. J. Eng. Sci., 41 (2003) 305–312.

[17] Y.Q. Zhang, G.R. Liu, X.Y. Xie, Free transverse vibrations of double-walled carbon nanotubes using a theory of nonlocal elasticity, Phys. Rev., B 71 (2005) 195404-195410.

[18] L.J. Sudak, Column buckling of multiwalled carbon nanotubes using nonlocal continuum mechanics, J. Appl. Phys., 94 (2003) 7281–7287.

[19] P. Lu, H.P. Lee, C. Lu, P.Q. Zhang, Dynamic properties of flexural beams using a nonlocal elasticity model, J. Appl. Phys., 99 (2006) 073510-073518.

[20] Y.Q. Zhang, G.R. Liu, J.S. Wang, Small-scale effects on buckling of multiwalled carbon nanotubes under axial compression, Phys. Rev., B 70 (2004) 205430-205435.

[21] C.M. Wang, S. Kitipornchai, C.W. Lim, M. Eisenberger, Beam Bending Solutions Based on Nonlocal Timoshenko Beam Theory, J. Eng. Mech., 134 (2008) 475–481.

[22] T. Murmu, S.C. Pradhan, Thermo-mechanical vibration of a single-walled carbon nanotube embedded in an elastic medium based on nonlocal elasticity theory, Comput. Mater. Sci., 46 (2009) 854-589.

[23] Y.M. Fu, J.W. Hong, X.Q. Wang, Analysis of nonlinear vibration for embedded carbon nanotubes, J. Sound Vib., 296 (2006) 746-756.

[24] Y.Q. Zhang, X. Liu, J.H. Zhao, Influence of temperature change on column buckling of multiwalled carbon nanotubes, Phys. Lett., A 372 (2008) 1676-1681.

[25] H. Jiang, B. Liu, Y. Huang, Thermal expansion of single wall carbon nanotubes, J. Eng. Mater. Technol., 126 (2004) 265-270.

[26] X.H. Yao, Q. Han, Buckling analysis of multiwalled carbon nanotubes under torsional load coupling with temperature change, J. Eng. Mater. Technol.,128 (2006) 419-427.

SESSION

COMPUTATIONAL SIMULATION AND MODELING

Chair(s)

TBA

Adaptive Data Structure Management for Grid Based Simulations in Engineering Applications

Jérôme Frisch, Ralf-Peter Mundani, and Ernst Rank

***Abstract*—This paper describes a hierarchical adaptive data structure management used for typical engineering simulations such as temperature diffusion problems or computational fluid dynamic problems. Sketches for using an adaptive non-overlapping block structured grid in a distributed manner are deployed and sample simulations are computed to underline the used concepts. Furthermore, a small outlook is given to future work planned in this area, how to improve the implemented version of the code, as well as how a parallel concept might look like.**

***Index Terms*—data management, adaptive grid, non-overlapping block structured grid, ghost cells, transient temperature diffusion equation, computational fluid dynamics**

I. Motivation

In modern engineering simulations of any kind, accurate geometric representation is playing a key role for describing a certain problem. Figure 1 shows a huge, detailed power plant model containing more than 12,5 million triangles. It can be seen that both very small but also large triangles are present. If a detailed computational fluid dynamics simulation around this power plant should be computed, there is the necessity of simulating a large volume of surrounding air in order to reduce the effects of boundary conditions from the enclosing domain to the plant itself. The easiest way to perform this task is to uniformly refine the complete computational domain until the smallest triangle is included or until a given geometric accuracy is reached. A reasonable resolution would contain more than $2 \cdot 10^9$ uniform hexahedral cells for which the solution of the CFD problem would take even on huge super computers a quite long time. The consequence of this uniform refinement is a very fine grid on places where it is not mandatory from a geometric point of view. A solution is an adaptive refinement only in areas where more information is necessary or helpful to increase simulation results, whereas a coarse grid can be used in areas of low information density. Unfortunately, this adaptive handling of data asks for a more complex data structure to manage geometry and boundary conditions.

In this paper, an adaptive data structure management framework based on non-overlapping block structured grids is presented, in which two engineering applications are tested. The construction of the block structured grid is based on a recursive hierarchical build-up. The concept is explained and demonstrated for a transient temperature distribution and for a computational fluid dynamics scenario.

This paper describes work in progress in order to construct a data structure and a software framework which is able to deal with grid refinement and is prepared in such a way that a future parallel distribution to multiple systems for running a massive parallel application is possible.

Adaptive grids are quite well studied in literature (c. f. Samet [2], Barequet et al. [3]) and applied to specific problems (c. f. Coelho et al. [4]), but in contrast to Coelho et al., the sub-grids are surrounded with so called ghost cells as described later in section II.A, even if they would not be necessary for a computation running on a machine using a shared memory approach. The term 'distributed' refers to the fact that the data structure is not allocated as one block in memory, but as a hierarchy of grids that are all maintained separately and that are 'coupled' via update functions (described in section II.B) between two hierarchical levels. In future, different grids reside on different machines using a distributed computing approach, as state-of-the-art solutions for engineering simulations, such as fluid dynamic problems, are almost always using a parallel approach in order to cope with the high data amount.

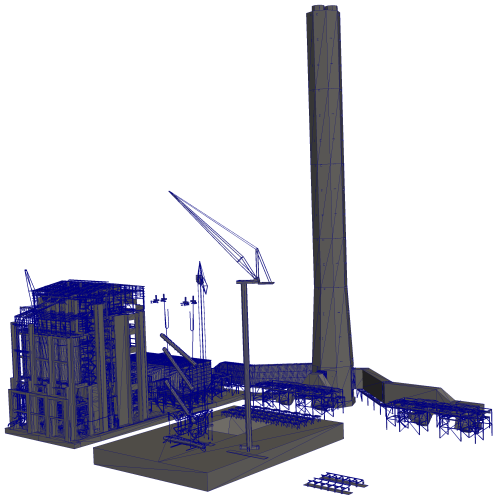


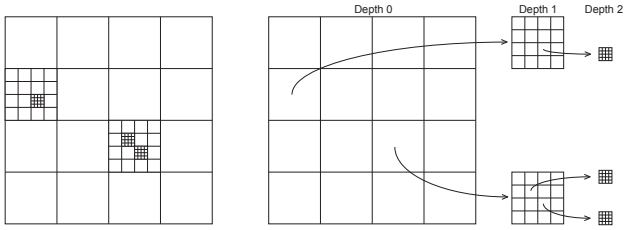
Figure 1. View of a power plant model [1] consisting of 12,748,510 triangular surfaces organised in 1,185 groups.

As geometric representation the authors chose a block structured approach as a trade-off between geometric accuracy and complexity in data handling. On the one hand, a fully detailed geometric description using unstructured grids can represent the geometry with a very high level of detail, using not too much cells. Unfortunately the data management handling is very complex and the performance is not so high. On the other hand, structured orthogonal grids have a very easy data handling and thus, very high performance regarding computation time but cannot represent the geometry quite accurate. Furthermore, the generation of input data for a structured block oriented mesh from an arbitrary surface mesh using an octree based space partition scheme is much easier to automatise than the generation of an arbitrary unstructured mesh.

Chair for Computation in Engineering, Technische Universität München, Arcisstrasse 21, 80290 München, Germany. Corresponding author contact: Phone: +49 89 289–25128, e-mail: frisch@tum.de

II. Adaptive Data Structure Management

The concept of the adaptive data structure management is based on non-overlapping block structured orthogonal grids. Each block is constructed out of orthogonal, equidistant pseudo-cells which can be regarded as real data cells describing fluids, solids, etc. or they may contain a link to a sub-grid. The possibility of local refinement gives the code the ability to adapt quite good to a complex geometry while still using orthogonal grids on which finite difference or finite volume schemes can be adapted fast.



(a) hierarchical 2D block oriented data structure

(b) recursive construction sequence of the 2D data structure from the left hand side

Figure 2. 2D block oriented data structure

The implemented code is designed for managing 3D grids. For the sake of simplicity some of the following examples are given for 2D grids only even if they can be applied to 3D grids. A scheme of the block structured grid can be seen in Figure 2. In this case, the main grid (identifiable by the depth zero), as well as the sub-grids, have a size of 4x4. These numbers result from a mere choice for an adequate visualisation. To reduce the overhead of grid management, a higher choice of cell amounts in the main grid level is reasonable. The arrows represent links through a pointer data structure from the respective pseudo-cell to the sub-grid and back.

In this case, a block structured approach is preferable to a standard octree, as a high depth would be necessary to acquire the desired detailed geometry. Furthermore a neighbouring search algorithm is called very often as a result of using finite difference stencils which is quite costly for octrees. Hence, we chose a non-overlapping block structured grid where the sub-grids are regular and neighbouring relations of finite difference stencils reduce to index shifting in data array access.

In order to keep the data structure as flexible as possible according to adaptive refinements, no links with pointers from single cells to neighbouring cells were established, but a ghost cell scheme was used.

A. Ghost Cell Scheme

The ghost cell scheme introduces one layer of cells all around the sub-grids as indicated in Figure 3 by gray-shaded cells around a 4x4 sub-cell grid. From depth zero to depth one there are two links to different sub-grids. The arrows from cells to ghost cells on the same depth level are not pointer links, but a mere indication which cell contents is copied during the update step described in the next sub-section.

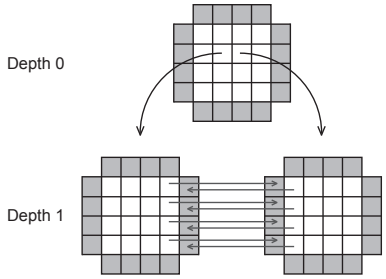


Figure 3. Example of the ghost cell scheme: ghost cells are marked in gray.

B. Update Step

The necessity of performing an update step and the ghost cell scheme resembles to a parallel computation approach. Using such a scheme for the design of the code – even in a serial case – is speeding up the later process of advancing to a parallel version.

The update step is performed after each computational step, meaning that after a specific computational algorithm has been performed on a sub-grid without following links to sub-cells e. g., the update function is called. Some inherent synchronisation is implied by the order of execution of update functions. Hence, to respect the order, it is necessary to treat the complete block structured grid in a bottom-up manner, starting on the deepest sub-grid and ending on the main grid at level zero.

To order the sub-grids in a bottom-up fashion, two data structures, namely a queue (FIFO) and a stack (LIFO) are used. At first, both stack and queue are empty and the main grid is added to the structures. While the queue is used for iterating through the different sub-grids into the depth, the stack is accumulating links to the complete data structure in such an order that the last elements pushed to the stack will be removed first, which delivers the bottom-up approach.

After the ordering of the pointers to the different sub-grids, different update procedures have to be chosen according to the computational desires. In case of a finite difference scheme, the total mean values for one sub-grid are computed and passed on to the corresponding parent cell for further processing. By starting from the deepest sub-grid, one can assure that only updated values are taken into account for performing computations on the current sub-grid.

Afterwards, cell values are copied to the corresponding neighbour ghost cells if they exist, as depicted by the arrows in depth one in Figure 3. Depending on a further subdivision of the neighbour cells, different copying techniques with or without averaging are applied. Having the surrounded cells as well as mean valued cells for the next step, a new calculation using only local values can be performed.

Thus, the main time stepping algorithm can be divided into two parts: one purely local part where only computation is taking place and one global part where communication is involved. Having built up the simulation in such a way, a parallelisation can be done without big changes.

III. Engineering Simulations

In the following section, the above mentioned basics will be applied to typical engineering simulations, namely temperature diffusion equation (III.A) for solving transient temperature distribution problems and Navier-Stokes equations (III.B) for solving computational fluid dynamics problems.

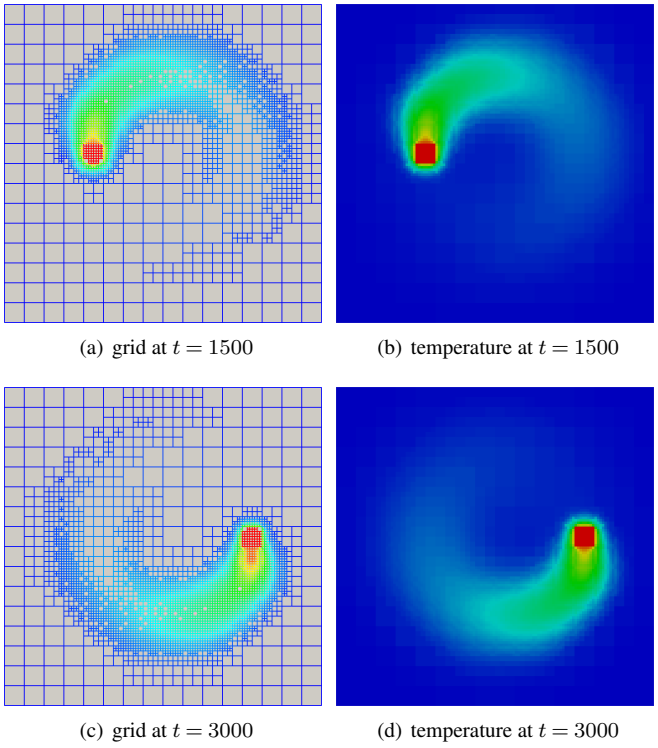
A. Temperature Diffusion Equation

One example of grid based engineering simulations is the time-dependent temperature diffusion equation

$$\frac{\partial}{\partial t}T = \alpha \cdot \Delta T \tag{1}$$

where T represents the temperature, depending on the time t and the spatial location, α the thermal diffusivity in [m^2/s] and Δ denoting the Laplace operator. If only a stationary solution is required, equation (1) reduces to the Laplace equation $\Delta T = 0$.

As numerical discretisation of equation (1), a forward Euler scheme in time and a central difference scheme in space is used, corresponding to a FTCS scheme.



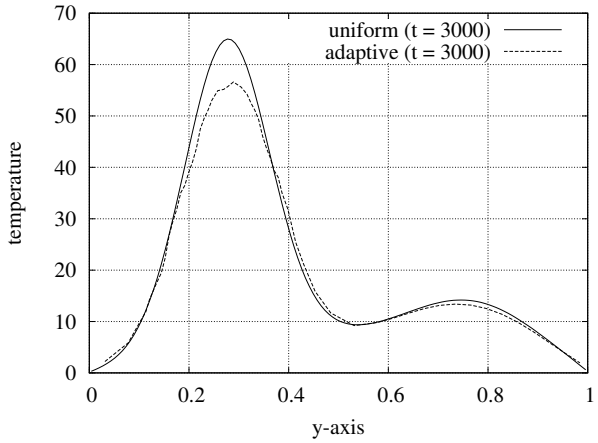
(a) grid at $t = 1500$ (b) temperature at $t = 1500$

(c) grid at $t = 3000$ (d) temperature at $t = 3000$

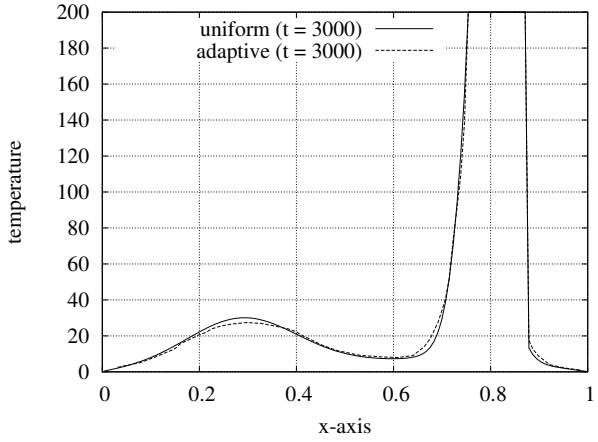
Figure 4. Computation of the time-dependent temperature diffusion equation (1) with adaptive grid refinement and time-dependent boundary conditions.

Example

An example of an adaptive computation can be seen in Figure 4, where the time-dependent temperature equation (1) is solved on a rectangular domain of 16x16x1. In this case, the computational domain is embedded in z-direction between two plates with Dirichlet boundary conditions $T = 0$. This setting was explicitly chosen over a setup with periodic boundary conditions in z-direction, as more energy is dissipated from the system and



(a) Temperature along a vertical line through the geometric centre of cavity



(b) Temperature along a horizontal line through the geometric centre of cavity

Figure 5. Comparison of temperature simulation results from a uniform computation (full lines) with results from an adaptive computation (dashed lines) at the time step $t = 3000$ of Figure 4(c) and Figure 4(d).

the adaptive coarsening can be better observed. For the sake of simplicity, only x and y coordinates are mentioned further on, even if a 3D computation was performed. As material, steel with a thermal diffusivity $\alpha = 1.172 \cdot 10^{-5}$ m^2/s was chosen.

The boundary cells of the domain have uniform temperature boundary conditions set to $T = 0$. Only a given obstacle of the size of one cell on the top level has a Dirichlet temperature boundary condition of $T = 200$. The boundary condition of the 'hot' cell is updated every time step t according to the relation $i_{\text{hot}} = 8 + 5 \cdot \lfloor \cos(2\pi t/1000) \rfloor$, $j_{\text{hot}} = 8 + 5 \cdot \lfloor \sin(2\pi t/1000) \rfloor$ and for all indices k_{hot}, which means that the 'hot' cell rotates counter-clockwise in the domain and has a period of $t_{\text{period}} = 1000$ s. If according to the index $(i_{\text{hot}}, j_{\text{hot}}, k_{\text{hot}})$ a new cell is selected, all other cells are set to 'free flow' meaning that the fixed Dirichlet boundary is removed, and the temperature of the cell can change again according to the values computed by the central difference stencil.

If a higher accuracy than 16x16x1 cells is desired, there are generally two possibilities. The easiest way is to increase the domain size uniformly to 128x128x1 e. g. As consequence, the domain consists now of 16,384 cells but no big changes to the code have to be made as only a different grid size has to be used.

A second, more complex way is an adaptive refinement as discussed in section II. Here only cells of interest get a higher resolution. In this example, cells from the top-level are refined by sub-grids of 2x2x1. This value is chosen for better visualisation results. From the point of view of computational overhead regarding grid management, it would be better to choose a higher value of cells per sub-grid.

According to the computation of the maximal and minimal temperature gradient between neighbouring cells, adaptive refinement or coarsening is applied to give a high accuracy regarding computational results using minimal cell amounts which reduces computational time. The example shown in Figure 4 uses three sub-levels of grids 2x2x1 with an average total amount of cells of 3,400. This is around 4.8 times less cell usage than in the uniform case at the same level of accuracy. Unfortunately the grid management is also more complicated and some averaging of values are applied in regions of coarsening.

Figure 5 shows a qualitative comparison between uniform and adaptive computation methods in terms of accuracy. The maximal temperature error from the adaptive to the uniform computation method in Figure 5(a) is around 12.5% and in Figure 5(b) about 8.0%. However, the adaptive version is approximately 1.5 times faster than the uniform computation.

These values can be even sped up by using a numerical more reasonable block size. As stated before, this example used a main grid size of 16x16x1 and the sub-grid size was chosen to 2x2x1 with three subdivision steps. When a sub-grid size of 4x4x1 is used with two subdivisions, compared to a uniform computation using 256x256x1 cells, the computation of the adaptive grid is 4.3 times faster than the uniform grid using 7.1 times less cells with a maximal error in temperature under 10%. Choosing a higher sub-grid size will be even more reasonable and give better results.

B. Navier-Stokes Equations

As a second example for grid based engineering simulations, an incompressible, isothermal Newtonian fluid flow without any acting external forces is simulated using the Navier-Stokes equations:

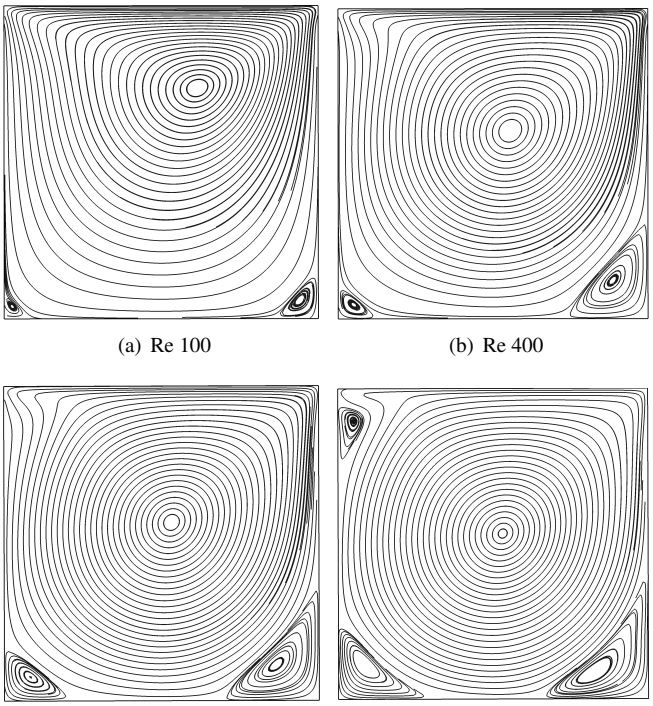
$$\vec{\nabla} \cdot \vec{u} = 0 \quad , \tag{2}$$

$$\frac{\partial}{\partial t}\vec{u} + \left(\vec{u} \cdot \vec{\nabla}\right)\vec{u} = -\frac{1}{\rho}\vec{\nabla}p + \nu\Delta\vec{u} \quad . \tag{3}$$

where $\vec{u}$ and p are the unknown velocities and pressure, t represents the time, ρ the density and ν the viscosity of the fluid. Further detailed information might be found in Hirsch [5] or Ferziger and Peric [6].

B.1 Numerical Discretisation Schemes

In the example at hand, a finite volume scheme is used for spatial discretisations and a finite difference scheme for temporal discretisations. For the sake of simplicity, the first tests are performed using an explicit Euler scheme for the temporal discretisations in order to test the above described adaptive block-oriented data structure. In a later stage it is planned to adopt a semi-implicit temporal discretisation. Furthermore, a fractional step or projection method is applied for solving the time-dependent incompressible flow equations. This method is based on a iterative procedure between velocity and pressure during one time step.



(a) Re 100 (b) Re 400

(c) Re 1000 (d) Re 3200

Figure 6. Streamline patterns in the lid-driven cavity with a grid resolution of 101x101 for different Reynolds numbers.

Omitting the pressure term, the momentum equations are solved for intermediate velocities $\vec{u}^*$:

$$\frac{\vec{u}^* - \vec{u}^n}{\Delta t} = -\left(\vec{u}^n \cdot \vec{\nabla}\right)\vec{u}^n + \nu\Delta\vec{u}^n \quad . \tag{4}$$

The superscript * denotes intermediate values and the superscript ^{n} values at time step n, which are fully known. In the second step, the pressure term at the next time step $n+1$ is used to correct the resultant intermediate velocity field leading to the velocity field at the new time step $n+1$:

$$\frac{\vec{u}^{n+1} - \vec{u}^*}{\Delta t} = -\frac{1}{\rho}\vec{\nabla}p^{n+1} \quad . \tag{5}$$

The divergence free velocity field at step $n+1$ can be guaranteed by computing the divergence of (5) and applying the continuity equation (2):

$$\frac{\rho}{\Delta t}\left(\vec{\nabla} \cdot \vec{u}^{n+1} - \vec{\nabla} \cdot \vec{u}^*\right) = -\Delta p^{n+1} \tag{6}$$

$$\Delta p^{n+1} = \frac{\rho}{\Delta t}\vec{\nabla} \cdot \vec{u}^* \quad . \tag{7}$$

Equation (7) represents a Poisson equation for the pressure, which has to be solved to compute the velocity field for the next time step, using (5).

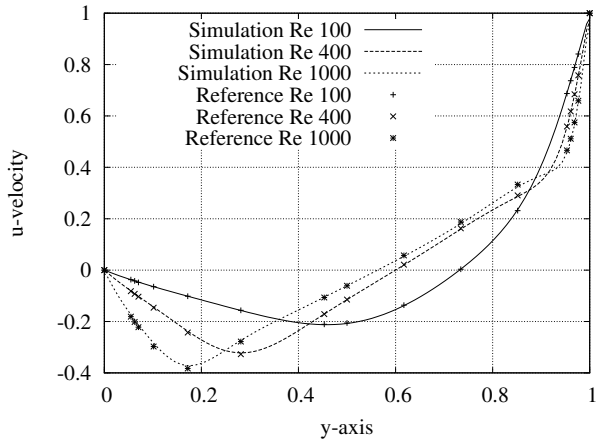
B.2 Staggered versus Collocated Grid Arrangements

While applying a spatial discretisation scheme, it is possible to choose between different settings. In a staggered grid

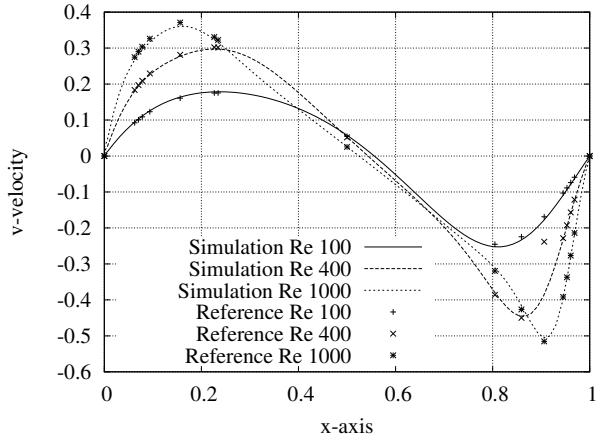
approach, not all variables are represented at the same point in space. Usually partially staggered arrangements are used, where pressure and other scalar terms are situated at the cell centre, whereas the velocities are positioned at the respective cell surfaces. This arrangement has the major advantage that the velocities are strongly coupled to the pressure values and no oscillations occur applying the projection method. Disadvantages are the more complicated handling in case of non-orthogonal grids or when applying multigrid solvers.

In collocated grids, all variables are defined in the cell centre which simplify the usage of non-orthogonal grids or advanced multigrid solvers. On the other hand, it can be shown that the solution of equation (7) leads to a so called odd-even decoupling which introduces non physical pressure oscillations if no special care is taken. More details can be found in Hirsch [5] or Ferziger and Peric [6].

The code described here uses a collocated grid arrangement due to the fact that in later phases of this project the authors plan to use sophisticated numerical solvers such as multigrid methods.



(a) u-velocities along a vertical line through the geometric centre of cavity



(b) v-velocities along a horizontal line through the geometric centre of cavity

Figure 7. Comparison of simulation results (lines) with results of Ghia et al. (points) for the lid-driven cavity

B.3 Validation Using the Lid-Driven Cavity Example

For the validation of the above described code, the lid-driven cavity example is used. This example consists of a square domain of unit length where the upper boundary wall moves with constant velocity $u = 1$. Thus, only shear driven forces from the no-slip boundaries are transferred to the initially resting fluid. Reference solutions for comparison where taken from Ghia et al. [7].

Figure 6 shows a streamline plot for different Reynolds numbers of $Re = 100$, $Re = 400$, $Re = 1000$, and $Re = 3200$ computed using the above mentioned code and a grid spacing of 101x101. All the validations and computations were done in a first step only in two dimensions, even if the data structure is designed for three dimensions.

For getting a better view of the numerical errors introduced by the discretisation technique e. g., detailed comparisons were made in Figure 7(a) and 7(b). It can be seen, that for Reynolds number $Re = 100$ the computed values match the reference values used by Ghia et al. quite well. But the higher the Reynolds number is, the higher the divergence between the computed values and the reference values gets, even if the characteristic behaviour can still be observed.

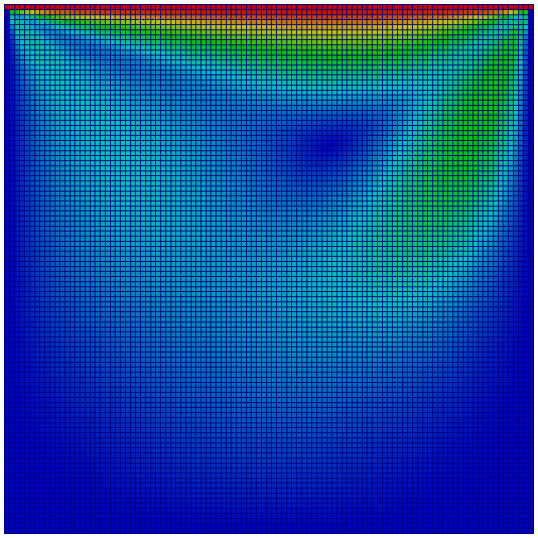
Computations with a higher geometric resolution and a smaller time step size for different Reynolds numbers show that a finer time step has a much higher impact as soon as the spatial discretisation is reasonably small. This is a numerical artefact of using the explicit Euler time scheme for temporal discretisation and shows that for simple tests of the data structure, the explicit scheme is adequate, but for later real case studies, a higher temporal discretisation technique has to be used.

Figure 8 shows the magnitude of the velocity vector $\vec{u}$ for an uniform and an adaptive computation of the lid-driven cavity example at $Re = 100$. The base grid is chosen to 21x21 and the sub-grid size to 5x5 for display reasons and the time step is set to 10^{-4} s.

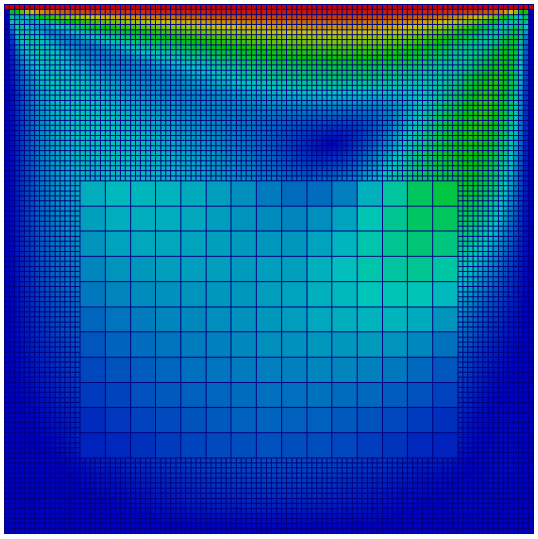
The accuracy of the computation is indicated in Figure 9. It can be seen, that the coarse grid of 21x21 is not reaching the reference values of Ghia et al., as the grid is too coarse to deliver accurate results. An adaptive mesh refinement as depicted in Figure 8(b) results in much better accuracy, even if the reference values are not quite reached. This is due to the interpolation effect of the grid changes from coarse to fine. In order to keep the computation algorithms as simple as possible, some trade-off was accepted and a numerical error was introduced. At the moment, the authors are working on reducing the numerical error while still keeping a simple scheme regarding numerical computation and data exchange from the different grid levels.

First parallel computations were done using a shared memory OpenMP concept. In this first implementation, only computational intensive nested loops of the Navier-Stokes equations were parallelised. Hence the update step mentioned in II.A is still running as serial procedure and is dominating the possible speedup as well as the parallel efficiency.

Computational results of the parallel speedup and efficiency are depicted in Figure 10. In order to get a better comparison, three different architectures and different grid sizes were used. The used architectures include an Intel Core 2 Quad Q9650 (3.00 GHz), an Intel Core i7 870 (2.93 GHz), and an Intel Xeon

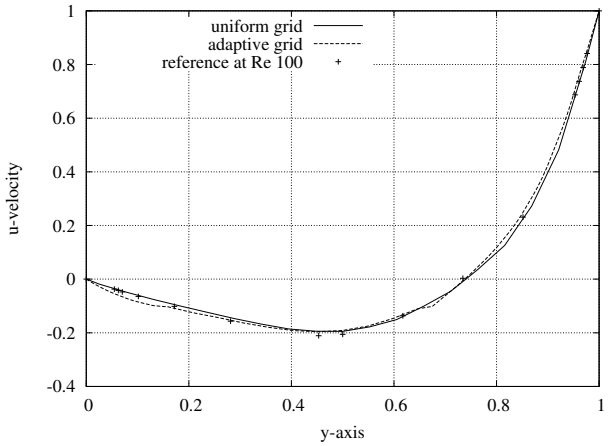


(a) uniform grid

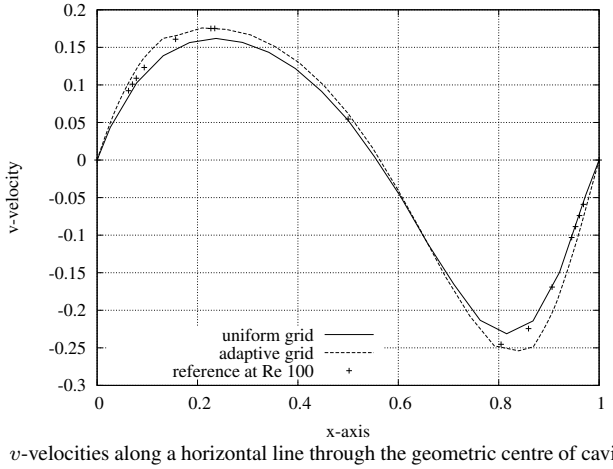


(b) adaptive grid

Figure 8. Magnitude of the velocity vector $\vec{u}$ of an example computation of the lid-driven cavity on a uniform grid using 105x105 cells with a time step of $\Delta t = 10^{-4}$ s and an adaptive computation using a base grid of 21x21 and sub-grids of size 5x5.



(a) u-velocities along a vertical line through the geometric centre of cavity



(b) v-velocities along a horizontal line through the geometric centre of cavity

Figure 9. Plots comparing velocities with the reference solution of Ghia et al. with an adaptive computation on a 21x21 coarse grid only (marked uniform grid in plots) and an adaptive computation depicted in Figure 8(b) (marked adaptive grid in plots)

E3-1245 (3.30 GHz). Furthermore the same optimisation flags were used for the Intel compiler on all architectures.

Figure 10 shows that this kind of parallelisation is not optimal as the efficiency is dropping quite fast as soon as more processes are used. Hence, another method has to be deployed when more cores or processes are involved, for which the data structure was designed to distribute the sub-grids to different processes using a message passing concept. This parallelisation will be subject to further investigations.

IV. Outlook to planned work for the future

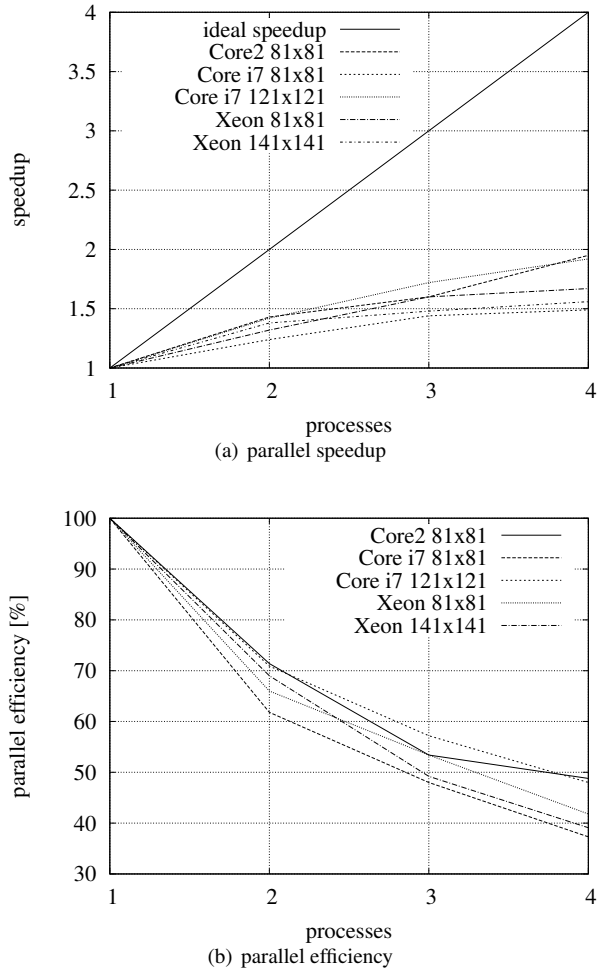
As this paper describes work in progress, the numerical error using an adaptive grid discretisation scheme has still higher errors than expected. The next steps will accordingly be, to improve the numerical scheme for the distribution of the values from one grid part to the other, especially in between coarse and fine cells.

As mentioned before, a higher order temporal discretisation scheme has to be implemented in order to increase the time step size and still get reasonable results.

A next step is to exploit the special design of the code in order to implement a parallel concept. As mentioned in section II, the local computations on the grid can be executed in parallel while the communication step needs global access and, thus, synchronisation between the sub-grids is necessary. A good distribution of sub-grids to different processes depending on the communication layout has to be chosen to ensure minimal communication effort. One master process should not be handling all the communications but delegate them to separate handlers who organise communication between the working nodes to ensure an excellent load balancing and efficient communication patterns as depicted by Mundani et al. [8].

V. Conclusion

In this paper, we have presented an adaptive data structure management for the simulation of engineering problems such as the temperature diffusion equation or the Navier-Stokes equa-



(a) parallel speedup

(b) parallel efficiency

Figure 10. parallel speedup and efficiency computed on shared memory machines using different architectures and different grid sizes

tions. Example applications were computed as far as the presented code is implemented at the moment. As soon as the tasks described in section IV have been finished, next steps will comprise the improvement of the numerical algorithms. The adaptive implementation of the finite difference grid as well as the finite volume grid have shown promising results and the authors look forward to further increase efficiency and handle real world problems rather than test case scenarios.

The ultimate goal is to compute an adaptive fluid-flow simulation around the power plant model introduced in the motivation part in order to compare the results in terms of accuracy and performance between a parallel adaptive computation versus a pure uniform one.

VI. Acknowledgment

This publication is based on work supported by Award No. UK-c0020, made by King Abdullah University of Science and Technology (KAUST).

References

[1] University of North Carolina at Chapel Hill, *Power Plant Model*, 2001.

[2] H. Samet, "The quadtree and related hierarchical data structures," *ACM Comput. Surv.*, vol. 16, pp. 187–260, June 1984.

[3] G. Barequet, B. Chazelle, L. J. Guibas, J. S.B. Mitchell, and A. Tal, "Boxtree: A hierarchical representation for surfaces in 3d," *Computer Graphics Forum*, vol. 15, no. 3, pp. 387–396, 1996.

[4] P. Coelho, J. C. F. Pereira, and M. G. Carvalho, "Calculation of laminar recirculating flows using a local non-staggered grid refinement system," *Int. J. Numer. Meth. Fluids*, vol. 12, no. 6, pp. 535–557, 1991.

[5] C. Hirsch, *Numerical Computation of Internal and External Flows, Volume 1*, Butterworth–Heinemann, 2nd edition edition, 2007.

[6] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer, 3rd, rev. ed edition, 2002.

[7] U. Ghia, K. N. Ghia, and C. T. Shin, "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method," *Journal of Computational Physics*, vol. 48, no. 3, pp. 387 – 411, 1982.

[8] R.-P. Mundani, A. Düster, J. Knezevic, A. Niggl, and E. Rank, "Dynamic load balancing strategies for hierarchical p-FEM solvers," in *Proc. of the 16th EuroPVM/MPI Conference*. 2009, Springer.

Simulated Docking of Darunavir with the HIV-1 L76V Mutant Protease Active Site

Jack K. Horner
P.O. Box 266
Los Alamos NM 87544 USA

Abstract

Darunavir is a second-generation HIV-1 protease inhibitor. Here I provide a computational docking analysis of Darunavir with the active site of the HIV-1 mutant L76V protease. The relatively low binding energy and high inhibition constant of Darunavir with the mutant protease suggests that Darunavir would not be as effective against this mutant as it would against other HIV-1 proteases. These results are consistent with clinical observations and demonstrate the utility of computational docking assessments of HIV-1 protease inhibitors.

Keywords: HIV-1, protease inhibitor, Darunavir, L76V

1.0 Introduction

Darunavir ([(1*R*,5*S*,6*R*)-2,8-dioxabicyclo[3.3.0]oct-6-yl] *N*-[(2*S*,3*R*)-4-[(4-aminophenyl)sulfonyl- (2-methylpropyl)amino]-3-hydroxy-1-phenyl-butan-2-yl] carbamate; [8]) is a second-generation protease inhibitor (PI), designed specifically to overcome problems with the older therapeutics in this class, such as indinavir ([7]), which had severe side effects, required a high therapeutic dose, was costly to manufacture, and showed a disturbing susceptibility to drug resistant mutations.

Darunavir was designed to form robust interactions with proteases from many strains of HIV, including those from treatment-experienced patients with multiple resistance mutations to PIs ([3], [9]). In the clinic, HIV-1 mutants containing the L76V protease have exhibited resistance to Darunavir ([6]).

2.0 Method

The general objective of this study was to computationally assess the binding energy of the active site of the crystallized HIV-1 protease L76V with Darunavir. Unless otherwise noted, all processing described in this section was performed on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 (clocked @ 2.33 GHz) and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)* operating environment.

Protein Data Bank (PDB) 3OY4 is a structural description of the crystallized HIV-1 L76V protease complexed with Darunavir ([1]).

3OY4 consists of two chains, designated Chain A and Chain B. 3OY4 was downloaded from PDB ([1]) on 22 February 2011. A PDB description of Darunavir was extracted from PDB 3OY4 using Microsoft *Word*. The automated docking suite *AutoDock Tools* v 4.2 (ADT, [2]) was used to perform the docking of Darunavir to the receptor. More specifically, in ADT, approximately following the rubric documented in [4]

-- The water in 3OY4 was deleted

-- The active site of the protease was extracted (PDB 3OY4 identifies the active site as 17 amides. Chain A contains nine of these amides: ASP 25, GLY27, ALA28, ASP30, VAL32, GLY48, GLY49, VAL82, ILE84. Chain B contains the

remaining eight amides: ASP25, GLY27, ALA28, ASP29, ASP30, GLY48, GLY49, ILE50.)

-- the hydrogens, charges, and torsions in the ligand and active site were adjusted in accordance with the ADT default recommendations, and finally, the ligand, assumed to be flexible wherever that assumption is physically possible, was auto-docked to the active site, assumed to be rigid, using the Lamarckian genetic algorithm implemented in ADT.

The atomic positions of the ligand determined by *AutoDock* were compared with the corresponding positions in PDB 3OY4.

The ADT parameters for the docking are shown in Figure 1. Most values are, or are a consequence of, ADT defaults.

```
autodock_parameter_version 4.2       # used by autodock to validate parameter set
outlev 1                             # diagnostic output level
intelec                              # calculate internal electrostatics
seed pid time                        # seeds for random generator
ligand_types A C NA OA N S           # atoms types in ligand
fld Darunavir_receptor.maps.fld      # grid_data_file
map Darunavir_receptor.A.map         # atom-specific affinity map
map Darunavir_receptor.C.map         # atom-specific affinity map
map Darunavir_receptor.NA.map        # atom-specific affinity map
map Darunavir_receptor.OA.map        # atom-specific affinity map
map Darunavir_receptor.N.map         # atom-specific affinity map
map Darunavir_receptor.S.map         # atom-specific affinity map
elecmap Darunavir_receptor.e.map     # electrostatics map
desolvmap Darunavir_receptor.d.map   # desolvation map
move Darunavir.pdbqt                 # small molecule
about 19.7094 29.7163 13.8994        # small molecule center
tran0 random                         # initial coordinates/A or random
axisangle0 random                    # initial orientation
dihe0 random                         # initial dihedrals (relative) or random
tstep 2.0                            # translation step/A
qstep 50.0                           # quaternion step/deg
dstep 50.0                           # torsion step/deg
torsdof 12                            # torsional degrees of freedom
rmstol 2.0                           # cluster_tolerance/A
extnrg 1000.0                        # external grid energy
e0max 0.0 10000                      # max initial energy; max number of retries
ga_pop_size 150                      # number of individuals in population
ga_num_evals 2500000                 # maximum number of energy evaluations
ga_num_generations 27000             # maximum number of generations
ga_elitism 1                         # number of top individuals to survive to
                                       next generation
ga_mutation_rate 0.02                # rate of gene mutation
ga_crossover_rate 0.8                # rate of crossover
ga_window_size 10                    #
ga_cauchy_alpha 0.0                  # Alpha parameter of Cauchy distribution
ga_cauchy_beta 1.0                   # Beta parameter Cauchy distribution
set_ga                               # set the above parameters for GA or LGA
sw_max_its 300                       # iterations of Solis & Wets local search
sw_max_succ 4                        # consecutive successes before changing rho
sw_max_fail 4                        # consecutive failures before changing rho
sw_rho 1.0                           # size of local search space to sample
sw_lb_rho 0.01                       # lower bound on rho
ls_search_freq 0.06                  # probability of performing local search on
                                       individual
set_psw1                             # set the above pseudo-Solis & Wets parameters
unbound_model bound                  # state of unbound ligand
ga_run 10                            # do this many hybrid GA-LS runs
analysis                             # perform a ranked cluster analysis
```

Figure 1. ADT parameters for the docking in this study

3.0 Results

The interactive problem setup, which assumes familiarity with the general L76V protease "landscape", took about 10 minutes in ADT; the docking proper, about 31 minutes on the platform described in Section 2.0 The platform's performance monitor suggested that the calculation was more or less uniformly distributed across the four processors at ~25% of peak per processor (with occasional bursts to 40% of peak), and required a constant 2.4 GB of memory.

Figure 2 shows the Darunavir/receptor energy and position summary produced by ADT. The estimated free energy of binding is ~ -4.5 kcal/mol; the estimated inhibition constant, ~453 microMolar at 298 K.

LOWEST ENERGY DOCKED CONFORMATION from EACH CLUSTER

```
Keeping original residue number (specified in the input PDBQ file) for outputting.

MODEL        1
USER    Run = 1
USER    Cluster Rank = 1
USER    Number of conformations in this cluster = 1
USER
USER    RMSD from reference structure       = 4.508 A
USER
USER    Estimated Free Energy of Binding    =   -4.56 kcal/mol  [=(1)+(2)+(3)-(4)]
USER    Estimated Inhibition Constant, Ki   =  453.28 uM (micromolar)  [Temperature =
298.15 K]
USER
USER    (1) Final Intermolecular Energy     =   -8.14 kcal/mol
USER        vdW + Hbond + desolv Energy     =   -7.85 kcal/mol
USER        Electrostatic Energy            =   -0.29 kcal/mol
USER    (2) Final Total Internal Energy     =   -2.49 kcal/mol
USER    (3) Torsional Free Energy           =   +3.58 kcal/mol
USER    (4) Unbound System's Energy  [=(2)] =   -2.49 kcal/mol
USER
USER
USER
USER    DPF = Darunavir.dpf
USER    NEWDPF move    Darunavir.pdbqt
USER    NEWDPF about   19.709400 29.716299 13.899400
USER    NEWDPF tran0   19.797619 32.365647 14.420779
USER    NEWDPF axisangle0     -0.294939 -0.728755 0.618002 -160.087643
USER    NEWDPF quaternion0    -0.290497 -0.717780 0.608695 -0.172895
USER    NEWDPF dihe0   137.80 57.18 -171.62 146.12 -43.47 -63.08 -98.83 141.49 -179.91 -
8.52 177.40 -31.58
USER
USER                              x       y       z    vdW   Elec        q     RMS
ATOM      1  C19 017 B 200      19.440  31.938  15.057 -0.17 +0.06    +0.145  4.508
ATOM      2  C17 017 B 200      18.349  31.103  14.365 -0.22 +0.10    +0.192  4.508
ATOM      3  O18 017 B 200      18.132  29.884  15.088 -0.22 -0.04    -0.220  4.508
ATOM      4  C16 017 B 200      18.664  30.820  12.896 -0.26 +0.08    +0.123  4.508
ATOM      5  N11 017 B 200      19.044  29.399  12.781 -0.18 -0.09    -0.174  4.508
ATOM      6  S8  017 B 200      17.926  28.489  12.024 -0.30 +0.16    +0.247  4.508
ATOM      7  O9  017 B 200      18.404  28.211  10.658 -0.23 -0.14    -0.206  4.508
ATOM      8  O10 017 B 200      16.705  29.314  11.992 -0.31 -0.17    -0.206  4.508
ATOM      9  C5  017 B 200      17.667  27.089  12.768 -0.20 +0.06    +0.099  4.508
ATOM     10  C4  017 B 200      17.856  25.888  12.092 -0.18 +0.01    +0.014  4.508
ATOM     11  C3  017 B 200      17.624  24.677  12.741 -0.19 +0.01    +0.022  4.508
ATOM     12  C2  017 B 200      17.206  24.666  14.067 -0.19 +0.08    +0.103  4.508
ATOM     13  N1  017 B 200      16.981  23.483  14.689 -0.16 -0.13    -0.148  4.508
ATOM     14  C7  017 B 200      17.015  25.862  14.754 -0.32 +0.01    +0.022  4.508
ATOM     15  C6  017 B 200      17.246  27.069  14.098 -0.29 +0.01    +0.014  4.508
ATOM     16  C12 017 B 200      20.423  29.176  12.280 -0.20 +0.05    +0.088  4.508
```

```
ATOM     17  C13 017 B 200      21.069  28.011  13.031 -0.23 +0.01    +0.017  4.508
ATOM     18  C14 017 B 200      22.578  27.929  12.804 -0.20 +0.00    +0.002  4.508
ATOM     19  C15 017 B 200      20.756  28.100  14.528 -0.31 +0.00    +0.002  4.508
ATOM     20  N20 017 B 200      18.702  32.887  15.876 -0.12 -0.11    -0.211  4.508
ATOM     21  C21 017 B 200      17.951  32.546  16.921 -0.11 +0.23    +0.384  4.508
ATOM     22  O22 017 B 200      17.791  31.427  17.386 -0.16 -0.11    -0.236  4.508
ATOM     23  O23 017 B 200      17.350  33.657  17.592 -0.16 -0.23    -0.283  4.508
ATOM     24  C24 017 B 200      16.328  34.362  16.908 -0.19 +0.15    +0.161  4.508
ATOM     25  C25 017 B 200      15.103  33.522  17.213 -0.25 +0.20    +0.175  4.508
ATOM     26  O26 017 B 200      14.749  33.954  18.529 -0.74 -0.54    -0.328  4.508
ATOM     27  C27 017 B 200      15.089  35.327  18.703 -0.17 +0.32    +0.238  4.508
ATOM     28  O28 017 B 200      15.743  35.561  19.951 -0.43 -0.51    -0.331  4.508
ATOM     29  C31 017 B 200      16.054  35.692  17.585 -0.15 +0.09    +0.089  4.508
ATOM     30  C29 017 B 200      17.110  35.944  19.739 -0.13 +0.13    +0.145  4.508
ATOM     31  C30 017 B 200      17.200  36.405  18.294 -0.10 +0.03    +0.040  4.508
ATOM     32  C32 017 B 200      20.345  31.137  15.992 -0.24 +0.00    +0.053  4.508
ATOM     33  C38 017 B 200      21.778  31.599  15.849 -0.14 +0.00    -0.020  4.508
ATOM     34  C33 017 B 200      22.395  31.523  14.609 -0.10 -0.00    -0.004  4.508
ATOM     35  C34 017 B 200      23.715  31.932  14.448 -0.07 +0.00    +0.000  4.508
ATOM     36  C35 017 B 200      24.431  32.401  15.543 -0.05 +0.00    +0.000  4.508
ATOM     37  C36 017 B 200      23.819  32.478  16.788 -0.08 +0.00    +0.000  4.508
ATOM     38  C37 017 B 200      22.495  32.069  16.947 -0.13 +0.00    -0.004  4.508
```

Figure 2. ADT's Darunavir energy and position predictions.

Figure 3 is a rendering of the active-site/inhibitor configuration computed in this study.

An analysis of the ligand's computed atomic positions showed all interatomic distances to the receptor were within 10% of the positions of those atoms as reported in PDB 3OY4 (this shows that the computed docking credibly approximated the original crystal structure).

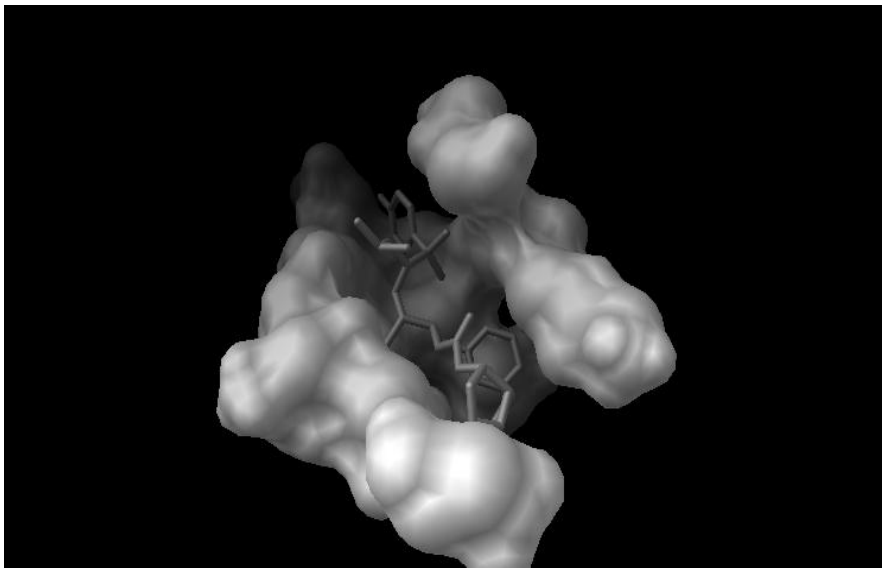


Figure 3. Rendering of Darunavir computationally docked with the active site of PDB 3OY4 (the HIV-1 L76V mutant protease). The inhibitor is shown in stick form. Only the interior, inhibitor-containing region of the molecular surface of the active site can be

compared to *in situ* data: the surface distal to the interior is a computational artifact, generated by the assumption that active site is detached from the rest of the receptor.

4.0 Discussion

The method described in Section 2.0 and the results of Section 3.0 motivate several observations:

1. The relatively small binding energy (~ -4.6 kcal/mol) and high inhibition constant (~453 microMolar) of Darunavir with the HIV-1 L76V mutant protease suggests that Darunavir would not be as effective against this mutant as it would against other HIV-1 proteases ([5]). This prediction is consistent with clinical observations ([6]), thus demonstrating the utility of computational docking analysis in predicting the efficacy of HIV-1 protease inhibitors.

2. The docking study reported here assumes that the receptor is rigid, and as a result, the calculation does not reflect any energy contributions of receptor "flexing" to the interaction of the ligand with native unliganded receptor.

3. The analysis described in Sections 2.0 and 3.0 assumes the protease is in a crystallized form (isolated at ~278 K). *In situ*, at physiologically normal temperatures (~310 K), the receptor is not in crystallized form. The ligand/receptor conformations *in situ*, therefore, may not be identical to their conformations in the crystallized form.

4. 3OY4 identifies highly similar sets of amides on each of Chains A and B as part of the active site. It is unclear whether the chains *in situ* collectively form the active site, or whether each chain has an standalone active site.

5. Minimum-energy search algorithms other than the Lamarckian genetic algorithm used in this work could be applied to this docking problem. Future work will use Monte Carlo/simulated annealing algorithms. In addition, a variety of torsion and charge models could be applied to this problem, and future work will do so.

5.0 Acknowledgements

This work benefited from discussions with Tony Pawlicki. For any problems that remain, I am solely responsible.

6.0 References.

[1] Schiffer CA, Nalivaika EA, Bandaranayake RM. Crystal structure of HIV-1 L76V protease in complex with the protease inhibitor Darunavir. PDB 3OY4. http://pdb202.rcsb.org/pdb/explore.do?structureId=3OY4

[2] Morris GM, Goodsell DS, Huey R, Lindstrom W, Hart WE, Kurowski S, Halliday S, Belew R, and Olson AJ. *AutoDock* v4.2. http://autodock.scripps.edu/. 2010.

[3] Clotet B, Bellos N, Molina J-M, Cooper D, Goffard J-C, Lazzarin A, Wöhrmann A, Katlama C, Wilkin T, Haubrich R, et al., Efficacy and safety of darunavir-ritonavir at week 48 in treatment-experienced patients with HIV-1 infection in POWER 1 and 2: a pooled subgroup analysis of data from two randomised trials. *The Lancet* 369 (7 April 2007-13 April 2007), 1169-1178.

[4] Huey R and Morris GM. *Using AutoDock 4 with AutoDock Tools: A Tutorial.* 8 January 2008. http://autodock.scripps.edu/.

[5] Cheng Y and Prusoff WH. Relationship between the inhibition constant (K_i) and the concentration of inhibitor which causes 50 per cent inhibition (I_{50}) of an enzymatic reaction. *Biochemical Pharmacology* 22 (December 1973), 3099–3108. doi:10.1016/0006-2952(73)90196-2.

[6] Young TP, Parkin NT, Stawiski E, Pilot-Matias T, Trinh R, Kempf DJ, and Norton M. Prevalence, mutation patterns, and effects on protease inhibitor susceptibility of the L76V mutation in HIV-1 protease. Antimicrobial Agents and Chemotherapy 54 (November 2010), 4903-4906. doi:10.1128/AAC.00906-10.

[7] The Drug Bank. *Indinavir*. http://www.drugbank.ca/drugs/APRD00069#.

[8] NCBI. PubChem. *Darunavir*. http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=213039.

[9] US Department of Health and Human Services. AidsInfo. *Darunavir*. http://www.aidsinfo.nih.gov/DrugsNew/DrugDetailT.aspx?int_id=397&ClassID=0&TypeID=0.

A Time-Series Model of Dinosauria Diversity

Jack K. Horner
P.O. Box 266
Los Alamos NM 87544 USA

Abstract

It is widely held that the dinosaurs were driven to near extinction because of the Chicxulub asteroid collision with the Earth about 65 million years ago (MA). Without doubt, dinosaur diversity in the fossil record after the collision was at most a percent of what it was prior to the collision. But whether the collision was the principal cause of the extinction is more difficult to assess. Here I compute and compare models of the time series of the number of genera of the Dinosauria and the Mollusca in the period 230 MA - 50 MA. The models are agnostic about whether specific events occurred, and in this sense do not require Chicxulub to explain the Dinosaurian diversity collapse.

Keywords: K-T boundary, time-series, dinosaur extinction, Chicxulub, Cretaceous-Paleogene boundary

1.0 Introduction

Approximately 65.5 million years ago, an asteroid hypothesized to have a speed of ~20 km/s and a diameter of ~10 km struck the ocean near present-day Chicxulub, Yucatan, Mexico ([1]-[3],[5]-[6]). The impact hurled molten rock and rock vapor into the atmosphere, creating, at least briefly, a crater nearly 100 km in diameter and 18 km deep ([7]), spawning fires across the planet. Debris from the impact and fires may have reduced sunlight for years. It is widely held that the largest land animals, including the dinosaurs of the time, were driven to near extinction by the event ([4],[6]).

Without doubt, dinosaur genera diversity (in this paper, defined as the number of genera in a time bin) in the fossil record after the collision is at most a percent of what it was before the collision. Whether the collision was the principal cause of the near extinction of the superorder Dinosauria ([19]; hereafter, "Dinosauria"), however, is problematic, because there is no definitive evidence that the Chicxulub event caused that demise. Less cataclysmic regimes, such as a small decrease in average annual surface temperature, or a persistent reduction in sunlight reaching the Earth's surface (perhaps caused by volcanic ejecta), could have made the planet untenable for the dinosaurs, who may have been highly sensitive to temperature-induced changes in food distribution ([22]). Unfortunately, even these hypotheses are not uniquely determined by the evidence.

A predictive time series model ([14]) of the Dinosauria genus-abundance data, because it is inherently agnostic about the occurrence of specific causes, would show that Chicxulub is not *necessary* to explain the observed decline in Dinosauria diversity. In such a series, an observation x_t is presumed to be a value of some random variable X_t; the time series $\{x_1, x_2, \ldots x_t, \ldots\}$, a single realization of a stochastic process (*i.e.*, a sequence of random variables) $\{X_1, X_2, \ldots X_t, \ldots\}$. A fundamental assumption of time series modeling is that the value of the series at time t, X_t, depends only on its previous values (deterministic part) and on a random disturbance (stochastic part). Furthermore, if the dependence of X_t on the previous p values is assumed to be linear, we can write ([21], p. 12)

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + \mathbf{D}_t \qquad \text{Eq. 1.1}$$

where $\{\varphi_1, \varphi_2, \ldots, \varphi_p\}$ are real constants. $\mathbf{D}_t$ is the stochastic disturbance at time *t*, and it is usually modeled as a linear combination of zero-mean, uncorrelated random variables or a zero-mean *white noise model* $\{D_t\}$

$$\mathbf{D}_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \ldots + \theta_q Z_{t-q}. \quad \text{Eq. 1.2}$$

($\{Z_t\}$ is a white noise model with mean 0 and variance σ^2 if and only if $E\ Z_t = 0$, $E\ (Z_t)^2 = \sigma^2$ for all *t*, and $E\ Z_s\ Z_t = 0$ if $s \neq t$, where *E* denotes the expectation.) Z_t is often referred to as the *random error* or *noise* at time *t*. The constants $\{ \varphi_1, \varphi_2, \ldots, \varphi_p \}$ and $\{ \theta_1, \theta_2, \ldots, \theta_q \}$ are called *autoregressive (AR) coefficients* and *moving average (MA) coefficients*, respectively.

Equations 1.1 and 1.2 jointly define a zero-mean *autoregressive moving average (ARMA) model* of orders *p* and *q*, or ARMA(*p*, *q*). If each of $\theta_1, \theta_2, \ldots, \theta_q$ are 0, Eqs. 1.1 and 1.2 define an *autoregressive* model of order *p*, or *AR(p)*.

The time-series analysis method used in this study assumes that a series of interest arises from a second-order or weak-stationary process. Roughly put, a process $\{X_t\}$ is stationary if its statistical properties do not change over time. (See [14], Chapter 2 or [21], p. 14, for a detailed account of weak stationarity).

2.0 Method

Two data sets were downloaded from *The Paleobiology Database* ([8]) on 3 January 2011 to a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 (clocked @ 2.33 GHz) and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)* operating environment and the *Mozilla Firefox* v3.6.13 browser, connected by a 1.5 Mbit/s DSL link to the Internet. The query values used to generate these data sets are shown in Figure 1.

Dinosauria data set, 230 MA - 50 MA ([10]):
Output data = occurrence list
Output delimiter = comma-delimited text
Taxon or taxa to include = Dinosauria
Oldest and youngest intervals = 230 - 50
Continents = Africa, Antarctica, Asia, Australia, Europe,
Indian Ocean, Oceania, North America, South America

Mollusca data set, 230 MA-50 MA ([11]):
Output data = occurrence list
Output delimiter = comma-delimited text
Taxon or taxa to include = Mollusca
Oldest and youngest intervals = 230 - 50
Continents = Africa, Antarctica, Asia, Australia, Europe,
Indian Ocean, Oceania, North America, South America

Figure 1. Query values used to generate the data sets used in this study.

The time interval for the data sets, 230 MA - 50 MA, spans the Dinosauria from their nominal first appearance in the fossil record ([19]) until well after the Chicxulub impact. The Mollusca ([20]) were chosen for comparison with the Dinosauria because there is a large body of Mollusca fossil data available, and the Mollusca are plausible as a "control" for Dinosauria analysis. By default, the query shown in Figure 1 returns data at the genus level.

The genus time-range files from each returned data set were saved. These (comma-separated format) files contain, among other things, the beginning (column *bottom of range),* and end (column *top of range*), times at which [8] reports a genus existed, one row per genus. These range files were imported by Microsoft *Excel* 2007 and the rows in the resulting spreadsheet were sorted in decreasing value of *top of range.* This sorting facilitated rapid visual identification of those genera that arose and perished in the sampled interval.

Excel functions were used to export columns *bottom of range* and *top of range* to a Windows text file. These text files were converted to UNIX textfile format using the Cygwin *dos2unix* utility. The *count_taxa* software ([12]) running under Cygwin environment (itself under Vista) was then used to count the resulting genera ranges into 1 MA bins, on the hardware described above. A genus was assumed to exist in given bin if the midpoint of that bin lay in the closed interval [*bottom of range, top of range*]. The output of *count_taxa* is a genera-abundance time-series that has a uniform distribution of time-differences (in this case, 1 MA) between adjacent ime-values.

The files output by *count_taxa* were imported to *Mathematica* using the *Time Series* add-on package ([9]). Both data sets were inspected to determine that they did not contradict with the hypothesis that the Chicxulub asteroid could have been the cause of the demise of these taxonomic groups.

Stationary linear time-series models of each of the data sets for the interval 230-50 MA were computed in *Mathematica* as follows (the general theory of the analysis can be found in [14], Chapters 2-3; the entire *Mathematica* script used in this study can be obtained as noted in [18]). Graphs of the time-series derived from *count_taxa* were inspected for trending. Each of the genus time-range data sets was then zero-meaned (i.e., the mean of the series was subtracted from each of the data values in the series). The Hannan-Rissanen method ([14], Section 5.1.4) was applied to the zero-meaned series to obtain six preliminary models of each of the series. The Akaike Information Criterion values for each of the models were then computed. (The AIC is a penalty function that, when minimized, balances the risks of over-, and under-, fitting ([13]; [14], p. 173).) The two models of each series with the lowest AIC values were selected for subsequent analysis. A conditional maximum likelihood estimate ([14], Section 5.2) of the parameters of each of the selected models was then obtained. The AIC values for the resulting refined models were then determined, and the model with the lowest AIC value was selected for further analysis. The residuals with respect to the selected model were then computed, and the correlation function of the residuals was computed and plotted to assess their convergence. Finally, the portmanteau statistic on the set of residuals for each series, and chi-square statistic for the 95th percentile, were determined; if the chi-square statistic was greater than the portmanteau statistic, the model was accepted ([14], pp. 26, 166-167, 352; [15]).

If *no* model were discovered by this method, it would be strong evidence that no stationary linear model of the time series exists, and, by implication, be a strong suggestion that a nonlinear or nonstationary event (e.g., a cataclysmic) event is required to explain the Dinosaurian time series.

The portion of the *Mathematica* script used to produce the time-series analysis for the Dinosauria genera-count in this study ([18])

is shown in Figure 2; the script for the Mollusca series is highly similar.

```
Needs["TimeSeries`TimeSeries`"];
dinodata =
  ReadList[ToFileName[{"C:", "cygwin", "species_abundance"},
    "Dino_230_50_counts.txt"], {Number, Number}];
n = Length[dinodata];
μdino = Mean[dinodata[[All, 2]]];
meanzerodino = (# - μdino) & /@ dinodata[[All, 2]];
dinomodels = HannanRissanenEstimate[meanzerodino, 5, 5, 5, 6];
If[Head[#] === ARMAModel, Head[#][Length[#[[1]]], Length[#[[2]]]],
   Head[#][Length[#[[1]]]]] & /@ dinomodels;
AIC[#, 181] &  /@ dinomodels;
{arm2} = ConditionalMLEstimate[meanzerodino, #] & /@
   Take[dinomodels, {2}];
AIC[#, 181] & /@ %;
{arm24} =    ConditionalMLEstimate[meanzerodino, #] & /@
   Take[dinomodels, {6}];
AIC[#, 181] & /@ %;
dinores = Residual[meanzerodino, arm2];
dinocorr = CorrelationFunction[dinores, 12];
plotcorr[corr_, opts___] :=
   ListPlot[corr, DataRange -> {0, Length[corr] - 1}, opts];
plotcorr[# &  /@ dinocorr, Joined -> True,
   AxesLabel -> {"k", "ρdino(k)"];
PortmanteauStatistic[dinores, 12];
Quantile[ChiSquareDistribution[11], 0.95];
```

Figure 2. The portion of the *Mathematica* script ([18], responses not shown) used to compute the model for the Dinosauria genera-count time-series used in this study. The analysis script for the Mollusca series is highly similar.

3.0 Results

The *Paleobiology Database* query described in Section 2.0 yielded 5377 occurrences distributed across 986 genus time-ranges for the Dinosauria data set, and 111412 occurrences distributed across 3658 genus time-ranges for the Mollusca data set. The total time to complete the queries and transmit files to the platform described in Section 2.0 was about one minute. Figure 3 shows the number of Dinosauria and Mollusca genera reported in [8], 230-50 MA, 1 MA binning. The shapes of the time series in Figure 3 are remarkably similar: roughly speaking, at any given time, on average the number of Mollusca genera is ~6 times the number of Dinosauria genera. The mean of the number of Dinosauria genera is 88.5; the mean of the number Mollusca genera, 535.1. The ratio of the mean Mollusca, to the mean Dinosauria, genera-count is 6.0.

The data underlying Figure 3 show a sharp decrease in Dinosauria genera-count near the nominal date of the Chicxulub impact (~65.5 MA) , appearing to lend weight to the hypothesis that the collision was a major contributor to the demise of the Dinosauria. But it is less clear that much of a causal connection can be inferred from this correlation. By way of comparison, at ~112 MA, 127 of the ~175, or ~70%, of the Dinosauria genera reported in [8] vanished, unassociated with any known asteroid collision with Earth. (New genera were being created at about the same rate as those that vanished, so the net genera count, as shown by Figure 3, decreased only ~10%.) This implies there is at least one apparently non-asteroidal genera-extinction rate in the Dinosauria fossil record whose magnitude is comparable to that which occurred at ~65.5 MA. The Dinosauria genus-diversity collapse at 65.5 MA, therefore, could be explained if the relative extinction rate of ~112 MA were in progress and few new Dinosauria genera were being created past 65.5 MA. Environmental changes less cataclysmic than Chicxulub, such as a small decrease in average annual surface temperature, or a decrease in sunlight reaching the Earth's surface (e.g., due to volcanic ejecta), could have catastrophically compromised the viability of the dinosaurs within a few years. Although the plausibility of these alternative hypothesis shows that the Chicxulub-extinction hypothesis is not uniquely determined by the evidence, the alternative hypotheses are themselves variously problematic.

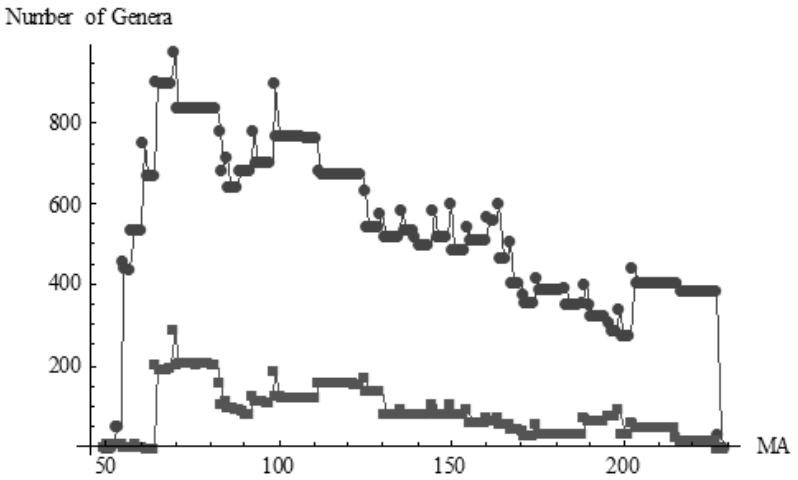


Figure 3. Number of Dinosauria (squares) and Mollusca (circles) genera in [8], 230-50 MA, 1 MA binning. Note the collapse of diversity in both groups beginning at ~65 MA. Note also the shape similarity of these time series. In the figure, time increases from the origin of the Dinosauria (~230 MA) from *right to left*. The figure was generated by the *Mathematica* ([9]) ListLinePlot command.

In any case, a time-series model is agnostic about whether specific events occurred, and such a model can be fitted to the Dinosauria data, showing that the Dinosauria time series does not require Chicxulub per se (nor does it prohibit such an event).

In *Mathematica*, the expression

```
ARMAModel[{φ1, φ2, … , φp},
    {θ1, θ2, … , θq}, σ^2]
```

specifies an ARMA(*p*, *q*) model with AR coefficients $\{\varphi_1, \varphi_2, \ldots, \varphi_p\}$, and MA coefficients $\{\theta_1, \theta_2, \ldots, \theta_q\}$, and noise variance σ^2. The *Mathematica* expression

```
ARModel[{φ1, φ2, … , φp}, σ^2]
```

specifies an AR(*p*) model with AR coefficients $\{\varphi_1, \varphi_2, \ldots, \varphi_p\}$, and noise variance σ^2.

One model for each series passed the portmanteau test described above:

Dinosauria model (ARM[2]):

```
ARModel[{0.786802,
  0.154315}, 8883.25]
```

Mollusca model (ARMA[1,1]):

```
ARMAModel[{0.963448},
{-0.118255},3634.09]
```

The existence of the first model demonstrates that Chicxulub is not necessary to explain the Dinosaurian demise. Note that the data did not require de-trending (e.g., by differencing; see [21] for details) in order to produce a model satisfying the test of significance described above.

The time to execute *count_taxa* on the platform described in Section 2.0 was less than 0.1 second per data set. The time to execute the *Mathematica* time-series analysis used in this study ([18]) was approximately three seconds on the same platform.

4.0 Discussion

Sections 2.0 and 3.0 motivate at least three observations:

1. In this paper, a genus was assumed to exist throughout the interval [*bottom of range, top of range*] reported for that genus in the taxonomic range files described in Section 2.0. More sophisticated existence tests that exploit various abundance weightings (such as *geometric mean abundance* reported in the taxonomic range files) are of course possible. Whether using these weightings would significantly affect the results reported in Section 3.0 will be the subject of future work.

2. The visual similarity of the times series in Figure 3 suggests that the relative creation and extinction rates of Dinosauria, and Mollusca, genera may have a common general dynamic, but the time series analysis models shown in Section 3.0 suggest otherwise.

3. One might conjecture that the variance in the Dinosauria time-series model is dominated by the diversity collapse at ~65 MA. However, careful inspection of Figure 3 shows that this is not the case. In particular, the variance by definition is essentially averaged over the number of points in the data set. Although the large standard deviation of the point at 65 MA contributes to this average, the average is dominated by standard deviations of 180 points whose standard deviations (the square root of the variance) are small compared to the standard deviation of the point at 65 MA. In addition, at ~84 MA, the number of Dinosauria genera increases by ~100, which is larger than the standard deviation in the time-series model.

5.0 Acknowledgements

This work benefited from discussions with Kris Krishtalka of the University of Kansas Biodiversity Institute, Tony Pawlicki, and George Hrabovsky of the Madison Area Science and Technology Institute for Scientific Computing. For any problems that remain, I am solely responsible.

6.0 References

[1] Baldwin EC, Vocadlo L, and Crawford IA. Influence of target yield stress on crater dimensions: a numerical approach based on Chicxulub. *Lunar and Planetary Science Conference* XXXVII (2006). Abstract 1887.

[2] Gisler G et al. Two- and three-dimensional simulations of asteroid ocean impacts. *Science of Tsunami Hazards* 21 (2003), 119-134.

[3] Morgan JV et al. Peak ring formation in large impact craters. *Earth and Planetary Science Letters* 183 (2000), 347-354.

[4] Alvarez LW et al. Extraterrestrial cause for the Cretaceous-Tertiary extinction. *Science* 208 (1980), 1095–1108.

[5] Ward W et al. Yucatan subsurface stratigraphy: implications and constraints for the Chicxulub impact. *Geology* 23 (1995), 873-876.

[6] Schulte P et al. The Chicxulub asteroid impact and mass extinction at the Cretaceous-Paleogene boundary. *Science* 327 (5 March 2010), 1214-1218.

[7] Horner JK. Sensitivity of maximum crater depth and diameter to bolide density in a Chicxulub-like asteroid collision. Proceedings of the 2007 International Conference on Scientific Computing, 200-208. CSREA Press. 2007.

[8] *The Paleobiology Database*. http://paleodb.org/cgi-bin/bridge.pl. 2010.

[9] Wolfram Research. *Mathematica Home Edition* v7.0 (2010) with the *Time Series* v1.4 add-on package (2007).

[10] The major contributors to this data set were Carrano M, Alroy J, Butler R, and Uhen M.

[11] The major contributors to this data set were Kiessling W, Hendy A, Carrano M, Alroy J, Aberhan M, Villier L, Harries P, Ivany L, Fursich F, Miller A, Patzkowsky M, Bottjer D, Pálfry J, McGowan A, Butler R, Wilf P, and Marshall C.

[12] Horner JK. *count_taxa*, a perl program to count taxa in a taxonomic range file. Available from the author on request.

[13] Akaike H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19 (1974), 716–723.

[14] Brockwell PJ and Davis RA. *An Introduction to Times Series and Forecasting*. Second Edition. Springer. 2002.

[15] Ljung GM and Box GEP. On a measure of lack of fit in time series models. *Biometrika* 65 (1978), 297–303.

[18] Horner JK. *Mathematica* notebook *Dino_Mollusca_230_50_time_series.nb*, available from the author on request. 2011.

[19] Weishampel DB, Dodson P, and Osmólska H, eds. *The Dinosauria: Second Edition.* University of California. 2004.

[20] Ponder W and Lindberg DR, eds. *Phylogeny and Evolution of the Mollusca*. University of California. 2008.

[21] Wolfram Research. *Mathematica Time Series* v1.4 manual. 2007. http://media.wolfram.com/documents/TimeSeriesDocumentation.pdf.

[22] Amstrup SC et al. Greenhouse gas mitigation can reduce sea-ice loss and increase polar bear persistence. *Nature* **468** (15 December 2010), 955-958. doi:10.1038/nature09653.

Noise and Oscillations in Chemically Reacting Systems

Silvana Ilie[1,2,3] **and Ronak Savani**[4]
[1]Department of Mathematics, Ryerson University, Toronto, Ontario, M5B 2K3, Canada
[4]Department of Aerospace Engineering, Ryerson University, Toronto, Ontario, M5B 2K3, Canada

Abstract— *Sensitivity analysis is a powerful technique which allows the exploration of a biochemical system dynamics. It is widely used in quantifying properties of the system, such as robustness with respect to perturbations in parameters. This is a critical problem in studying biochemical systems, and in particular cellular dynamics, as often some parameters for the kinetics of interaction are poorly known. In this paper, we investigate a parametric sensitivity analysis for mathematical models of well-stirred biochemical systems which exhibit an oscillatory behavior. This analysis will enable the identification of the key reaction rate parameters and it may give important biological insight into the mechanism that generates the oscillatory dynamics. Numerical results on two realistic models show the excellent performance of the proposed method.*

Keywords: Stochastic modeling, sensitivity analysis, stochastic simulation, Gillespie algorithm, oscillatory biochemical systems.

1. Introduction

Stochastic modeling and computer simulations have been successfully used for explaining many important cellular processes [3], [6], [16], [18]. There are currently several levels of refinement used for modeling biochemically reacting systems. Often chemical kinetic models represent cellular processes as systems of chemical reactions. Traditionally, these processes were modeled as continuous deterministic systems, by ordinary differential equations. However, the small number of some molecular species within a cell invalidate the hypothesis of continuity and the stochastic fluctuations which are captured by the experiments are neglected by such models [24], [25]. Therefore stochastic discrete models are needed to accurately describe the dynamics of many biological processes at the cellular level, where some species have small molecular populations [3], [7], [20].

Many cellular processes have a periodic dynamics. Oscillatory behavior is important in the study of circadian rhythms, of the cell cycle or of periodic neuronal signals [13]. Goodwin introduced an enzymatic control model exhibiting an oscillatory behavior [15]. Circadian rhythms were studied, among others, by Bagheri et al. [1], Leloup & Goldbeter [17], Forger & Peskin [4], [5], Ruoff et al. [21] (see also the review by Goldbeter [13]). Many key genetic networks exhibit an oscillatory behavior. Elucidating the mechanism which generates the oscillatory behavior is critical for understanding the cellular dynamics. Such a mechanism may be very complex. Often, cellular processes involve a large number of molecular species coupled in many regulatory interactions. Thus mathematical models and computer simulations are essential for understanding the nature of these interactions. Moreover, it has been observed that the network topology alone may not be sufficient for explaining the qualitative behavior of the system. In several applications, for the same network of interactions the oscillatory behavior may be present or absent, depending on the ranges of values considered for the kinetic parameters. Thus, knowing the appropriate network of interactions as well as the range of such kinetic parameters are critical for an accurate description of the system dynamics.

In this paper we introduce a new method to investigate the sensitivity with respect to system parameters in (bio)-chemically reacting systems. Sensitivity analysis plays a central role in the study of biochemical systems, being an important aid in their model construction, investigation and validation. It quantifies the dependence of the solution of the mathematical model on the model parameters, such as initial molecular numbers or kinetic constants [23]. Knowledge of robustness or sensitivity of a biochemical system with respect to system parameters is very important, as theses parameters may be poorly known, or they can be subject to change with environmental or intracellular conditions. In this case, sensitivity analysis indicates which parameters are important and thus need to be estimated with higher accuracy. Classical sensitivity analysis [23] focuses on studying the steady-state of biochemical reaction systems. While this remains an important problem, so is the parametric sensitivity of their dynamic behavior, and in particular of their oscillatory behavior. Stochastic biochemical systems are computationally very challenging to simulate and analyze, therefore designing efficient methods which will help validate the model and study its robustness with respect to perturbations would be an important advance.

2. Stochastic modeling and simulation of chemical kinetics

Studying the stochastic behavior of the well-stirred biochemical systems is a difficult task. An accurate model of

[2]Corresponding author: silvana@ryerson.ca
[3]Research partially supported by a grant from Natural Sciences and Engineering Research Council of Canada (NSERC).

biochemical reaction networks is based on the Chemical Master Equation [12]. Most biochemical systems of practical interest typically involve many components interconnected in a complex manner.

We consider a biochemical system containing N reacting species $X_1, \cdots, X_N$ which interact through M elemental reaction channels $R_1, \cdots, R_M$. The system is assumed well-stirred and at thermal equilibrium, in a constant volume. The vector of states is denoted by $\mathbf{x}(t) = (x_i(t)_{i=1,\cdots,N})$, where $x_i(t)$ is the number of X_i molecules at time t. It is a stochastic process. To determine the dynamics of the system, one needs to compute the state vector $\mathbf{x}(t)$, given that at the initial time, $t = t_0$, the system was in the state $\mathbf{x}(t_0) = \mathbf{x}_0$.

A reaction R_j is characterized by its propensity function $a_j(\mathbf{x})$. The propensity function in a given state $\mathbf{x}$, is defined by $a_j(\mathbf{x})dt$ is the the probability that one R_j reaction will take place in the infinitesimal interval $[t, t + dt)$ given that $\mathbf{x}(t) = \mathbf{x}$. The basic elemental reactions have the following propensities:

$X_1 \xrightarrow{c_1}$ products, propensity $a_1(\mathbf{x}) = c_1 x_1$,
$X_1 + X_2 \xrightarrow{c_2}$ products, propensity $a_2(\mathbf{x}) = c_2 x_1 x_2$,
$X_1 + X_1 \xrightarrow{c_3}$ products, propensity $a_3(\mathbf{x}) = \frac{c_3}{2} x_1(x_1 - 1)$.

Denote by $\boldsymbol{\nu}_j \equiv (\nu_{1j}, \ldots, \nu_{Nj})'$ the state change vector corresponding to the reaction R_j. More precisely, ν_{ij} is the change in the number of X_i molecules after one reaction R_j occurs. Thus, each reaction R_j causes a change $\mathbf{x} \to \mathbf{x} + \boldsymbol{\nu}_j$. The matrix $\mathbf{S} = (\nu_{ij})_{1 \le i \le N, 1 \le j \le M}$ is the stoichiometric matrix.

Chemical Master Equation: We define the probability that the state vector at time $t > t_0$ is $\mathbf{x}(t) = \mathbf{x}$ given that at time t_0 was $\mathbf{x}(t_0) = \mathbf{x}_0$ by $P(\mathbf{x}, t | \mathbf{x}_0, t_0) = \text{Prob}\{\mathbf{x}(t) = \mathbf{x}, \text{ given } \mathbf{x}(t_0) = \mathbf{x}_0\}$. The Chemical Master Equation is the most refined model of stochastic chemical kinetics of well-stirred isothermal systems. It is given by

$$\begin{aligned} \frac{d}{dt} P(\mathbf{x}, t | \mathbf{x}_0, t_0) &= \sum_{j=1}^{M} P(\mathbf{x} - \boldsymbol{\nu}_j, t | \mathbf{x}_0, t_0) a_j(\mathbf{x} - \boldsymbol{\nu}_j) \\ &- \sum_{j=1}^{M} P(\mathbf{x}, t | \mathbf{x}_0, t_0) a_j(\mathbf{x}). \end{aligned}$$

It is a (very) large system of ordinary differential equations, one for each possible state of the system subject to the M reaction channels.

Gillespie's Direct Method: Gillespie proposed two methods of Monte Carlo type to compute the solution of the Chemical Master Equation: the Direct Method and the Next Reaction Method [10], [11]. Sample paths $\mathbf{x}(t)$ are generated by computing each reaction, one at a time. Such trajectories are calculated by specifying the reactions and the times of these reactions with their exact probability distribution, as given by the Chemical Master Equation. A good approximation of the statistics for the solution of the Chemical Master Equation is obtained when sufficiently many such sample trajectories are simulated. We present below Gillespie's Direct Method:

1). Calculate the propensity functions, $a_k(\mathbf{x})$, for $1 \le k \le M$, for the current state of the system, $\mathbf{x}(t) = \mathbf{x}$, and the sum of all propensities,
$$a_0(\mathbf{x}) = \sum_{k=1}^{M} a_k(\mathbf{x}) \ .$$
2). Generate two independent unit-interval uniform random numbers r_1 and r_2.
3). Calculate the time to the next reaction by
$$\tau = (1/a_0(\mathbf{x})) \ln(1/r_1) \ .$$
4). Compute the index of the next reaction, the integer j such that
$$\sum_{k=1}^{j-1} a_k(\mathbf{x}) < r_2 a_0(\mathbf{x}) \le \sum_{k=1}^{j} a_k(\mathbf{x}).$$
5). Update the state of the system after one reaction R_j occurred, $\mathbf{x}(t + \tau) = \mathbf{x}(t) + \boldsymbol{\nu}_j$ and set $t = t + \tau$.

3. Sensitivity analysis

Sensitivity analysis has been well-developed mainly in the framework of ordinary differential equations. In this paper, we explore a procedure for sensitivity analysis applied to stochastic biochemical system models with respect to the kinetic parameters (the reaction rate constants).

Denote the solution at time t of the mathematical model by $[x_i(t, \mathbf{p})]_{i=1,\cdots,N}$. The solution depends on the vector of external parameters $\mathbf{p} = (p_1, \cdots, p_m)$. The first order sensitivity matrix has as elements the local sensitivities $\partial x_i(t, \mathbf{p}) / \partial p_j$. These sensitivities give a measure of how the system dynamics is affected by changes in the values of a parameter. A large sensitivity indicates that the system's behavior can change a lot with changes in that parameter. By contrast, a small sensitivity shows that small changes in the parameter value do not have a significant impact on the dynamics of the system. We describe below a *forward sensitivity analysis* for biochemical systems.

The reaction rate equations, which are ordinary differential equations, may be written in the form

$$\frac{d\mathbf{x}}{dt} = \mathbf{S}\mathbf{v}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) \tag{1}$$

where $\mathbf{v}(\mathbf{x}_i(t, \mathbf{p}), \mathbf{p})$ is the M-valued vector of the reaction rates and $\mathbf{S}$ is the stoichiometric matrix. A direct derivation of the variation in time of sensitivities is obtained by differentiating (1) with respect to each p_j for any $j = 1, 2, \ldots, m$:

$$\frac{d}{dt} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} = \mathbf{S} \left[\frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{v}}{\partial \mathbf{p}}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) \right] \ . \tag{2}$$

The sensitivities can then be obtained from solving simultaneously the system for the species (1) coupled with the

auxiliary equations for the sensitivities (2). This is a forward sensitivity analysis of the *full system*.

Often, biochemical systems involve a large number of molecular species interacting through many reaction channels. For these large systems, the above analysis may be computationally expensive. A more efficient method for parametric sensitivity is presented below, based on a reduced form of the reaction rate system which eliminates the redundant state variables.

Denote by N_0 the row rank of the stoichiometric matrix $\mathbf{S}$. A permutation of the rows of the stoichiometric matrix may be performed such that the first N_0 rows become linearly independent. Thus, one obtains $\mathbf{S} = \mathbf{L}\mathbf{S}_r$,where $\mathbf{L}$ is an $N \times N_0$ matrix such that $\mathbf{L} = \begin{bmatrix} \mathbf{I}_{N_0} \\ \mathbf{L}_0 \end{bmatrix}$ and $\mathbf{I}_{N_0}$ is the $N_0 \times N_0$ identity matrix. The reduced stoichiometric matrix $\mathbf{S}_r$ is obtained from $\mathbf{S}$ by deleting the last $(N - N_0)$ linearly dependent rows. With the same reordering of the entries in the state vector $\mathbf{x}$ we obtain $\mathbf{x} = [\mathbf{x}_I, \mathbf{x}_D]'$, where $\mathbf{x}_I$ is the N_0-vector containing the linearly independent species and $\mathbf{x}_D$ is the $(N - N_0)$-vector storing the linearly dependent species. One obtains from (1) that

$$\frac{d\mathbf{x}_I}{dt} = \mathbf{S}_r \mathbf{v}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) \ , \quad \frac{d\mathbf{x}_D}{dt} = \mathbf{L}_0 \frac{d\mathbf{x}_I}{dt} \ .$$

Therefore $\mathbf{x}_D(t) = \mathbf{L}_0 \mathbf{x}_I(t) + \mathbf{T}$ for all $t \geq 0$, thus

$$\mathbf{T} = \mathbf{x}_D(0) - \mathbf{L}_0 \mathbf{x}_I(0) \ .$$

We remark that $\partial \mathbf{T}/\partial \mathbf{p} = 0$ since the initial conditions do not depend on the kinetic constants, stored in the vector $\mathbf{p}$.

We only need to study the reduced reaction rate system containing the linearly independent equations

$$\frac{d\mathbf{x}_I}{dt} = \mathbf{S}_r \mathbf{v}((\mathbf{x}_I(t, \mathbf{p}), \mathbf{L}_0 \mathbf{x}_I(t, \mathbf{p}) + \mathbf{T}), \mathbf{p}) \ . \tag{3}$$

Differentiating (3) with respect to the kinetic parameters $\mathbf{p}$ and noticing that $\partial \mathbf{T}/\partial \mathbf{p} = 0$, we derive

$$\begin{aligned} \frac{d}{dt}\frac{\partial \mathbf{x}_I}{\partial \mathbf{p}} &= \mathbf{S}_r \left[\frac{\partial \mathbf{v}(t)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} + \frac{\partial \mathbf{v}(t)}{\partial \mathbf{p}} \right] \\ &= \mathbf{S}_r \left[\frac{\partial \mathbf{v}(t)}{\partial \mathbf{x}} \mathbf{L} \frac{\partial \mathbf{x}_I(t)}{\partial \mathbf{p}} + \frac{\partial \mathbf{v}(t)}{\partial \mathbf{p}} \right] \end{aligned} \tag{4}$$

The initial conditions for the local sensitivities are

$$\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(0) = 0 \ .$$

We shall calculate the local sensitivities by solving the ordinary differential equations (3) together with the system for sensitivities (4) to find $\mathbf{x}(.)$ and $\partial \mathbf{x}(.)/\partial \mathbf{p}$. We obtained thus a forward sensitivity analysis for the *reduced system*. More details on the sensitivity analysis applied to the deterministic models of chemical kinetics can be found in [8].

While parametric sensitivity of ordinary differential equation models is a well-established research area, the parametric sensitivity of stochastic models has been much less investigated. We propose below an efficient method for sensitivity analysis for a class of stochastic models of biochemical kinetics.

Algorithm:

1). Construct the reaction rate equations for the system of reactions.

2). Perform a sensitivity analysis, as described above, on the reaction rate equations for the species of interest with respect to the reaction rate parameters.

3). Use the sensitivity analysis performed on the reaction rate equations as an approximation of the sensitivity analysis of the stochastic system described by the Chemical Master Equation.

This heuristic algorithm is based on the observation that the investigation of the deterministic reaction rate models for many biochemically reacting systems provides important insight in studying their more general stochastic models, given by the Chemical Master Equation. Our proposed method applies to this class of biochemical systems. As mentioned before, the intrinsic noise arises in well-stirred (bio)chemically reacting systems due to low population numbers of some molecular species. The intrinsic noise may be enhanced if some reactant species are present in very small molecular populations. By contrast, if all species have large population numbers (at least in the thousands) for the time interval under consideration, then the intrinsic noise is small. In such cases, we expect the periodic dynamics predicted by the stochastic model to be well approximated by the oscillatory dynamics obtained for the deterministic model. This happens, for example, for systems which are robust with respect to molecular noise [14]. Indeed, for the circadian rhythm model considered in [14] it was noticed that the intrinsic noise affected only the maxima of the oscillations, not their period, for a range of realistic values of molecular population numbers.

Remark 1: A quantitative analysis of the stochastic system is computationally very expensive in most realistic applications, compared to its deterministic approximation. In practice, this procedure for sensitivity analysis based on the deterministic part only, can be much more efficient than a full analysis of the stochastic system.

Remark 2: Typically, the biochemical systems arising in applications are quite large, with many reacting species and many reaction channels. Then, identifying and analyzing the mechanism which controls the oscillations, is a challenging task. Often, it can not be done based on intuition and thus a quantitative approach is required to deal with the large amount of data.

Sensitivity analysis is an important technique for studying this problem. First, the sensitivity of the key species involved

in the oscillatory behavior with respect to the problem parameters will be computed. If these key reactant species are robust (less sensitive) with respect to certain reaction rate parameters, then the reactions corresponding to those parameters will not have a strong impact on the oscillations and may be eliminated. Based on this, a subsystem can be identified, which is responsible with controlling the oscillations. Such a simplified system will be easier to analyze and simulate. For example, it may be simpler to find the regions in the parameter space for which the system maintains its oscillatory dynamics.

4. Numerical results

Lotka-Volterra model: The predator-prey model, due to Lotka and Volterra, involves two interacting species [25]. The biological population Y_1 represents the prey, while the population Y_2 is the predator. However, these populations can be interpreted as species of a reacting biochemical system. The reaction system and its kinetic formulation are described in Table 1. The reactions correspond to the prey reproduction, the predator-prey interaction and the predator death, respectively. The initial conditions are $Y_1(0) = Y_2(0) = 100$. We study the behavior of this system for the time interval $[0, 50]$. The numerical results show a good agreement of the period as predicted by the local sensitivities and the period of oscillations for both the deterministic solution and the solution of the Chemical Master Equation.

Brusselator model: The model of chemically reacting systems in Table 2 has the property that, for any initial molecular populations of the reacting species Y_1 and Y_2, the system will start oscillating after a short time. This system is called the Brusselator, and it is an example of a limit cycle chemical oscillator. The species $A_1 = 1$ and $A_2 = 1$ are kept at a constant value. The initial conditions are $Y_1(0) = 1000$ and $Y_2(0) = 2000$. The system is integrated on the interval $[0, 10]$. This system is stiff. A trajectory obtained with Gillespie's algorithm requires the simulation of order $\mathcal{O}(10^6)$ reactions, therefore it is computationally quite expensive. Similarly to the previous example, the parametric sensitivities match the period of the solution for the reaction rate model as well as of the one for the stochastic model. In addition, by applying the method described in this paper, we approximated these parametric sensitivities quite efficiently.

5. Conclusion and future work

We presented in this paper a new sensitivity analysis technique for stochastic models of biochemical kinetics. This method extends a parametric sensitivity technique designed for (deterministic) reaction rate equation models to the more general stochastic discrete models of oscillating biochemical systems. The numerical results on two mathematical models arising in applications show that our method performed well on the finite time-interval of interest. Sensitivity analysis may give important biological insight by identifying the key components of the system which lead to a certain behavior, such as sustained oscillations. We plan to work on designing efficient and accurate methods for finding the sensitivity with respect to the period of oscillatory systems.

References

[1] N. Bagheri, J. Stelling, F.J. Doyle, Quantitative performance metrics for robustness in circadian rhythms, *Bioinformatics* **23**(3), 358 – 364, 2007.

[2] M.B. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators, *Nature* **403**, 335–338, 2000.

[3] M.B. Elowitz, A.J. Levine, E.D. Siggia, P.S. Swain, Stochastic gene expression in a single cell, *Science* **297**, 1183–1186, 2002.

[4] D.B. Forger, C.S. Peskin, A detailed predictive model of the mammalian circadian clock, *Proc. Natl. Acad. Sci. USA* **100**, 14806 – 14811, 2003.

[5] D.B. Forger, C.S. Peskin, Stochastic simulation of the mammalian circadian clock, *Proc. Natl. Acad. Sci. USA* **102**, 321 – 324, 2005.

[6] J. Hasty, D. McMillen, F. Isaacs, J.J. Collins, Computational studies of gene regulatory networks: in numero molecular biology, *Nature Reviews Genetics* **2**, 268–279, 2001.

[7] S. Ilie, W.H. Enright, K.R. Jackson, Numerical solution of stochastic models of biochemical kinetics, *Canadian Applied Mathematics Quarterly*, **17**(3), 523 – 554, 2009.

[8] B.P. Ingalls, H.M. Sauro, Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories, *J. Theor. Biol.*, **222**, 23–36, 2003.

[9] C.W. Gardiner, *Stochastic methods: a handbook for the natural and social sciences*, Springer, Berlin, 2009.

[10] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, *J. Comp. Phys.* **22**, 403–434, 1976.

[11] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.* **81**, 2340–2361, 1977.

[12] D.T. Gillespie, A rigorous derivation of the chemical master equation, *Physica A*, **188**, 402–425, 1992.

[13] A. Goldbeter, Computational approaches to cellular rhythms, *Nature* **420**, 238 – 245, 2002.

[14] D. Gonze, J. Halloy, A. Goldbeter, Emergence of coherent oscillations in stochastic models of circadian rhythms, *Physica A* **342**, 221 – 233, 2004.

[15] B.C. Goodwin, Oscillatory behavior in enzymatic control processes, *Adv. Enzyme Regul.* **3**, 425 – 438, 1965.

[16] H. de Jong, Modeling and simulation of genetic regulatory systems, *J. Comput. Biol.* **9**(1), 67–103, 2002.

[17] J.C. Leloup, A. Goldbeter, Toward a detailed computational model for the mammalian circadian clock, *Proc. Natl. Acad. Sci. USA* **100**, 7051 – 7056, 2000.

[18] H. McAdams, A. Arkin, Stochastic mechanism in gene expression, *PNAS. USA* **94**, 814–819, 1997.

[19] I. Prigogine, R. Lefever, Symmetry breaking instabilities in dissipative systems, *J. Chem. Phys.*, **48**, 1968.

[20] J.M. Raser, E.K. O'Shea, Control of stochasticity in eukaryotic gene expression, *Science*, **304**, 1811–1814, 2004.

[21] P. Ruoff, M.K. Christensen, V.K. Sharma, PER/TIM-mediated amplification, gene dosage effects and temperature compensation in an interlocking feedback loop model of the *Drosophila* circadian clock, *J. Theor. Biol.* **237**, 41 – 57, 2005.

[22] M. Samoilov, S. Plyasunov, A.P. Arkin, Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations, *PNAS. USA* **102**(7), 2310–2315, 2004.

[23] A. Varma, M. Morbidelli, H. Wu, *Parametric sensitivity in chemical systems*, Cambridge University Press, 1999.

[24] T.E. Turner, S. Schnell, K. Burrage, Stochastic approaches for modelling in vivo reactions, *Comput. Biol. Chem.*, **28**, 165–178, 2004.

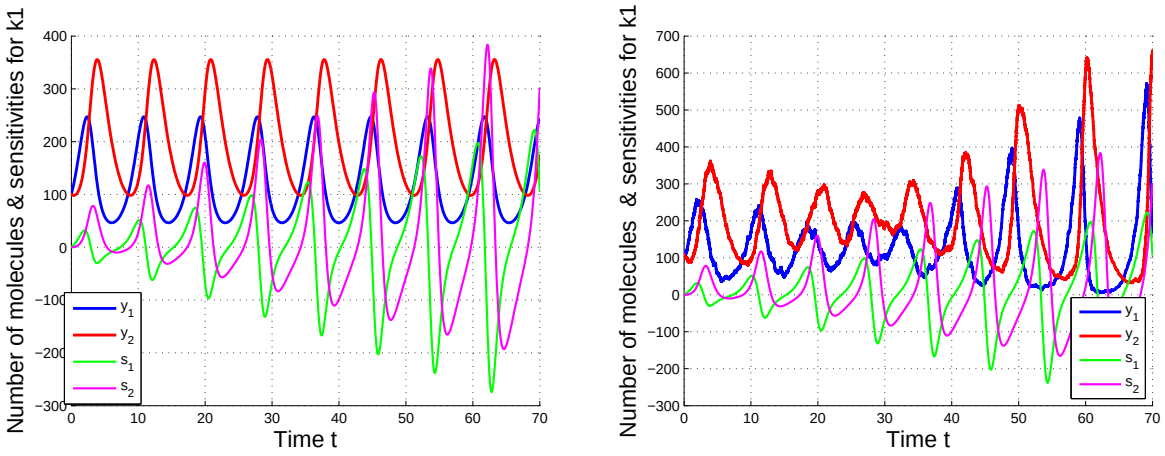


Fig. 1: Sensitivity analysis with respect to the parameter k_1 for the Lotka-Volterra model: the deterministic simulation (left) and the stochastic simulation (right). The sensitivities are scaled by 10^{-1}.

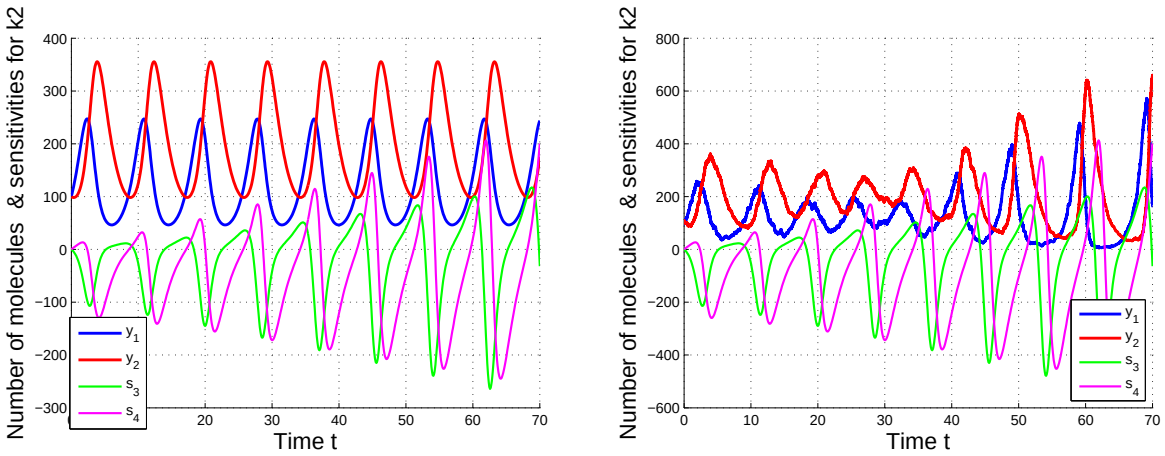


Fig. 2: Sensitivity analysis with respect to the parameter k_2 for the Lotka-Volterra model: the deterministic simulation (left) and the stochastic simulation (right). The sensitivities are scaled by 2×10^{-3}.

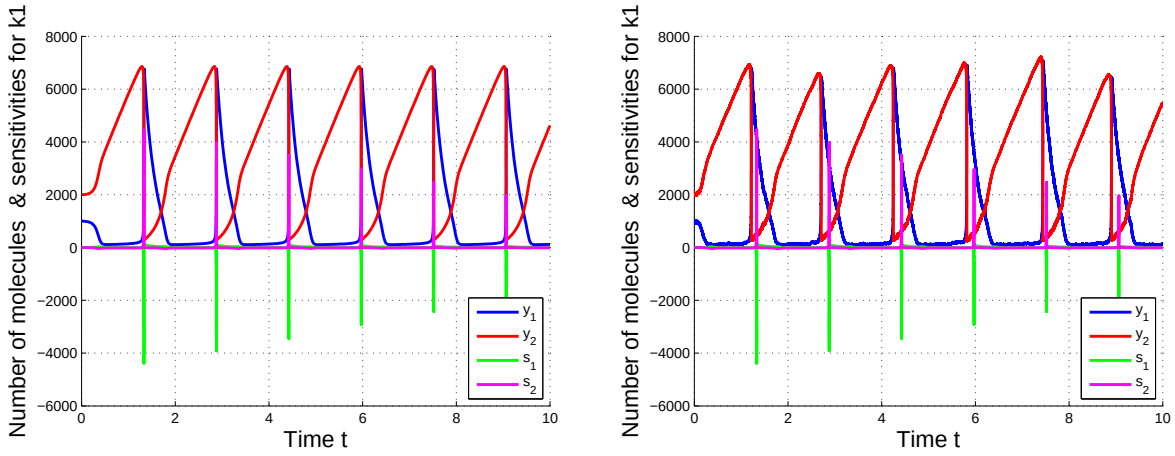


Fig. 3: Sensitivity analysis with respect to the parameter k_1 for the Busselator model: the deterministic simulation (left) and the stochastic simulation (right).

Table 1: The Lotka-Volterra model.

	Reaction channel	Reaction propensity	Reaction rate
$R1$	$Y_1 \xrightarrow{k_1} 2Y_1$	$a_1 = k_1 Y_1$	$k_1 = 1$
$R2$	$Y_1 + Y_2 \xrightarrow{k_2} 2Y_2$	$a_2 = k_2 Y_1 Y_2$	$k_2 = 0.005$
$R3$	$Y_2 \xrightarrow{c_3} \emptyset$	$a_3 = k_3 Y_2$	$k_3 = 0.6$

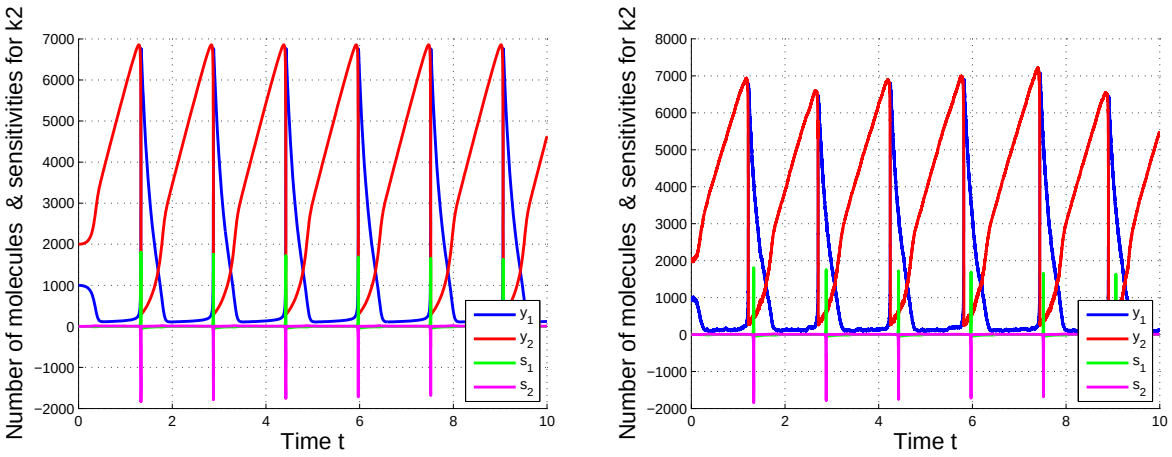


Fig. 4: Sensitivity analysis with respect to the parameter k_2 for the Busselator model: the deterministic simulation (left) and the stochastic simulation (right). The sensitivities are scaled by 10^{-3}.

Table 2: The Brusselator model.

	Reaction channel	Reaction propensity	Reaction rate
R_1	$A_1 \xrightarrow{k_1} Y_1$	$k_1 A_1$	$k_1 = 5000$
R_2	$A_2 + Y_1 \xrightarrow{k_2} Y_2 + B_1$	$k_2 A_2 Y_1$	$k_2 = 50$
R_3	$2Y_1 + Y_2 \xrightarrow{k_3} 3Y_1$	$k_3 Y_1 (Y_1 - 1) Y_2 / 2$	$k_3 = 5 \cdot 10^{-5}$
R_4	$Y_1 \xrightarrow{k_4} B_2$	$k_4 Y_1$	$k_4 = 5$

[25] D.J. Wilkinson, *Stochastic modelling for systems biology*, Chapman & Hall/CRC, 2006.

A Simple Relaxation based Circuit Simulator for VLSI Circuits with Emerging Devices

Balwinder Kumar, Yogesh Dilip Save, H. Narayanan, and Sachin B. Patkar
Electrical Engineering Department, Indian Institute of Technology - Bombay, Mumbai 400076.

Abstract— *This paper presents a circuit simulator based on look-up table approach for simulation of VLSI digital circuits with emerging devices which currently cannot be simulated with existing commercial simulators. We have used a point relaxation based circuit simulator which is suitable for digital circuits. To validate our circuit simulator, we have simulated standard circuits with MOS devices of SPICE level 49 for 0.18 micron technology and results are in very good agreement with a commercial simulator. Further, we have successfully simulated standard circuits with an emerging device, FinFET inverter, chain of FinFET inverters, 31-stage ring oscillator of FinFET inverters and read-write operations for FinFET 6-T SRAM cell. Our results match with that of the device simulator. Finally, we have been able to simulate digital circuits with approximately* 1.6 *million FinFETs (*512×512 *SRAM memory array) using less than* 1.7*GB working memory (Pentium-4, 3.0GHz).*

Keywords: Look-up Table, Spline interpolation, FinFET, Relaxation, Digital Circuit, Circuit Simulator

1. Introduction

During the last two decades, the number of transistors that can be placed on a chip has been approximately doubling every year. In the case of digital circuit, especially memory, this relationship holds strongly. With increased size of VLSI circuits, the simulation of large circuits becomes an important issue in circuit design. Conventional circuit simulators are impractical for such large circuits. Also with development in technology, new devices are emerging frequently. In order to evaluate the performance of circuits with these emerging devices, an accurate device model is required. The analytical model development of emerging devices is a difficult task as the model becomes increasingly complex with decrease in size of the device. An alternative to the analytical model based methods is a look-up table (LUT) method, which solves the above mentioned problems. The LUT method involves generation of the device characteristics tables either from direct measurements or through device simulators and uses interpolation techniques to obtain values at the intermediate points. The data tables and the interpolation method determines the accuracy of the LUT approach. With the availability of good process and device simulators, it is possible to generate the device characteristics curves even before fabricating the actual device. The LUT approach does not require analytical models. Further, unlike analytical models, improvement in data can be easily accommodated. The accuracy of the LUT based simulator depends on the number of points taken on the characteristics. Higher accuracy can be achieved with larger number of data points. But this requires larger memory for storage. Circuit simulators based on the direct method for solving a linear system of equations are not suitable with LUT approach as they require substantially more memory for storing the system coefficient matrix. Those based on iterative methods are most suitable for the LUT based simulators due to their reduced memory requirements. The point relaxation based simulators have already shown usefulness in handling large digital circuits [1]. Simulators based on this method require memory essentially to store the data structure. There is no need to build an explicit coefficient matrix for the equations. Thus a circuit simulator which uses a combination of LUT and point relaxation method is useful in handling large digital circuits.

The use of LUT was first proposed in [2] to simulate digital circuits. In the last three decades, many improvement schemes for interpolation optimization [3], reducing memory storage requirement [4], effective construction of derivative information [5] etc. have been proposed. In [6], a new LUT method is presented for simulation of FinFET circuits but that circuit has not more than a few hundreds of transistors. The main aim of our work is to show use of relaxation method to solve large digital circuits with millions of emerging devices.

The outline of the present paper is as follows. We discuss the look-up table approach for emerging devices in section 2. In section 3, cubic spline interpolation method is described which is used to interpolate the current variables and the capacitances. The overview of relaxation based circuit simulator is given in section 4. In section 5, we validate our circuit simulator through simulation of standard circuits (CMOS inverter, buffer chain of inverters and 31-stage ring oscillator) and compare it with commercial simulator HSPICE for MOS devices of SPICE level 49 for 0.18 micron technology. Section 6 is devoted to circuits with emerging devices. Standard circuits, FinFET inverter, chain of FinFET inverters, 31-stage ring oscillator of FinFET inverters and read-write operations for six-transistor FinFET SRAM cell are simulated using our simulator and results are verified using the device simulator. The usefulness of our simulator

for large size digital circuit is shown through simulation of FinFET based SRAM circuits containing millions of transistors.

2. Look-Up Table Approach for emerging devices

In order to simulate a circuit, the information about input-output characteristics of devices is required. The current through device is a function of terminal voltages and the geometry of the device e.g., in MOS transistors, drain current is a function of gate source voltage V_{GS}, drain source voltage V_{DS} and bulk source voltage V_{BS} and the dimensions width and length of the device. If we consider all the n parameters and generate n-dimensional look-up tables, this leads to large storage and evaluation time. Instead of this one can use the knowledge of the device physics and technology to reduce dimension of tables [7] as smaller dimensional tables require moderate storage and less evaluation time. To describe the transient and AC analysis accurately, in addition to static current of the device, the relation with ac current is also required. In VLSI circuits the ac current is mainly because of device capacitance. If all the capacitances are considered then the size of the table becomes large. Instead of that one can consider the capacitances that significantly contribute. There are different techniques available to extract the capacitances of emerging devices. One can employ transient analysis or AC analysis based methods to extract the capacitances using device simulator. Many device simulators provide the capacitance data tables directly. Once the look-up tables are available, an interpolation scheme can be used to obtain intermediate values in the data tables.

3. Spline Interpolation

There are many different interpolation methods in numerical analysis that can be used in the LUT approach. The choice of the interpolation method depends upon the requirements like accuracy and smoothness. The spline interpolation method is the popular choice. This method employs polynomial functions with certain degree of smoothness to interpolate the data. For MOS circuit simulations, first derivative continuity is required [8]. Hence the cubic spline method has been used for the interpolation of data as it uses third degree polynomials and ensures the continuity upto the second derivative.

Consider, a table consisting of values of independent variable x and corresponding values of dependent variable y. Let S_i denote the cubic polynomial that will be used on the sub interval $[t_i, t_{i+1}]$. In cubic spline, polynomial values at knots and, first and second derivatives of polynomial at interior of knots should be continuous. These conditions lead to $2(n-1)+2(n-2) = 4n-6$ equations, but we have $4(n-1)$ coefficients. Additional 2 equations are obtained from boundary conditions. Based on the choice of these conditions, we get different types of cubic splines. We have used natural cubic spline [9] where end points conditions are

$$S''(t_1) = S''(t_n) = 0 \quad (1)$$

The cubic polynomial for $x \in [t_i, t_{i+1}]$ is given by

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^2 + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1}\right)(x-t_i) + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i\right)(t_{i+1}-x) \quad (2)$$

where

$$z_i = S''(t_i)$$

The coefficients of various polynomial segments are stored to use for computation.

4. Circuit Simulator BREMICS

We have used circuit simulator BREMICS [1] which is based on Point Relaxation method. In this method, the given circuit is broken into smaller sub-circuits, in case of BREMICS each sub-circuit is actually a single node. So a circuit of n nodes is essentially treated as n sub-circuits, each sub-circuit containing the concerned node and the adjacent nodes. In point relaxation method, while solving for a node potential, it is assumed that potential of all other nodes are known. This process is iterated over all the nodes. Gauss-Seidel iterations for all the nodes continues until convergence is reached. At each node, Newton-Raphson technique is used to linearize the nonlinear element and then it is solved for node potential. Convergence in BREMICS is dependent on unidirectionality of signal flow and weak coupling between sub circuits. If coupling between the sub-circuits is significant and signal flow is bidirectional then the rate of convergence is slow. MOS device is a unidirectional device and MOS digital circuits usually have minimal feedback in practice. So, this relaxation technique is useful in simulating MOS integrated circuits [10].

5. Validation of approach

To validate correctness of our simulator, we have simulated standard circuits with MOS devices and compared results with a commercial simulator. We have used Spice model LEVEL 49, VERSION = 3.1 for 0.18 micron technology for MOS devices.

5.1 $I_D - V_{DS}$ curve interpolation

Initially we have the interpolated values of $I_D - V_{DS}$ with spice generated values. As the second derivative is nearly zero close to start and end point of $I_D - V_{DS}$ characteristics, natural cubic spline is used as interpolation technique. Figure 1 shows the interpolated $I_D - V_{DS}$ for $V_{GS} = 1.2V$.

There is always a trade-off between memory usage and accuracy. Figure 2 shows the relationship between accuracy and the number of data points. The error is exponentially

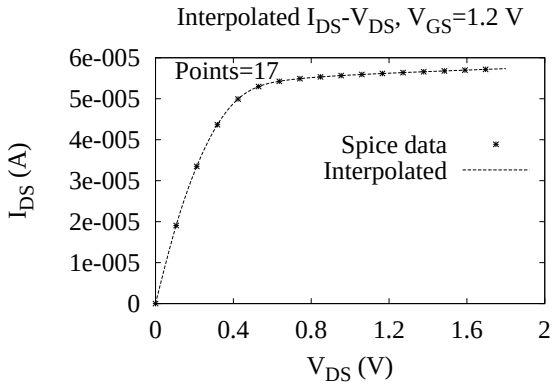


Fig. 1: $I_D(V_{DS})$ interpolated curve at V_{GS} = 1.2 V

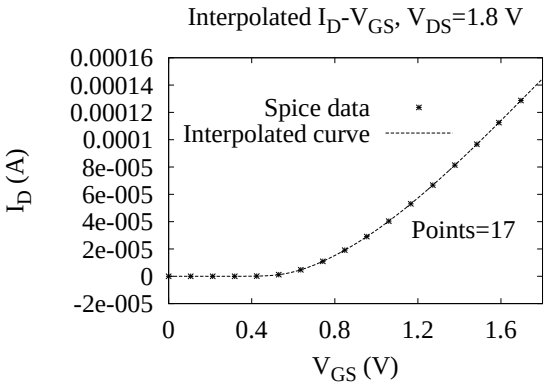


Fig. 3: $I_D(V_{GS})$ interpolated curve at V_{DS} = 1.8 V

decreasing with the number of data points. For 3% accuracy we need with only 18 data points but to achieve 1% accuracy more than 50 data points are required. For optimal accuracy and memory usage we have used 18 data points.

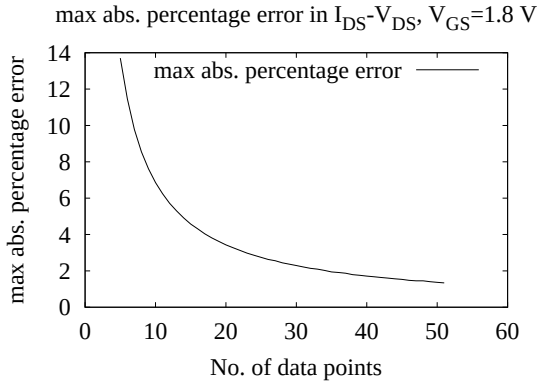


Fig. 2: Maximum error Vs No. of data points

5.2 $I_D - V_{GS}$ curve interpolation

For $I_D - V_{GS}$ curve, simple polynomial interpolation is good only for non sub-threshold regions. In sub-threshold regions a large number of data points are required to get accurate results. Thus we have used combined interpolation technique which uses exponential interpolation for sub-threshold region, polynomial interpolation for strongly inverted region and combined interpolation in transition region for $I_D - V_{GS}$ curves interpolation. Figure 3 shows $I_D - V_{GS}$ interpolation curves at $V_{DS} = 1.8V$.

5.3 Capacitance interpolation

Figure 4 shows the drain capacitance curve for $V_{DS} = 0.2V$. We have used Sentaurus device simulator to generate data points. The interpolated values match closely with that of TCAD generated. It can be observed that interpolation is not very accurate around the peak occurring in capacitance curves. This problem can be resolved by choosing a data point at peak and two close points on both side on the peak point.

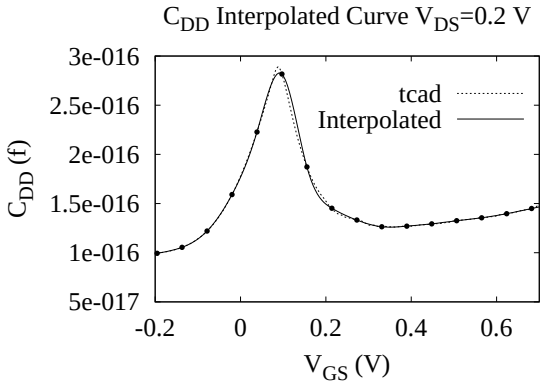


Fig. 4: $C_{DD}(V_{GS})$ interpolated curve

5.4 Inverter DC transfer characteristics

We have simulated CMOS inverter using look-up table in BREMICS and compared our results with HSPICE. Figure 5 shows transfer curve of inverter. Table 1 shows the comparison of switching voltage of inverter as computed by BREMICS and HSPICE for different device dimensions. We have found excellent agreement with HSPICE results. Our simulated curves match with SPICE very closely and shows the accuracy of the interpolation routines used.

5.5 Buffer chain of inverters

Buffer chain of CMOS inverters has been simulated using BREMICS and results are shown in Figure 6. We have simulated buffer chains of different number of inverter stages. The results are compared with that of HSPICE. In all cases the accuracy is within 2.4%.

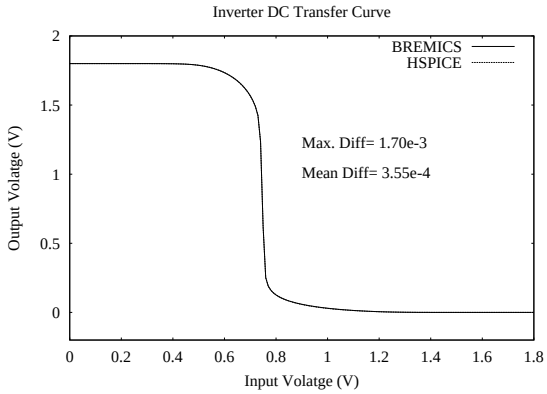


Fig. 5: Inverter DC transfer curve

Table 1: Switching voltage for different device dimensions

Dimensions $W_P = 3.0 * W_N$	Switching voltage		error (%)
	HSPICE	BREMICS	
L = 0.5 μ, W_N = 1.0 μ	0.8808 V	0.8808 V	0
L = 0.5 μ, W_N = 5.0 μ	0.8844 V	0.8843 V	0.01
L = 1.0 μ, W_N = 4.0 μ	0.8663 V	0.8663 V	0
L = 5.0 μ, W_N = 10 μ	0.8521 V	0.8521 V	0
L = 5.0 μ, W_N = 20 μ	0.8521 V	0.8521 V	0

5.6 Ring oscillator simulation

We have simulated a ring oscillator circuit with BREMICS and HSPICE and results are shown in Figure 7. Using LUT routine, we have simulated the circuit with different number of stages and, for different device dimensions for 31-stage ring oscillator. The results are shown in table 3. In all cases difference is less than 2 $\%$.

6. Results and Discussions

As an application of the use of look-up table method, we have simulated circuits having FinFET devices. FinFETs are emerging devices whose accurate analytical models are not presently available. This makes SPICE like simulators incapable of simulating circuits involving such devices as they employ device analytical models. However, with integration of LUT routines into BREMICS, it is capable of simulating such circuits. To simulate a circuit having FinFETs, BREMICS requires to have look-up table for its drain current I_D and various capacitances such as gate source capacitance C_{gs}, gate drain capacitance C_{gd}, drain source capacitance C_{ds}. Device simulator SENTARUS has been used to generate look-up table for drain current I_D and various capacitances of FinFETs. The section discusses the results obtained in simulation of FinFET inverter and buffer chain of FinFET inverters, 31-stage ring oscillator of FinFET inverters and read-write operations for FinFET 6-T SRAM cell.

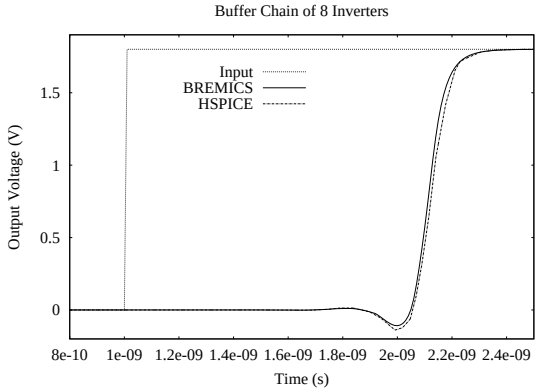


Fig. 6: Buffer chain of 8 MOSFET inverters

Table 2: Buffer chain with different number of stages

Stages	Delay Time		error(%)
	HSPICE	BREMICS	
8	1.138 ns	1.130 ns	0.70
12	1.741 ns	1.755 ns	0.80
16	2.341 ns	2.375 ns	1.45
20	2.940 ns	3.000 ns	2.04
24	3.540 ns	3.625 ns	2.40

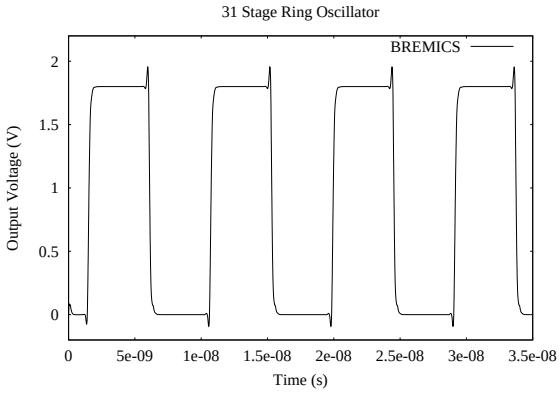


Fig. 7: 31 stage ring oscillator of MOSFET inverter

Table 3: Ring oscillator with different number of stages

No. of Stages	Oscillation Frequency		Error(%)
	HSPICE	BREMICS	
11	300 MHz	306 MHz	2.0
15	221 MHz	223 MHz	0.8
19	175 MHz	178 MHz	1.9
23	144 MHz	146 MHz	1.3
27	122 MHz	124 MHz	1.4
31	108 MHz	109 MHz	1.2

6.1 FinFET inverter

We have simulated FinFET inverter by using look-up tables for its drain current I_D in BREMICS. Figure 8 shows the DC transfer curve of FinFET inverter. We have got excellent agreement with device simulator results with difference less than 10^{-4} for all data points.

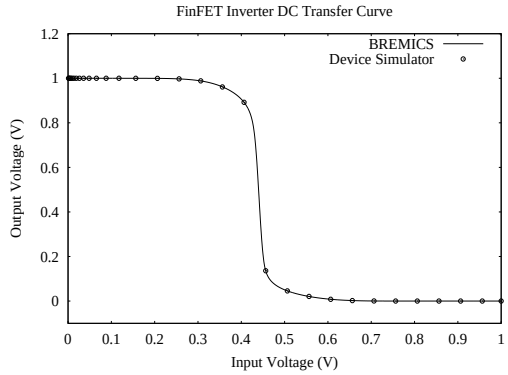


Fig. 8: FinFET inverter DC transfer curve

6.2 Chain of FinFET inverters

We have simulated chain of 3 FinFET inverter by using look-up tables for its drain current I_D and capacitances in BREMICS. Delay for chain of 3 FinFETs comes out to be 4.5 ps. We have got good agreement with device simulator results. The first stage output of the circuit is shown in Figure 9.

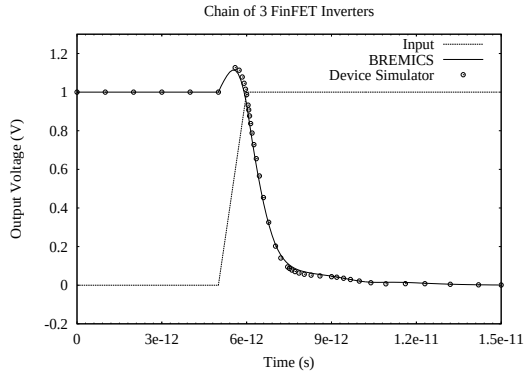


Fig. 9: First stage output of chain of 3 FinFET inverters

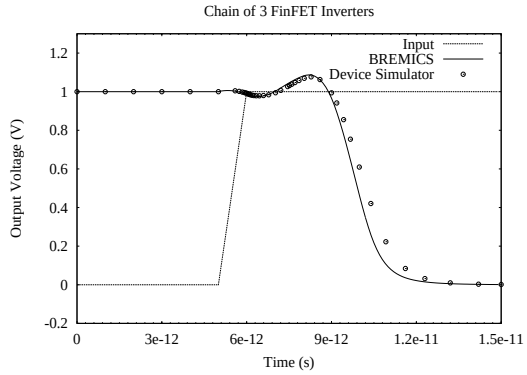


Fig. 10: Third stage output of chain of 3 FinFET Inverters

Figure 10 shows that there is some mismatch between the LUT enabled BREMICS output and device simulator data. The possible reason could be the very high frequency of operation and breakdown of quasi-static model. We have used capacitance data generated at frequency 1 MHz, but the operational frequency is order of few GHz.

6.3 FinFET ring oscillator

We have simulated 31-stage ring oscillator with FinFET inverters and result is shown in Figure 11. Due to convergence problem, device simulator could not simulate the circuit. As a result we can not validate the accuracy of our simulation directly. However, we have compared single stage delay calculated from the frequency of the ring oscillator with the delay obtained from simulation of a chain of FinFET inverter using device simulator. The oscillation frequency comes out to be 8.85 GHZ. It gives single stage delay of 1.8 ps which matches with the result obtained from device simulator. This proves the ability of LUT enabled BREMICS to simulate circuits involving emerging devices like FinFETs correctly.

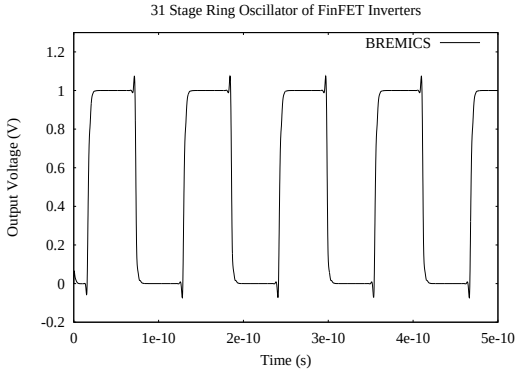


Fig. 11: 31 stage ring oscillator of FinFET inverters

6.4 6-T SRAM with FinFET

We have simulated the six-transistor FinFET SRAM cell. The bit line capacitance values is taken as 10 fF. In case of read operations, both the bit lines were pre-charged to 1.0 V. When the word line goes high, one of the bit lines is discharged, depending upon whether 0 or 1 is stored in cell. Figure 12 shows the read operation, when 1 is stored in cell. In case of write operations, if we want to write 0 into cell, we charge bit line to 0 and bit line bar to 1 and vice-versa. When the word line goes high, desired value is written into cell. Figure 13 shows the write operation for changing the cell value from 1 to 0.

Table 4 shows a performance Bremics for read operation in terms of memory and simulation time. From the table it is clear that the timing performance and memory usage of Bremics is linear. We have simulated SRAM memory array

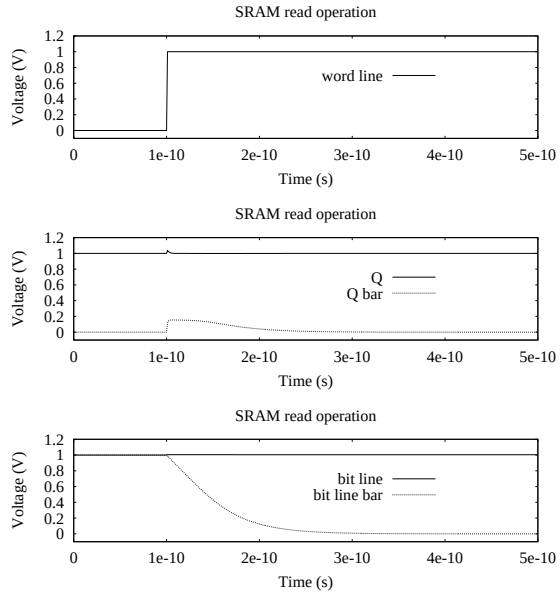


Fig. 12: FinFET SRAM read operation

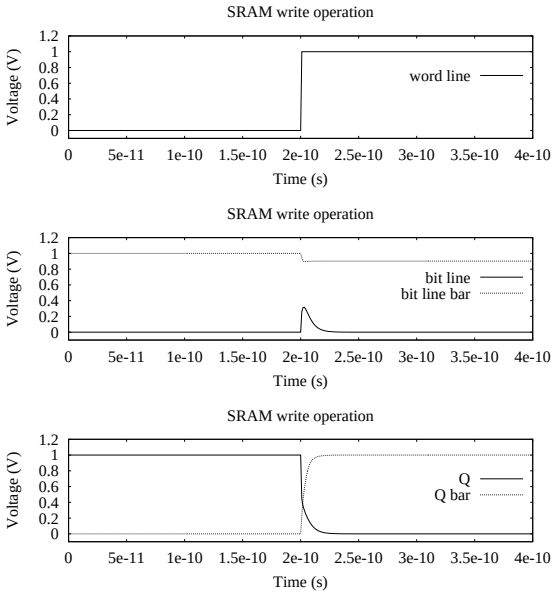


Fig. 13: FinFET SRAM write operation

Table 4: Memory and Timing Performance of BREMICS for SRAM (memory read operation)

SRAM size	Nodes	BREMICS	
		Memory Usage(MB)	Time(sec)
8x8	154	1.0	49
16x16	550	2.5	162
32x32	2030	7.2	582
64x64	7824	26.0	1870
128x128	30754	102	7174
256x256	120924	422	27032
512x512	480652	1589	107458

of size 512×512 (1.6 million transistors) using only 1.7GB working memory even though LUT requires large memory to store the characteristics. This is possible because of the use of a point relaxation method for a solution of a system of equations as there is no need to store an explicit coefficient matrix for the equations.

7. Conclusion

The primary aim of the present work is to demonstrate that using an elementary relaxation technique and the look-up method a large circuit (about 1 million FinFETs) with emerging devices can be simulated in reasonable time. We have presented results to support this claim by analyzing an SRAM memory array of size 512×512 (1.6 million transistors) in 30 hours using only 1.7GB working memory. This shows that the capability of integration of a point relaxation method with LUT to handle a large size digital circuits with emerging devices. Also, as point relaxation method is inherently parallelizable, one can easily extend the use this simulator to simulate entire chip using parallelization techniques. The look-up tables used in this work are based on data provided by device simulators. One can use actual experimental data also to build look-up tables without any change in the approach.

References

[1] S. Roy, Y. Save, H. Narayanan, and S. B. Patkar, "Large scale VLSI circuit simulation using point relaxation," in *International Conference on Scientific Computing, CSC 2010*, 2010, pp. 343–347.

[2] B. Chawla, H. Gummel, and P. Kozak, "Motis-an MOS timing simulator," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 12, pp. 901–910, Dec 1975.

[3] J. Barby, J. Vlach, and K. Singhal, "Polynomial splines for MOSFET model approximation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 5, pp. 557–566, May 1988.

[4] P. Meijer, "Fast and smooth highly nonlinear multidimensional table models for device modeling," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 335 –346, Mar. 1990.

[5] A. Rofougaran and A. Abidi, "A table lookup FET model for accurate analog circuit simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 2, pp. 324 –335, Feb. 1993.

[6] R. A. Thakker, C. Sathe, A. B. Sachid, M. S. Baghini, V. R. Rao, and M. B. Patil, "A novel table-based approach for design of FinFET circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 7, pp. 1061–1070, 2009.

[7] P. Subramaniam, "Modeling MOS VLSI circuits for transient analysis," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 2, pp. 276–285, Apr 1986.

[8] J. Coughran, W.M., E. Grosse, and D. Rose, "Cazm: A circuit analyzer with macromodeling," *IEEE Transactions on Electron Devices*, vol. 30, no. 9, pp. 1207 – 1213, Sep 1983.

[9] W. Cheney and D. Kincaid, "Numerical mathematics and computing," *Brooks Cole Publishing Company*, pp. 271–276, 1985.

[10] W. Mokari and D. Smart, "Robust VLSI circuit simulation techniques based on overlapped waveform relaxation,," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 14, no. 4, pp. 510–518, 1995.

Simulation of Darcian evaporation through a heterogeneous soil layer

A.R.Kacimov
Department of Soils, Water and Agricultural Engineering, Sultan Qaboos University, Muscat, Oman

Abstract - *Steady, Darcian, 1-D, 1-phase evaporation through a vertically inhomogeneous soil from a horizontal water table is simulated as an optimal control problem, i.e. the hydraulic properties of the porous medium (relative permeability of moisture and capillary water retention curve) are considered as control functions. The Richards nonlinear ordinary differential equation is integrated twice as a boundary value problem on a given interval (vadose zone thickness) and the top soil suction head is taken as a cost function in optimization. The Gardner conductivity-pressure relation and the Ahuja et al. (1989) correlation between saturated conductivity and sorptive number are assumed. A continuous vertical change in permeability is optimized.*
Key words: *nonlinear optimization, boundary value problem for nonlinear ODE, Richards equation, control of heterogeneity.*

1 Introduction

Evaporation from a shallow phreatic surface of artificial recharge (AR) schemes in arid regions is an important factor in secondary salinization of soil and gross-losses of water. Common natural soils superjacent to a water table and capillary fringe are heterogeneous in their hydraulic properties (soil-water retention, SWR, and unsaturated hydraulic conductivity, UHC), which control evaporation from the topsoil (see, e.g., [1]). Collection of soil samples from boreholes or pedons is a tedious, costly and plant-invasive procedure. Core samples taken from the field for further laboratory experiments require a pressure-plate apparatus (or other device) to coin a SWR by relating the capillary pressure head, H, and the moisture content, Θ. UHC, K, a function of Θ is very seldom obtained in laboratory because the involved experiments are expensive and lengthy. Consequently, simulation is often the only feasible method to predict evaporation. A crucial step in conceptual modeling of moisture dynamics in soils is the Bourdine or Mualem type conversion of SWR to UHC by simple integration [2,3] he real (experimental) or surrogate (obtained by the mentioned integration or from pedotransfer functions) pair (SWR,UHC) is needed for each layer of the soil horizon.

In this paper we study a steady evaporation (exfiltration) from a stationary fully saturated plane to a relatively dry soil surface – a common situation in arid climates. For example, in coastal and internal areas [4], which are now identified as potential AR sites, a shallow (1-5 meters deep) water table intensively evaporates to a hot and dry playa surface and the catchment-scale hydrological balance calls for the corresponding vertical flux.

The mathematical models, on which we base our analysis, are presented in [5-6], where the soil properties were fixed. We utilize a synthetic approach, i.e. instead of simulation (stochastic or deterministic) of a given soil heterogeneity we consider it as an entity to be designed with respect to its hydraulic functions (see, e.g., [7]). The objective of our design is the suction head, H_s, at the soil surface. The standard Richards Equation (RE) serves as a state equation, and its coefficients (SWR, UHC) are control functions. In the vernacular of the optimal control theory (see e.g., [8]), the peculiarity of our problem is in the inherent dependence of the controls on the "trajectory" (solution of the RE), i.e. a strongly nonlinear feedback is stipulated by soil physics (constitutive relations, e.g., given Averyanov or Van Genuchten functions $K \sim \Theta$).

Clearly, either natural or engineering constraints are imposed in optimization. In the admissible class of control functions the synthesis of soil heterogeneity may not be achievable. Then one usually either modifies the cost function/functional, the class of controls or recurs to the famous Hilbert recipe of giving an extended interpretation to the very meaning of "solution" of a variational problem. In this manner, [9] –when faced the non-existence of a "crispy" optimum - systematically used so-called "quasi-solutions". Hilbert's "fuzzification" dates back to Euler's prophecy that any phenomenon of this world possesses a certain (not always obvious) minimum or maximum. Generally, in optimization an algorithmically found extremum is almost always a local one, while a hydrologic engineer or agronomist prefer global optima. So, the solvability, uniqueness and globality should be kept in mind and addressed, whenever possible.

2 Hydraulic Model

We assume that the soil heterogeneity is purely vertical, i.e. we ignore the planar mosaic of (SWR,UHC). This allows us to consider a vertical cross-section in Fig.1

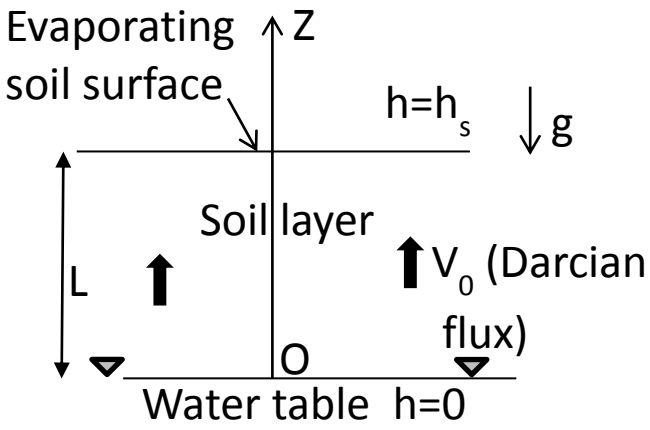


Fig.1 Vertical cross-section of a soil profile with a continuous vertical heterogeneity.

where the origin of Cartesian coordinates coincides with the water table and the vertical coordinate OZ is oriented upward against gravity. We assume that the water table does not fluctuate. We assume that the surface Z=L is also at a constant suction (moisture content), which is determined by atmospheric conditions. The one-phase ascent of moisture from Z=0 to Z=L through a soil layer is steady-state and governed by the second order, nonlinear ODE (RE) as the following boundary value problem (BVP):

$$\vec{V} = -K\nabla H_t,\ K = K_s k_r,\ H_t = Z + H$$
$$\frac{d}{dz}\left(K_s(z)k_r(z,h)\frac{dh_t}{dz} \right) = 0, \quad 0 < z < 1 \qquad (1)$$
$$h(0) = 0, h(1) = h_s \quad \text{or } \theta(0) = 1, \theta(1) = \theta_s$$

where H_t is the total hydraulic head, V is the Darcian velocity, $K_s(z)$ is saturated hydraulic conductivity, $k_r(z,h)$ is the relative conductivity of water. Dimensionless values in eqn.(1) are introduced as $(z, h_t, h)=(Z, H_t, -H)/L$, $\theta = (\Theta - \Theta_i)/(\sigma - \Theta_i)$ where the irreducible moisture content Θ_i and porosity σ for simplicity (but without any loss of generality) are assumed to be z-independent and hence vanish from the solution. We note that the capillary pressure head, h, positive in the flow domain of Fig.1, is now used. Capillarity in eqn.(1) hoists moisture and viscosity with gravity resist the drive.

Obviously, without root water uptake, the second line in eqns.(1) is immediately integrated:

$$\frac{dh(z)}{dz} = 1 + \frac{V_0}{K_s(z)k_r(z,h(z))}, \qquad (2)$$

where the first constant of integration V_0 is a dimensional Darcian flux (evaporation rate) through the layer. The third line in eqn.(1) makes the boundary conditions for the first order ODEs (2) with respect to either h(z) or θ(z).

3 Optimization

We state the following:

Problem A. Given the value of V_0 determine UHC which minimizes h_s.

Agronomically, h_s (actually, h(z) within the rhizosphere lamina $0<L_r<z<L$) determines the root comfort with respect to the liquid-gas environment of the matrix, and hence solution to *Problem A* can guide in creating a soil horizon hydraulically and aerationally favourable to the plants, given the net quantity of conveyed moisture from the AR storage.

We select the Gardner UHC $K = K_s(z)e^{[-\alpha(z)h(z)]}$ where α is the sorptive number. As is well-known, on the level of a pore network of r-radius parallel tubes, K_s is determined by the Hagen-Poiseuille conductance of the tubes ($\sim r^2$) making the capillary bundle and α depends on the Laplacian menisci ($\sim r$) of the tubes. As is also well-known [10], α and K_s are interrelated on the level of the soil continuum.

Separation of variables implemented [5-6] is impossible in eqn.(2). Consequently, we recur to the Runge-Kutta method. We solved *Problem A* in the class of functions $K_s=K_0(1+b\ z)$, where $b>-1$ is a given constant and K_0 is conductivity at the water table (Fig.1) (known from, for instance, a pumping test of the unconfined aquifer beneath the water table). This linear increase or decrease of conductivity induces the corresponding variation of capillarity that we quantify by regression $\alpha = cK_s^s$ where c and s are empiric constants [10]. Actually, in [10] K_s and the Green-Ampt front pressure were correlated. This pressure – according to the known Bouwer integral relation – is proportional to $1/\alpha$.

Now in the selected class, *Problem A* is reduced to minimizing h_s as the boundary condition of what we get from eqn. (2) in the following BVP:

$$\frac{dh(z)}{dz} = 1 + \frac{V_0}{K_0(1+bz)\exp[-c(K_0(1+bz))^s h(z)]},\ 0 < z < 1$$
$$h(0) = 0, h(1) = h_s \qquad (3)$$

In order to find the minimum of h_s (b), we solve eqn.(3) as a Cauchy problem by the *NIntegrate* routine of [11].

We selected the *Lakeland* soil from [10], see their Fig.2, for which c=1/35.36, s=0.415. Then we assumed K_0=1cm/h and L=25 cm. The graphs of h_s(b) – the endpoints of the "trajectory" of ODE (5) - are shown in our Fig.2 for V_0/K_0=0.3, 0.5 and 0.7 (curves 1-3, correspondingly). As is evident from Fig.2, the curves possess a single global minimum. These minima as pairs (h_{sm}, b_m) are found by the *FindRoot* routine of [11]. For the three cases of Fig.2 they are: (1.09, 55.9), (1.17, 50.42) and (1.28,44.52), correspondingly. Computations for other soils from [10] (i.e. other c and s) and other L, K_0, V_0 showed that in some cases the minima exists and in others do not.

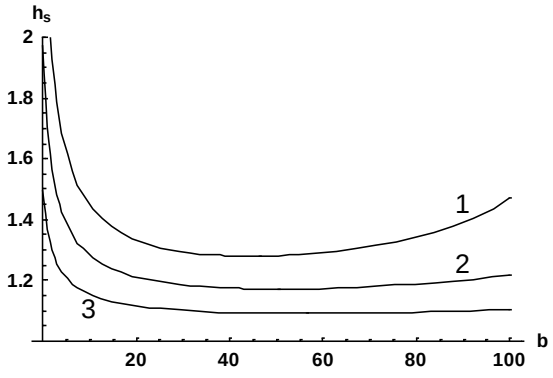


Fig.2 Top soil capillary head as a function of the rate of linear increase of saturated conductivity for the Gardner soil with Ahuja et al. conductivity-sorptive number correlation.

Along with a z-linearly increasing conductivity we also tried quadratic and exponential K_s(z) functions. For several cases we found minima similar to ones shown in Fig.2. So, when is *Problem A* solvable? The question, generally remains open, although for each particular set of soil parameters we can easily answer it, because the Runge-Kutta algorithm routinely reconstructs the "trajectory" h(z). A systematic optimization (i.e. with a proof of a necessary and sufficient optimum conditions) of its endpoint locus, h(1), even for simple K(z) seems gloomy.

3 Conclusions

Optimal moisture/suction conditions and evaporation fluxes of the topsoil are vital for the natural and cultivated vegetation and for efficient cooling operation of a green roof in hot climates. Using computer algebra tools and extensive simulations we solved BVP for the Richards 1-D ODE and detected the soil heterogeneities that optimize the dynamics of soil water in vertical evaporation. Clearly, in *Problem A* we can swap h_s and V_0 as a criterion and constraint. We fixed the total soil thickness L but it can be also used as a criterion in optimal design, if the pedotransfer functions involve the soil bulk density and the mass of a dry soil layer is fixed (rather than volume which we fixed by keeping L in *Problem A* constant). Then - in the language of the optimal control theory - the corresponding optimization will be analogous to the Feldbaum-Bushaw problem of "fastest delivery of a material point to the origin of coordinates" (we recall that time, initial position and velocity in the dynamics of a Newtonian particle are similar to the Z-coordinate, water table pressure and the Darcian velocity in terms of the RE). Solution to *Problem A* is relevant to the "inverse texture phenomenon" [12] i.e. ecologically better performance of moisture transpiring plants in relatively coarse soils as compared to finer substrates.

Acknowledgments

This work was supported by His Majesty grant SR/AGR/SWAE/09/01 ''Feasibility of Managed Aquifer Recharge Using Excess Treated Wastewater in Oman''.

4 References:

[1] D.B.Stephens, *Vadose Zone Hydrology*. CRC Press, 1996.

[2] M.T.Van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Am. J*., 44(5), 892-898, 1980.

[3] H.Vereecken, M.Weynants, M.Javaux, Y.Pachepsky, M.G.Schaap, M.T. Van Genuchten. Using pedotransfer functions to estimate the van Genuchten–Mualem soil hydraulic properties:A Review. *Vadose Zone J*., 9(4), 795-820, 2010.

[4] A.R.Kacimov, M.M.Sherif, J.S.Perret, A. Al-Mushikhi. Control of sea-water intrusion by salt-water pumping: Coast of Oman. Hydrogeology J., 2009, 17, 541-548.

[5] A.Warrick. Additional solutions for steady-state evaporation from a shallow water table. *Soil Science*, 146(2), 63–66, 1988.

[6] J.Zhu, B.Mohanty. Effective hydraulic parameters for steady state vertical flow in heterogeneous soils. *Water Resources Res.,* 39(8), 1227, 2003.

[7] A.R.Kacimov, N.D.Yakimov. Minimal advective travel time along arbitrary streamlines of porous media flows: the Fermat-Leibnitz-Bernoulli problem revisited. *J.Hydrology,* 375, 356-362, 2009.

[8] L.S.Pontryagin, V.G.Boltyanski, R.V.Gamkrelidze, E.F.Mishchenko. *Mathematics of Optimal Processes.* Nauka, Moscow (in Russian), 1976.

[9] A.M.Elizarov, A.R.Kacimov, D.V. Maklakov. *Optimal Shape Design Problems in Aerohydrodynamics*. Fizmatlit, Moscow (in Russian), 2008.

[10] L.R.Ahuja, D.L. Nofziger, D.Swartzendruber, J.D.Ross. Relationship between Green and Ampt parameters based on scaling concepts and field-measured hydraulic data. *Water Resources Res*., 25(7). 1766-1770, 1989.

[11] S.Wolfram. *Mathematica.A System for Doing Mathematics by Computer. Addison-Wesley*, Redwood City, 1991.

[12] I.Noy-Meir. Desert ecosystems: environment and producers. *Ann. Rev. Ecology and Systematics*, 4, 25–51, 1973.

Spatial Pattern Formation of a Modified Leslie-Gower Predator-Prey Model Incorporating Prey Refuge

Sunita Gakkhar, and Dawit Melese
Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, Uttrakhand, India
sungkfma@gmail.com/mahifikir@gmail.com

Abstract – *In this paper, the spatiotemporal dynamics of a modified Leslie-Gower predator-prey system with Beddington - DeAngelis functional response incorporating constant proportion of prey refuge under homogeneous Neumann boundary condition is investigated. The local and global asymptotic stability of the unique positive homogeneous steady state in the absence of diffusion are discussed. Furthermore, we perform a series of numerical simulations and the results of the numerical simulations reveal that the typical dynamics of population density variation is the formation of isolated groups, i.e., stripe like or spotted or coexistence of both..*

Keywords: Leslie-Gower; Prey Refuge; Turing Patterns.

1 Introduction

The dynamic relationships between species and their complex properties are at the heart of many ecological and biological processes [1]. One of such relationships is the dynamical relationship between a predator and their prey which has long been and will continue to be one of the dominant themes in both ecology and mathematical ecology due to its universal existence and importance [2].

We live in a spatial world and spatial patterns are ubiquitous in nature. The issue of spatial and spatiotemporal pattern formation in biological communities is probably one of the most exciting problems in modern biology and ecology [3], [4].

On the other hand, predator-prey interactions often exhibit spatial refugia which afford the prey some degree of protection from predation. Such refugia can help in prolonging prey-predator interactions by reducing the chance of extinction due to predation [5] - [7], and damp prey predator oscillations [8]. In the literatures studies show that refuges have both stabilizing [9] and destabilizing effect [10], [11].

The effect of the use of refuges by the prey population on the temporal dynamics of a prey-predator system has been investigated by many people [7],[12]-[16]. However, to the best of our knowledge, little attention has been given to the dynamics of a spatiotemporal prey-predator system incorporating prey refuges.

The main aim of this paper is to study the effect of prey refuge on the stability property of the coexistence equilibrium point and the Turing pattern formation of a modified Leslie-Gower prey-predator model incorporating constant proportion of prey refuge with Beddington - DeAngelis functional response.

The organization of the paper is as follows. Section two devotes to the local and global asymptotical stability of the unique positive equilibrium point in the absence of diffusion. In section 3, the emergence of Turing patterns via numerical simulations are shown. At last, conclusion is presented in section 4.

2 Temporal System

A modified Leslie-Gower predator-prey system incorporating a constant proportion of prey refuge with Beddington-DeAngelis functional response in homogeneous environment is governed by the following system of non-linear ordinary differential equations

$$\begin{aligned}\frac{dU}{d\tau} &= r_1\left(1-\frac{U}{K}\right)U - \frac{c(1-m)U}{B+(1-m)U+\varpi V}V,\\ \frac{dV}{d\tau} &= r_2\left(1-\frac{V}{s_1+s(1-m)U}\right)V,\end{aligned} \tag{2.1}$$

subject to initial conditions $U(0)\geq 0$ and $V(0)\geq 0$. $U=U(\tau)$ and $V=V(\tau)$ represents the prey and predator densities respectively. The reaction parameters $r_1, K, c, B, \varpi, r_2, s, s_1$ are positive constants which stand for the intrinsic growth rate of the prey, the environmental carrying capacity of prey population, a maximum consumption rate, a saturation constant, predator interference, maximum per capita growth rate of the predator, the conversion factor of prey into predator, normalization constant respectively.

Note that, system (2.1) incorporates constant proportion of prey refuge $mU, m\in[0,1)$, which leaves $(1-m)U$ of the prey available for predation.

Using the following scaling:

$$u=\frac{U}{K},\ v=\frac{V}{sK},\ t=r_1\tau,$$

and, the parameters

$$\alpha=\frac{cs}{r_1},\beta=\frac{B}{K},\omega=s\varpi,\gamma=\frac{r_2}{r_1},b=\frac{s_1}{sK}.$$

system (2.1) takes the non dimensional form

$$\begin{aligned}\frac{du}{dt}&=(1-u)u-\frac{\alpha(1-m)uv}{\beta+(1-m)u+\omega v}=G_1(u,v)\\ \frac{dv}{dt}&=\gamma\left(1-\frac{v}{b+(1-m)u}\right)v=G_2(u,v)\end{aligned}\qquad(2.2)$$

Lemma 2.1: All solutions of (2.2) initiating in $\mathbb{R}^2_+$ are bounded.

The system (2.2) has at most four biologically feasible equilibrium points: $(0,0),(1,0),(0,b)$ and $E(\tilde{u},\tilde{v})$, which is the solution of $(1-u)(\beta+(1-m)u+\omega v)=\alpha(1-m)v$ and $v=b+(1-m)u$.

It may be observed that E is positive and unique if

$$(\alpha(1-m)-\omega)b<\beta\qquad(2.3)$$

Linearization of system (2.2) at E yields the coefficients of the Jacobean matrix J as:

$$a_{11}=1-2\tilde{u}-\frac{\alpha(1-m)\tilde{v}(\beta+\omega\tilde{v})}{(\beta+(1-m)\tilde{u}+\omega\tilde{v})^2},a_{21}=(1-m)\gamma>0$$

$$a_{12}=-\frac{\alpha(1-m)\tilde{u}(\beta+(1-m)\tilde{u})}{(\beta+(1-m)\tilde{u}+\omega\tilde{v})^2}<0,\quad a_{22}=-\gamma<0$$

Theorem 2.1: The unique positive equilibrium point E is locally asymptotically stable provided

$$\begin{cases} m<\dfrac{\alpha(b+\gamma)-1}{\alpha\gamma}; & \beta>b\\ 1-\dfrac{\beta+\omega b}{\sqrt{\alpha(b-\beta)}}<m<\dfrac{-1+\alpha(b+\gamma)}{\alpha\gamma}; & \beta<b\end{cases}\qquad(2.4)$$

Proof: The trace and determinant of the Jacobean matrix J at E are simplified as

$$trace(J)=\frac{\tilde{u}(\tilde{u}^2-p\tilde{u}+q)-\alpha b\gamma}{\alpha(b+(1-m)\tilde{u})}$$

and $\det(J)=\gamma\left(\dfrac{z+n(2(\beta+\omega b)+n\tilde{u})\tilde{u}}{(\beta+(1-m)\tilde{u}+\omega\tilde{v})^2}\right)\tilde{u}$ with

$$p=(\alpha(1-m)+2),\ q=1-\alpha(b+(1-m)\gamma),$$

$z=\alpha(1-m)^2(-b+\beta)+(\beta+\omega b)^2$ and $n=(1-m)(1+\omega)$.

The trace is negative provided $q<0$ *i.e.* $m<\dfrac{-1+\alpha(b+\gamma)}{\alpha\gamma}$

and the determinant is positive if

$z>0$ *i.e.* $\beta>b$ *or* $b>\beta$ *and* $m>1-\dfrac{\beta+\omega b}{\sqrt{\alpha(b-\beta)}}$.

Theorem 2.2: The system (3.1) does not admit any periodic solution when $m>1-(\omega/\alpha)$.

Proof: Let $(u(t),v(t))$ be a positive solution of (2.2) and define a Dulac function $H=\dfrac{\beta+(1-m)u+\omega v}{u^2v^2}$ from system (2.2), we have

$$\begin{aligned}Q&=\frac{\partial(HG_1)}{\partial u}+\frac{\partial(HG_2)}{\partial v}\\&=-H(u,v)\frac{(1-m)u^2+\beta+v(\omega-\alpha(1-m))}{(1-m)u+\beta+\omega v}\\&\quad-H(u,v)\gamma\left(\frac{((1-m)u+\beta)((1-m)u+b)+\omega v^2}{((1-m)u+\beta+\omega v)((1-m)u+b)}\right)u\end{aligned}$$

Therefore by Dulac criterion, we see that if $m>1-(\omega/\alpha)$ then system has no non-trivial positive periodic solutions.

Theorem 2.3: If $m>1-(\omega/\alpha)$ then the local stability of system (2.2) ensures its global stability around the unique positive interior equilibrium point $E(\tilde{u},\tilde{v})$.

Proof: The unique equilibrium point $E(\tilde{u},\tilde{v})$ is the only stable point in the uv plane. The boundedness of the solutions of the system together with the non existence of periodic solutions establishes the global stability.

3 The Spatiotemporal System : Turing Bifurcation

In the predator-prey model (2.2), the prey and predator species are assumed to be spatially independent and dispersion is not included. However, in reality, prey and predator populations are heterogeneously distributed over the habitat. Taking into account the mobility of the prey and predator population within a bounded habitat, the governing model (2.2) is modified as the following system of reaction-diffusion equations, after an appropriate scaling of spatial coordinates:

$$\frac{\partial u}{\partial t} = \Delta u + (1-u)u - \frac{\alpha(1-m)uv}{\beta+(1-m)u+\omega v},$$
$$\frac{\partial v}{\partial t} = \delta\Delta v + \gamma\left(1-\frac{v}{b+(1-m)u}\right)v \qquad (3.1)$$

The system is subjected to the homogeneous Neumann boundary condition and non-negative initial condition. The operator Δ represents the Laplacian operator in two spatial domains and δ is the ratio of the diffusion coefficient of predator to prey.

To linearize the dynamic system (3.1) around the spatially homogeneous equilibrium point $E(\tilde{u}, \tilde{v})$ for small space and time dependent fluctuations, set

$$u(x,y,t) = \tilde{u} + \bar{u}(x,y,t); \left|\bar{u}(x,y,t)\right| \le \tilde{u}$$
$$v(x,y,t) = \tilde{v} + \bar{v}(x,y,t); \left|\bar{v}(x,y,t)\right| \le \tilde{v}$$

Let us assume solutions of the form

$$\begin{pmatrix}\bar{u}(x,y,t)\\ \bar{v}(x,y,t)\end{pmatrix} = \begin{pmatrix}\rho_0\\ \rho_1\end{pmatrix} e^{\lambda t}\cos(k_x x)\cos(k_y y),$$

where λ is the growth rate of perturbation in time *t*, $\rho_i\ (i=0,1)$ represent the amplitudes, k_x and k_y are the wave number of the solutions. The corresponding linearized system has the characteristic equation

$$\left|J - k^2 D - \lambda I\right| = 0. \qquad (3.2)$$

Here $D = diag(1,\delta), k^2 = k_x^2 + k_y^2$ and k represents the wave numbers.

The characteristic polynomial corresponding to $E(\tilde{u}, \tilde{v})$ is

$$\lambda^2 - tr_k\lambda + \Delta_k = 0, \qquad (3.3)$$

where

$tr_k = (a_{11} + a_{22} - k^2(1+\delta))$,

$\Delta_k = a_{11}a_{22} - a_{21}a_{12} - k^2(a_{22} + \delta a_{11}) + \delta k^4$

The roots of equation (3.3) give the dispersion

$$\lambda_{1,2} = \frac{1}{2}\left(tr_k \pm \sqrt{tr_k^2 - 4\Delta_k}\right).$$

The reaction-diffusion systems have led to the characterization of two basic types of symmetry-breaking bifurcations responsible for the emergence of spatiotemporal patterns. The space-independent Hopf bifurcation $[\mathrm{Im}(\lambda(k^2)) \neq 0, \mathrm{Re}(\lambda(k^2)) = 0 \ at\ k^2 = 0]$ breaks the temporal symmetry of a system and gives rise to oscillations that are uniform in space and periodic in time. The (stationary) Turing bifurcation $[\mathrm{Im}(\lambda(k^2)) = 0, \mathrm{Re}(\lambda(k^2)) = 0 \ at\ k^2 = k_T^2 > 0]$ breaks spatial symmetry, leading to the formation of patterns that are stationary in time and oscillatory in space.

Linear stability analysis of system (3.1) yields the bifurcation diagram with $\alpha = 1.18, \beta = 0.15, \omega = 0.2, b = 0.1, \gamma = 0.2$.

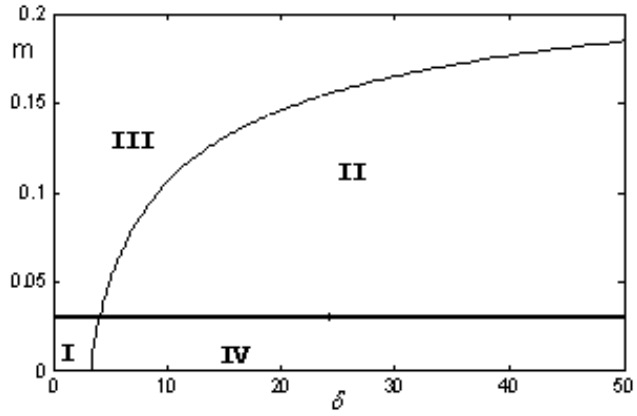


Fig. 1. Bifurcation diagram of system (3.1) with $\alpha = 1.18, \beta = 0.15, \omega = 0.2, b = 0.1, \gamma = 0.2$. Hopf and Turing bifurcation lines separate the parameter space into four domains.

In Fig.1, the Hopf bifurcation line and the Turing bifurcation curve, which separate the parametric space into four distinct domains, for the spatiotemporal system (3.1) are shown in the $\delta - m$ plane. Domain I and II are the region of pure Hopf instability and pure Turing instability respectively. In domain IV, located above all two bifurcation lines, the steady state is the only stable solution of the system whereas in domain III, which is located below all two bifurcation lines, both Hopf and Turing instability occur. Domain II is the Turing space.

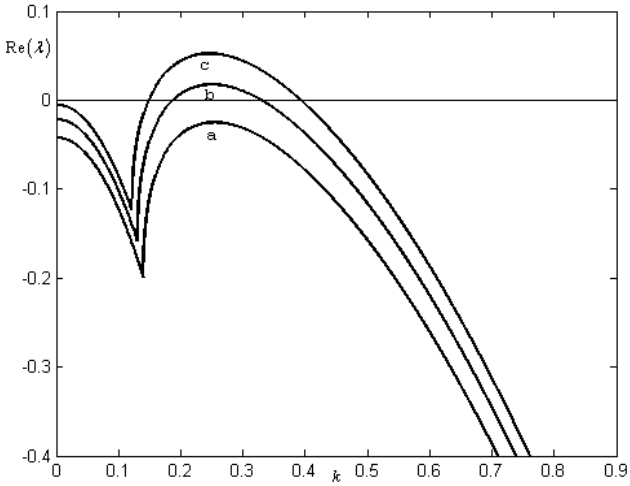


Fig. 2. The Real part of the dispersion relation corresponding to the characteristics equation (3.3) with $\alpha = 1.18, \beta = 0.15, \omega = 0.2, b = 0.1, \gamma = 0.2, \delta = 15$ a) m=0.15, b) m=0.1, c) m=0.05

Figure 2 shows the dispersion relation corresponding to three values of the bifurcation parameter m while keeping the others parameters fixed as

$$\alpha = 1.18, \beta = 0.15, \omega = 0.2, b = 0.1, \gamma = 0.2, \delta = 15. \quad (3.4)$$

In Fig. 2, the curve (a) corresponds to the case when the value of the refugee parameter m equals 0.15, which is

greater than the critical Turing value $m_{cr} = 0.12175$. The curves (b) and (c) show the occurrence of Turing instability for $m = 0.1$ *and* $m = 0.05$ respectively. From Fig. 2, it can be seen that a decrease in the value of m increases the available Turing modes [$\mathrm{Re}(\lambda) > 0$], and it also further enhances the available modes.

The spatiotemporal system (3.1) is solved numerically in two-dimensional space using a finite difference approximation for the spatial derivatives and an explicit Euler method for the time integration [17]. In order to avoid numerical artifacts the values of the time and space steps have been chosen sufficiently small. This method finally results to a sparse, banded linear system of algebraic equations. The linear system obtained is then solved by using GMRES algorithm [17].

All our numerical simulations employ the zero-flux (Neumann) boundary conditions in a square habitat of size 200x200. Iterations are performed for different step sizes in time and space until the solution seems to be invariant. The time step size of $\Delta t = 0.1$ and space step size $h = 0.5$ are chosen. The initial density distribution corresponds to random perturbations around the stationary state $E(\tilde{u}, \tilde{v})$ in the spatially extended system (3.1), which seems to be more general from the biological point of view.

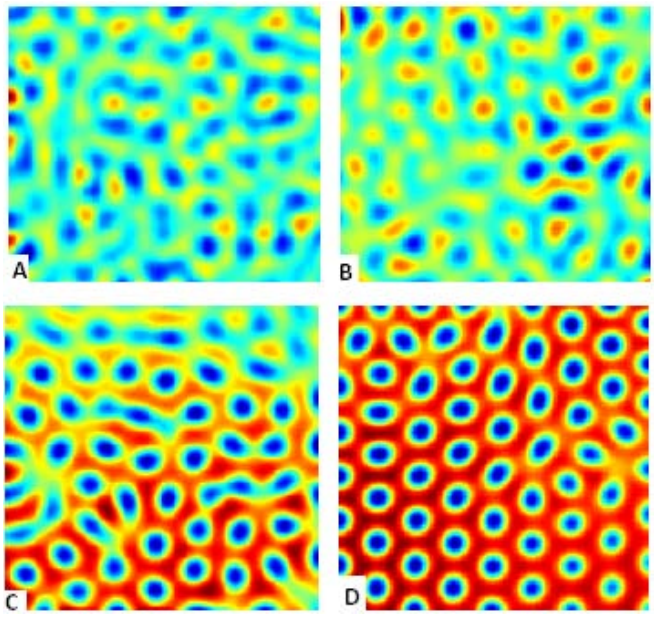


Fig. 3. Snapshots of contour pictures of the time evolution of the prey at different instants with $m = 0.1$. Other parameters are fixed as (3.4). Iterations number: (A) 500, (B) 2000, (C) 5000, (D) 10000.

In the numerical simulations, different types of dynamics are observed and it is found that the distributions of predator and prey are always of the same type. Consequently, we can restrict our analysis of pattern formation to one distribution. In this section, we show the distribution of prey u, for instance. And the parameter m is located in the Turing space, domain II (cf., Fig. 1.). We have taken some snapshots with red (blue) corresponding to the high (low) value of prey u.

Fig. 3 shows the time evolution of spatial pattern formation for the spatiotemporal system (3.1) at 500, 2000, 5000, 150000 iterations when the value of the refuge constant m is 0.1 and other parameters are given in (3.4). In this case, one can see that a small random perturbation to the homogeneous state $(\tilde{u}, \tilde{v}) = (0.222771, 0.300494)$ of the system (3.1) leads to the formation of stripes and spots (cf., Fig. 3(B) and (C)). However, at the final stage patterns of blue spots on a red background are formed. They can be called as ''cold spots'' (cf., Fig. 3(D)) which are isolated zones with low prey density.

From the snapshot in Fig. 4, one can see that a decrease in m to 0.075 while keeping the other parameters fixed as in (3.4) finally leads to the coexistence of spotted patterns and the stripe-like patterns. However, a further decrease in m to 0.05 finally results in the formation of stripe-like patterns only (cf., Fig. 5)

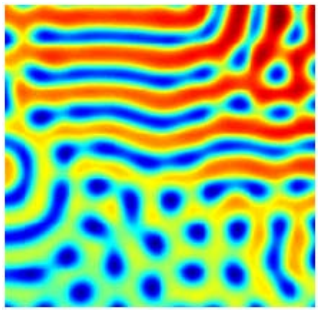


Fig. 4. Snapshot of contour picture of the prey at $m = 0.075$ (10000 iterations). Other parameters are fixed as (3.4).

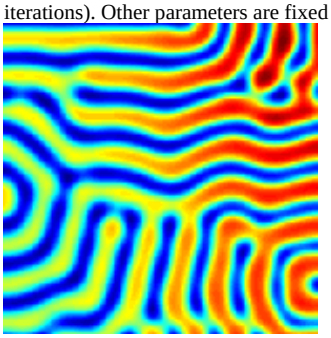


Fig. 5. Snapshot of contour picture of the prey at $m = 0.05$ (10000 iterations). Other parameters are fixed as (3.4).

4 Conclusion

The numerical simulation results indicate that the effect of the prey refuge for pattern formation is remarkable. More specifically, as the value of the prey refuge constant is increased, the stripe like patterns breaks down and ultimately form spotted like. This may enrich the dynamics of the effect of refuge on the predator-prey system.

5 References

[1] Baba I. Camara, Moulay A. Aziz-Alaoui, "Complexity in a prey predator model, " 2007 International Conference in Honor of Claude Lobry

[2] Berryman AA. "The origins and evolutions of predator–prey theory, "Ecology, 1992; 73:1530–1535.

[3] Wang Lin and Zhenbing Zeng, "Computer aided analysis of pattern formation in a semi-ratio-dependent predator-prey model, " 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), V12, pp 558-560

[4] Hongxia Shen and Zhen Jin, "Two Dimensional Pattern Formation of Prey-predator System, "Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, DOI 10.1109/SNPD.2007.215 343-346

[5] Zeng Zhijun, "Optimization Problem of a Predator-prey System with Holling II Functional Response, "Proceedings of the 29th Chinese Control Conference , July 29-31, 2010, Beijing, China, pp 5483-5485

[6] Huang, F. Chen, L. Zhong, "Stability analysis of a prey-predator model with Holling type III response function incorporating a prey refuge, "Appl. Math. Comput., 182, pp. 672–683, 2006

[7] T.K. Kar, "Stability analysis of a prey-predator model incorporating a prey refuge," Commun. Nonlinear Sci. Numer. Simul., 10, pp. 681–691, 2005

[8] J. B.Collins, "Bifurcation and stability analysis of a temperature-dependent mite predator-prey interaction model incorporating prey refuge, "Bulletin of Mathematical Biology, 57 (1995) 63-76

[9] M.P. Hassel, The Dynamics of Arthropod Predator-Prey Systems, Princeton Univ. Press, Princeton, 1974.

[10] J.N. McNair, "The effects of refuges on predator_prey interactions: A reconsideration, " Theoretical Population Biology 29 (1986) 38_63

[11] Taylor, R.J.1984.Predation. New York: Chapman & Hall

[12] Eduardo González-Olivares, Rodrigo Ramos-Jiliberto, "Dynamic consequences of prey refuges in a simple model system: more prey, fewer predators and enhanced stability, " Ecological Modelling 166 (2003) 135–146

[13] J.N. McNair, "Stability effects of prey refuges with entry-exit dynamics, " Journal of Theoretical Biology 125 (1987) 449_464

[14] Youde Tao , Xia Wang , Xinyu Song , Effect of prey refuge on a harvested predator–prey model with generalized functional response , Commun Nonlinear Sci Numer Simulat 16(2011) 1052-1059

[15] Fengde Chena, Liujuan Chenb, Xiangdong Xie , "On a Leslie-Gower predator-prey model incorporating a prey refuge, " Nonlinear Analysis: Real World Applications 10 (2009), pp 2905-2908

[16] Lili Ji , Chengqiang Wu, "Qualitative analysis of a predator prey model with constant-rate prey harvesting incorporating a constant prey refuge, "Nonlinear Analysis: Real World Applications 11 (2010) 22852295

[17] Marcus R. Garvie **, "**Finite-Difference Schemes for Reaction–Diffusion Equations Modeling Predator–Prey Interactions in MATLAB," Bulletin of Mathematical Biology (2007) 69: 931–956, DOI 10.1007/s11538-006-9062-3

BILINEAR GARCH TIME SERIES MODELS

Mahmoud Gabr, Mahmoud El-Hashash

Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt

Department of Mathematics and Computer Science, Bridgewater State University, Bridgewater, MA, USA

Abstract

In this paper the class of BL-GARCH (Bilinear General AutoregRessive Conditional Heteroskedasticity) models is introduced. The proposed model is a modification to the BL-GARCH model proposed by Storti and Vitale (2003). Stationary conditions and autocorrelation structure for special cases of these new models are derived. Maximum likelihood estimation of the model is also considered. Some simulation results are presented to evaluate our algorithm.

Keywords : Time series, ARCH models, GARCH models, Bilinear models, weak dependence, .

1. Introduction

A lot of time series encountered in empirical applications are nonlinear and non-stationary. Their structures such as means and variances may vary over time. The problem of nonlinear time series identification and modeling has attracted considerable attention for the past 30 years in diverse fields such as financial econometrics, biometrics, socioeconomics, transportation, electric power systems, and aeronautics which exhibit nonlinear process. A good nonlinear model should be able to capture some of the nonlinear phenomena in the data. Moreover, it should also have some intuitive appeal. Therefore a number of wide classes of non-linear time series models have been proposed, investigated and studied. One of these classes which has received a great deal of attention is that of bilinear models. Bilinear time series models and its statistical and probabilistic properties have been extensively studied by [7] Granger and Andersen (1978), [14]Subba Rao (1981), [5] Gabr (1992) and comprehensively surveyed by [15] Subba Rao and Gabr (1984) and [11] Pham (1993).

A class of non-linear model, called a bilinear class, may be regarded as a plausible non-linear extension of ARMA, rather than of the AR model. Bilinear models incorporate cross-product terms involving lagged values of the time series and of the innovation process. The model may also incorporate ordinary AR and MA terms. The general form of a bilinear time series $\{X_t, t = 0, \pm 1, \pm 2, ...\}$ denoted by BL(p, q, P, Q) is defined by

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \sum_{j=1}^{q} c_j e_{t-j} + \sum_{i=1}^{P}\sum_{j=1}^{Q} b_{ij} X_{t-i} e_{t-j} + e_t \quad (1)$$

where $\{e_t\}$ is a set of independent random variables. We define the model (1) as a bilinear time series model BL (p,r,m,k) and the process **$\{X_t\}$** as a bilinear process.

In econometrics, a vast literature is devoted to the study of autoregressive conditionally heteroskedastic (ARCH) models for financial data. One of the best-known model is the GARCH model (Generalized Autoregressive Conditionally Heteroskedastic) introduced by [3] Engle (1982) and [1] Bollerslev (1986). The classical GARCH(p,q) model is given by the equations

$$\varepsilon_t = \sigma_t Z_t \,,\; h_t = \sigma_t^2$$

$$h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_q \varepsilon_{t-q}^2 + \beta_1 h_{t-1} + \cdots + \beta_p h_{t-p}$$

$$= \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j} \quad (2)$$

where

$$\alpha_0 > 0 \,,\; \alpha_i \geq 0 \,,\quad \beta_j \geq 0 \,,\quad q \geq 0 \,,\quad p \geq 0$$

are model parameters and $\{Z_j, j=1, 2, 3, \ldots\}$ are independent identically distributed (i.i.d.) random variables with zero mean and variance 1. The variables ε_t, σ_t, Z_t in (2) are usually interpreted as financial (log) returns (**ε_t**), their volatilities or conditional standard deviations (σ_t), and so-called innovations or shocks (Z_t), respectively; the innovations are supposed to follow a certain fixed distribution (e.g., standard normal). Later, a number of modifications of (4.1) were proposed, which account for asymmetry, leverage effect, heavy tails and other" stylized facts".

Under some additional conditions, similarly as in the case of ARMA models, the GARCH model can be written as ARCH(∞) model i.e., h_t can be represented as a moving average of the past squared returns ε_s^2, s < t, with exponentially decaying coefficients (see [1] Bollerslev, 1986) and absolutely summable exponentially decaying autocovariance function. For instance, the GARCH(p, q) process of (2) can be written as

$$\varepsilon_t = \sigma_t Z_t\,,\; h_t = \sigma_t^2\,,$$

$$h_t = \left[1-\beta(1)\right]^{-1}\alpha_0 + \left[1-\beta(B)\right]^{-1}\alpha(B)\varepsilon_t^2$$

where $\beta(B) = \beta_1 B + \cdots + \beta_p B^p$ and B stands for the back-shift operator, $B^k X_t = X_{t-k}$. This leads to the ARCH(∞) representation;

$$\varepsilon_t = \sigma_t Z_t\,,\; h_t = \sigma_t^2$$

$$h_t = b_0 + \sum_{i=1}^{\infty} b_i\,\varepsilon_{t-i}^2 \qquad \textbf{(3)}$$

with $b_0 = \left[1-\beta(1)\right]^{-1}\alpha_0$ and with positive exponentially decaying weights b_i, $i \geq 1$ defined by the generating function $\alpha(y)/\left[1-\beta(y)\right] = \sum_{i=1}^{\infty} b_i\, y^i$. It is interesting to note that the non-negativity of the regression coefficients $\boldsymbol{\alpha_j}$, $\boldsymbol{\beta_j}$ in (2) is not necessary for non-negativity of b_j in the corresponding ARCH(∞) representation, see [10] Nelson and Cao (1992).

Clearly, if $E(Z_t / \varepsilon_s\,,\, s < t) = 0$, $E(Z_t^2/\varepsilon_s\,,\, s < t) = 1$ then $\boldsymbol{\varepsilon_t}$ has conditional mean zero and a random conditional variance σ_t^2, i.e.

$$E(\varepsilon_t / \varepsilon_s\,,\, s < t) = 0,\; \mathrm{var}(\varepsilon_t/\varepsilon_s\,,\, s < t) = \sigma_t^2 = h_t$$

The general framework leading to the model (2) was introduced by [12] Robinson (1991) in the context of testing for strong serial correlation and has been subsequently studied by [8] Kokoszka and Leipus (2000) in the change-point problem context. The class of ARCH(∞) models includes the finite order ARCH and GARCH models of [3] Engle (1982) and [2]Bollerslev (1986).

2. The Bilinear ARCH Models

Formally, the classes AR, ARCH, LARCH (at least, their finite memory counterparts ARMA, GARCH, ARCH) all belong to the general class of bilinear model (1). [6] Giraitis and Surgailis (2002) studied the heteroscedastic bilinear equation

$$X_t = Z_t\left(\alpha_0 + \sum_{i=1}^{\infty}\alpha_i X_{t-i}\right) + \beta_0 + \sum_{i=1}^{\infty}\beta_i X_{t-i} \qquad \textbf{(4)}$$

where $\{Z_t,\ t=1, 2, 3, \ldots\}$ are i.i.d. random variables, with zero mean and variance 1, and α_j , β_j , $j \geq 0$ are real (not necessary nonnegative) coefficients. Equation (4) appears naturally when studying the class of processes with the property that the conditional mean

$$\mu_t = E(X_t / X_s,\, s < t)$$

is a linear combination of X_s, $s < t$, and the conditional variance

$$h_t^2 = \sigma_t^2 = \mathrm{Var}(X_t / X_s,\, s < t)$$

is the square of a linear combinations of $\mathbf{X_s}$, $s < t$, as it is in the case of (4): i.e.

$$\mu_t = E\left[X_t / X_s\,, s < t\right] = \beta_0 + \sum_{i=1}^{\infty}\beta_i X_{t-i}$$

$$h_t^2 = \sigma_t^2 = \mathrm{var}\left[X_t / X_s\,, s < t\right] = \left(\alpha_0 + \sum_{i=1}^{\infty}\alpha_i X_{t-i}\right)^2$$

Clearly, the case $\alpha_j \equiv 0$, $j \geq 1$ gives the linear AR(∞) equation, while $\beta_j \equiv 0$ ($j \geq 0$) results in the Linear ARCH (LARCH) model, introduced by [12] Robinson (1991), defined by the equation

$$\varepsilon_t = \sigma_t Z_t\,,\; h_t = \sigma_t \qquad h_t = \alpha + \sum_{j=1}^{\infty} c_j\,\varepsilon_{t-j}$$

The main advantage of LARCH is that it allows modeling of long memory as well as some characteristic asymmetries (the "leverage effect"). Both these properties cannot be modeled by the classical ARCH(∞) with finite fourth moment. The coefficients c_i satisfy

$$c_j \sim k\, j^{d-1} \quad \text{for some} \quad 0 < d < ½\,,\; k > 0$$

which implies the condition

$$\sum_{j=1}^{\infty} c_j^2 < \infty$$

Neither α nor the c_j are assumed positive and, unlike in (4.3), $\boldsymbol{\sigma_t}$ (not σ_t^2), is a linear combination of the past values of $\boldsymbol{\varepsilon_t}$, rather than their squares.

[4] Engle and Ng (1993) introduced a nonlinear asymmetric GARCH model which captures asymmetry by means of interactions between past returns and volatilities In the simple (p=1,q=1) case the conditional variance equation is given by

$$\varepsilon_t = \sigma_t Z_t ,$$

$$\sigma_t^2 = h_t^2 = \alpha_0 + a_1(\varepsilon_{t-j} + \xi_1 h_{t-1})^2 + b_1 h_{t-1}^2 \quad \textbf{(5)}$$

with the model becoming asymmetric when the coefficient $\boldsymbol{\xi_1}$ is equal to zero. [13] Starti and Vitale (2003) have generalized this model to the following BL-GARCH model

$$\varepsilon_t = \sigma_t Z_t ,$$

$$\sigma_t^2 = h_t^2 = \alpha_0 + \sum_{j=1}^{p} a_j \varepsilon_{t-j}^2 + \sum_{j=1}^{p} b_j h_{t-j}^2 + \sum_{j=1}^{p} c_j \varepsilon_{t-j} h_{t-j} \quad \textbf{(6)}$$

where a_j, b_j, c_j j=1, 2, ..., p are constants. This model has the advantage of being characterized by a more flexible parametric structure In this model leverage effects are explained by the interactions between past observations and volatilities To see the positivity of the conditional variance in equation (6), we can write

$$a_j \varepsilon_{t-j}^2 + b_j h_{t-j}^2 + c_j \varepsilon_{t-j} h_{t-j} = \left(\sqrt{a_j}\, \varepsilon_{t-j} + \sqrt{b_j}\, h_{t-j} \right)^2 + \left(c_j - 2\sqrt{a_j b_j} \right) \varepsilon_{t-j} h_{t-j}$$

j=1, 2, …, p

Hence for, $\boldsymbol{\alpha_0 > 0}$, a sufficient condition for $\boldsymbol{h_t^2 > 0}$, in (6), is given by

$$4 a_j b_j \le c_j^2 \quad \text{for } j=1, 2, \ldots, p \quad \textbf{(7)}$$

Model (6) with the condition (7) leads us to introduce a simpler reduced parameter Bilinear GARCH model in the form;

$$\varepsilon_t = \sigma_t Z_t ,$$

$$\sigma_t^2 = h_t^2 = \alpha_0 + \sum_{j=1}^{p} \left(\alpha_j \varepsilon_{t-j} + \beta_j h_{t-j} \right)^2 \quad \textbf{(8)}$$

From (6) and (8), we can see that the two models contain exactly the same number of terms, although the number of parameters required by each model is different. In fact the number of parameters in (8) is less than that in (6) by p parameters. Moreover, we do not need the condition of positivity of the parameters α_j, β_j of model (8).

Theorem

The Bilinear GARCH process (8) is stationary in wide sense if and only if the roots u_i of the polynomial

$$\Phi(u) = 1 - \sum_{i=1}^{p} \phi_i u^i$$

where $\phi_i = \alpha_i^2 + \beta_i^2$, lie outside the unit circle.

Proof

The Bilinear GARCH process (8) can be rewritten as

$$h_t^2 = \alpha_0 + \sum_{j=1}^{p} \left(\alpha_j h_{t-j} Z_{t-j} + \beta_j h_{t-j} \right)^2 = \alpha_0 + \sum_{j=1}^{p} \eta_j h_{t-j}^2 \quad \textbf{(9)}$$

which is a random coefficient autoregressive representation for h_t^2 where

$$\eta_j = (\alpha_j Z_{t-j} + \beta_j)^2 \quad \textbf{(10)}$$

Taking in consideration the properties of $\{Z_t\}$, the expectation of (9) is given by

$$E\ h_t^2 = \alpha_0 + \sum_{j=1}^{p} E\ \eta_j h_{t-j}^2 = \alpha_0 + \sum_{j=1}^{p} E\ \eta_j\ E\ h_{t-j}^2$$

$$= \alpha_0 + \sum_{j=1}^{p} \left(\alpha_j^2 + \beta_j^2 \right) E\ h_{t-j}^2 = \alpha_0 + \sum_{j=1}^{p} \phi_j E\ h_{t-j}^2$$

Since,

$$E\ h_t^2 = E\left(E\ \varepsilon_t^2 / \psi_{t-1} \right) = E\ \varepsilon_t^2$$

it follows that,

$$Y_t = \alpha_0 + \sum_{j=1}^{p} \phi_j Y_{t-j} \qquad \textbf{(11)}$$

where $Y_t = E\ \varepsilon_t^2$. Letting B be the backward shift operator defined by $B^k Y_t = Y_{t-k}$, equation (11) can be rewritten as,

$$Y_t = \frac{\alpha_0}{\left(1 - \sum_{j=1}^{p} \phi_j B^j\right)}$$

Therefore Y_t in (11) converges to a finite value if and only if all the roots u_i of the polynomial $\Phi(u) = 1 - \sum_{i=1}^{p} \phi_i u^i$ lie outside the unit circle which completes the proof.

The simplest but often very useful Bilinear GARCH process is that of order 1 given by

$$\varepsilon_t / \psi_{t-1} \sim N(0, h_t^2) \qquad \textbf{(12)}$$

where

$$\mathbf{h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1} + \beta_1 h_{t-1}}^{\ 2} \qquad \textbf{(13)}$$

with $\alpha_0 > 0$. The unconditional variance is

$$\begin{aligned} E(\varepsilon_t^2) &= E[E(\varepsilon_t^2 / \psi_{t-1})] \\ &= E(h_t^2) \\ &= E(\alpha_0 + \alpha_1^2 \varepsilon_{t-1}^2 + \beta_1^2 h_{t-1}^2 + 2\alpha_1 \beta_1 \varepsilon_{t-1} h_{t-1}) \\ &= \alpha_0 + \alpha_1^2 E(\varepsilon_{t-1}^2) + \beta_1^2 E(h_{t-1}^2) \\ &= \alpha_0 + \alpha_1^2 E(\varepsilon_{t-1}^2) + \beta_1^2 E(\varepsilon_{t-1}^2) \\ &= \alpha_0 + \alpha_1^2 + \beta_1^2 \ E(\varepsilon_{t-1}^2) \end{aligned}$$

which implies that

$$E(\varepsilon_t^2) = \frac{\alpha_0}{1 - \alpha_1^2 - \beta_1^2} \qquad \textbf{(14)}$$

The sequence of variances converges to the constant if $\alpha_1^2 + \beta_1^2 < 1$ suffices for wide sense stationarity.

Under normality assumption

$$\begin{aligned} E(\varepsilon_t^4) &= E(E(\varepsilon_t^4 / \psi_{t-1})) = 3E(h_t^4) \\ &= 3E\ \alpha_0 + \alpha_1^2 \varepsilon_{t-1}^2 + \beta_1^2 h_{t-1}^2 + 2\alpha_1 \beta_1 \varepsilon_{t-1} h_{t-1}^{\ 2} \\ &= 3E\ \alpha_0 + \alpha_1^2 \varepsilon_{t-1}^2 + \beta_1^2 h_{t-1}^2 + 2\alpha_1 \beta_1 \varepsilon_{t-1} h_{t-1}^{\ 2} \end{aligned}$$

From which we obtain,

$$E(\varepsilon_t^4) = \frac{3\alpha_0^2 (1 + \alpha_1^2 + \beta_1^2)}{(1 - \alpha_1^2 - \beta_1^2)(1 - 3\alpha_1^4 - \beta_1^4 - 6\alpha_1^2 \beta_1^2)} \qquad \textbf{(15)}$$

The necessary and sufficient condition for the existence of the fourth moment is

$$3\alpha_1^4 + \beta_1^4 + 6\alpha_1^2 \beta_1^2 < 1 \qquad \textbf{(16)}$$

The coefficient of Kurtosis is

$$\frac{E(\varepsilon_t^4)}{E\ \varepsilon_t^2}^{\ 2} = \frac{3\left(1 - \alpha_1^2 + \beta_1^2\right)^{\ 2}}{1 - 3\alpha_1^4 - \beta_1^4 - 6\alpha_1^2 \beta_1^2}$$

(2.78)

In fact it is typically found that the GARCH (1,1) model yields an adequate description of many financial time series data , see, for example , [2] Bollerslev,Chou, and Kroner (1992). A data set which requires a model of order greater than GARCH (1, 2) or GARCH (2, 1) is very rare.

A series of size N=300 is generated from the simple BL-GARCH model

$$\varepsilon_t = z_t h_t,\ h_t^2 = \alpha_0 + (\alpha_1 \varepsilon_{t-1} + \beta_1 h_t)^2$$

With

$$\alpha_0 = 0.8, \quad \alpha_1 = 0.5, \quad \beta_1 = 0.4$$

The series $\{z_t\}$ is a sequence of i.i.d. N(0, 1). The initial values are chosen as $h_1 = 1$and $\varepsilon_1 = z_1 h_1$. The graph of the series $\{\varepsilon_t\}$ and $\{h_t\}$ are presented in figures (1) and (2) respectively

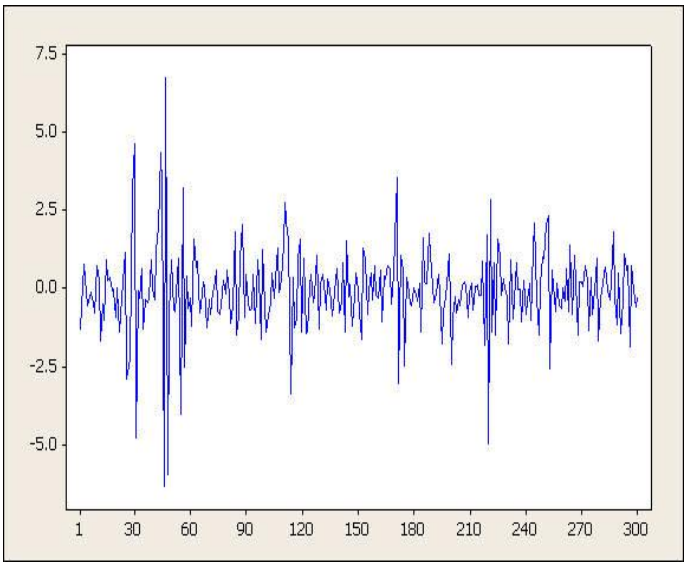


Figure (1)

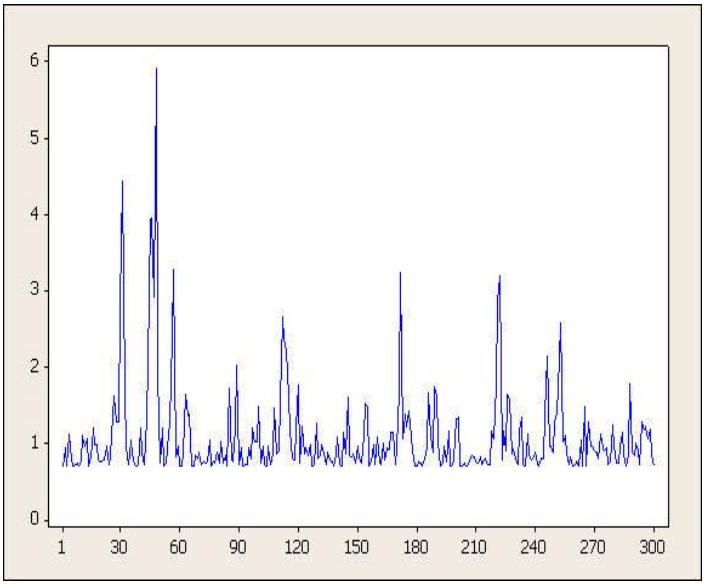


Figure (2)

3. MLE of BL-GARCH Parameters

We now consider the maximum likelihood estimation of the parameters in the BL-GARCH model (8).

Let $\underline{\theta} = (\alpha_0\ \alpha_1 \ldots \alpha_p\ \beta_1 \ldots \beta_p)'$ and suppose that we have the observations $\mathbf{X}_{-m+1}, \ldots, \mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_n$ for the time series $\{x_t\}$. Under a reasonable assumption that we have known the σ-field $\sigma\{\varepsilon_{-m+1}, \ldots, \varepsilon_{-1}, \varepsilon_0\}$, we can obtain the joint conditional density function of $x_1, \ldots, x_n$ given the **σ**-field

$$\sigma\{x_{-m+1}, \ldots, x_0,\ \varepsilon_{-m+1}, \ldots, \varepsilon_{-1}, \varepsilon_0\}$$

as follows

$$\begin{aligned} & f(x_1, \ldots, x_n / x_0, \ldots, x_{-m+1}, \varepsilon_0, \ldots, \varepsilon_{-m+1}) \\ &= \\ & f(x_2, \ldots, x_n / x_1,\ x_0, \ldots, x_{-m+1}, \varepsilon_0, \ldots, \varepsilon_{-m+1}) \\ & f(x_1 / x_0, \ldots, x_{-m+1}, \varepsilon_0, \ldots, \varepsilon_{-m+1}) \\ & \qquad \ldots\ldots\ldots \\ &= \prod_{t=1}^{n} f(x_i / x_{t-1}, \ldots, x_{-m+1}, \varepsilon_{t-1}, \ldots, \varepsilon_{-m+1}) \\ &= \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi}\, h_t} \exp\left\{ -\frac{\varepsilon_t^2}{2h_t^2} \right\} \end{aligned}$$

Thus the MLE $\hat{\underline{\theta}}$ of the parameter vector $\underline{\theta}$ is the value of $\underline{\theta}$ which maximizes the logarithm likelihood function

$$\begin{aligned} L(\underline{\theta}) &= \ln\ f(x_1, \ldots, x_n / x_0, \ldots, x_{-m+1}, \varepsilon_0, \ldots, \varepsilon_{-m+1}) \\ &= \sum_{t=1}^{n} \left\{ -\ln(h_t) - \frac{\varepsilon_t^2}{2h_t^2} \right\} - \frac{n}{2} \ln(2\pi) \\ &= \sum_{t=1}^{n} Q_t(\underline{\theta}) + C \end{aligned}$$

Using the recursive Newton-Raphson iteration algorithm, the MLE $\hat{\underline{\theta}}$ can be obtained by the following iteration:

$$\underline{\theta}^{(k+1)} = \underline{\theta}^{(k)} - H^{-1}(\underline{\theta}^{(k)}) G(\underline{\theta}^{(k)})$$

where $\underline{\theta}^{(k)}$ is the set of estimates obtained at the k^{th} stage of iteration. $G(\underline{\theta})$ is the gradient vector of partial derivatives,

$$G(\underline{\theta}) = \left[\frac{\partial L(\underline{\theta})}{\partial \theta_1} \ \ldots\ \frac{\partial L(\underline{\theta})}{\partial \theta_{2p+1}} \right]'$$

and $H(\underline{\theta})$ is the Hessian matrix of second order partial derivatives,

$$H(\underline{\theta}) = \left[\frac{\partial^2 L(\underline{\theta})}{\partial \theta_i\, \partial \theta_j} \right]$$

The first order partial derivatives are given by

$$\frac{\partial Q_t}{\partial \theta_i}=\frac{\partial Q_t}{\partial h_t}\frac{\partial h_t}{\partial \theta_i}=\left\{-\frac{1}{h_t}+\frac{\varepsilon_t^2}{h_t^3}\right\}\frac{\partial h_t}{\partial \theta_i}, \qquad i=1,2,\ldots\ldots,2p+1$$

where $\frac{\partial h_t}{\partial \theta_i}$ are calculated recursively from the equations:

$$\frac{\partial h_t}{\partial \alpha_0}=\frac{1}{2h_t}\left\{1+2\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial h_{t-i}}{\partial \alpha_0}\right\}$$

$$\frac{\partial h_t}{\partial \alpha_k}=\frac{1}{h_t}\left\{(\alpha_k\varepsilon_{t-k}+\beta_k h_{t-k})\varepsilon_{t-k}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial h_{t-i}}{\partial \alpha_k}\right\} \quad k=1,2,\ldots,p$$

$$\frac{\partial h_t}{\partial \beta_k}=\frac{1}{h_t}\left\{(\alpha_k\varepsilon_{t-k}+\beta_k h_{t-k})h_{t-k}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial h_{t-i}}{\partial \beta_k}\right\} \quad k=1,2,\ldots,p$$

The second order derivatives are given by:

$$\frac{\partial^2 Q_t}{\partial \theta_i^2}=\left\{-\frac{1}{h_t}+\frac{\varepsilon_t^2}{h_t^3}\right\}\frac{\partial^2 h_t}{\partial \theta_i^2}+\left\{\frac{1}{h_t^2}-\frac{3\varepsilon_t^2}{h_t^4}\right\}\left(\frac{\partial h_t}{\partial \theta_i}\right)^2, \qquad i=1,2,\ldots\ldots,2p+1$$

where

$$\frac{\partial^2 h_t}{\partial \alpha_0^2}=\frac{1}{h_t}\left\{\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \alpha_0^2}+\sum_{i=1}^{p}\beta_i^2\left(\frac{\partial h_{t-i}}{\partial \alpha_0}\right)^2-\left(\frac{\partial h_t}{\partial \alpha_0}\right)^2\right\}$$

$$\frac{\partial^2 h_t}{\partial \alpha_0\partial \alpha_k}=\frac{1}{h_t}\left\{(\beta_i\varepsilon_{t-k})\frac{\partial h_{t-k}}{\partial \alpha_0}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \alpha_0\partial \alpha_k}+\sum_{i=1}^{p}\beta_i^2\frac{\partial h_{t-i}}{\partial \alpha_0}\frac{\partial h_{t-i}}{\partial \alpha_k}-\frac{\partial h_t}{\partial \alpha_0}\frac{\partial h_t}{\partial \alpha_k}\right\}$$

$$\frac{\partial^2 h_t}{\partial \alpha_0\partial \beta_k}=\frac{1}{h_t}\left\{\alpha_k\varepsilon_{t-k}+2\beta_k h_{t-k}\frac{\partial h_{t-k}}{\partial \alpha_0}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \alpha_0\partial \beta_k}+\sum_{i=1}^{p}\beta_i^2\frac{\partial h_{t-i}}{\partial \alpha_0}\frac{\partial h_{t-i}}{\partial \beta_k}-\frac{\partial h_t}{\partial \alpha_0}\frac{\partial h_t}{\partial \beta_k}\right\}$$

$$\frac{\partial^2 h_t}{\partial \alpha_k^2}=\frac{1}{h_t}\left\{\varepsilon_{t-k}^2+2\beta_k\varepsilon_{t-k}\frac{\partial h_{t-k}}{\partial \alpha_k}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \alpha_k^2}+\sum_{i=1}^{p}\beta_i^2\left(\frac{\partial h_{t-i}}{\partial \alpha_k}\right)^2-\left(\frac{\partial h_t}{\partial \alpha_k}\right)^2\right\}$$

$$\frac{\partial^2 h_t}{\partial \alpha_k\partial \alpha_m}=\frac{1}{h_t}\left\{\beta_k\varepsilon_{t-k}\frac{\partial h_{t-k}}{\partial \alpha_m}+\beta_m\varepsilon_{t-m}\frac{\partial h_{t-m}}{\partial \alpha_k}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \alpha_k\partial \alpha_m}+\sum_{i=1}^{p}\beta_i^2\frac{\partial h_{t-i}}{\partial \alpha_k}\frac{\partial h_{t-i}}{\partial \alpha_m}-\frac{\partial h_t}{\partial \alpha_k}\frac{\partial h_t}{\partial \alpha_m}\right\}$$

$$\frac{\partial^2 h_t}{\partial \alpha_k\partial \beta_m}=\frac{1}{h_t}\left\{\beta_k\varepsilon_{t-k}\frac{\partial h_{t-k}}{\partial \beta_m}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \alpha_k\partial \beta_m}+(\alpha_m\varepsilon_{t-m}+2\beta_m h_{t-m})\frac{\partial h_{t-m}}{\partial \alpha_k}+\sum_{i=1}^{p}\beta_i^2\frac{\partial h_{t-i}}{\partial \alpha_k}\frac{\partial h_{t-i}}{\partial \beta_m}-\frac{\partial h_t}{\partial \alpha_k}\frac{\partial h_t}{\partial \beta_m}\right\}$$

$$\frac{\partial^2 h_t}{\partial \beta_k^2}=\frac{1}{h_t}\left\{h_{t-k}^2+2(\alpha_k\varepsilon_{t-k}+2\beta_k h_{t-k})\frac{\partial h_{t-k}}{\partial \beta_k}+\sum_{i=1}^{p}\beta_i^2\left(\frac{\partial h_{t-i}}{\partial \beta_k}\right)^2+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \beta_k^2}-\left(\frac{\partial h_{t-i}}{\partial \beta_k}\right)^2\right\}$$

$$\frac{\partial^2 h_t}{\partial \beta_k\partial \beta_m}=\frac{1}{h_t}\left\{(\alpha_k\varepsilon_{t-k}+2\beta_k h_{t-k})\frac{\partial h_{t-k}}{\partial \beta_m}+\sum_{i=1}^{p}(\alpha_i\varepsilon_{t-i}+\beta_i h_{t-i})\beta_i\frac{\partial^2 h_{t-i}}{\partial \beta_k\partial \beta_m}+(\alpha_m\varepsilon_{t-m}+2\beta_m h_{t-m})\frac{\partial h_{t-m}}{\partial \beta_k}+\sum_{i=1}^{p}\beta_i^2\frac{\partial h_{t-i}}{\partial \beta_k}\frac{\partial h_{t-i}}{\partial \beta_m}-\frac{\partial h_t}{\partial \beta_k}\frac{\partial h_t}{\partial \beta_m}\right\}$$

Note that the estimated the Hessian matrix $\hat{\mathbf{H}}(\underline{\theta})$ may be singular and some numerical problems may arise. One common way to deal with this problem is the Levenberg-Marquardt procedure [9] (Marquardt(1963)).

4. Monte Carlo Simulation

The Newton-Raphson with Marquordt algorithm, described in the previous section were tried successfully on many sets of data simulated from several stationary BL-GARCH models. We shall consider here the following model

$$\varepsilon_t=\sigma_t z_t, \quad \sigma_t^2=h_t^2=\alpha_0+(\alpha_1\varepsilon_{t-1}+\beta_1 h_{t-1})$$

with $\alpha_0 = 0.8,\ \alpha_1 = 0.5$ and $\beta_1 = 0.4$. The series $\{z_t\}$ is a sequence of i.i.d. N(0, 1). The initial values are chosen as $h_1 = 1$ and $\varepsilon_1 = z_1 h_1$. The Newton-Raphson algorithm is applied at the above model with sample size N=300 and replicate simulations 100 times. The results from the Monte-Carlo study shows, clearly, that the mean of each parameter estimates is close the true value. The standard deviations of the estimates are small indicating that the estimators are consistent.

Parameter estimates	N=100		
	α_0	α_1	β_1
True value	0.8	0.5	0.4
Mean	0.809	0.481	0.376
S.D.	0.087	0.092	0.095

References

[1] Bollerslev, T. (1986) Generalized autoregressive conditional heteroskedasticity. J. Econometrics 31, 307–327.

[2] Bollerslev,Chou, and Kroner (1992) Arch Modeling in Finance: A Review of the Theory and Empirical Evidence. Journal of Econometrics, April-May 1992, pp. 5-59.

[3] Engle, R.F. (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 50, 987–1008.

[4] Engle , R.F. and V.Ng (1993) Measuring and testing the Impact of News on Volatility. Journal of Finance 48, 1749-1778.

[5] Gabr, M.M. (1992) Recursive estimation of Bilinear Time Series Models". Commun. Statist.: Theory & Meth., 21(8), 2261-2277.

[6] Giraitis, L. and Surgailis, D. (2002) ARCH-type bilinear models with double long memory. Stoch. Process. Appl. 100, 275–300.

[7] Granger, C.W.J. and Andersen, A.P. (1978) An Introduction to bilinear time series Models. Gottengen: Vendenhoek and Ruprechet.

[8] Kokoszka, P. and Leipus, R. (2000) Change-point estimation in ARCH models. Bernoulli 6, 513–539.

[9] Maquardt, D. (1963) "An algorithm for least squares estimation of nonlinear parameters". *J. Soc. Ind. Appl. Aath., pp 431-441*.

[10] Nelson, D. B. and Cao, C. Q. (1992) Inequality constraints in the univariate GARCH model. Journal of Business & Economic Statistics, 10, 229–235.

[11] Pham, D.T. (1993) Bilinear Times Series Models. In Dimension Estimation and od-els (ed. H. Tong), 191-223. Singapore: World Scienti.c Publishing Co.

[12] Robinson, P. M. (1991). Testing for strong serial correlation and dynamic conditional heteroskedasticity in multiple regressions. Journal of Econometrics, 47,67–84.

[13] Storti and Vitale (2003). BL-GARCH models and asymmetries volatility. Statistical methods & applications 12:19-39

[14] Subba Rao, T. (1981) On the theory of Bilinear Models. J. Roy. Statis. Soc. B, (43), 244-255.

[15] Subba Rao, T. and Gabr, M. M. (1984) An Introduction to Bispectral Analysis and Bilinear Time Serie[14] Models. Lecture Notes in Statistics, volume 24. Springer Verlag, New York.

Computational Simulation of Backward Facing Step Flow Using Immersed Boundary Method

S. Jayaraj[1], A. Shaija, and C.A. Saleel
Mechanical Engineering Department, National Institute of Technology, Calicut-673601, India

Abstract- *The present numerical method is based on a finite volume approach on a staggered grid together with a fractional step approach. Backward facing step is treated as an immersed boundary and both momentum forcing and mass source terms are applied on the step to satisfy the no-slip boundary condition and also to satisfy the continuity for the mesh containing the immersed boundary. In the immersed boundary method, the necessity of an accurate interpolation scheme satisfying the no-slip condition on the immersed boundary is important, because the grid lines generally do not coincide with the immersed boundary. The numerically obtained velocity profiles, and stream line plots in the channel with backward facing step shows excellent agreement with the published results in various literatures. Results are presented for different Reynolds numbers with respect to channel length and height.*

Keywords: IBM, Momentum Forcing, Mass Source/ Sink.

1 Introduction

Numerical simulations are now recognized to be a part of the computer-aided engineering (CAE) spectrum of tools used extensively today in all industries, and its approach to modeling fluid flow phenomena allows equipment designers and technical analysts to have the power of a virtual wind tunnel on their desktop computer. Numerical simulation software has evolved far beyond what Navier, Stokes or Da Vinci could ever have imagined. It has become an indispensable part of the aerodynamic and hydrodynamic design process for planes, trains, automobiles, rockets, ships, submarines, MEMS, Lab-on-Chip (LOC) devices and so on; and indeed any moving craft or manufacturing process that mankind has devised so far. The advantage of numerical simulation with respect to experimentation is conceptually tabulated in Table 1.

The ability to handle complex geometries has been one of the main issues in computational simulations because most engineering problems have complex geometries. So far, two different approaches for simulating complex flow have been developed: the unstructured grid method and the immersed-boundary method (IBM). In this paper, numerical simulation of backward facing step flow problem is being performed using IBM, an alternative CFD simulation technique. It is an approach to model and simulate mechanical systems in which elastic structures (or membranes) interact with fluid flows. Treating the coupling of the structure deformations and the fluid flow poses a number of challenging problems for numerical simulations. In the immersed boundary method approach the fluid is represented in an Eulerian coordinate frame and the structures in a Lagrangian coordinate frame.

1.1 Immersed Boundary Method

The term "immersed boundary method" (now known in abbreviated form as 'IBM') was first used in reference to a method developed by Peskin in 1972 [1]. Originally this method was used to simulate cardiac mechanics and associated blood flow. The distinguished feature of this method was that, the entire simulation was carried out on a Cartesian grid, which did not conform to the geometry of the heart. Hence, a novel procedure was simulated for imposing the effect of the immersed boundary (IB) on the flow. That is, imposing the boundary conditions is not straight forward in IBM. Since Peskin introduced this method, numerous modifications and refinements have been proposed and a number of variants of this approach now exist. The main advantages of the IBM include memory and CPU time savings. Also easy grid generation is possible with IBM compared to the unstructured grid method. Even moving boundary problems can be handled using IBM without regenerating grids in time, unlike the structured grid method.

Table 1. Comparison of Numerical Simulation and Experimentation

Parameter	Numerical Simulation	Experimentation
Cost	Cheap	Expensive
Time	Short	Long
Scale	Any	Small/Middle
Information	All	Measured Points
Repeatability	All	Some
Security	Safe	Some Dangerous

It is to be noted that generating body conformal structured or unstructured grid is usually very cumbersome. Imposition of

[1]Presenting Author
Tel: +919447312732; Email: sjayaraj@nitc.ac.in

boundary conditions on the IB is the key factor in developing an IB algorithm and distinguishes one IB method from another. In the former approach, which is termed as "continuous forcing approach", the forcing function is incorporated in to the continuous equations before discretization, where as in the latter approach, which can be termed the "discrete forcing approach", the forcing function is introduced after the equations are discretized. An attractive feature of the continuous forcing approach is that it is formulated independent of the underlying spatial discretization. On the other hand, the discrete forcing approach very much depends on the discretization method. However, this allows direct control over the numerical accuracy, stability, and discrete conservation properties of the solver.

A review about Immersed Boundary Methods (IBM) encompassing all variants is cited by Mittal and Iaccarino [2]. The Immersed Boundary Finite Volume Method [3] used to simulate the present problem (i.e., to simulate the backward facing step flow problem) is based on a finite volume approach on a staggered mesh together with a fractional step method. The backward facing step is treated as an immersed boundary (IB). Both momentum forcing and mass source are applied on the body surface or inside the body to suit the no-slip boundary condition on the immersed boundary and also to satisfy the continuity for the cell containing the immersed boundary. In the immersed boundary method, the choice of an accurate interpolation scheme satisfying the no-slip condition on the IB is important because the grid lines generally do not concur with the IB. Therefore, a stable second order interpolation scheme for evaluating the momentum forcing on the body surface is also used.

1.2 Backward Facing Step Flows

The study of backward-facing step flows constitutes an important branch of fundamental fluid mechanics. Flow geometry of the same is very significant for investigating separated flows. This flow is of particular interest because it facilitates the study of the reattachment process by minimizing the effect of the separation process, while for other separating and reattaching flow geometries there may be a stronger interaction between the two. The principal flow features of the backward facing step flow are illustrated in Figure 1[4].

The phenomenon of flow separation is a problem of great importance for fundamental and industrial reasons. For instance it often corresponds to drastic losses in aerodynamic performances of airfoils or automotive vehicles. The backward-facing step is an extreme example of separated flows that occur in aerodynamic devices such as high-lift airfoils at large angles of attack. In these flows separation may be created by a strong adverse pressure gradient rather than a geometric perturbation, but the flow topology is similar. It is important in heat exchangers and gas turbines also. Since the location of the reattachment zone and its flow structure also determine the local heat and mass transport properties of the flow. This geometry has been received attention for half a century. Many researchers considered different aspects of this geometry from the flow pattern point of view and heat transfer. In some numerical simulations the backward facing step flow problem is a benchmark for validating the computational simulation algorithm.

The research in such a flow was intensified with the experimental and numerical work of Armaly *et al*. [5]. They presented a detailed experimental investigation in backward-facing step geometry for an expansion ratio (*H*/*h)* of 1.9423, an aspect ratio (*W*/*h)* of 35 and Reynolds numbers (Re_D) up to 8000. Here D=2h denotes the hydraulic diameter of the inlet channel with height *h*, *H* the channel height in the expanded region and *W* the channel width. When Reynolds number exceeds 400; it has been noticed that the flow appeared to be three-dimensional, a discrepancy in the primary recirculation length between the experimental results and the numerical predictions and a secondary recirculation zone was observed at the channel upper wall. Armaly *et al*. [5] conjectured that the discrepancy between the experimental measurements and the numerical prediction was due to the secondary recirculation zone that perturbed the two-dimensional character of the flow. The normalized value of the reattachment length showed a peak at Re_D=1,200. The decrease in recirculation length beyond a Reynolds number of 1,200 was attributed to the effect of Reynolds stresses.

Kim and Moin [6] numerically simulated the flow over a backward-facing step using a method that is second-order accurate in both space and time. Their results are (variation of the reattachment length on Reynolds number) in good agreement with the experimental data of Armaly *et al*. [5] up to about Re_D = 500. At Re_D = 600 the computed results of started to deviate from the experimental values. The discrepancy was due to the three-dimensionality of the experimental flow around a Reynolds number of 600.

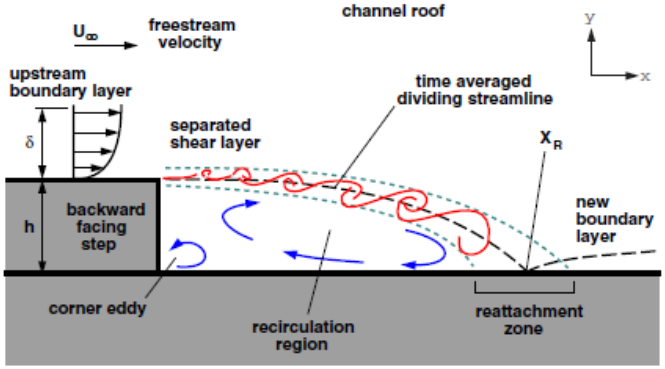


Fig.1 Detailed flow features of the backward facing step flow

The bifurcation of two-dimensional laminar flow to three-dimensional flow was identified by Kaiktsis *et al*. [7]. This is the primary source of discrepancies appearing in comparisons of numerical predictions and experimental data. From their valuable work, it has also been observed that irrespective of

the accuracy of the numerical schemes, the experimentally measured recirculation lengths (Armaly *et al*. [5] were consistently underestimated above a Reynolds number of Re_D =5600. They found that all unsteady states of the flow are three-dimensional and develop for Reynolds number $Re_D > Re_c$ =700. Furthermore, they detected that the downstream flow region is excited through the upstream shear layer with a characteristic frequency f_1. The supercritical states (Re_D > 700) were found to be periodic with another incommensurate frequency, f_2.

Kaiktsis *et al*. [8] revisited the backward-facing step flow and found that the unsteadiness in step flow was created by convective instabilities. Another important conclusion of this study is that the upstream-generated small disturbances propagate downstream at exponentially amplified amplitude with a space-dependent speed in the range $700<Re_D<2500$.

Heenan and Morrison [9] conducted experiments for a Reynolds number (Re_S) based on the step height *S* of $1.9X10^5$ and suggested that while the flow is likely to be convectively unstable over a large region, the global unsteadiness, driven by the impingement of large eddies at reattachment is the cause of low frequency oscillations called *flapping*.

Erturk *et al*.[10] have have presented a new, efficient and stable numerical method for the solution of stream function and vorticity equations. With this method they have presented steady solutions of driven cavity flow at very high Reynolds numbers (up to Re=21,000) using very fine grid mesh. They have analysed the nature of the cavity flow at high Reynolds numbers.

2 Problem Specification

To explain the concept of immersed boundary method, consider the simulation of flow past a solid body shown in Fig. 2*a*. The body occupies the volume Ω_b with boundary Γ_b. The body has a characteristic length scale *L*, and a boundary layer of thickness δ develops over the body.

The conventional approach to this would employ structured or unstructured grids that conform to the body. Generating these grids proceeds in two sequential steps. First, a surface grid covering the boundaries Γ_b *is* generated. This is then used as a boundary condition to generate a grid in the volume Ω_f occupied by the fluid. If a finite-difference method is employed on a structured grid, then the differential form of the governing equations is transformed to a curvilinear coordinate system aligned with the grid lines [11]. Because the grid conforms to the surface of the body, the transformed equations can then be discretized in the computational domain with relative ease. If a finite-volume technique is employed, then the integral form of the governing equations is discretized and the geometrical information regarding the grid is incorporated directly into the discretization. If an unstructured grid is employed, then either a finite-volume or a finite-element methodology can be used. Both approaches incorporate the local cell geometry into the discretization and do not resort to grid transformations.

Now consider employing a non body conformal Cartesian grid for this simulation, as shown in Figure 2*b*. In this approach the immersed boundary (IB) would still be represented through some means such as a surface grid, but the Cartesian volume grid would be generated with no regard to this surface grid. Thus, the solid boundary would cut through this Cartesian volume grid. Because the grid does not conform to the solid boundary, incorporating the boundary conditions would require modifying the equations in the vicinity of the boundary. Precisely what these modifications are is the subject matter of IBM. However, assuming that such a procedure is available, the governing equations would then be discretized using a finite-difference, finite-volume, or a finite-element technique without resorting to coordinate transformation or complex discretization operators.

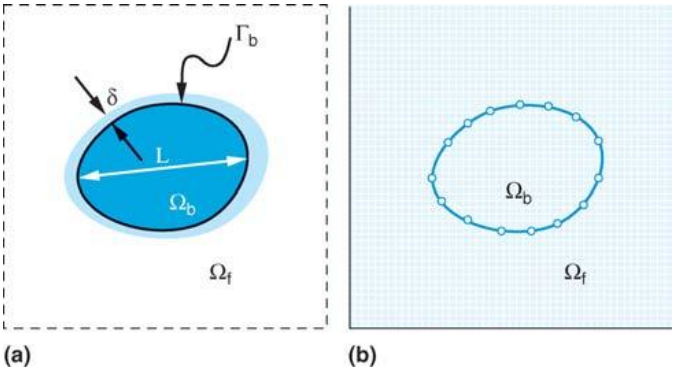


Fig. 2 (*a*) Schematic showing a generic body past which flow is to be simulated. (*b*) Schematic of body immersed in a Cartesian grid on which the governing equations are discretized.

2.1 Governing Equations

The governing equations for unsteady incompressible viscous flow between parallel plates are

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \qquad (1)$$

$$\frac{\partial u_i}{\partial x_i} - q = 0 \qquad (2)$$

where x_i are the Cartesian coordinates, u_i are the corresponding velocity components, p is the pressure, f_i s are the momentum forcing components defined at the cell faces on the immersed boundary or inside the body, and q is the mass source/sink defined at the cell center on the immersed boundary or inside the body. All the variables are non-dimensionalized by the bulk (average) velocity of the inlet flow, U_b and length scales, by H (channel height at the downstream), and the only dimensionless number appearing in the governing equations is the Reynolds number. For the flow problem considered, the following definition is used for the Reynolds number, Re.

$$\mathrm{Re} = \frac{\rho U_b H}{\mu} \quad (3)$$

Where ρ and μ are the density and the dynamic viscosity, respectively

2.2 Geometry of Flow Domain and Boundary Conditions

Figure 3 depicts the two-dimensional channel with a backward facing step with finite distance in between the channel, which is small compared to its length and width. Hence the flow through this channel is assumed to be two dimensional. In addition, the flow is assumed as steady and laminar. Buoyant forces are negligible compared with viscous and pressure forces.

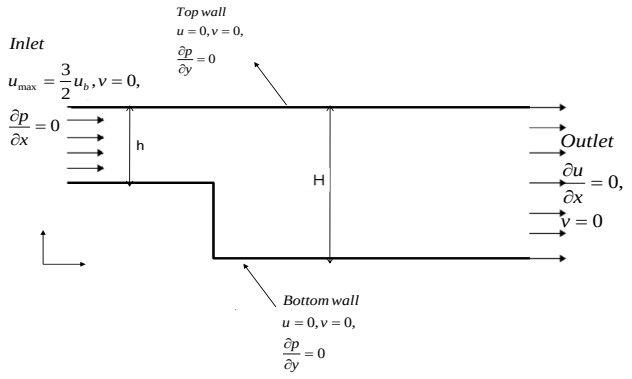


Fig.3. Sketch of the flow configuration and definition of length scales.

Inlet: In order to simulate a fully developed laminar channel flow upstream of the step and to eliminate the corner effects, a standard parabolic velocity profile with a maximum velocity $U_{max}=(3/2)U_b$ is prescribed at the channel inlet for the present model. Cross stream velocity is equal to zero. The Neumann boundary condition is assumed for pressure.

Outlet: Fully developed velocity profile is assumed at the outlet. Pressure boundary condition is not specified.

Walls: No slip condition (u=0 and v=0) for velocity and Neumann boundary condition for pressure.

To ease the comparison of the results obtained by the numerical simulation using IBM, the geometry of the flow problem was chosen in accordance to the experimental setup of Armaly *et al*. [5]. The expansion ratio is defined by $\frac{H}{h} = 1 + \frac{S}{h}$,

i.e., by the ratio of the channel height *H* downstream of the step to the channel height *h* of the inflow channel, where *S* denotes the step height. The results are generated for an expansion ratio of 1.9423. This expansion ratio was considered in the experimental study by Armaly *et al* [5] and the same value has been used for a set of numerical computations at the Reynolds numbers 0.0001, 1,100 to compare the results with Biswas *et al.* [14] results which is in turn agreeing with the Armaly *et al* [5]. An incompressible Newtonian fluid with constant fluid properties is assumed.

3 Solution Procedure

For the spatial discretization of Equations (1) and (2) an immersed-boundary method (IBM) based on finite volume approach on a staggered grid together with a fractional step method was employed. Being a CFD method, the finite volume method (FVM) describes mass, momentum and energy conservation for solution of the set of differential equations considered. The approximated equations for the FVM can be obtained by two approaches. The first consists in applying balances for the elementary volumes (finite volumes), and the second consists in the integration spatial-temporal of the conservation equations. In this work, the latter approach is followed.

The momentum forcing and the mass source/sink are applied on the body surface or inside the body to satisfy the no-slip boundary condition on the immersed boundary (step) and the continuity for the cell containing the immersed boundary, respectively. A linear interpolation scheme is used to satisfy the no-slip velocity on the immersed boundary, which is numerically stable regardless of the relative position between the grid and the immersed boundary.

The time-integration method used to solve the above equations is based on a factional step method where a pseudo-pressure is used to correct the velocity field so that the continuity equation is satisfied at each computational time step. In this study, a second-order semi-implicit time advancement scheme (a third order Runge-Kutta method (RK3) for the convection terms and a second order Crank-Nicholson method for the diffusion terms).

The convection and diffusion terms were evaluated using a central differencing scheme of second-order accuracy. Solution of non-dimensional u and v are made possible in powerful and accurate TDMA (Tri-diagonal Matrix Algorithm) with ADI (Alternating Direction Implicit) approximate factorization method. The pressure solver is SOR (Successive Over Relaxation) method. The numerical code is developed using Digital Visual FORTRAN (DVF) and a detailed flow chart is shown in Figure 4 which leads to the development of code.

4. Results and Discussions

In order to ensure whether the predicted results are grid independent, extensive refinement studies were carried out. Finally, the non-dimensional stream wise velocity at the centre of the channel outlet for Re=1.0 is tabulated in Table 2. It is seen that for the computational stencil of 252x102, percentage

change with respect to previous stencil is least. Hence the same stencil is being selected for the code execution

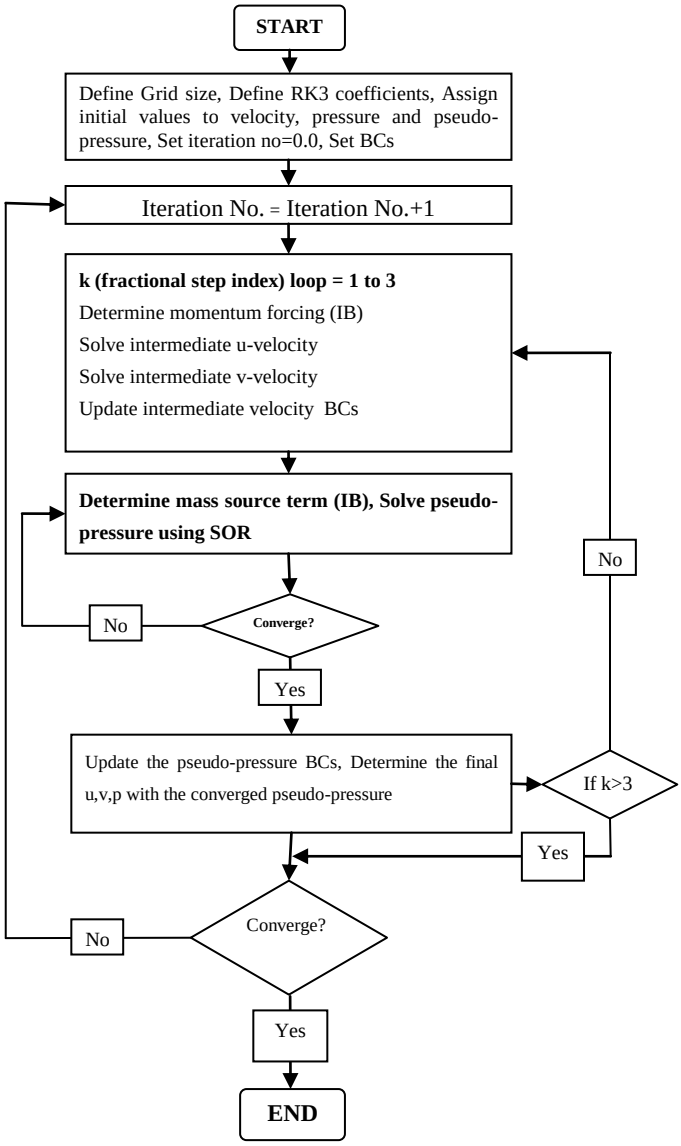


Fig.4 Flow chart for the Immersed Boundary Method

It has been observed that at low Reynolds numbers the flow separates at the sharp corner and then reattaches itself to the lower boundary further downstream forming a single primary re-circulating eddy. The reattachment length increases almost linearly with Reynolds number, the slight non-linear trend being attributed to viscous drag along the upper boundary. Computed non-dimensionalised reattachment lengths against inlet Reynolds number are shown in Table 3, to compare the same with the results of Biswas *et al.* [12].

The determination of the separation and reattachment locations thus offers a severe bench-mark test for any hydrodynamic model because of the highly non-linear flow kinematics in the vicinity of the step. It is evident from plots and stream lines that as the Reynolds number increases there is a backward flow occurring at the step, which is result of the negative pressure developed due to separation occurring at high velocity due to high Reynolds number.

Table 2. Maximum non-dimensional stream wise velocity at the centre of the channel for different number of grids in horizontal and vertical directions at Re=1.0

Maximum non-dimensional stream wise velocity at the channel exit	**Number of grids in stream wise direction**	**Number of grids in cross stream direction**
0.681687	27	7
0.763075	52	22
0.765939	102	42
0.766465	152	62
0.766649	202	82
0.766798	252	102

Figures 5-9 show the stream wise velocity contours and cross stream velocity contours for the Reynolds number range $10^{-4}<Re<10^{2}$. It is being observed that the maximum velocity is at upstream side of the channel. A vortex is also visible at the concave corner behind the step. Stream wise velocity is being fully developed far downstream of the channel. It is being noted that immediately after the concave vortex, the fluid adjacent to the walls decelerates due to the formation of the two hydrodynamic boundary layers and backward pressure. Consequently, as a result of continuity principle, fluid outside these two boundary-layers accelerates. Due to this action, a transverse velocity component is engendered, which is clearly visible from the cross stream velocity contour, that sends the fluid away from the two plates outside the two boundary-layers and towards the centerline between the two walls. However, this action gradually decays with further increase in the axial distance downstream the entrance and finally vanishes when the flow becomes hydro dynamically fully developed.

Figures 10 and 11 show streamlines of the steady state flow field for an expansion ratio H/h=1.9423 and a Reynolds numbers range 10^{-4} and 10^{2}. The plots well agree with literature especially commensurate with the experiments of Armaly *et al.* [5] which reveals that flow over the backward-facing step is purely two dimensional and non-oscillatory in the considered region.

The streamline patterns for Re $=10^{-4}$ depict that the flow follows the upper convex corner without revealing a flow separation. Furthermore, a corner vortex is found in the concave corner behind the step. In this range of very small Reynolds numbers (10^{-4}), the size of this vortical structure is nearly constant varying between x_1/h=0.3491(for Re=10^{-4}) and 0.3647(for Re=1), where x_1 referred to as reattachment

length. Under these conditions, the effect of inertia forces can be assumed to be negligible compared with viscous forces often denoted as molecular transport. Hence the flow resembles the Stokes flow.

Table 3. Comparison of the results

Reynolds Number	Size of the corner vortex (x_1/H)	Size of the corner vortex (x_1 / h)	Size of the corner vortex (x_1 / h)
	Present work		Biswas *et al.*[12]
0.0001	0.180	0.3491	0.350
1.0	0.188	0.3647	0.365
100	1.45	2.8128	2.8

The validation of the numerical model with respect to backward-facing step flow problem, which is one of the most fundamental geometries causing flow separation and has been extensively investigated in both the laboratory and as a standard 'bench-mark' test for numerical simulations, ascertain that IBM is a successful alternative CFD technique. This ensures a test of the stability and accuracy of the present algorithms.

5 Conclusions

Immersed-boundary method is adopted to validate a relevant fluid mechanics bench mark problem, the backward facing step flow problem. The present algorithm is ideally suited to low Reynolds number flows also. Predictions from the numerical model have been compared against experimental data of different Reynolds numbers of flow past backward-facing step geometries. In addition, computed reattachment and separation lengths have been compared against alternative numerical predictions. The immersed boundary method with both the momentum forcing and mass source/sink is found to gives realistic velocity profiles and reattachment lengths downstream of the step demonstrating the accuracy of the method.

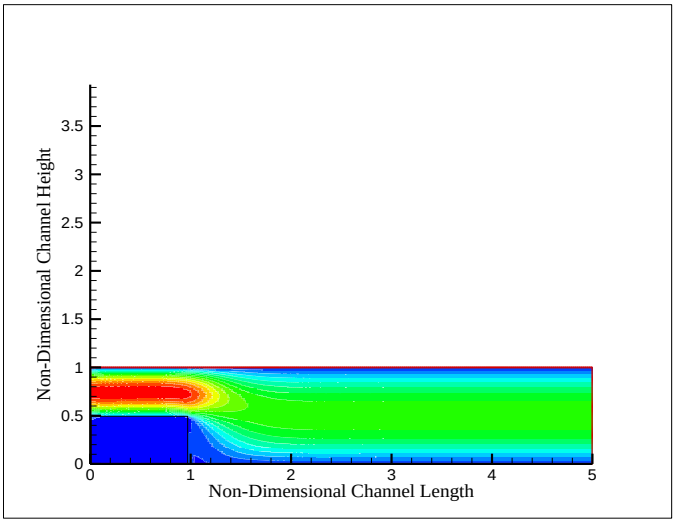


Fig.5 Stream wise Velocity contours for Re=0.0001

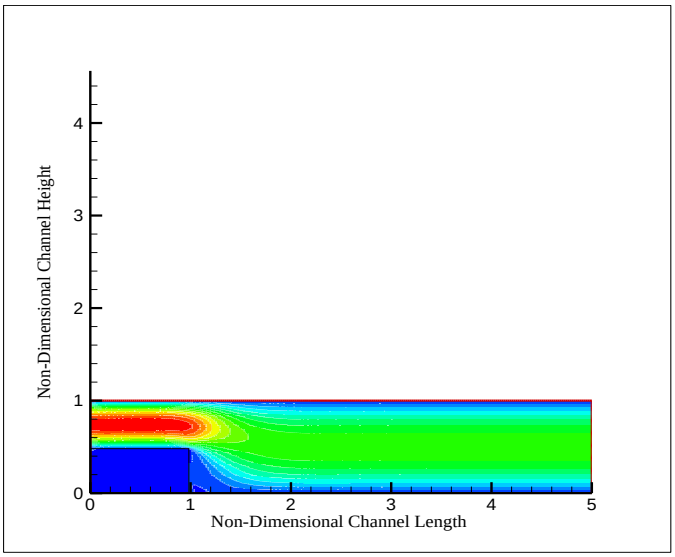


Fig.6 Stream wise Velocity contours for Re=1.0

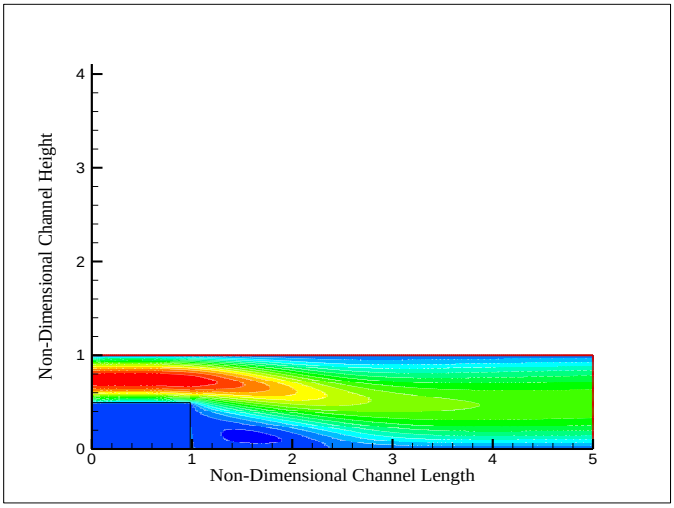


Fig.7 Stream wise Velocity contours for Re=100.0

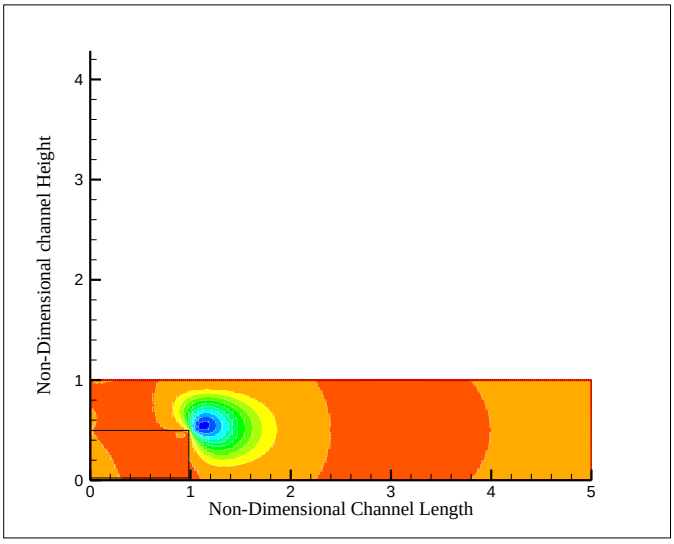


Fig.8 Cross Stream Velocity contours for Re=0.0001

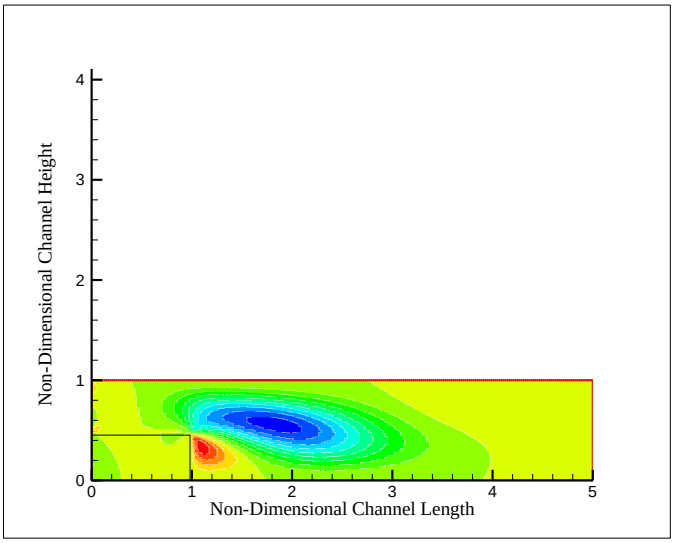


Fig.9 Cross Stream Velocity contours for Re=100.0

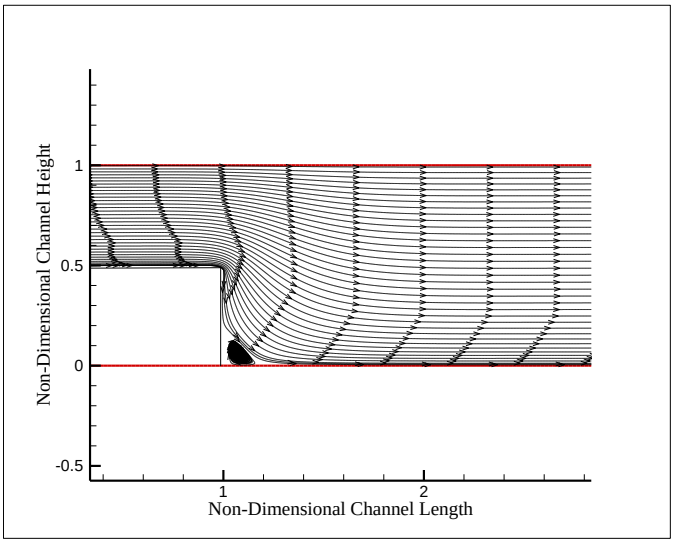


Fig.10 Streamlines in the vicinity of the step for Re=0.0001

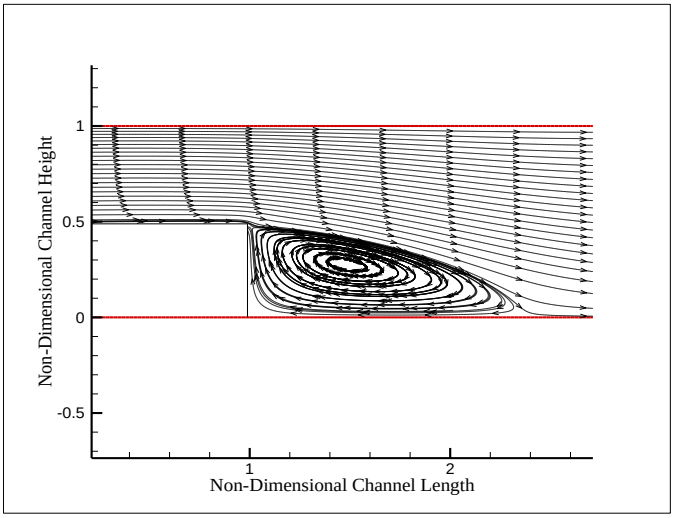


Fig.11 Streamlines in the vicinity of the step for Re=100.0

6 References

[1] Peskin C.S., "*Flow patterns around heart valves: a numerical method*" J. of Computational Physics, **10**, pp 252-271, 1972

[2] Mittal R. and Iaccarino G. "*Immersed Boundary Methods*" in Annual Review of Fluid Mechanics, **37**, pp. 239-261, 2005.

[3] Kim J., Kim D., and Choi H., "*An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries*" J. of Computational Physics **171,** 132–150, 2001

[4] Kostas, J., Soria,J.,Chong M/ S, "*A study of backward facing step flow at two Reynolds numbers"*, 14th Australian Fluid Mechanics Conference, Adelaide University, Adelaide, Australia, 10-14 December 2001

[5] Armaly, B. F., Durst, F., Peireira, J. C. F., Scho¨nung, B., 1983, "*Experimental and theoretical investigation of backward-facing step flow"*, J. Fluid Mech., **127**, pp. 473–496.

[6] Kim, J., and Moin, P., 1985, "*Application of a fractional-step method to incompressible Navier-Stokes equations"*, J. Comput. Phys., **59**, pp. 308–323.

[7] Kaiktsis, L., Karniadakis, G. E., and Orszag, S. A., 1991, "*Onset of three dimensionality ,equilibria, and early transition in flow over a backward-facing step"*, J. Fluid Mech., **231**, pp. 501–528.

[8] Kaiktsis, L., Karniadakis, G. E., and Orszag, S. A., 1996, "*Unsteadiness and convective instabilities in a two-dimensional flow over a backward-facing step"*, J. Fluid Mech., **321**, pp. 157–187.

[9] Heenan, A. F. and Morrison, J. F., 1998, "*Passive control of back step flow"*, Exp.Therm. Fluid Sci., **16**, pp. 122–132.

[10] Erturk, E., Corke, T.C. and Gokcol, C., 2005, "*Numerical Solutions of 2-D Steady Incompressible Driven Cavity Flow at High Reynolds Numbers*", Int. J. for Numerical Methods in Fluids, **48**, pp 747-774

[11] Ferziger J.H. and Peric M. 1996. "*Computational Methods in Fluid Dynamics",* Springer-Verlag, New York

[12] Biswas, G., Breuer, M. and Durst, F., 2004, "*Backward-Facing Step Flows for Various Expansion Ratios at Low and Moderate Reynolds Numbers"*, J. Fluid Engg., **126**, pp. 362–374.

SESSION

NUMERICAL METHODS + APPROXIMATION AND ESTIMATION TECHNIQUES + SOFTWARE TOOLS AND SYSTEMS + OPTIMIZATION METHODS

Chair(s)

TBA

Optimization of New-Sample and Within-Sample Prediction Intervals for Order Statistics

N. A. Nechval[1], K. N. Nechval[2], V. Danovich[1], and T. Liepins[1]
[1]Statistics Department, EVF Research Institute, University of Latvia, Riga, Latvia
[2]Applied Mathematics Department, Transport and Telecommunication Institute, Riga, Latvia

Abstract - *Prediction intervals for order statistics are widely used for reliability problems and other related problems. The determination of these intervals has been extensively investigated. But the optimality property of these intervals has not been fully explored. In this paper, in order to discuss this problem, a risk function is introduced to compare prediction intervals. In particular, new-sample prediction based on a previous sample (i.e., when for predicting the future observation in a new sample there are available the data only from a previous sample), and within-sample prediction based on the early observed data from a current experiment (i.e., when for predicting the future observation in a sample there are available the early observed data only from that sample). We restrict attention to families of distributions invariant under location and/or scale changes. The technique used here for optimization of prediction intervals based on censored data emphasizes pivotal quantities relevant for obtaining ancillary statistics. It allows one to solve the optimization problems in a simple way. An illustrative example is given.*

Keywords: Order Statistic, Prediction Interval, Risk Function, Optimization

1 Introduction

Prediction of an unobserved random variable is a fundamental problem in statistics. Patel [1] provides an extensive survey of literature on this topic. In the areas of reliability and life-testing, lifetime data are often modeled via the Exponential and the Weibull in order to make predictions about future observations. Prediction intervals are constructed to have a reasonably high probability of containing a specified number of such future observations. These limits may be helpful in establishing warranty policy, determining maintenance schedules, etc. For a very readable discussion of prediction limits and related intervals, see Hahn and Meeker [2]. Many authors have reported their efforts for constructing prediction limits for the Weibull and for the related extreme value distributions (see Patel [1]). Mann and Saunders [3] proposed prediction limits for the Weibull which make use of only two or three order statistics (see also Mann [4]). Antle and Rademaker [5] used simulation to produce a table of factors to use with ML estimates to obtain prediction limits. Lawless [6] proposed prediction limits based on a conditional confidence approach; his limits require both determination of the ML estimates and numerical integration. Engelhardt and Bain [7-8] and Fertig, Meyer and Mann [9] have proposed various approximate prediction limits for the Weibull. Mee and Kushary [10] provided a simulation based procedure for constructing prediction intervals for Weibull populations for Type II censored case. This procedure is based on maximum likelihood estimation and requires an iterative process to determine the percentile points. Bhaumik and Gibbons [11] and Krishnamoorthy et al. [12] proposed approximate methods for constructing upper prediction limits for a gamma distribution. Consider the following examples of practical problems which often require the computation of prediction bounds and prediction intervals for future values of random quantities: (i) a consumer purchasing a refrigerator would like to have a lower bound for the failure time of the unit to be purchased (with less interest in distribution of the population of units purchased by other consumers); (ii) financial managers in manufacturing companies need upper prediction bounds on future warranty costs; (iii) when planning life tests, engineers may need to predict the number of failures that will occur by the end of the test to predict the amount of time that it will be take for a specified number of units to fail. Some applications require a two-sided prediction interval that will, with a specified high degree of confidence, contain the future random variable of interest. In many applications, however, interest is focused on either an upper prediction bound or a lower prediction bound (e.g., the maximum warranty cost is more important than the minimum, and the time of the early failures in a product population is more important than the last ones). Conceptually, it is useful to distinguish between 'new-sample' prediction and 'within-sample' prediction. For new-sample prediction, data from a past sample are used to make predictions on a future unit or sample of units from the same process or population. For example, based on previous (possibly censored) life test data, one could be interested in predicting the time to failure of a new unit, time until r failures in a future sample of m units, or number of failures by time $t^{\bullet}$ in a future sample of m units. For within-sample prediction, the problem is to predict future events in a sample or process based on early data from that sample or process. If, for example, n units are followed until $t_{\bullet}$ and there are k observable failures, $X_1 < X_2 < \cdots < X_k$, one could be interested in predicting the time of the next failure, $X_{(k+1)}$; time until l additional failures, $X_{(k+l)}$; number of additional failures in a

future interval ($t_\bullet$,$t^\bullet$). In general, to predict a future realization of a random quantity one needs the following:

1) *A statistical model to describe the population or process of interest*. This model usually consists of a distribution depending on a vector of parameters $\boldsymbol{\theta}$. In this paper, attention is restricted to families of distributions which are invariant under location and/or scale changes. In particular, the case may be considered where a previously available complete or type II censored sample is from a continuous distribution with cdf $F((x-\mu)/\sigma)$, where $F(\cdot)$ is known but both the location (μ) and scale (σ) parameters are unknown. For such family of distributions the decision problem remains invariant under a group of transformations (a subgroup of the full affine group) which takes μ (the location parameter) and σ (the scale) into $c\mu + b$ and $c\sigma$, respectively, where b lies in the range of μ, $c > 0$. This group acts transitively on the parameter space.

2) *Information on the values of components of the parametric vector* $\boldsymbol{\theta}$. It is assumed that only the functional form of the distribution is specified, but some or all of its parameters are unspecified. In such cases ancillary statistics and pivotal quantities, whose distribution does not depend on the unknown parameters, are used.

The technique used here for constructing prediction intervals (or bounds) emphasizes pivotal quantities relevant for obtaining ancillary statistics. It represents a simple procedure that can be utilized by non-statisticians, and which provides easily computable explicit expressions for both prediction bounds and prediction intervals. The technique is a special case of the method of invariant embedding of sample statistics into a performance index (see, e.g., Nechval et al. [13-18]) applicable whenever the statistical problem is invariant under a group of transformations, which acts transitively on the parameter space.

2 Within-sample prediction problem

For within-sample prediction, the problem is to predict future events in a sample or process based on early data from that sample or process. For example, if n units are followed until t_k and there are k observed failures, $t_1, \ldots, t_k$, one could be interested in predicting the time of the next failure t_{k+1}; time until l additional failures, t_{k+l}; number of additional failures in a future interval.

2.1 Location-scale family of density functions

Consider a situation described by a location-scale family of density functions, indexed by the vector parameter $\boldsymbol{\theta}=(\mu,\sigma)$, where μ and σ (>0) are respectively parameters of location and scale. For this family, invariant under the group G of positive linear transformations: $x \to ax+b$ with $a>0$, we shall assume that there is obtainable (from some informative experiment) the first k order statistics $X_1<X_2<\cdots<X_k$ from a random sample of size n with cumulative distribution function

$$F(x \mid \mu,\sigma) \equiv F\left(\frac{x-\mu}{\sigma}\right),$$

$$(-\infty)\ \mu < x < \infty, \quad -\infty < \mu < \infty, \quad \sigma > 0. \tag{1}$$

If Y is an independent future observation from the same sample of size n, then $W=(Y-X_k)/S_k$ (or $W=(Y-X_k)/X_k$) is an invariant statistic, the distribution of which does not depend on (μ,σ); S_k is a sufficient statistic (or a maximum likelihood estimator $\hat{\sigma}_k$) for σ based on $\mathbf{X}=(X_1, X_2, \ldots, X_k)$.

2.2 Piecewise-linear loss function

We shall consider the interval prediction problem for the rth order statistic X_r, $k<r\le n$, in the same sample of size n for the situation where the first k observations $X_1 < X_2 < \cdots < X_k$, $1\le k<n$, have been observed. Suppose that we assert that an interval $\mathbf{d}=(d_1,d_2)$ contains X_r. If, as is usually the case, the purpose of this interval statement is to convey useful information we incur penalties if d_1 lies above X_r or if d_2 falls below X_r. Suppose that these penalties are $c_1(d_1- X_r)$ and $c_2(X_r-d_2)$, losses proportional to the amounts by which X_r escapes the interval. Since c_1 and c_2 may be different the possibility of differential losses associated with the interval overshooting and undershooting the true μ is allowed. In addition to these losses there will be a cost attaching to the length of interval used. For example, it will be more difficult and more expensive to design or plan when the interval $\mathbf{d}=(d_1,d_2)$ is wide. Suppose that the cost associated with the interval is proportional to its length, say $c(d_2-d_1)$. In the specification of the loss function, σ is clearly a 'nuisance parameter' and no alteration to the basic decision problem is caused by multiplying all loss factors by $1/\sigma$. Thus we are led to investigate the piecewise-linear loss function

$$r(\boldsymbol{\theta},\mathbf{d}) = \begin{cases} \dfrac{c_1(d_1-X_r)}{\sigma} + \dfrac{c(d_2-d_1)}{\sigma} & (X_r < d_1), \\ \dfrac{c(d_2-d_1)}{\sigma} & (d_1 \le X_r \le d_2), \\ \dfrac{c(d_2-d_1)}{\sigma} + \dfrac{c_2(X_r-d_2)}{\sigma} & (X_r > d_2). \end{cases} \tag{2}$$

The decision problem specified by the informative experiment density function (1) and the loss function (2) is invariant under the group G of transformations. Thus, the problem is to find the best invariant interval predictor of X_r,

$$\mathbf{d}^* = \arg\min_{\mathbf{d}\in\mathscr{D}} R(\boldsymbol{\theta},\mathbf{d}), \tag{3}$$

where $\mathscr{D}$ is a set of invariant interval predictors of X_r, $R(\boldsymbol{\theta},\mathbf{d})=E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d})\}$ is a risk function.

2.3 Transformation of the loss function

It follows from (2) that the invariant loss function, $r(\boldsymbol{\theta},\mathbf{d})$, can be transformed as follows:

$$r(\boldsymbol{\theta},\mathbf{d}) = \ddot{r}(\mathbf{V},\boldsymbol{\eta}), \tag{4}$$

where

$$\ddot{r}(\mathbf{V},\boldsymbol{\eta})=\begin{cases} c_1(-V_1+\eta_1V_2)+c(\eta_2-\eta_1)V_2 & (V_1<\eta_1V_2),\\ c(\eta_2-\eta_1)V_2 & (\eta_1V_2\le V_1\le\eta_2V_2),\\ c_2(V_1-\eta_2V_2)+c(\eta_2-\eta_1)V_2 & (V_1>\eta_2V_2),\end{cases} \tag{5}$$

$$\mathbf{V}=(V_1,V_2),\quad V_1=(X_r-X_k)/\sigma,\quad V_2=S_k/\sigma;$$

$$\boldsymbol{\eta}=(\eta_1,\eta_2),\quad \eta_1=(d_1-X_k)/S_k,\quad \eta_2=(d_2-X_k)/S_k. \tag{6}$$

2.4 Risk function

It follows from (5) that the risk associated with **d** and **θ** can be expressed as

$$\begin{aligned} R(\boldsymbol{\theta},\mathbf{d}) &= E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d})\}=E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}\\ &= c_1\int_0^{\infty}\int_0^{\eta_1v_2}(-v_1+\eta_1v_2)\,f(v_1,v_2)dv_1dv_2\\ &+c_2\int_0^{\infty}\int_{\eta_2v_2}^{\infty}(v_1-\eta_2v_2)\,f(v_1,v_2)dv_1dv_2\\ &+c(\eta_2-\eta_1)\int_0^{\infty}\int_0^{\infty}v_2\,f(v_1,v_2)dv_1dv_2, \end{aligned} \tag{7}$$

which is constant on orbits when an invariant predictor (decision rule) **d** is used, where $f(v_1,v_2)$ is defined by the joint probability density of the first k observations $X_1 < X_2 < \cdots < X_k$ and X_r,

$$\begin{aligned} f(x_1,x_2,\ldots,x_k,x_r\mid\mu,\sigma) &= \frac{n!}{(r-k-1)!(n-r)!}\\ \times[F(x_r\mid\mu,\sigma)-F(x_k\mid\mu,\sigma)]^{r-k-1}&[1-F(x_r\mid\mu,\sigma)]^{n-r}\\ \times\prod_{i=1}^{k}f(x_i\mid\mu,\sigma)&f(x_r\mid\mu,\sigma). \end{aligned} \tag{8}$$

2.5 Risk minimization and optimal predictors

The following theorem gives the central result in this section.

Theorem 1 (*Optimal predictor of X_r based on* **X**). Suppose that (u_1, u_2) is a random vector having density function

$$u_2\,f(u_1,u_2)\left[\int_0^{\infty}\int_0^{\infty}u_2\,f(u_1,u_2)du_1du_2\right]^{-1}\quad(u_1,u_2>0), \tag{9}$$

where f is defined by $f(v_1,v_2)$, and let Q be the probability distribution function of u_1/u_2.

(i) If $c/c_1+c/c_2<1$ then the optimal invariant linear-loss interval predictor of X_r based on **X** is $\mathbf{d}^*=(X_k+\eta_1S_k,\ X_k+\eta_2S_k)$, where

$$Q(\eta_1)=c/c_1,\quad Q(\eta_2)=1-c/c_2. \tag{10}$$

(ii) If $c/c_1+c/c_2\ge1$ then the optimal invariant linear-loss interval predictor of X_r based on **X** degenerates into a point predictor $X_k+\eta_\bullet S_k$, where

$$Q(\eta_\bullet)=c_2/(c_1+c_2). \tag{11}$$

Proof. From (7)

$$\begin{aligned} &\frac{\partial E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}}{\partial\eta_1}\\ &=c_1\int_0^{\infty}\int_0^{\eta_1v_2}v_2\,f(v_1,v_2)dv_1dv_2-c\int_0^{\infty}\int_0^{\infty}v_2\,f(v_1,v_2)dv_1dv_2\\ &=\int_0^{\infty}\int_0^{\infty}v_2\,f(v_1,v_2)dv_1dv_2[c_1Q(\eta_1)-c], \end{aligned} \tag{12}$$

and

$$\frac{\partial E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}}{\partial\eta_2}=\int_0^{\infty}\int_0^{\infty}v_2\,f(v_1,v_2)dv_1dv_2[-c_2(1-Q(\eta_2))+c], \tag{13}$$

where

$$Q(\eta)=\int_0^{\eta}q(w)dw, \tag{14}$$

$$q(w)=\frac{\int_0^{\infty}v_2^2\,f(wv_2,v_2)dv_2}{\int_0^{\infty}\int_0^{\infty}v_2\,f(v_1,v_2)dv_1dv_2}, \tag{15}$$

$$W=V_1/V_2. \tag{16}$$

Now $\partial E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}/\partial\eta_1=\partial E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}/\partial\eta_2=0$ if and only if (10) hold. Thus, $E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}$ provided (10) has a solution with $\eta_1<\eta_2$ and this is so if $1-c/c_2>c/c_1$. It is easily confirmed that this $\boldsymbol{\eta}=(\eta_1,\eta_2)$ gives the minimum value of $E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}$. Thus (i) is established.

If $c/c_1+c/c_2\ge1$ then the minimum of $E\{\ddot{r}(\mathrm{v},\eta)\}$ in the region $\eta_2\ge\eta_1$ occurs where $\eta_1=\eta_2=\eta_\bullet$, $\eta_\bullet$ being determined by setting

$$\partial E\{\ddot{r}(V,(\eta_\bullet,\eta_\bullet))\}/\partial\eta_\bullet=0 \tag{17}$$

and this reduces to

$$c_1Q(\eta_\bullet)-c_2[1-Q(\eta_\bullet)]=0, \tag{18}$$

which establishes (ii). □

Corollary 1.1 (*Minimum risk of the optimal invariant predictor of X_r based on* **X**). The minimum risk is given by

$$\begin{aligned} R(\boldsymbol{\theta},\mathbf{d}^*) &= E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d}^*)\}=E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}\\ &=-c_1\int_0^{\infty}\int_0^{\eta_1v_2}v_1\,f(v_1,v_2)dv_1dv_2+c_2\int_0^{\infty}\int_{\eta_2v_2}^{\infty}v_1\,f(v_1,v_2)dv_1dv_2 \end{aligned} \tag{19}$$

for case (i) with $\boldsymbol{\eta}=(\eta_1,\eta_2)$ as given by (10) and for case (ii) with $\eta_1=\eta_2=\eta_\bullet$ as given by (11).

Proof. These results are immediate from (7) when use is made of $\partial E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}/\partial\eta_1=\partial E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}/\partial\eta_2=0$ in case (i) and $\partial\mathrm{E}\{\ddot{r}(\mathbf{V},(\eta_\bullet,\eta_\bullet))\}/\partial\eta_\bullet=0$ in case (ii). □

The underlying reason why $c/c_1+c/c_2$ acts as a separator of interval and point prediction is that for $c/c_1+c/c_2\ge1$ every interval predictor is inadmissible, there existing some point predictor with uniformly smaller risk.

Theorem 2 (*Optimal invariant predictor of X_r based on X_k*). Suppose that μ=0 and

$$\mathbf{V}=(V_1,V_2),\quad V_1=(X_r-X_k)/\sigma,\quad V_2=X_k/\sigma;$$

$$\boldsymbol{\eta}=(\eta_1,\eta_2),\quad \eta_1=(d_1-X_k)/X_k,\quad \eta_2=(d_2-X_k)/X_k. \quad (20)$$

Let us assume that (u_1, u_2) is a random vector having density function

$$u_2 f_0(u_1,u_2)\left[\int_0^\infty\int_0^\infty u_2 f_0(u_1,u_2)du_1du_2\right]^{-1} \quad (u_1,u_2>0), \quad (21)$$

where f_0 is defined by $f_0(v_1,v_2)$, and let Q_0 be the probability distribution function of u_1/u_2.

(i) If $c/c_1+c/c_2<1$ then the optimal invariant linear-loss interval predictor of X_r based on X_k is $\mathbf{d}^*=((1+\eta_1)X_k, (1+\eta_2)X_k)$, where

$$Q_0(\eta_1)=c/c_1,\quad Q_0(\eta_2)=1-c/c_2. \quad (22)$$

(ii) If $c/c_1+c/c_2\geq 1$ then the optimal invariant linear-loss interval predictor of X_r based on X_k degenerates into a point predictor $(1+\eta_\bullet)X_k$, where

$$Q_0(\eta_\bullet)=c_2/(c_1+c_2). \quad (23)$$

Proof. For the proof we refer to Theorem 1. □

Corollary 2.1 (*Minimum risk of the optimal invariant predictor of X_r based on X_k*). The minimum risk is given by

$$R(\boldsymbol{\theta},\mathbf{d}^*)=E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d}^*)\}=E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}$$

$$=-c_1\int_0^\infty\int_0^{\eta_1 v_2} v_1 f_0(v_1,v_2)dv_1dv_2+c_2\int_0^\infty\int_{\eta_2 v_2}^\infty v_1 f_0(v_1,v_2)dv_1dv_2 \quad (24)$$

for case (i) with $\boldsymbol{\eta}=(\eta_1,\eta_2)$ as given by (22) and for case (ii) with $\eta_1=\eta_2=\eta_\bullet$ as given by (23).

Proof. For the proof we refer to Corollary 1.1. □

3 Equivalent confidence coefficient

For case (i) when we obtain an interval predictor for X_r we may regard the interval as a confidence interval in the conventional sense and evaluate its confidence coefficient. The general result is contained in the following theorem.

Theorem 3 (*Equivalent confidence coefficient for $\mathbf{d}^*$ based on* **X**). Suppose that $\mathbf{V}=(V_1,V_2)$ is a random vector having density function $f(v_1,v_2)$ ($v_1,v_2>0$) where f is defined by (8) and let H be the distribution function of $W=V_1/V_2$, i.e., the probability density function of W is given by

$$h(w)=\int_0^\infty v_2 f(wv_2,v_2)dv_2. \quad (25)$$

Then the confidence coefficient associated with the optimum prediction interval $\mathbf{d}^*=(d_1,d_2)$, where $d_1=X_k+\eta_1S_k$, $d_2=X_k+\eta_2S_k$, is

$$\Pr\{\mathbf{d}^*: d_1<X_r<d_2\mid\mu,\sigma\}$$

$$=H[Q^{-1}(1-c/c_2)]-H[Q^{-1}(c/c_1)]. \quad (26)$$

Proof. The confidence coefficient for $\mathbf{d}^*$ corresponding to (μ,σ) is given by

$$\Pr\{(X_k,S_k): X_k+\eta_1S_k<X_r<X_k+\eta_2S_k\mid\mu,\sigma\}$$

$$=\Pr\{(v_1,v_2):\eta_1<v_1/v_2<\eta_2\}$$

$$=H(\eta_2)-H(\eta_1)=H[Q^{-1}(1-c/c_2)]-H[Q^{-1}(c/c_1)]. \quad (27)$$

This is independent of (μ,σ). □

Theorem 4 (*Equivalent confidence coefficient for $\mathbf{d}^*$ based on X_k*). Suppose that $\mathbf{V}=(V_1,V_2)$ is a random vector having density function $f_0(v_1,v_2)$ (v_1 real, $v_2>0$), where f_0 is defined by

$$f(x_k,x_r\mid\mu,\sigma)=\frac{1}{\mathrm{B}(k,r-k)\mathrm{B}(r,n-r+1)}$$

$$\times[F(x_k\mid\mu,\sigma)]^{r-1}[F(x_r\mid\mu,\sigma)-F(x_k\mid\mu,\sigma)]^{r-k-1}$$

$$\times[1-F(x_r\mid\mu,\sigma)]^{n-r}f(x_k\mid\mu,\sigma)f(x_r\mid\mu,\sigma), \quad (28)$$

and let H_0 be the distribution function of $W=V_1/V_2$, i.e., the probability density function of W is given by

$$h_0(w)=\int_0^\infty v_2 f_0(wv_2,v_2)dv_2. \quad (29)$$

Then the confidence coefficient associated with the optimum prediction interval $\mathbf{d}^*=(d_1,d_2)$, where $d_1=(1+\eta_1)X_k$, $d_2=(1+\eta_2)X_k$, is

$$\Pr\{\mathbf{d}^*: d_1<X_r<d_2\mid\mu,\sigma\}$$

$$=H_0[Q_0^{-1}(1-c/c_2)]-H_0[Q_0^{-1}(c/c_1)]. \quad (30)$$

Proof. For the proof we refer to Theorem 3. □

The way in which (26) (or (30)) varies with c, c_1 and c_2, and the fact that c_1 and c_2 are the factors of proportionality associated with losses from overshooting and undershooting relative to loss involved in increasing the length of interval, provides an interesting interpretation of confidence interval prediction.

4 New-sample prediction problem

For new-sample prediction, data from a past sample are used to make predictions on a future unit or sample of units from the same process or population. For example, based on previous (possibly censored) life test data, one could be interested in predicting the time to failure of a new item, time until l failures in a future sample of m units, or number of failures by time $t_\bullet$ in a future sample of m units.

4.1 Location-scale family of density functions

Consider a situation described by a location-scale family of density functions, indexed by the vector parameter $\boldsymbol{\theta}=(\mu,\sigma)$, where μ and σ (>0) are respectively parameters of location and scale. For this family, invariant under the group of positive linear transformations: $x\rightarrow ax+b$ with $a>0$, we shall assume that there is obtainable from some informative experiment (the first k order statistics $X_1<X_2<\cdots<X_k$ from a

random sample of size n) a sufficient statistic (M_k,S_k) (or a maximum likelihood estimator $(\hat{\mu}_k, \hat{\sigma}_k)$) for (μ,σ) based on $\mathbf{X}=(X_1, X_2,\ldots, X_k)$ with density function

$$p(m_k, s_k \mid \mu,\sigma) = \sigma^{-2} p_0[(m_k-\mu)/\sigma, s_k/\sigma]$$
$$-\infty < m_k < \infty,\ \ 0 < s_k < \infty,\ \ -\infty < \mu > \infty,\ \ \sigma > 0. \quad (31)$$

We are thus assuming that for the family of density functions an induced invariance holds under the group G of transformations: $m_k \to am_k+b$, $s_k \to as_k$ or $\hat{\mu}_k \to a\hat{\mu}_k+b$, $\hat{\sigma}_k \to a\hat{\sigma}_k$ ($a>0$). The family of density functions satisfying the above conditions is, of course, the limited one of normal, negative exponential, Weibull and gamma (with known index) density functions. The structure of the problem is, however, more clearly seen within the general framework. Let Y be an independent future observation from a new sample. If Y is invariantly predictable then $W=(Y-M_k)/S_k$ (or $W=(Y-\hat{\mu}_k)/\hat{\sigma}_k$) is a maximal invariant pivotal, conditional on $\mathbf{X}$.

4.2 Piecewise-linear loss function

We shall consider the interval prediction problem for the sth order statistic Y_s, $1 \le s \le m$, in a future sample of size m for the situation where the first k observations $X_1 < X_2 < \cdots < X_k$, $1 \le k < n$, from a past sample of size n have been observed. Suppose that we assert that an interval $\mathbf{d}=(d_1,d_2)$ contains Y_s. If, as is usually the case, the purpose of this interval statement is to convey useful information we incur penalties if d_1 lies above Y_s or if d_2 falls below Y_s. Suppose that these penalties are $c_1(d_1- Y_s)$ and $c_2(Y_s -d_2)$, losses proportional to the amounts by which Y_s escapes the interval. Since c_1 and c_2 may be different the possibility of differential losses associated with the interval overshooting and undershooting the true μ is allowed. In addition to these losses there will be a cost attaching to the length of interval used. For example, it will be more difficult and more expensive to design or plan when the interval $\mathbf{d}=(d_1,d_2)$ is wide. Suppose that the cost associated with the interval is proportional to its length, say $c(d_2-d_1)$. In the specification of the loss function, σ is clearly a 'nuisance parameter' and no alteration to the basic decision problem is caused by multiplying all loss factors by $1/\sigma$. Thus we are led to investigate the piecewise-linear loss function

$$r(\boldsymbol{\theta},\mathbf{d}) = \begin{cases} \dfrac{c_1(d_1-Y_s)}{\sigma} + \dfrac{c(d_2-d_1)}{\sigma} & (Y_s < d_1), \\ \dfrac{c(d_2-d_1)}{\sigma} & (d_1 \le Y_s \le d_2), \\ \dfrac{c(d_2-d_1)}{\sigma} + \dfrac{c_2(Y_s-d_2)}{\sigma} & (Y_s > d_2). \end{cases} \quad (32)$$

The decision problem specified by the informative experiment density function (31) and the loss function (32) is invariant under the group G of transformations. Thus, the problem is to find the optimal interval predictor of Y_s,

$$\mathbf{d}^* = \arg\min_{\mathbf{d}\in\mathscr{D}} R(\boldsymbol{\theta},\mathbf{d}), \quad (33)$$

where $\mathscr{D}$ is a set of invariant interval predictors of Y_s, $R(\boldsymbol{\theta},\mathbf{d})=E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d})\}$ is a risk function.

4.3 Transformation of the loss function

It follows from (32) that the invariant loss function, $r(\boldsymbol{\theta},\mathbf{d})$, can be transformed as follows:

$$r(\boldsymbol{\theta},\mathbf{d}) = \ddot{r}(\mathbf{V},\boldsymbol{\eta}), \quad (34)$$

where

$$\ddot{r}(\mathbf{V},\boldsymbol{\eta}) = \begin{cases} c_1(-V_1+\eta_1V_2)+c(\eta_2-\eta_1)V_2 & (V_1 < \eta_1V_2), \\ c(\eta_2-\eta_1)V_2 & (\eta_1V_2 \le V_1 \le \eta_2V_2), \\ c_2(V_1-\eta_2V_2)+c(\eta_2-\eta_1)V_2 & (V_1 > \eta_2V_2), \end{cases} \quad (35)$$

$$\mathbf{V}=(V_1,V_2),\ \ V_1=(Y_s-M_k)/\sigma,\ \ V_2=S_k/\sigma;$$
$$\boldsymbol{\eta}=(\eta_1,\eta_2),\ \ \eta_1=(d_1-M_k)/S_k,\ \ \eta_2=(d_2-M_k)/S_k. \quad (36)$$

4.4 Risk function

It follows from (35) that the risk associated with $\mathbf{d}$ and $\boldsymbol{\theta}$ can be expressed as

$$R(\boldsymbol{\theta},\mathbf{d}) = E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d})\} = E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}$$
$$= c_1\int_0^{\infty}\int_{-\infty}^{\eta_1v_2}(-v_1+\eta_1v_2)\,f(v_1,v_2)\,dv_1dv_2$$
$$+ c_2\int_0^{\infty}\int_{\eta_2v_2}^{\infty}(v_1-\eta_2v_2)\,f(v_1,v_2)\,dv_1dv_2$$
$$+ c(\eta_2-\eta_1)\int_0^{\infty}\int_{-\infty}^{\infty}v_2\,f(v_1,v_2)\,dv_1dv_2, \quad (37)$$

which is constant on orbits when an invariant predictor (decision rule) $\mathbf{d}$ is used, where $f(v_1,v_2)$ is defined by the joint probability density of the first k observations $X_1 < X_2 < \cdots < X_k$ from the past random sample of size n and the sth order statistic Y_s in the future sample of size m,

$$f(x_1,x_2,\ldots,x_k,y_s \mid \mu,\sigma) = \frac{n!}{(n-k)!}\frac{m!}{(s-1)!(m-s)!}$$
$$\times\prod_{i=1}^{k} f(x_i \mid \mu,\sigma)[1-F(x_k \mid \mu,\sigma)]^{n-k}$$
$$\times[F(y_s \mid \mu,\sigma)]^{s-1}[1-F(y_s \mid \mu,\sigma)]^{m-s}\,f(y_s \mid \mu,\sigma). \quad (38)$$

4.5 Risk minimization and optimal predictors

The following theorem gives the central result in this section.

Theorem 5 (*Optimal invariant predictor of Y_s based on* $\mathbf{X}$). Suppose that $(u_1,\ u_2)$ is a random vector having density function

$$u_2\,f(u_1,u_2)\left[\int_0^{\infty}\int_{-\infty}^{\infty}u_2\,f(u_1,u_2)\,du_1du_2\right]^{-1} \quad (u_1 \text{ real},\ u_2>0), \quad (39)$$

where f is defined by $f(v_1,v_2)$, and let Q be the probability distribution function of u_1/u_2.

(i) If $c/c_1+c/c_2<1$ then the optimal invariant linear-loss interval predictor of Y_s based on $\mathbf{X}$ is $\mathbf{d}^*=(M_k+\eta_1 S_k, M_k+\eta_2 S_k)$, where

$$Q(\eta_1)=c/c_1, \quad Q(\eta_2)=1-c/c_2. \tag{40}$$

(ii) If $c/c_1+c/c_2\geq 1$ then the optimal invariant linear-loss interval predictor of Y_s based on $\mathbf{X}$ degenerates into a point predictor $M_k+\eta_\bullet S_k$, where

$$Q(\eta_\bullet)=c_2/(c_1+c_2). \tag{41}$$

Proof. For the proof we refer to Theorem 1. □

Corollary 5.1 (*Minimum risk of the optimal invariant predictor of Y_s based on* $\mathbf{X}$). The minimum risk is given by

$$R(\boldsymbol{\theta},\mathbf{d}^*)=E_{\boldsymbol{\theta}}\{r(\boldsymbol{\theta},\mathbf{d}^*)\}=E\{\ddot{r}(\mathbf{V},\boldsymbol{\eta})\}$$

$$=-c_1\int_0^\infty\int_{-\infty}^{\eta_1 v_2} v_1 f(v_1,v_2)dv_1dv_2+c_2\int_0^\infty\int_{\eta_2 v_2}^\infty v_1 f(v_1,v_2)dv_1dv_2 \tag{42}$$

for case (i) with $\boldsymbol{\eta}=(\eta_1,\eta_2)$ as given by (40) and for case (ii) with $\eta_1=\eta_2=\eta_\bullet$ as given by (41).

Proof. For the proof we refer to Corollary 1.1. □

Theorem 6 (*Equivalent confidence coefficient for* $\mathbf{d}^*$ *based on* $\mathbf{X}$). Suppose that $\mathbf{V}=(V_1,V_2)$ is a random vector having density function $f(v_1,v_2)$ (v_1 real, $v_2>0$) where f is defined by (38) and let H be the distribution function of $W=V_1/V_2$, i.e., the probability density function of W is given by

$$h(w)=\int_0^\infty v_2 f(wv_2,v_2)dv_2. \tag{43}$$

Then the confidence coefficient associated with the optimum prediction interval $\mathbf{d}^*=(d_1,d_2)$, where $d_1=M_k+\eta_1 S_k$, $d_2=M_k+\eta_2 S_k$, is

$$\Pr\{\mathbf{d}^*: d_1<X_r<d_2\mid\mu,\sigma\}$$

$$=H[Q^{-1}(1-c/c_2)]-H[Q^{-1}(c/c_1)]. \tag{44}$$

Proof. For the proof we refer to Theorem 3. □

5 Example

5.1 Within-sample prediction

Exponential distribution. Let $X_1<X_2<\cdots<X_n$ be order statistics of size n from the exponential distribution with the density

$$f(x\mid\sigma)=\frac{1}{\sigma}\exp\left(-\frac{x}{\sigma}\right), \quad x>0, \quad \sigma>0. \tag{45}$$

We shall consider the prediction problem of X_r for the situation where the first k observations $X_1<X_2<\cdots<X_k$, $1\leq k<r\leq n$, have been observed. Let G be the group of transformations $x_i=ax_i$ ($i=1,\ldots,k,r,n$, $a>0$) We are now concerned with optimization of the prediction interval for X_r under the loss function (2). Let $\mathbf{X}=(X_1,X_2,\ldots,X_k)$ and $X_r>X_k$ for $r\leq n$. Then the joint probability density function of $\mathbf{X}$ and X_r is given by

$$f(x_1,x_2,\ldots,x_k,x_r\mid\sigma)=\frac{n!}{(r-k-1)!(n-r)!}$$

$$\times[F(x_r\mid\sigma)-F(x_k\mid\sigma)]^{r-k-1}[1-F(x_r\mid\sigma)]^{n-r}$$

$$\times\prod_{i=1}^k f(x_i\mid\sigma)f(x_r\mid\sigma)$$

$$=\frac{n!}{(r-k-1)!(n-r)!}\frac{1}{\sigma^{k+1}}\exp\left(-\frac{\sum_{i=1}^k x_i+(n-k)x_k}{\sigma}\right)$$

$$\times\left[1-\exp\left(-\frac{x_r-x_k}{\sigma}\right)\right]^{r-k-1}\left[\exp\left(-\frac{x_r-x_k}{\sigma}\right)\right]^{n-r+1}. \tag{46}$$

Let

$$V_1=\frac{X_r-X_k}{\sigma}, \quad V_2=\frac{S_k}{\sigma}=\frac{\sum_{i=1}^k X_i+(n-k)X_k}{\sigma}. \tag{47}$$

Using the invariant embedding technique [13-18], we then find in a straightforward manner that the joint density of V_1, V_2 is

$$f(v_1,v_2)=f_1(v_1)f_2(v_2), \tag{48}$$

where

$$f_1(v_1)=\frac{[1-e^{-v_1}]^{r-k-1}[e^{-v_1}]^{n-r+1}}{\mathrm{B}(r-k,n-r+1)}$$

$$=\frac{1}{\mathrm{B}(r-k,n-r+1)}\sum_{j=0}^{r-k-1}\binom{r-k-1}{j}(-1)^j e^{-v_1(n-r+1+j)},$$

$$v_1>0, \tag{49}$$

and

$$f(v_2)=\frac{1}{\Gamma(k)}v_2^{k-1}e^{-v_2}, \quad v_2>0. \tag{50}$$

It follows from (15) and (49) that

$$q(w)=\frac{\int_0^\infty v_2^2 f(wv_2,v_2)dv_2}{\int_0^\infty\int_0^\infty v_2 f(v_1,v_2)dv_1dv_2}=\frac{1}{k}\int_0^\infty v_2^2 f_1(wv_2)f_2(v_2)dv_2$$

$$=\frac{k+1}{\mathrm{B}(r-k,n-r+1)}$$

$$\times\sum_{j=0}^{r-k-1}\binom{r-k-1}{j}(-1)^j\frac{1}{[1+w(n-r+1+j)]^{k+2}}. \tag{51}$$

It follows from (25) and (48) that

$$h(w)=\int_0^\infty v_2 f(wv_2,v_2)dv_2=\int_0^\infty v_2 f_1(wv_2)f_2(v_2)dv_2$$

$$=\frac{k}{\mathrm{B}(r-k,n-r+1)}\sum_{j=0}^{r-k-1}\binom{r-k-1}{j}(-1)^j\frac{1}{[1+w(n-r+1+j)]^{k+1}}. \tag{52}$$

If $c/c_1+c/c_2<1$ then the optimal invariant linear-loss interval predictor of X_r based on $\mathbf{X}$ is given by

$$\mathbf{d}^*=(X_k+\eta_1 S_k,\ X_k+\eta_2 S_k), \tag{53}$$

where

$$\eta_1=\arg\left(\int_0^{\eta_1} q(w)dw=\frac{c}{c_1}\right),\quad \eta_2=\arg\left(\int_0^{\eta_1} q(w)dw=1-\frac{c}{c_2}\right). \tag{54}$$

The confidence coefficient associated with the optimum prediction interval $\mathbf{d}^*=(d_1,d_2)$, where $d_1=X_k+\eta_1 S_k$, $d_2=X_k+\eta_2 S_k$, is given by

$$\Pr\{\mathbf{d}^* : d_1 < X_r < d_2 \mid \mu,\sigma\}$$

$$= H[\eta_2]-H[\eta_1]=\int_{\eta_1}^{\eta_2} h(w)dw. \tag{55}$$

6 Conclusions

In many statistical decision problems it is reasonable confine attention to rules that are invariant with respect to a certain group of transformations. If a given decision problem admits a sufficient statistic, it is well known that the class of invariant rules based on the sufficient statistic is essentially complete in the class of all invariant rules under some assumptions. This result may be used to show that if there exists a minimax invariant rule among invariant rules based on sufficient statistic, it is minimax among all invariant rules.

7 References

[1] J. K. Patel. "Prediction Intervals – A Review"; Communications in Statistics –Theory and Methods, Vol. 13, pp. 2393 –2465, 1989.

[2] G. J. Hahn and W. Q. Meeker. "Statistical Intervals: A Guide for Practitioners". New York: Wiley, 1991.

[3] N. R. Mann and S. C. Saunders. "On Evaluation of Warranty Assurance when Life Has a Weibull Distribution"; Biometrika, Vol. 56, pp. 615 –625, 1969.

[4] N. R. Mann. "Warranty Periods Based on Three Ordered Sample Observations from a Weibull Population"; IEEE Transactions on Reliability, Vol. R–19, pp. 167 –171, 1970.

[5] C. E. Antle and F. Rademaker. "An Upper Confidence Limit on the Maximum of m Future Observations from a Type I Extreme-Value Distribution"; Biometrika, Vol. 59, pp. 475–477, 1972.

[6] J. F. Lawless. "On the Estimation of Safe Life when the Underlying Life Distribution is Weibull"; Technometrics, Vol. 15, pp. 857 –865, 1973.

[7] M. Engelhardt and L. J. Bain. "Prediction Limits and Two-Sample Problems with Complete or Censored Weibull Data"; Technometrics, Vol. 21, pp. 233 –237, 1979.

[8] M. Engelhardt and L. J. Bain. "On Prediction Limits for Samples from a Weibull or Extreme-Value Distribution"; Technometrics, Vol. 24, pp. 147 –150, 1982.

[9] K. W. Fertig, M. Mayer, and N. R. Mann. "On Constructing Prediction Intervals for Samples from a Weibull or Extreme Value Distribution"; Technometrics, Vol. 22, pp. 567–573, 1980.

[10] R. W. Mee and D. Kushary. "Prediction Limits for the Weibull Distribution Utilizing Simulation"; Computational Statistics & Data Analysis, Vol. 17, pp. 327 –336, 1994.

[11] D. K. Bhaumik and R. D. Gibbons. "One-Sided Prediction Intervals for at Least *p* of *m* Observations from a Gamma Population at Each of *r* Locations"; Technometrics, Vol. 48, pp. 112 –129, 2006.

[12] K. Krishnamoorthy, T. Mathew, and S. Mukherjee. "Normal Based Methods for a Gamma Distribution: Prediction and Tolerance Intervals and Stress-Strength Reliability"; Technometrics, Vol. 50, pp. 69 –78, 2007.

[13] N. A. Nechval and E. K. Vasermanis. "Improved Decisions in Statistics". Riga: SIA "Izglitibas soli", 2004.

[14] N. A. Nechval, G. Berzins, M. Purgailis, and K. N. Nechval. "Improved Estimation of State of Stochastic Systems via Invariant Embedding Technique"; WSEAS Transactions on Mathematics, Vol. 7, pp. 141–159, 2008.

[15] N. A. Nechval, M. Purgailis, G. Berzins, K. Cikste, J. Krasts, and K. N. Nechval. "Invariant Embedding Technique and Its Applications for Improvement or Optimization of Statistical Decisions"; in Analytical and Stochastic Modeling Techniques and Applications, K. Al-Begain, D. Fiems, and W. Knottenbelt (Eds.). LNCS, Vol. 6148, Berlin, Heidelberg: Springer-Verlag, 2010, pp. 306 –320.

[16] N. A. Nechval, M. Purgailis, K. Cikste, G. Berzins, U. Rozevskis, and K. N. Nechval. "Prediction Model Selection and Spare Parts Ordering Policy for Efficient Support of Maintenance and Repair of Equipment"; in Analytical and Stochastic Modeling Techniques and Applications, K. Al-Begain, D. Fiems, and W. Knottenbelt (Eds.). LNCS, Vol. 6148, Berlin, Heidelberg: Springer-Verlag, 2010, pp. 321–338.

[17] N. A. Nechval, M. Purgailis, K. Cikste, G. Berzins, and K. N. Nechval. "Optimization of Statistical Decisions via an Invariant Embedding Technique"; in Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2010, WCE 2010, 30 June - 2 July, 2010, London, U.K., pp. 1776 –1782.

[18] N. A. Nechval and M. Purgailis. "Improved State Estimation of Stochastic Systems via a New Technique of Invariant Embedding"; in Stochastic Control, Chris Myers (Ed.). Croatia, India, Publisher: Sciyo, 2010, pp. 167 –193.

Numerical Computation Method in Solving Integral Equation by Using the Second Chebyshev Wavelets

L. Zhu, Y. X. Wang, and Q. B. Fan
School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei, China

Abstract— *In this paper, a numerical method for solving the Fredholm and Volterra integral equations is presented. The method is based upon the second Chebyshev wavelet approximation. The properties of the second Chebyshev wavelet are first presented and then operational matrix of integration of the second Chebyshev wavelets basis and product operation matrix of it are derived. The second Chebyshev wavelet approximation method is then utilized to reduce the integral equation to the solution of algebraic equations combining Galerkin method. Some comparative examples are included to demonstrate superiority of operational matrix of the second Chebyshev wavelets to those of Legendre wavelets and CAS wavelets. It shows higher accuracy of the second Chebyshev wavelets method.*

Keywords: The second Chebyshev wavelets, Operational matrix of integration, Product operational matrix, Integral equation

1. Introduction

In recent years, wavelets have found their way into many different fields of science and engineering, particularly, wavelets are very successfully used in signal analysis for waveform representation and segmentations, time-frequency analysis and fast algorithms for easy implementation. Wavelets permit the accurate representation of a variety of functions and operators. Moreover, wavelets establish a connection with fast numerical algorithms[1]. The main advantage of wavelet method for solving the integral equation and differential equation is after discreting the coefficients matrix of algebraic equations is spare[2]. So, the computational cost is low.

Several wavelets methods for approximating the solution of the integral equations and differential equations are known. Haar wavelets method was presented in [3-5]. CAS wavelets method was developed in [6,7]. Harmonic wavelets method of successive approximation was introduced in [8]. In [9,10], E. Babolian applied operational matrix of integration of Chebyshev wavelets basis to the integral equations and differential equations and it was used in solving a nonlinear fractional differential equation in [11]. K. Maleknejad[12] introduced Legendre wavelets method for Fredholm and Volterra integral equations, while in [13] the integral and differential equations were solved by Legendre wavelets.

Here we will construct the second Chebyshev wavelets on the interval $[0,1]$. The wavelets basis are suitable for numerical solutions of the integral equation.

2. Properties of the second Chebyshev wavelets

Wavelets constitute a family of functions constructed from dilation and translation of a single function $\psi(x)$ called the mother wavelet. When the dilation parameter a and the translation parameter b vary continuously we have the following family of continuous wavelets as[2]

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}}\psi\left(\frac{t-b}{a}\right),\ a,\ b \in R,\quad a \neq 0.$$

If we restrict the parameters a and b to discrete values as $a = a_0^{-k},\ b = nb_0a_0^{-k},\ a_0 > 1,\quad b_0 > 0$, we have the following family of discrete wavelets

$$\psi_{k,n}(t) = |a_0|^{\frac{k}{2}}\psi(a_0^k t - nb_0),\quad k, n \in \mathbb{Z},$$

where $\psi_{k,n}$ form a wavelet basis for $L^2(R)$. In particular, when $a_0 = 2$ and $b_0 = 1$ then $\psi_{k,n}(t)$ form an orthonormal basis.

The second Chebyshev wavelets $\psi_{n,m}(t) = \psi(k,n,m,t)$ involve four arguments, $n = 1, ..., 2^{k-1}$, k is assumed any positive integer, m is the degree of the second Chebyshev polynomials and t is the normalized time. They are defined on the interval $[0,1)$ as

$$\psi_{nm}(t) = \begin{cases} 2^{\frac{k}{2}}\tilde{\mathrm{U}}_{\mathrm{m}}(2^k t - 2n + 1), & \frac{n-1}{2^{k-1}} \leqslant t < \frac{n}{2^{k-1}}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where

$$\tilde{\mathrm{U}}_{\mathrm{m}}(t) = \sqrt{\frac{2}{\pi}}\mathrm{U}_{\mathrm{m}}(t), \tag{2}$$

and $m = 0, 1, \ldots, M-1$. In Eq. (2) the coefficients are used for orthonormality. Here $U_m(t)$ are the second Chebyshev polynomials of degree m which respect to the weight function $\omega(t) = \sqrt{1-t^2}$ on the interval $[-1,1]$ and satisfy the following recursive formula

$$U_0(t) = 1,\quad U_1(t) = 2t,$$

$$U_{m+1}(t) = 2tU_m(t) - U_{m-1}(t),\quad m = 1, 2, \cdots$$

We should note that in dealing with the second Chebyshev wavelets the weight funtion $\tilde{\omega}(t) = \omega(2t-1)$ have to be dilate and translate as

$$\omega_n(t) = \omega(2^k t - 2n + 1),$$

A function $f(t)$ defined over $[0,1)$ may be expanded as

$$f(t) = \sum_{n=0}^{\infty} \sum_{m \in z} c_{nm} \psi_{nm}(t), \tag{3}$$

where

$$c_{nm} = (f(t), \psi_{nm}(t))_{\omega_n} = \int_0^1 \omega_n(t) \psi_{nm}(t) f(t)\, \mathrm{d}t, \tag{4}$$

in which $(\cdot\ ,\ \cdot)$ denotes the inner product in $L^2_{\omega_n}[0,1]$. If the infinite series in Eq. (3) is truncated, then it can be written as

$$f(t) \simeq \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(t) = C^T \Psi(t), \tag{5}$$

where C and $\Psi(t)$ are $2^{k-1}M \times 1$ matrices given by

$$C = [c_{10}, c_{11}, \ldots, c_{1(M-1)}, c_{20}, \ldots, c_{2(M-1)}, \ldots, c_{2^{k-1}0}, \ldots, c_{2^{k-1}(M-1)}]^T \tag{6}$$

and

$$\Psi(t) = [\psi_{10}, \psi_{11}, \ldots, \psi_{1(M-1)}, \psi_{20}, \ldots, \psi_{2(M-1)}, \ldots, \psi_{2^{k-1}0}, \ldots, \psi_{2^{k-1}(M-1)}]^T. \tag{7}$$

Similarly, a function $k(x,t) \in L^2_{\omega_n}([0,1] \times\ [0,1])$ may be approximated as

$$k(x,t) = \Psi(x)^T K \Psi(t), \tag{8}$$

where K is $2^{k-1}M \times 2^{k-1}M$ matrix with

$$K_{ij} = (\psi_i(x), (k(x,t), \psi_j(t))). \tag{9}$$

3. Operational matrix of integration and product operation matrix

In this section we will first derive the operational matrix P of integration[14–16] which plays a great role in dealing with the problem of integro-differential equations and Volterra integral equations. First we construct the 6×6 matrix P for $k = 2$ and $M = 3$. In this case, the six basis functions are given by

$$\left.\begin{aligned} \psi_{10}(t) &= 2\sqrt{\tfrac{2}{\pi}}, \\ \psi_{11}(t) &= 2\sqrt{\tfrac{2}{\pi}}(8t-2), \\ \psi_{12}(t) &= 2\sqrt{\tfrac{2}{\pi}}(64t^2 - 32t + 3), \end{aligned}\right\} \quad 0 \leqslant t < \frac{1}{2}, \tag{10}$$

$$\left.\begin{aligned} \psi_{20}(t) &= 2\sqrt{\tfrac{2}{\pi}}, \\ \psi_{21}(t) &= 2\sqrt{\tfrac{2}{\pi}}(8t-6), \\ \psi_{22}(t) &= 2\sqrt{\tfrac{2}{\pi}}(64t^2 - 96t + 35), \end{aligned}\right\} \quad \frac{1}{2} \leqslant t < 1. \tag{11}$$

By integrating (10) and (11) from 0 to t and representing it to the matrix form, we obtain

$$\begin{aligned} \int_0^t \psi_{10}(t')\, \mathrm{d}t' &= \begin{cases} 2\sqrt{\frac{2}{\pi}}t, & 0 \leqslant t < \frac{1}{2}, \\ \sqrt{\frac{2}{\pi}}, & \frac{1}{2} \leqslant t < 1. \end{cases} \\ &= \frac{1}{4}\psi_{10}(t) + \frac{1}{8}\psi_{11}(t) + \frac{1}{2}\psi_{20}(t) \\ &= \left[\frac{1}{4}, \frac{1}{8}, 0, \frac{1}{2}, 0, 0\right] \Psi_6(t), \end{aligned}$$

$$\begin{aligned} \int_0^t \psi_{11}(t')\, \mathrm{d}t' &= \begin{cases} 4\sqrt{\frac{2}{\pi}}(2t^2 - t), & 0 \leqslant t < \frac{1}{2}, \\ 0, & \frac{1}{2} \leqslant t < 1, \end{cases} \\ &= -\frac{3}{16}\psi_{10}(t) + \frac{1}{16}\psi_{12}(t) \\ &= \left[-\frac{3}{16}, 0, \frac{1}{16}, 0, 0, 0\right] \Psi_6(t). \end{aligned}$$

Similarly we have

$$\begin{aligned} \int_0^t \psi_{12}(t')\, \mathrm{d}t' &= \frac{1}{12}\psi_{10}(t) - \frac{1}{24}\psi_{11}(t) \\ &= \left[\frac{1}{12}, -\frac{1}{24}, 0, 0, 0, 0\right] \Psi_6(t), \end{aligned}$$

$$\begin{aligned} \int_0^t \psi_{20}(t')\, \mathrm{d}t' &= \frac{1}{4}\psi_{20}(t) + \frac{1}{8}\psi_{21}(t) \\ &= \left[0, 0, 0, \frac{1}{4}, \frac{1}{8}, 0\right] \Psi_6(t), \end{aligned}$$

$$\begin{aligned} \int_0^t \psi_{21}(t')\, \mathrm{d}t' &= -\frac{3}{16}\psi_{20}(t) + \frac{1}{16}\psi_{22}(t) \\ &= \left[0, 0, 0, -\frac{3}{16}, 0, \frac{1}{16}\right] \Psi_6(t), \end{aligned}$$

$$\begin{aligned} \int_0^t \psi_{22}(t')\, \mathrm{d}t' &= \frac{1}{12}\psi_{20}(t) - \frac{1}{24}\psi_{21}(t) \\ &= \left[0, 0, 0, \frac{1}{12}, -\frac{1}{24}, 0\right] \Psi_6(t). \end{aligned}$$

Thus

$$\int_0^t \Psi_6(t')\, \mathrm{d}t' = P_{6\times 6} \Psi_6(t), \tag{12}$$

where

$$\Psi_6(t) = [\psi_{10}, \psi_{11}, \psi_{12}, \psi_{20}, \psi_{21}, \psi_{22}]^T$$

and

$$P_{6\times 6} = \frac{1}{4} \begin{bmatrix} 1 & \frac{1}{2} & 0 & 2 & 0 & 0 \\ -\frac{3}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{3} & -\frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{3} & -\frac{1}{6} & 0 \end{bmatrix}.$$

In Eq. (12) the subscript of $P_{6\times6}$ and Ψ_6 denote the dimension. In fact the matrix $P_{6\times6}$ can be written as

$$P_{6\times6} = \frac{1}{4}\begin{bmatrix} L_{3\times3} & F_{3\times3} \\ O_{3\times3} & L_{3\times3} \end{bmatrix},$$

where

$$L_{3\times3} = \begin{bmatrix} 1 & \frac{1}{2} & 0 \\ -\frac{3}{4} & 0 & \frac{1}{4} \\ \frac{1}{3} & -\frac{1}{6} & 0 \end{bmatrix}, \qquad F_{3\times3} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

For general case, we have

$$\int_0^t \Psi(t')\,\mathrm{d}t' = P\Psi(t), \tag{13}$$

where P is a $2^{k-1}M \times 2^{k-1}M$ matrix for integration and is given as

$$P = \frac{1}{2^k}\begin{bmatrix} L & F & F & \cdots & F & F \\ O & L & F & \cdots & F & F \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \cdots & L & F \\ O & O & O & \cdots & O & L \end{bmatrix},$$

where F and L are $M \times M$ matrices given by

$$F = \begin{bmatrix} 2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

and

$$L = \begin{bmatrix} 1 & \frac{1}{2} & 0 & \cdots & 0 \\ -\frac{3}{4} & 0 & \frac{1}{4} & \cdots & 0 \\ \frac{1}{3} & -\frac{1}{6} & 0 & \cdots & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{8} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-1)^{M-2}\frac{1}{M-1} & 0 & 0 & \cdots & \frac{1}{2(M-1)} \\ (-1)^{M-1}\frac{1}{M} & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Then we will obtain the product operation matrix, which is important for solving Volterra integral equations.

Let

$$\Psi(t)\Psi^T(t)C \simeq \tilde{C}\Psi(t), \tag{14}$$

where $\widetilde{C}$ is $2^{k-1}M \times 2^{k-1}M$ product operation matrix. To illustrate the calculation procedure we choose $k = 2, M = 3$. Thus we have

$$C = [c_{10}, c_{11}, c_{12}, c_{20}, c_{21}, c_{22}]^T, \tag{15}$$

$$\Psi(t) = [\psi_{10}, \psi_{11}, \psi_{12}, \psi_{20}, \psi_{21}, \psi_{22}]^T, \tag{16}$$

where, the six basis functions are given by Eq. (10) and Eq. (11).

So we get

$$\Psi(t)\Psi^T(t) = \begin{bmatrix} \psi_{10}\psi_{10} & \psi_{10}\psi_{11} & \psi_{10}\psi_{12} & \psi_{10}\psi_{20} & \psi_{10}\psi_{21} & \psi_{10}\psi_{22} \\ \psi_{11}\psi_{10} & \psi_{11}\psi_{11} & \psi_{11}\psi_{12} & \psi_{11}\psi_{20} & \psi_{11}\psi_{21} & \psi_{11}\psi_{22} \\ \psi_{12}\psi_{10} & \psi_{12}\psi_{11} & \psi_{12}\psi_{12} & \psi_{12}\psi_{20} & \psi_{12}\psi_{21} & \psi_{12}\psi_{22} \\ \psi_{20}\psi_{10} & \psi_{20}\psi_{11} & \psi_{20}\psi_{12} & \psi_{20}\psi_{20} & \psi_{20}\psi_{21} & \psi_{20}\psi_{22} \\ \psi_{21}\psi_{10} & \psi_{21}\psi_{11} & \psi_{21}\psi_{12} & \psi_{21}\psi_{20} & \psi_{21}\psi_{21} & \psi_{21}\psi_{22} \\ \psi_{22}\psi_{10} & \psi_{22}\psi_{11} & \psi_{22}\psi_{12} & \psi_{22}\psi_{20} & \psi_{22}\psi_{21} & \psi_{22}\psi_{22} \end{bmatrix}.$$

Expanding each product by wavelet basis we have

$$\Psi(t)\Psi^T(t) = 2\sqrt{\frac{2}{\pi}}\begin{bmatrix} \psi_{10} & \psi_{11} & \cdots & 0 \\ \psi_{11} & \psi_{10}+\psi_{12} & \cdots & 0 \\ \psi_{12} & \psi_{11} & \cdots & 0 \\ 0 & 0 & \cdots & \psi_{22} \\ 0 & 0 & \cdots & \psi_{21} \\ 0 & 0 & \cdots & \psi_{20}+\psi_{22} \end{bmatrix}.$$

By using the vector C, the $\tilde{C}$ is

$$\tilde{C} = 2\sqrt{\frac{2}{\pi}}\begin{bmatrix} \tilde{C}_1 & O \\ O & \tilde{C}_2 \end{bmatrix},$$

where $\tilde{C}_i (i = 1, 2)$ are 3×3 matrices given by

$$\tilde{C}_i = \begin{bmatrix} c_{i0} & c_{i1} & c_{i2} \\ c_{i1} & c_{i0}+c_{i2} & c_{i1} \\ c_{i2} & c_{i1} & c_{i0}+c_{i2} \end{bmatrix}.$$

4. Solving linear integral equation

First, consider the following integral equation

$$y(x) = \int_0^1 k(x,t)y(t)\,\mathrm{d}t + f(x), \qquad x \in [0,1], \tag{17}$$

where $f(x) \in L^2_\omega([0,1], k(x,t) \in L^2_\omega([0,1] \times [0,1])$ are known and $y(t)$ is the unknown function to be determined. If we approximate f, y and k by the way mentioned before

$$\begin{aligned} y(x) = \Psi(x)^T C, \quad f(x) = \Psi(x)^T F, \\ k(x,t) = \Psi(x)^T K \Psi(t). \end{aligned} \tag{18}$$

Substitute Eq. (18) into Eq. (17), we have

$$\begin{aligned} \Psi(x)^T C &= \Psi(x)^T F + \int_0^1 \Psi(x)^T K \Psi(t)\Psi(t)^T C \mathrm{d}t \\ &= \Psi(x)^T F + \Psi(x)^T K \left(\int_0^1 \Psi(t)\Psi(t)^T \mathrm{d}t\right) C \\ &= \Psi(x)^T (F + KDC), \end{aligned}$$

then

$$(I - KD)C = F, \tag{19}$$

where

$$D = \int_0^1 \Psi(t)\Psi(t)^T\,\mathrm{d}t.$$

Then, for the following Volterra integral equation

$$y(x) - \int_0^x k(x,t)y(t)\,\mathrm{d}t = f(x), \qquad x \in [0,1], \tag{20}$$

with Eq. (5), Eq. (9), Eq. (13) and Eq. (14) we have

$$\begin{aligned}\int_0^x k(x,t)y(t)\,\mathrm{d}t &\simeq \int_0^x \Psi(x)^T K\Psi(t)\Psi(t)^T C\mathrm{d}t \\ &= \Psi(x)^T K\left(\int_0^x \Psi(t)\Psi(t)^T C\mathrm{d}t\right) \\ &= \Psi(x)^T K\int_0^x \tilde{C}\Psi(t)\mathrm{d}t \\ &= \Psi(x)^T K\tilde{C}P\Psi(x).\end{aligned}$$

Then

$$\Psi^T(x)C = f(x) + \Psi(x)^T K\tilde{C}P\Psi(x). \tag{21}$$

By evaluating this equation in $2^{k-1}M$ points $\{x_i\}_{i=1}^{2^{k-1}M}$ in interval $[0,1)$ we have a system of linear equations

$$\Psi(x_i)^T C = f(x_i) + \Psi(x_i)^T K\tilde{C}P\Psi(x_i). \tag{22}$$

In calculating the elements of matrices of Galerkin method we often need to calculate the inner products of functions and the second Chebyshev wavelets basis. Here we discuss some formulae. By using p-points closed Gauss Chebyshev quadrature rule we have

$$\begin{aligned}&(f,\psi_{n,m})_{\omega_n} \\ &= \int_0^1 \omega_n(t)\psi_{n,m}(t)f(t)\,\mathrm{d}t \\ &= \int_{\frac{n-1}{2^{k-1}}}^{\frac{n}{2^{k-1}}} f(t)2^{\frac{k}{2}}\sqrt{\frac{2}{\pi}}\mathrm{U}_m(2^k t-2n+1)\omega(2^k t-2n+1)\,\mathrm{d}t \\ &= 2^{-\frac{k}{2}}\sqrt{\frac{2}{\pi}}\int_{-1}^1 f\left(\frac{t+2n-1}{2^k}\right)\mathrm{U}_m(t)\omega(t)\,\mathrm{d}t \\ &\simeq 2^{-\frac{k}{2}}\sqrt{\frac{2}{\pi}}\frac{\pi}{p+1}\sum_{l=1}^{p} f\left(\frac{\cos(l\pi/(p+1))+2n-1}{2^k}\right) \\ &\sin\left[\frac{(m+1)l\pi}{p+1}\right]\sin\frac{l\pi}{p+1}.\end{aligned}$$

for $n = 1,\cdots,2^{k-1}$, $m = 0,1,\cdots,M-1$.

5. Numerical examples

For showing efficiency of our numerical method, we consider the following examples.

Example 1. Consider the Fredholm integral equation of the second kind

$$y(x) - x\int_0^1 t^2 y(t)\,\mathrm{d}t = \sin x - x(\cos 1 + 2\sin 1 - 2), \tag{23}$$

with the exact solution $y(x) = \sin x$. Table 1 shows the comparison of the absolute error between exact solution and approximate solution for $k = 2, M = 3$ among Legendre wavelets(Leg for short), CAS wavelets and the second Chebyshev wavelets(Che for short) methods. Where y and y_n in the Table 1 denote the exact solution and the numerical solution, respectively.

Table 1: Numerical results of Example 1

x_r	$\lvert y-y_n\rvert$		
	Che	Leg	CAS
0.0	0.001269	0.001013	0.123699
0.2	0.000235	0.000280	0.008219
0.4	0.000199	0.000358	0.008096
0.6	0.000181	0.000284	0.008002
0.8	0.000160	0.000219	0.006031
1.0	0.000936	0.000734	0.080036

Example 2. Consider the following equation

$$y(x) - x\int_0^1 ty(t)\,\mathrm{d}t = e^x - x, \tag{24}$$

with exact solution $y(x) = e^x$. Table 2 shows the comparison of the absolute error between exact solution and approximate solution for $k = 2, M = 4$ among Legendre wavelets(Leg for short), CAS wavelets and the second Chebyshev wavelets(Che for short) methods.

Table 2: Numerical results of Example 2

x_r	$\lvert y-y_n\rvert$		
	Che	Leg	CAS
0.0	0.000064	0.000047	0.140991
0.2	0.000007	0.000011	0.012311
0.4	0.000015	0.000020	0.004524
0.6	0.000025	0.000032	0.003513
0.8	0.000015	0.000019	0.023426
1.0	0.000114	0.000081	0.303905

Example 3. Consider the following Volterra integral equation[10,12]

$$y(x) - x\int_0^x (xt^2 - t)y(t)\,\mathrm{d}t = -\frac{3}{4}x^6 + \frac{1}{3}x^5 + x^4 - \frac{1}{2}x^3 + 3x - 1, \tag{25}$$

with exact solution $y(x) = 3x - 1$. Table 3 shows the comparison of the absolute error between exact solution and approximate solution for $k = 2, M = 3$ among Legendre wavelets(Leg for short), Chebyshev wavelets[10] and the second Chebyshev wavelets(Che for short) methods.

Table 3: Numerical results of Example 3

x_r	$\lvert y-y_n\rvert$		
	Che	Leg	method in [10]
0.0	0.003479	0.000000	0.0000e-1
0.2	0.000182	0.000401	0.0234e-1
0.4	0.000124	0.001107	0.1084e-1
0.6	0.001746	0.002979	0.1743e-1
0.8	0.000055	0.003141	0.3524e-1
1.0	0.019876	0.009363	0.5923e-1

The results of Example 1 and Example 2 show that the second Chebyshev wavelets method is the same or sightly better than the Legendre case and is more better than the CAS wavelets method. Because CAS wavelets is a period wavelets, it is suitable for the periodic problems. The table of example 3 shows that the degree of accuracy of the second Chebyshev wavelets operational matrix method used for solving the Volterra integral equation is better than the Chebyshev wavelets and Legendre wavelets operational matrix method.

6. Conclusions

The second Chebyshev wavelets operational matrix of integration and its product operational matrix have been obtained in general and used for solving the integral equations. The present method reduces an integral equation into a set of algebraic equations. Some examples are included to demonstrate the superiority of our method. Moreover, the method in this paper can also be used for nonlinear integral equations and integro-differential equations.

References

[1] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms I," *Commun. Pure Appl. Math.*, vol. 44, pp. 141-183, Mar. 1991.

[2] Q. B. Fan, *Wavelet Analysis*. Wuhan, China: Wuhan University Press, 2008.

[3] K. Maleknejad, F. Mirzaee, "Using rationalized Haar wavelet for solving linear integral equations," *Appl. Math. Comp.*, vol. 160, pp. 579-587, Jan. 2005.

[4] E. Babolian, A. Shahsavaran, "Numerical solution of nonlinear Fredholm integral equations of the second kind using Haar wavelets," *Journal of Computational and Applied Mathematics*, vol. 225, pp. 87-95, Mar. 2009.

[5] Ü. Lepik, "Numerical solution of differential equations using Haar wavelets," *Mathematics and Computers in Simulation*, vol. 68, pp. 127-143, Apr. 2005.

[6] S. Yousefi, A. Banifatemi, "Numerical solution of Fredholm integral equations by using CAS wavelets," *Appl. Math. Comp.*, vol. 183, pp. 458-463, Dec. 2006.

[7] D. F. Han, X. F. Shang, "Numerical solution of integro-differential equations by using CAS wavelet operational matrix of integration," *Appl. Math. Comp.*, vol. 194, pp. 460-466, Dec. 2007.

[8] C. Cattani, A. Kudreyko, "Harmonic wavelet method towards solution of the Fredholm type integral equations of the second kind," *Appl. Math. Comp.*, vol. 215, pp. 4164-4171, Feb. 2010.

[9] E. Babolian, F. Fattahzadeh, "Numerical solution of differential equations by using Chebyshev wavelet operational matrix of integration," *Appl. Math. Comp.*, vol. 188, pp. 417-426, May. 2007.

[10] E. Babolian, F. Fattahzadeh, "Numerical computation method in solving integral equations by using Chebyshev wavelet operational matrix of integration," *Appl. Math. Comp.*, vol. 188, pp. 1016-1022, May. 2007.

[11] Y. L. Li, "Solving a nonlinear fractional differential equation using Chebyshev wavelets ," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 2284-2292, Sept. 2010.

[12] K. Maleknejad, M. T. Kajani, Y. Mahmoudi, "Numerical solution of linear Fredholm and Volterra integral equation of the second kind by using Legendre wavelets," *Kybernetes, Int. J. Sys. Math.*, vol. 32, pp. 1530-1539, 2003.

[13] X. Y. Zheng, X. F. Yang, "Techniques for solving integral and differential equations by Legendre wavelets," *International Journal of Systems Science.*, vol. 40, pp. 1127-1137, Nov. 2009.

[14] M. Razzaghi, S. Yousefi, " The Legendre wavelets operational matrix of integration," *Int. J. Syst. Sci.*, vol. 32, pp. 495-502, 2001.

[15] M. Razzaghi, S. Yousefi, "Legendre wavelets method for the solution of nonlinear problems in the calculus of variations," *Mathematical and Computer Modelling*, vol. 34, pp. 45-54, July. 2001.

[16] M. T. Kajani, A. H. Vencheh, and M. Ghasemi, "The Chebyshev wavelets operational matrix of integration and product operation matrix," *International Journal of Computer Mathematics*, vol. 86, pp. 1181-1125, July. 2009.

A New Method Based on Operational Matrices of Bernstein Polynomials for Nonlinear Integral Equations

K. Maleknejad[1], B. Basirat[1,2], and E. Hashemizadeh[1]
[1]Department of Mathematics, Karaj Branch, Islamic Azad University, Karaj, Iran
[2]Department of Mathematics, Birjand Branch, Islamic Azad University, Birjand, Iran

Abstract—*An approximation method based on operational matrices of Bernstein polynomials used for the solution of Hammerstein integral equations. The operational matrices of these functions are utilized to reduce a nonlinear Hammerstein and Volterra Hammerstein integral equation to a system of nonlinear algebraic equations. The method is computationally very simple and attractive, and applications are demonstrated through illustrative examples. The results obtained are compared by the known results.*

Keywords: operational matrix; Bernstein polynomial; Hammerstein integral equation.

1. Introduction

In this work, we consider the nonlinear integral equations of Hammerstein and Volterra-Hammerstein types that take the following forms respectively

$$u(x) = f(x) + \lambda \int_0^1 k(x,s)\psi(s,u(s))ds, \quad 0 \le x \le 1 \quad (1)$$

$$u(x) = f(x) + \lambda \int_0^x k(x,s)\psi(s,u(s))ds, \quad 0 \le x \le 1 \quad (2)$$

where λ is a real known constant and f, g and k are assume to be in L^2, with $\psi(x,u(x))$ nonlinear in u. We assume that Eqs.(1) and (2) have a unique solution $u(x)$ to be determined.
The nonlinear Hammerstein integral equations (1) arise as a reformulation of two-point boundary value problems with a certain nonlinear boundary condition, [1]. Many problems in mathematical physics, contact problems in the theory of elasticity, and mixed boundary value problems are transformed into Volterra Hammerstein integral equations (2), see ([2]-[4]).

Bernstein polynomials play a prominent role in various areas of mathematics. These polynomials have been frequently used in the solution of integral equations, differential equations and approximation theory; see, e.g., ([5]-[8]).
Recently Yousefi and Behroozifar derived the operational matrices of Bernestain polynomials [9], in this work we proposed a method based on operational matrices of Bernstein polynonials for numerical solution of Hammerstein integral equation (1) and Volteral Hammerstein integral equation (2).

2. Bernstein polynomials and their properties

2.1 Definition of Bernstein polynomials

The Bernstein basis polynomial of degree n are defined by

$$B_{i,n}(x) = \binom{n}{i} x^i (1-x)^{n-i}, \quad (3)$$

By using binomial expansion of $(1-x)^{n-i}$, we have

$$\binom{n}{i} x^i (1-x)^{n-i} = \sum_{k=0}^{n-i} (-1)^k \binom{n}{i}\binom{n-i}{k} x^{i+k}. \quad (4)$$

Now, we define

$$\Phi(x) = [B_{0,n}(x), B_{1,n}(x), ..., B_{n,n}(x)]^T, \quad (5)$$

where we can have

$$\Phi(x) = AT_n(x), \quad (6)$$

that A is an $(n+1)\times(n+1)$ upper triangular matrix with rows

$$A_{i+1} = \Big[\overbrace{0,0,...,0}^{i \text{ times}}, (-1)^0\tbinom{n}{i}\tbinom{n-i}{0}, (-1)^1\tbinom{n}{i}\tbinom{n-i}{1}, ..., (-1)^{m-i}\tbinom{n}{i}\tbinom{n-i}{n-i}\Big],$$

and $T_n(x)$ is an $(n+1)\times 1$ matrix as follows

$$T_n(x) = \begin{bmatrix} 1 \\ x \\ \vdots \\ x^n \end{bmatrix}.$$

2.2 Function approximation

Suppose that $H = L^2[0,1)$ is a Hilbert space with the inner product that is defined by $(f,g) = \int_0^1 f(x)g(x)dx$ and $Y = Span\{B_{0,n}(x), B_{1,n}(x), ..., B_{n,n}(x)\}$ is a finite dimensional and closed subspace, therefore Y is a complete subspace of H. So, if f is an arbitrary element in H, it has a unique best approximation out of Y such as y_0, that is [10]

$$\exists y_0 \in Y \quad s.t \quad \forall y \in Y \quad \|f - y_0\| \le \|f - y\|,$$

this implies that

$$\forall y \in Y \quad (f - y_0, y) = 0, \quad (7)$$

since $y_0 \in Y$ so there exist coefficients $c_0, c_1, ..., c_n$ such that

$$y_0 = c^T \Phi(x),$$

where

$$c^T = [c_0, c_1, ..., c_n]. \tag{8}$$

By (7)

$$(f - c^T \Phi(x), B_{i,n}(x)) = 0, \qquad i = 0, 1, \ldots, n.$$

For simplicity we write

$$c^T (\Phi(x), \Phi(x)) = (f, \Phi(x)), \tag{9}$$

where vector$(f, \Phi(x)) = \int_0^1 f(x)\Phi(x)dx$, and $(\Phi(x), \Phi(x))$ is an $(n+1) \times (n+1)$ matrix and is said dual matrix of $\Phi(x)$. Let

$$\mathbf{D} = (\Phi(x), \Phi(x)) = A[\int_0^1 T_n(x) T_n^T(x) dx] A^T = AHA^T, \tag{10}$$

H is a Hilbert matrix. We can specify the element of $\mathbf{D}$ as:

$$\mathbf{D}_{(i+1),(j+1)} = \int_0^1 B_{i,n}(x) B_{j,n}(x) dx = \frac{\binom{n}{i}\binom{n}{j}}{(2n+1)\binom{2n}{i+j}}, \tag{11}$$

where $i, j = 0, 1, ..., n$. Any function $f(x) \in L^2[0, 1]$ can be expand in Bernstein basis as $f(x) \simeq c^T \Phi(x)$, where from Eqs.(9) and (10), we obtain

$$c = \mathbf{D}^{-1}(f, \Phi(x)). \tag{12}$$

We can also approximate the function $k(x, s) \in L^2([0, 1] \times [0, 1])$ as follows

$$k(x, s) \simeq \Phi^T(x) K \Phi(s), \tag{13}$$

where K is an $(n+1) \times (n+1)$ matrix that

$$K_{ij} = \frac{(\Phi_i(x), (k(x, s), \Phi_j(s)))}{(\Phi_i(x), \Phi_i(x))(\Phi_j(s), \Phi_j(s))}, \tag{14}$$

for $i, j = 1, 2, ..., n$. From (10) we can have

$$K = \mathbf{D}^{-1}\big(\Phi(x), (k(x, s), \Phi(s))\big)\mathbf{D}^{-1}. \tag{15}$$

2.3 Operational matrix of integration

The integration of the vector $\Phi(x)$ defined in Eq.(5) is given by

$$\int_0^x \Phi(x')dx' \simeq \mathbf{P}\Phi(x), \tag{16}$$

where $\mathbf{P}$ is the $(n+1) \times (n+1)$ operational matrix for integration and is given in [9] as

$$\int_0^x \Phi(x')dx' = A_p X_p, \tag{17}$$

where A_p is an $(n+1) \times (n+1)$ matrix,

$$A_p = A \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & \frac{1}{2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \frac{1}{n+1} \end{bmatrix} \quad \text{and} \quad X_p = \begin{bmatrix} x \\ x^2 \\ \vdots \\ x^{n+1} \end{bmatrix}. \tag{18}$$

Now, we approximate the elements of vector X_p in terms of $\Phi(x)$. By (6), we have $T_n(x) = A^{-1}\Phi(x)$ then for $k = 0, 1, \ldots, n$,

$$x^k = A^{-1}_{[k+1]}\Phi(x),$$

where $A^{-1}_{[k+1]}$ is $k+1$-th row of A^{-1} for $k = 0, 1, \ldots, n$. We just need to approximate $x^{n+1} \simeq c^T_{n+1}\Phi(x)$. By using (12) and (11), we have

$$c_{n+1} = \mathbf{D}^{-1} \int_0^1 x^{n+1}\Phi(x)dx = \frac{\mathbf{D}^{-1}}{2n+2} \begin{bmatrix} \frac{\binom{n}{0}}{\binom{2n+1}{n+1}} \\ \frac{\binom{n}{1}}{\binom{2n+1}{n+2}} \\ \vdots \\ \frac{\binom{n}{n}}{\binom{2n+1}{2n+1}} \end{bmatrix}.$$

Let

$$B = \begin{bmatrix} A^{-1}_{[2]} \\ A^{-1}_{[3]} \\ \vdots \\ A^{-1}_{[n+1]} \\ c^T_{n+1} \end{bmatrix}, \tag{19}$$

then,$X_p \simeq B\Phi(x)$. Therefore we have the operational matrix of integration $\mathbf{P} = A_p B$.

2.4 Product operational matrix

It is always necessary to evaluate the product of $\Phi(x)$ and $\Phi(x)^T$, that is called the product matrix of Bernstein polynomials basis. Let

$$\mathbf{\Pi}(x) = \Phi(x)\Phi(x)^T, \tag{20}$$

where $\mathbf{\Pi}(x)$ is an $(n+1) \times (n+1)$ matrix. By multiplying the matrix $\mathbf{\Pi}(x)$ in vector c that is defined in Eq.(8) we obtain

$$c^T \mathbf{\Pi}(x) = \Phi(x)^T \widehat{\mathbf{C}}, \tag{21}$$

where $\widehat{\mathbf{C}}$ is an $(n+1) \times (n+1)$ matrix and called the coefficient matrix. So we have

$$c^T \mathbf{\Pi}(x) = \Big[\textstyle\sum_{i=0}^n c_i B_{i,n}(x), \sum_{i=0}^n c_i x B_{i,n}(x), \ldots, \sum_{i=0}^n c_i x^n B_{i,n}(x) \Big] A^T. \tag{22}$$

Now, we approximate all functions $x^k B_{i,n}(x)$ in terms of $\Phi(x)$. Let

$$e_{k,i} = \begin{bmatrix} e_0^{k,i} \\ e_1^{k,i} \\ \vdots \\ e_n^{k,i} \end{bmatrix}, \tag{23}$$

by (12), we have $x^k B_{i,n}(x) \simeq e_{k,i}\Phi(x)$, $i,k = 0,1,\ldots,n$. By using (12) and (11) for $i,k = 0,1,\ldots,n$, we have

$$e_{k,i} = \mathbf{D}^{-1}\int_0^1 x^k B_{i,n}(x)\Phi(x)dx = \frac{\mathbf{D}^{-1}\binom{n}{i}}{2n+k+1}\begin{bmatrix} \frac{\binom{n}{0}}{\binom{2n+k}{i+k}} \\ \frac{\binom{n}{1}}{\binom{2n+k}{i+k+1}} \\ \vdots \\ \frac{\binom{n}{n}}{\binom{2n+k}{i+k+n}} \end{bmatrix}. \tag{24}$$

Therefore

$$\begin{aligned} &\textstyle\sum_{i=0}^n c_i x^k B_{i,n}(x) \simeq \sum_{i=0}^n c_i(\sum_{j=0}^n e_j^{k,i} B_{j,n}(x)) \\ &= \Phi(x)^T \begin{bmatrix} \sum_{i=0}^n c_i e_0^{k,i} \\ \sum_{i=0}^n c_i e_1^{k,i} \\ \vdots \\ \sum_{i=0}^n c_i e_n^{k,i} \end{bmatrix} \\ &= \Phi(x)^T[e_{k,0}, e_{k,1}, \ldots, e_{k,n}]c = \Phi(x)^T E_{k+1}c, \end{aligned} \tag{25}$$

where E_{k+1} is an $(n+1)\times(n+1)$ matrix, that has vectors $\widetilde{e_{k,i}}, k = 0,1,...,n$ for each column's. Then we define $\widetilde{E_{k+1}} = E_{k+1}c$ for $k = 0,1,\ldots,n$. If we choose an $(n+1)\times(n+1)$ matrix $\widetilde{C} = [\widetilde{E_1}, \widetilde{E_2}, \ldots, \widetilde{E_{n+1}}]$, then by (22) and (25) we have

$$c^T\mathbf{\Pi}(x) \simeq \Phi(x)^T\widetilde{C}A^T, \tag{26}$$

and therefore we have the coefficient matrix as $\widehat{\mathbf{C}} = \widetilde{C}A^T$.

3. Solution of Hammerstein Integral Equations

For solving Hammerstein integral equation (1), we let

$$z(s) = \psi(s,u(s)), \quad 0 \le s \le 1, \tag{27}$$

then we get

$$u(x) = f(x) + \int_0^1 k(x,s)z(s)ds. \tag{28}$$

Substituting (28) in (27) results,

$$z(x) = \psi(x, f(x) + \int_0^1 k(x,s)z(s)ds). \tag{29}$$

We approximate this equation as

$$z(x) = Z^T\Phi(x), \tag{30}$$

which Z and $\Phi(x)$ are defined with (5) and (8). By use of (10), (13) and (30) we have

$$\begin{aligned} \textstyle\int_0^1 k(x,s)z(s)ds &\simeq \int_0^1 \Phi^T(x)K\Phi(s)\Phi(s)^T Z ds \\ &= \Phi^T(x)K\int_0^1 \Phi(s)\Phi(s)^T ds Z \\ &= \Phi^T(x)K\mathbf{D}Z. \end{aligned} \tag{31}$$

Via Eqs.(29), (30) and (31) we get

$$Z^T\Phi(x) = \psi(x, f(x) + \lambda\Phi^T(x)K\mathbf{D}Z). \tag{32}$$

In order to find Z we collocate Eq.(32) in n nodal points of Newton–Cotes as,

$$x_p = \frac{2p-1}{2n}, \quad p = 1,2,\ldots,n, \tag{33}$$

then we have equation (32) as follows

$$Z^T\Phi(x_p) = \psi(x_p, f(x_p) + \lambda\Phi^T(x_p)K\mathbf{D}Z), \; p = 1,2,\ldots,n. \tag{34}$$

We can calculate the unknown vector Z from the above nonlinear system of equations. The Newton's iterative method is suitable for solving this nonlinear system. We used Mathematica 7 for obtaining our solutions. The required approximated solution $u(x)$ for our Hammerstein intgral equation (1), can be obtained by using Eqs.(28) and (30) as follows

$$u(x) = f(x) + \lambda\Phi^T(x)K\mathbf{D}Z. \tag{35}$$

4. Solution of nonlinear Volterra Integral equations

Consider the nonlinear Volterra integral equations given in (2). For solving these equations like previous section, we let $z(s) = \psi(s,u(s))$ for $0 \le s \le 1$. Then from Eq.(2) we get

$$z(x) = \psi(x, f(x) + \int_0^x k(x,s)z(s)ds). \tag{36}$$

By use of (21), (13), (30) and (16) we can write

$$\begin{aligned} \textstyle\int_0^x k(x,s)z(s)ds &\simeq \int_0^x \Phi^T(x)K\Phi(s)\Phi(s)^T Z ds \\ &= \Phi^T(x)K\int_0^x \mathbf{\Pi}(s)Z ds \\ &= \Phi^T(x)K\widehat{\mathbf{Z}}^T\int_0^x \Phi(s)ds \\ &= \Phi^T(x)K\widehat{\mathbf{Z}}^T\mathbf{P}\Phi(x). \end{aligned} \tag{37}$$

After using Eqs.(37) and (36) we get

$$Z^T\Phi(x) = \psi(x, f(x) + \lambda\Phi^T(x)K\widehat{\mathbf{Z}}^T\mathbf{P}\Phi(x)). \tag{38}$$

By collocating Eq.(38) in n nodal points (33) we have,

$$Z^T\Phi(x_p) = \psi(x_p, f(x_p) + \lambda\Phi^T(x_p)K\widehat{\mathbf{Z}}^T\mathbf{P}\Phi(x_p)), \tag{39}$$

for $p = 1,2,\ldots,n$. After solving nonlinear system (39) we get Z, and by use of $z(x) = Z^T\Phi(x)$ we will have the approximation solution of Volterra–Hammerstein integral equations (2) as

$$u(x) = f(x) + \int_0^x k(x,s)z(s)ds, \tag{40}$$

and from above we know that Eq.(40) can be evaluated by

$$u(x) = f(x) + \lambda\Phi^T(x)K\widehat{\mathbf{Z}}^T\mathbf{P}\Phi(x). \tag{41}$$

5. Error estimation

The Bernstein polynomials can be expressed in terms of some orthogonal polynomials, such as Chebychev polynomial $\chi_n(x)$ of second kind ([7], [11]). It can be shown that

$$B_{i,n}(x) = \frac{1}{2^n}\binom{n}{i}\sum_{s=0}^{n} d_s^{i,n}\frac{1}{2^s}\sum_{m=0}^{[\frac{s}{2}]}(\binom{s}{m} - \binom{s}{m+1})\chi_{s-2m}(x), \tag{42}$$

$$d_s^{i,n} = \sum_k (-1)^{s-k}\binom{i}{k}\binom{n-i}{s-k}.$$

Expand $f(x)$ in the approximated form of Bernstein polynomials

$$f(x) \simeq p_n(x) = \sum_{i=0}^{n} a_i B_{i,n}(x), \tag{43}$$

Thus, it is eventually expressed as

$$p_n(x) = \sum_{j=0}^{n} b_j \chi_j(x), \tag{44}$$

where b_j can be expressed in terms of a_i, $i, j = 0, ..., n$. If $u_j(x) = \sqrt{\frac{2}{\pi}}\chi_j(x)$, then $u_j(x)$, $j = 0, ..., n$, form an orthonormal polynomial basis in $[-1, 1]$ with respect to weight function $\omega(x) = (1 - x^2)^{\frac{1}{2}}$, that can be mapped to $[0, 1]$. Therefor, this procedure yields

$$p_n(x) = \sum_{j=0}^{n} \sqrt{\frac{\pi}{2}} b_j u_j(x), \tag{45}$$

Golberg and Chen ([12]) proved that when we are approximating a continuously differentiable function ($f \in C^r$, $r > 0$) by Chebychev polynomials, then

$$\|f - p_n\|_\omega < c_0 n^{-r}, \tag{46}$$

where c_0 is some constant.

We assume throughout this paper, the following conditions on k, f, and ψ for Eqs.(1) and (2), unless stated otherwise. Define $k_x \equiv k(x, s)$ for $x, s \in [0, 1]$ to be the x section of k:

1. $\lim_{x \longrightarrow \tau} \|k_x - k_\tau\| = 0$, $\tau \in [0, 1]$;
2. $M \equiv \sup_{0 \le x,s \le 1} |k(x, s)| < \infty$;
3. $f \in C[0, 1]$;
4. $\psi(s, x)$ is continuous in $s \in [0, 1]$ and Lipschitz continuous in $x \in R$, i.e., there exists a constant $C_1 > 0$ for which

$$|\psi(s, x_1) - \psi(s, x_2)| \le C_1 |x_1 - x_2| \; for \; all \; x_1, x_2 \in R.$$

We denote the $u_n(x)$ and $u(x)$ show the approximate and exact solutions of the integral equations respectively.

Theorem. The solution of Hammerstein and Volterra–Hammerstein integral equation by using Bernstein basis converges if $0 < \alpha < 1$.

Proof. For Hammerstein integral equation by assumptions 2 and 4, we see that there exists a constant $\alpha = |\lambda| M C_1 > 0$ such that

$$\|u_n(x) - u(x)\| = \max_{x \in [0,1]} |u_n(x) - u(x)|$$

$$\le \max_{x \in [0,1]} |\lambda| \int_0^1 |k(x, s)| |\psi(s, u_n(s)) - \psi(s, u(s))| ds$$

$$\le |\lambda| M C_1 \max_{x \in [0,1]} |u_n(x) - u(x)| \le \alpha \max_{x \in [0,1]} |u_{nm}(x) - u(x)|.$$

We get $(1-\alpha)\|u_n(x) - u(x)\| \le 0$ and choose $0 < \alpha < 1$, by increasing n, it implies $\|u_n(x) - u(x)\| \longrightarrow 0$ as $n \longrightarrow \infty$.

The similar proof for Volterra–Hammerstein integral equations can be obtained, because $\int_0^x |k(x, t)| |dt \le \int_0^1 |k(x, t)| dt$, for $0 < x < 1$. □

6. Numerical examples

6.1 Example 1

Consider the nonlinear Fredholm integral equation

$$u(x) = f(x) + \int_0^1 2x^2 s \ln(u(s)) ds, \tag{47}$$

where

$$f(x) = 1 + x + \Big(1 - \frac{3}{2}\ln(3) + \frac{\sqrt{3}}{6}\pi\Big)x^2,$$

the exact solution is $1 + x + x^2$. Table 1 shows the present method results for example 1 in comparison with method of [13]. The superiority of Bernstein operational matrices method compared with Taylor polynomial method is clear here, because with the same number of basis functions we get very better results.

Table 1: Approximate and exact solutions for Example 1.

x_i	Present method $n = 6$	Method of [13] $N = 6$	Exact solution
0.0	1.000000	1.000000	1
0.2	1.239999	1.238432	1.24
0.4	1.559999	1.553726	1.56
0.6	1.959999	1.945884	1.96
0.8	2.439999	2.414905	2.44
1.0	2.999999	2.960788	3

6.2 Example 2

Consider the Hammerstein integral equation

$$u(x) = f(x) + \int_0^1 \sin(x+s)\ln(u(s))ds, \quad 0 \le x \le 1, \quad (48)$$

where $f(x) = e^x - 0.382\sin(x) - 0.301\cos(x)$, and the exact solution is $u(x) = e^x$ [14]. The computational results are obtained by the present method with $n = 5$, we compared our results by results of method [14]. In this comparison the number of present method basis functions is 5 but the number of basis functions for method of [14] are 32 and the results have almost same accuracy, so Bernestain method is superior than hybrid Legendre and Block–Pulse method for solving Hammerstein integral equation.

Table 2: Approximate and exact solutions for Example 2.

x	Present method with $n = 5$	Method of [14] with $m = 4$, $n = 8$	Exact
0.0	1.0001824226	1.0001817942	1
0.2	1.2215473608	1.2215472834	1.2214027582
0.4	1.4919260348	1.4919261952	1.4918246976
0.6	1.8221730268	1.8221731864	1.8221188004
0.8	2.2255460310	2.2255459923	2.2255409285
1.0	2.7182380285	2.7182373557	2.7182818285

6.3 Example 3

Consider the nonlinear Volterra integral equation

$$u(x) = \frac{3}{2} - \frac{1}{2}e^{-2x} + \int_0^x (u^2(s) + u(s))ds, \quad (49)$$

where the exact solution is e^{-x}.
For this example we consider the L^2-norm of errors which can be shown by

$$E_2 = \Big(\int_0^1 \big[u(x) - u_n(x)\big]^2 dx \Big)^{1/2}.$$

The comparison among the Bernstein operational matrices errors E2 with $n = 4, 8, 16, 32$ beside triangular function [15] errors with $m = 4, 8, 16, 32$ are shown in Table 3. The primacy of present method compared with triangular function method is obvious here because by the same number of basis function present method $E2$ errors are very lower.

Table 3: Errors E2 for Example 3.

$n\|m$	Present method	Method of [15]
4	0.000068193353	0.003738014268
8	0.000000084293	0.000937018240
16	0.000000000000	0.000234412613
32	0.000000000000	0.002374324588

7. Conclusions

This work present a numerical approach for solving Hammerstein and Volterra Hammerstein integral equations by the operational matrices of Bernstein polynomials. The dual matrix $\mathbf{D}$, operational matrix of integration $\mathbf{P}$, product matrix $\mathbf{\Pi}$ and coefficient matrix $\widehat{\mathbf{C}}$ beside collocation method were used for transform the Hammerstein and Volterra Hammerstein integral equations to a nonlinear system of algebraic equations that can be solved by Newton's method. The Bernstein polynomials operational matrices method is very simple and attractive. The implementation of current approach in analogy to existed methods is more convenient and the accuracy is high. The numerical examples that have been presented in the paper and the compared results support our claim.

References

[1] L. M. Delves and J. L. Mohamed, *Computational Methods for Integral Equations*, Cambridge University Press, Cambridge, 1985.

[2] M. A. Abdou, "On the solution of linear and nonlinear integral equation," *Applied Mathematics and Computation*, 146 (2-3), 857–871, 2003.

[3] M. Ganesh and M. C. Joshi, "Numerical solvability of Hammerstein integral equations of mixed type," *IMA Journal of Numerical Analysis*, 11 (1), 21–31, 1991.

[4] R. K. Miller, *Nonlinear Volterra Integral Equations,Mathematics Lecture Note Series*, Mathematics Lecture Note Series, W. A. Benjamin, California, 1971.

[5] K. Maleknejad, E. Hashemizadeh and R. Ezzati, "A new approach to the numerical solution of Volterra integral equations by using Bernsteins approximation," *Commun Nonlinear Sci Numer Simulat*, 16, 647–655, 2011.

[6] E. H. Doha, A. H. Bhrawy and M. A. Saker, "Integrals of Bernstein polynomials: An application for the solution of high even-order differential equations," *Applied Mathematics Letters*, 24, 559–565, 2011.

[7] B. N. Mandal and S. Bhattacharya, "Numerical solution of some classes of integral equations using Bernstein polynomials," *Appl Math Comput*, 190, 1707–1716, 2007.

[8] T. J. Rivlin, *An introduction to the approximation of functions*, New York: Dover Publications, 1969.

[9] S. A. Yousefi and M. Behroozifar, "Operational matrices of Bernstein polynomials and their applications," *International Journal of Systems Science*, 41(6), 709–716, 2010.

[10] E. Kreyszig, *Introductory Functional Analysis with Applications,*, New York: John Wiley and sons. Inc, 1978.

[11] M. A. Snyder, *Chebychev Methods in Numerical Approximation*, Prentice-Hall: Englewood Clifs, NJ, 1996.

[12] M. A. Golberg and C. S. Chen, "Discrete Projection Methods for Integral Equations," *Southampton: Computational Mechanics Publications*, 178–306, 1997.

[13] Y. Mahmoudi, "Taylor polynomial solution of non–linear Volterra-Fredholm integral equation," *International Journal of Computer Mathematics*, 82 (7), 881–887, 2005.

[14] K. Maleknejad, E. Hashemizadeh and B. Basirat, "Numerical solvability of Hammerstein integral equations based on hybrid Legendre and Block-Pulse functions," *In'l Conf. Par. and Dist. Proc. Tech. and Appl. PDPTA'10*, 1, 172–175, 2010.

[15] K. Maleknejad, H. Almasieh, M. Roodaki, "Triangular functions (TF) method for the solution of nonlinear Volterra–Fredholm integral equations," *Commun Nonlinear Sci Numer Simulat*, 15, 3293–3298, 2010.

Bootstrap Tail Thickness Estimation for Symmetric Alpha-Stable Random Variables

Brandon Franzke and Bart Kosko
Signal and Image Processing Institute
Department of Electrical Engineering
University of Southern California, Los Angeles, California 90089

Abstract—*A new theorem shows that a bootstrap algorithm can estimate the impulsivity or tail thickness of symmetric a-stable (SαS) signals. SαS bell curves include the Gaussian bell curve as a special but non-impulsive case. Signals grow more impulsive as the bell curve's tail thickness increases or as the tail-thickness parameter α falls from 2 toward 0. The thin-tailed Gaussian bell curve has $\alpha = 2$. The algorithm computes a statistic from SαS samples and then matches the test statistic against a continuum of precomputed values to the find the estimated tail thickness $\widehat{\alpha}$. The theorem and a corollary show that the statistic is invertible because it is a continuous bijection. So the boostrapped $\widehat{\alpha}$ is a consistent estimator of α in general. Simulations show that $\widehat{\alpha}$ is robust for signals with $\alpha \in [0.2, 2]$ and that the estimator error decreases as the number of samples increases.*

1. Robust Estimation of Symmetric α-Stable Tail Thickness.

We show that a bootstrap algorithm can estimate the impulsiveness of a sequence of symmetric α-stable (SαS) random samples [1], [2], [3], [4], [5], [6]. The algorithm estimates the tail-thickness parameter α by interpolating a sample statistic between precomputed values. Figure 6 and Table 1 show that the algorithm applies to SαS random variables with $\alpha \in [0.2, 2]$. A theorem shows that each α corresponds to a unique value of a sample statistic $\tau(X_\alpha)$. A corollary to the theorem shows that the map is a bijection. The algorithm estimates α by calculating $\tau(X_\alpha)$ and then inverting the bijection (Figure 5). The estimator $\widehat{\alpha}$ applies to all finite sequences of independent and identically distributed (i.i.d.) SαS random variables.

Symmetric α-stable random variables have thick power-law tails and generalize the Gaussian probability density function (pdf) [7], [8], [9], [10], [11], [12], [13], [14]. The tail-thickness parameter α lies in $(0, 2]$ and controls the impulsivity of samples drawn from the random variable X_α. Figure 1 shows the inverse relation between α and the thickness of the bell-curve tails. Figure 2 shows how α controls impulsiveness of samples from the distribution. The Gaussian pdf takes $\alpha = 2$. The thicker-tailed Cauchy pdf takes $\alpha = 1$. The moments of an α-stable random variable are finite only up to order k for $k < \alpha$. Only the Gaussian random variable has finite second and higher-order moments. The usual central limit theorem states that a standardized sum of finite-variance random variables converges in distribution to the standard normal distribution $Z \sim N(0, 1)$ [15], [16]. The *g*eneralized central limit theorem states a similar result for infinite-variance α-stable random variables [17], [18]: a standardized sum of α-stable random variables converges in distribution to an α-stable random variable with the same α. It also shows this holds only for α-stable random variables.

Brandon Franzke and Bart Kosko are with the Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, USA (email: kosko@usc.edu)

Real noise tends to be impulsive. Natural sources of impulsive signals include condensed and soft matter physics [19], [20], [21], geophysics [22], meteorology [23], biology [24], economics [25], [26], [27], fractional kinetics [28], [29], and communications [14], [30], [31], [32], [33]. Many random models assume that the dispersion of a random variable is equal to its *finite* variance or its squared-error from the population mean. Impulsive signals violate this finite variance assumption in general. So these model may wrongly dismiss important "rare" events as outliers. Selection of a bell-curve signal model requires empirical tests to estimate the actual tail thickness. Finding the optimal SαS bell-curve for a given symmetric signal pattern is an open research problem.

The α-Stable Estimation Algorithm in Section 3 estimates α from a sequence of observed SαS random samples. The algorithm computes the estimator $\widehat{\alpha}$ in two stages: (1) it constructs a bijective τ-map between a test statistic $\tau(X_\alpha)$ and α and (2) it computes $\tau(X_\alpha)$ for observed random samples and then uses the τ^{-1}-map to find $\widehat{\alpha}$. The α-Stable Estimation Map Theorem in Section 2 ensures that the τ-map is unique. A corollary shows that the τ-map is a bijection and thus has an inverse $\widehat{\alpha} = \tau^{-1}(\tau(X_\alpha))$. The τ-map is also continuous. Thus $\widehat{\alpha}$ converges in probability to α and so is consistent because $\widehat{\alpha}$ is a consistent estimator of α since $\widehat{\tau(X_\alpha)}$ is a consistent estimator of $\tau(X_\alpha)$ [15]. Section 4 presents simulations to show that $\widehat{\alpha}$ is a good estimator for $\alpha \in [0.2, 2]$.

2. The α-Stable Estimation Map Theorem

The α-Stable Estimation Algorithm computes an estimator $\widehat{\alpha}$ of the tail-thickness parameter α. It computes a sample statistic $\tau(X_\alpha)$ from a sequence of observed SαS samples and then estimates α through the τ^{-1}-map that maps from $\tau(X_\alpha)$ to α. The α-Stable Estimation Map Theorem guarantees that on average the τ-map generates distinct $\tau(X_\alpha)$ for two independent sequences of SαS independent and identically distributed (i.i.d.) random variables X_{α_1} and X_{α_2} if $\alpha_1 \neq \alpha_2$. The α-Stable Estimation Algorithm uses a corollary to ensure the inverse exists. The corollary thus allows the algorithm to estimate α through the τ^{-1}-map.

The algorithm computes a test statistic for X_α that resembles a vector p-norm. The test statistic is finite because the p^{th}-sample moment of a finite sequence of such realizations is finite for any finite $p > 0$. Suppose X_α is a sequence of N i.i.d. SαS random variables X_α with pdf $f_\alpha(x)$. Suppose the random variable has $\alpha \in (0, 2]$ and has unit dispersion: $\gamma = 1$. Suppose further that N is finite. Then the sample maximum

$$H(x_\alpha) = \max_{1 \leq k \leq N} |x_\alpha[k]| < \infty \tag{1}$$

almost surely since $\lim_{x \to \infty} f_\alpha(x) = 0$.

Define g_p by the length-normalized sample p-norm for finite

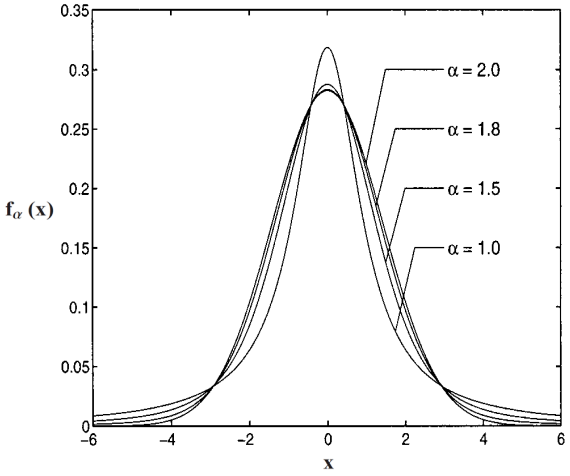


Fig. 1. Symmetric α-stable probability density functions. The figure shows SαS probability density functions for $\alpha = 2.0$ (Gaussian), 1.8, 1.5, and 1.0 (Cauchy). The thickness of the bell curve tails increases as α decreases. Thicker tails correspond to more impulsive samples. The Gaussian bell curve is the only SαS distribution with finite moments of order $k \geq 2$. Choosing α for a given application is an empirical question.

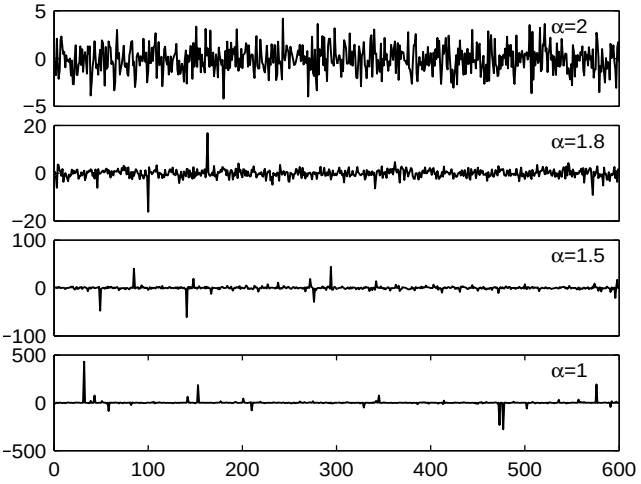


Fig. 2. Impulsive samples from SαS random variables with unit dispersion. The figure shows SαS realizations for $\alpha = 2$ (Gaussian), 1.8, 1.5, and 1 (Cauchy). The scale differs by two orders of magnitude between $\alpha = 2$ and $\alpha = 1$. Only the Gaussian samples have finite variance and no impulsiveness.

$p > 0$:

$$g_p(x_\alpha) = \frac{1}{N}\|x_\alpha\|_p^p \tag{2}$$

where $\|\cdot\|_p$ is the usual p-norm on $\mathbb{R}^N$:

$$\|x\|_p^p = \sum_{k=1}^{N} |x_k|^p = |x_1|^p + |x_2|^p + \cdots + |x_N|^p . \tag{3}$$

The α-Stable Estimation Map Theorem shows that g_p is injective (1-to-1) with respect to α. The corollary that follows shows that g_p is a continuous bijection. It also shows how to construct the inverse g_p^{-1}. The α-Stable Estimation Algorithm uses this result to justify that $\widehat{\alpha}$ is a consistent estimator of α. The proof of the theorem is lengthy and appears in the Appendix.

α-Stable Estimation Map Theorem. *Suppose X_{α_1} and X_{α_2} are two independent sequences of N i.i.d. SαS random variables with probability density functions $f_{\alpha_1}(x)$ and $f_{\alpha_2}(x)$. Suppose $\alpha_i \in (1, 2]$ with unit dispersion: $\gamma = 1$. Fix $p > 0$ and define $g_p(X)$ as*

$$g_p(X) = \frac{1}{N}\|X\|_p^p = \frac{1}{N}\sum_{k=1}^{N} |X_k| . \tag{4}$$

Define the maximum function $H(X)$ as

$$H(X) = \max_{1 \leq k \leq N} |X_k| . \tag{5}$$

Then there exists an N_0 such that

$$E\left[g_p(X_{\alpha_1}) | H(X_{\alpha_1}) = h\right] = E\left[g_p(X_{\alpha_2}) | H(X_{\alpha_2}) = h\right] \tag{6}$$

for $h > 2$ and all sequences X_{α_1} and X_{α_2} with length $N \geq N_0$ only if $\alpha_1 = \alpha_2$.

The proof of the theorem relies on the following two lemmas to show that $E\left[g_p(X_\alpha) | H(X_\alpha) = h\right]$ strictly decreases on α. The first lemma shows that a denominator term for $E\left[g_p(X_\alpha) | H(X_\alpha) = h\right]$ strictly increases. The second lemma shows that a numerator term for $E\left[g_p(X_\alpha) | H(X_\alpha) = h\right]$ strictly increases. The proofs for the lemmas appear in the Appendix.

Lemma 1. *Suppose X_{α_1} and X_{α_2} are two independent sequences of N i.i.d. SαS random variables with probability density functions $f_{\alpha_1}(x)$ and $f_{\alpha_2}(x)$ and cumulative distribution functions $F_{\alpha_1}(x)$ and $F_{\alpha_2}(x)$. Suppose $\alpha_i \in (1, 2]$ with $\alpha_1 \neq \alpha_2$ and unit dispersion: $\gamma = 1$. Then there exists $N_0 < \infty$ such that $P\left[\max\left|X_{\alpha_1}\right| = h\right] < P\left[\max\left|X_{\alpha_2}\right| = h\right]$ for all $N \geq N_0$ and $h > 2$.*

Lemma 2. *The function $A(\alpha) = \int_{-H}^{H} |x|^p f_\alpha(x)\, dx$ strictly decreases on $\alpha \in (1, 2]$ for $p > 0$ if $H > 2$.*

The α-Stable Estimation Algorithm uses the corollary below to show that the expected value of the τ-map is a bijection between $\alpha \in (1, 2]$ and $\tau(X_\alpha)$. The algorithm exploits this fact to estimate α with the τ^{-1}-map. The proof of the corollary is lengthy and appears in the Appendix.

τ-map Invertibility Corollary. *Define $g_p(X)$ by* (4) *and the maximum function $H(X)$ by* (5)*. Define X_{α_1} and X_{α_2} as in the α-Stable Estimate Map Theorem. Suppose that the conditions hold such that a finite N_0 exists. Suppose further that $h < \infty$. Then $\tau(X_\alpha) = \mathscr{G}(\alpha) = E\left[g_p(X_\alpha) | H(X_\alpha) = h\right]$ is a bijection from $\alpha \in (1, 2]$ onto $\tau(X_\alpha) \in [\mathscr{G}(1), \mathscr{G}(2)]$.*

3. The α-Stable Estimation Algorithm

The τ-map Invertibility Corollary guarantees that the test statistic maps to a unique $\widehat{\alpha}$ on average. But the finite length of X_α means that outliers may dominate the calculation of $\tau(X_\alpha)$. The α-Stable Estimation Algorithm bootstraps to reduce the sensitivity of the algorithm to outliers. The τ-map Invertibility Corollary also establishes that the τ-map is continuous for $\alpha \in (1, 2]$. Thus the bootstrap estimator $\widehat{\alpha}$ is a consistent estimator of α in general [34], [35], [36] because $\widehat{\tau(X_\alpha)}$ is a consistent estimator of $\tau(X_\alpha)$.

The α-Stable Estimation Algorithm consists of two stages: (1) it uses randomly generated sequences of SαS observations to construct a bijective τ-map between $\tau(X_\alpha)$ and α and (2) it computes $\tau(X_\alpha)$ for the observed unknown samples and then uses the τ^{-1}-map to find $\widehat{\alpha}$

Stage 1 does not depend on the particular unknown random sequence X_α. So the algorithm preconstructs the τ-map. The algorithm constructs the τ-map by computing $\tau(X_\alpha)$ for representative

SαS sequences with $0 < \alpha \leq 2$. Stage 2 uses $\tau(X_\alpha)$ from stage 1 to characterize the unknown signal X_α. It then maps $\tau(X_\alpha)$ to $\widehat{\alpha}$ with the τ^{-1}-map. The τ-map is unique and so stage 2 can use the map to calculate $\tau(X_\alpha)$ for arbitrary X_α.

Figure 3 shows results from the test statistic $\tau(X_\alpha)$ computation for $\alpha \in [0.4, 2]$. The brackets show 90% confidence bands for $\tau(X_\alpha)$ from 50 independent sequences for each α tested. The blue line shows the median of $\tau(X_\alpha)$.

3.1 Stage 1: Construct τ-map

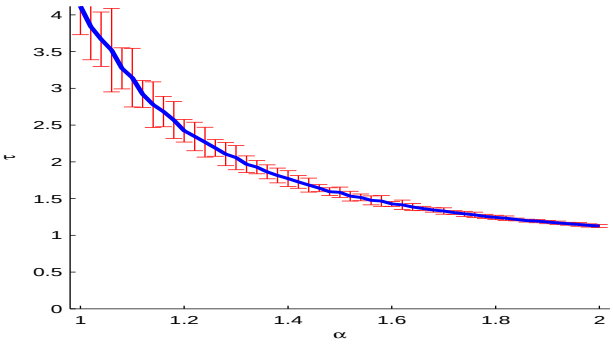
Stage 1 constructs the τ-map from α to $\tau(X_\alpha)$. The algorithm uses the τ^{-1}-map to find $\widehat{\alpha}$ from $\tau(X_\alpha)$ since the τ-map Invertibility Corollary shows that the τ-map is a bijection for $\alpha \in (1, 2]$.

The τ-map takes a finite sequence of i.i.d. SαS random variables to a real number:

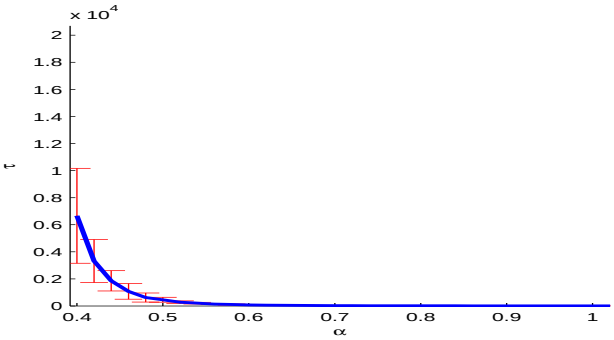
$$\tau(X_\alpha) : \mathbb{R}^N \to \mathbb{R}^+ \tag{7}$$

$$X_\alpha \mapsto g(X_\alpha). \tag{8}$$

The algorithm computes $\tau(X_\alpha)$ for a representative set of α values: $0 < \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_M < 2$. It then interpolates to find $\tau(X_\alpha)$ for $\alpha \in (0, 2]$ in general.



(a) $1.0 \leq \alpha \leq 2.0$



(b) $0.4 \leq \alpha \leq 1.0$

Fig. 3. The τ-map takes α to $\tau(X_\alpha)$ and its inverse takes $\tau(X_\alpha)$ to $\widehat{\alpha}$. This figure shows the median and 90% confidence bands for the τ-map on $\alpha \in [0.4, 2]$. The algorithm uses the inverse map to determine $\widehat{\alpha}$ from an sequence of unknown SαS random observations. The α-Stable Estimation Map Theorem shows that the error bars will shrink toward the mean as the sequence length increases. The corollary to the theorem establishes that this map is continuous from α to $\tau(X_\alpha)$ and that the inverse function exists. The τ-map also exists for all $\alpha \in (0, 0.4)$ but we omit the figure because the double-exponential scaling would obscure the figure.

Let $\widetilde{\tau(X_\alpha)}$ be the log log transformed $\tau(X_\alpha)$:

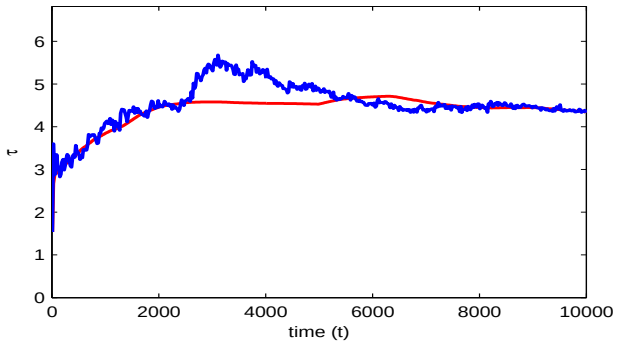
$$\widetilde{\tau(X_\alpha)} = \log\log \tau(X_\alpha). \tag{9}$$

Figure 5 shows a linear relation between α and $\widetilde{\tau(X_\alpha)}$. Least squares linear regression gives the relation

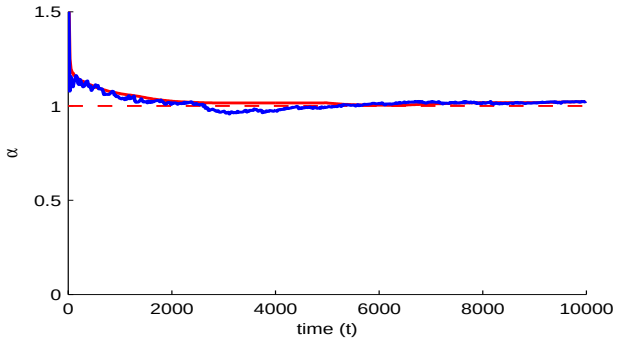
$$\widehat{\alpha} = 0.3969 \cdot \widetilde{\tau(X_\alpha)} + 1.1764. \tag{10}$$

This gives the relation between $\tau(X_\alpha)$ and $\widehat{\alpha}$ as

$$\widehat{\alpha} = -0.3969 \cdot \exp\left[e^{\tau(X_\alpha)}\right] + 1.1764. \tag{11}$$



(a) $\tau(X_\alpha)$



(b) $\widehat{\alpha}(t)$

Fig. 4. Stage 2 of the α-Stable Estimation Algorithm. (a) The blue line shows $\tau(X_\alpha)$ as a function of time for an i.i.d. Cauchy random sequence. The smooth red line shows the mean $\tau(X_\alpha)$. Both will converge to the same value as t increases. (b) The α-Stable Estimation Algorithm uses the τ-map in figure 3 to compute $\widehat{\alpha}$.

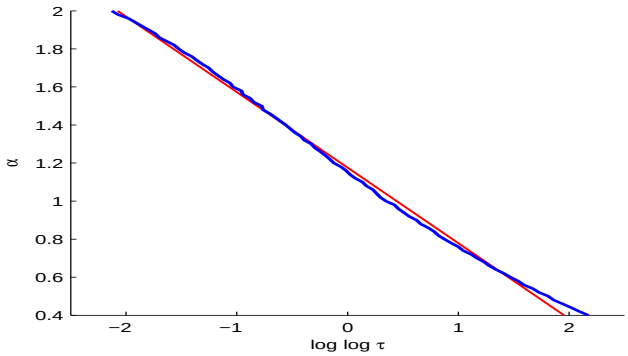


Fig. 5. The α-Stable Estimation Algorithm can use a linear map to estimate α. The blue line shows the log log transform of the τ-map in figure 3 The red line represents the least-squares linear approximation to the transformed τ-map: $\widehat{\alpha} = 0.3969 \log\log \tau(X_\alpha) + 1.1764$. This linear approximation fails for $\alpha < 0.4$ because $\tau(X_\alpha)$ increases to ∞ as α decreases to 0. The figure shows evidence of this by showing the transformed τ-map bend away from the line for $\alpha \approx 0.4$. The algorithm can use other representations of the τ-map such as a piecewise linear approximation or a lookup table to correct this weakness. The algorithm could also use a hill-climbing technique to find $\widehat{\alpha}$ since the τ-map is strictly decreasing.

3.2 Stage 2: Estimate α from an unknown SαS noise source

Stage 2 estimates α from the sequence of unknown SαS random observation. Algorithm 1 below specifies the estimation procedure. The process first computes $\tau(X_\alpha)$ and then maps $\tau(X_\alpha)$ to $\widehat{\alpha}$ by

$$\widehat{\alpha} = -0.3969 \cdot \log\left[\log \tau(X_\alpha)\right] + 1.1764. \tag{12}$$

The algorithm may also use others representations of the τ-map such as a piecewise linear approximation or a lookup table.

4. Experimental Results for $\widehat{\alpha}$

Simulations in this section show the α-Stable Estimation Algorithm applied to four observed SαS sequences with $\alpha \in [0.2, 2]$. Figure 6 shows the evolution of $\widehat{\alpha}$ as the number of observations increases with SαS signals with $\alpha = 2$, 1, 0.5, and 0.2. Table 1 summarizes the performance of the estimator in the trials. The figures show that if $\alpha \geq 0.2$ then $\widehat{\alpha}$ is a robust estimator for α. We did not simulate the algorithm for $\alpha < 0.2$ because α-stable random number generators often produce numeric overflows or fail for very low α [37]. The experiments simulated random SαS samples for $\alpha \geq 0.2$ with the STABLE MATLAB TOOLBOX [38].

Table 1
PERFORMANCE OF THE α-STABLE ESTIMATION ALGORITHM

α	$\widehat{\alpha}$	$\widehat{\alpha}$	$\widehat{\alpha}$
2 (Gaussian)	2.1057	1.9644	1.9618
1 (Cauchy)	0.9681	0.9756	0.9875
0.5	0.3887	0.4796	0.4888
0.2	0.1796	0.1797	0.1921
	$N = 1000$	$N = 5000$	$N = 10000$

Table 1 shows that the algorithm underestimates α in general. The error appears to arise from the linear approximation to the τ-map. $\tau(X_\alpha)$ tends toward ∞ as α decreases to 0. This means that the log log transform will curve upward and so the linear approximation overestimates $\tau(X_\alpha)$. So the algorithm appears to underestimate α on average since it computes $\widehat{\alpha}$ through the inverse map.

5. Conclusion

A bootstrap algorithm can estimate the impulsiveness in a sequence of observed SαS random samples. The α-Stable Estimation Algorithm consists of two stages: (1) it constructs an invertible map from the test statistic $\tau(X_\alpha)$ to α and (2) it computes $\tau(X_\alpha)$ for the unknown SαS observations and then estimates $\widehat{\alpha} = \tau^{-1} - map(\tau(X_\alpha))$. The α-Stable Estimation Theorem shows that the τ-map is 1-to-1 with α. A corollary shows that the τ-map is a bijection and so it has an inverse. Simulations show that the algorithm estimates $\widehat{\alpha}$ if $\alpha \geq 0.2$. Extensions of the α-Stable Estimation Algorithm may well estimate α for non-symmetric α-stable random variables or SαS random variables with unknown dispersion γ. Adaptive and other algorithms may be able to normalize or center the τ-map or they may be able to compute additional test statistics to estimate the parameters. Future research will extend the theorem and corollary to sub-Cauchy ($\alpha < 1$) SαS random variables. It will also study the accuracy of $\widehat{\alpha}$ as a function of the bootstrap parameters: (1) the resampling size and (2) the number resampling iterations.

Algorithm 1 The α-Stable Estimation Algorithm

```
1:  procedure ESTIMATEALPHA(X_α, R, s)
2:      τ ← COMPUTET(X_α, R, s)
3:      α̂ ← MAPTTOALPHA(τ)
4:      return α̂

5:  procedure MAPTTOALPHA(τ(X_α))
6:      m ← 0.3969
7:      b ← 1.1764
8:      α̂ ← b − m · log(log τ)
9:      return α̂

10: procedure COMPUTET(X_α, R, s)
11:     N ← LENGTH(X_α)
12:     for k ← 1, R do
13:         S ← SUBSAMPLE(X_α, s)
14:         V ← 0
15:         for j ← 1, s do
16:             V ← V + |S[j]|
17:         𝒢[k] ← V/N
18:     return CENTER(𝒢)
```

References

[1] A. Patel and B. Kosko, "Stochastic resonance in continuous and spiking neuron models with Lévy noise," *IEEE Trans. on Neural Networks*, vol. 19, no. 12, pp. 1993–2008, December 2008.

[2] B. Kosko and S. Mitaim, "Robust stochastic resonance for simple threshold neurons."

[3] S. Mitaim and B. Kosko, "Adaptive stochastic resonance in noisy neurons based on mutual information," *IEEE Trans. Neural Netw*, pp. 1526–1540, 2004.

[4] S. Mitaim and B. Kosko, "Adaptive stochastic resonance," in *Proceedings of the IEEE: Special Issue on Intelligent Signal Processing*, 1998, pp. 2152–2183.

[5] M. M. Wilde and B. Kosko, "Quantum forbidden-interval theorems for stochastic resonance," *Journal of Physics A: Mathematical and Theoretical*, vol. 42, pp. 465 309–465 331, 2009.

[6] B. Kosko and S. Mitaim, "Stochastic resonance in noisy threshold neurons," *Neural Networks*, vol. 16, pp. 755–761, June 2003.

[7] H. M. Kim and B. Kosko, "Fuzzy prediction and filtering in impulsive noise," *Fuzzy Sets and Systems*, vol. 77, pp. 15–33, January 1996.

[8] B. Kosko and S. Mitaim, "Robust stochastic resonance: Signal detection and adaptation in impulsive noise," *Physical Review E*, vol. 64, no. 5, p. 051110, Oct 2001.

[9] B. Kosko, *Noise*, 2006.

[10] K. Penson and K. Górska, "Exact and explicit probability densities for one-sided Lévy stable distributions," *Phys. Rev. Lett.*, vol. 105, no. 21, p. 210604, Nov 2010.

[11] L. Breiman, *Probability*, 1st ed., Reading, Massachusets, 1968.

[12] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed., 1968, vol. II.

[13] M. Grigoriu, *Applied non-Gaussian processes: examples, theory, simulation, linear random vibration, and MATLAB solutions*, 1995.

[14] C. L. Nikias and M. Shao, *Signal processing with alpha-stable distributions and applications*, New York, NY, USA, 1995.

[15] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed., 2001.

[16] R. Durrett, *Probability: Theory and Examples*, 3rd ed., 2005.

[17] A. R. Dabrowski and A. Jakubowski, "Stable limits for associated random variables," *The Annals of Probability*, vol. 22, no. 1, pp. 1–16, 1994.

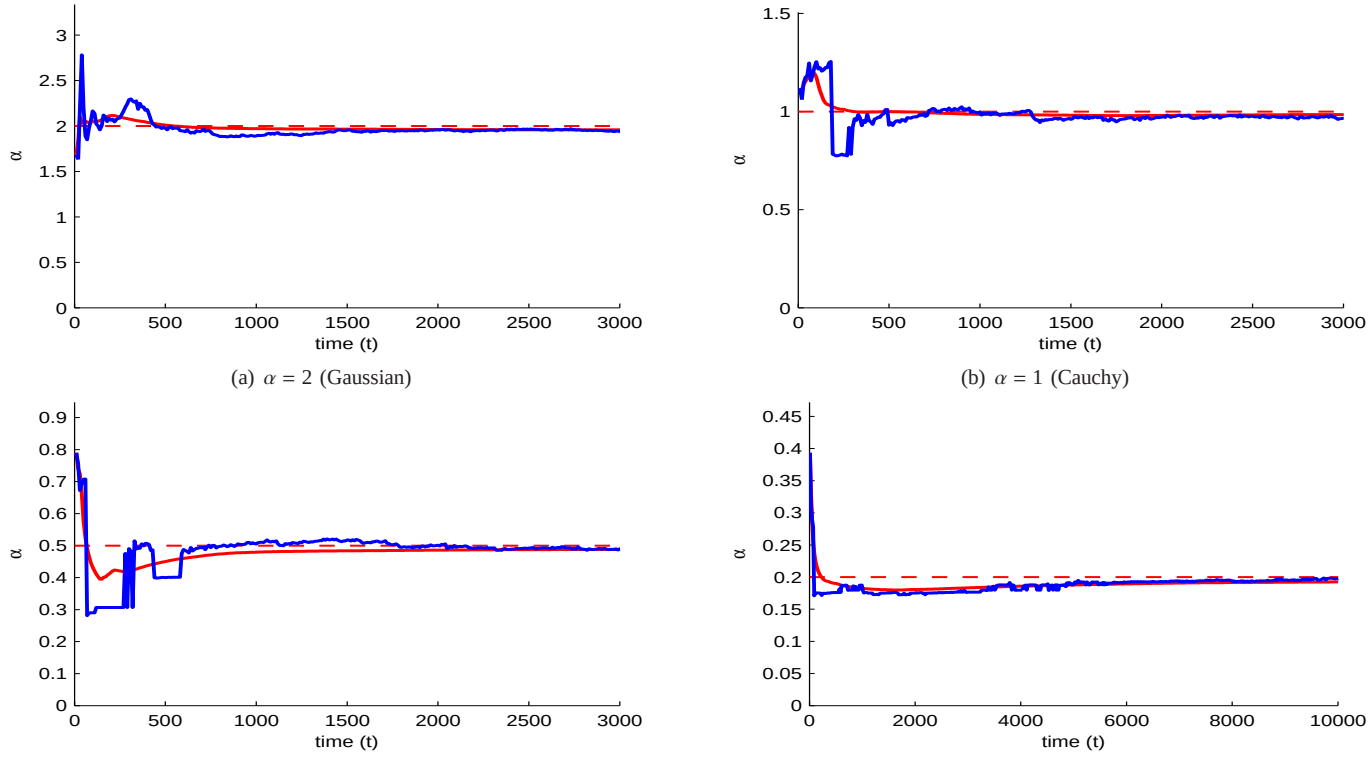
[18] J. P. Nolan, *Stable Distributions — Models for Heavy Tailed Data*. Boston: Birkhauser, 2011.

[19] R. Metzler and J. Klafter, "The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics," *Journal of Physics A: Mathematical and General*, vol. 37, no. 31, pp. R161–R208, aug 2004.

[20] R. Hilfer, "H-function representations for stretched exponential relaxation and non-Debye susceptibilities in glassy systems," *Phys. Rev. E*, vol. 65, no. 6, p. 061510, Jun 2002.

[21] P.-G. de Gennes, "Relaxation anomalies in linear polymer melts," *Macromolecules*, vol. 35, no. 9, pp. 3785–3786, 2002.

[22] O. Sotolongo-Costa, J. C. Antoranz, A. Posadas, F. Vidal, and A. Vazquez, "Lévy flights and earthquakes," *Geophys. Res. Lett.*, vol. 27, pp. 1965–1967, May 2002.



(a) $\alpha = 2$ (Gaussian)

(b) $\alpha = 1$ (Cauchy)

(c) $\alpha = 0.5$

(d) $\alpha = 0.2$

Fig. 6. $\widehat{\alpha}$ estimated from i.i.d. SαS random observations with $\alpha = 2$ (Gaussian), 1 (Cauchy), 0.5, and 0.2. The blue line shows $\widehat{\alpha}$ as a function of time. The dotted red line shows the actual α. The figures show rapid convergence of $\widehat{\alpha}$ to α. Table 1 shows the accuracy of $\widehat{\alpha}$ at $t = 1000$, 5000, and 10000. The algorithm computes $\widehat{\alpha}$ every 10 samples using the cumulative vector $\{x_{\alpha,k} : 1 \le k \le t\}$. The solid red line shows the average value of $\widehat{\alpha}$ as a function of time: $\frac{1}{t}\sum_{k=1}^{t}\widehat{\alpha}[k]$.

[23] G. Samorodnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes: Stochastic models with infinite variance*. Boca Raton, FL: Chapman and Hall/CRC, 2000.

[24] B. Dybiec, A. Kleczkowski, and C. A. Gilligan, "Modelling control of epidemics spreading by long-range interactions," *J R Soc Interface*, vol. 6, no. 39, pp. 941–50, 2009.

[25] H. J. Mcculloch, "Financial applications of stable distributions," in *Statistical methods in finance*, ser. Handbook of Statistics. Amsterdam: North-Holland, 1996, vol. 14, pp. 393–425.

[26] R. N. Mantegna and H. E. Stanley, "Scaling behavior in the dynamics of an economic index," *Nature*, vol. 376, pp. 46–49, 1995.

[27] S. Rachev and S. Mittnik, *Stable Paretian Models in Finance*. New York: Wiley, 2000.

[28] I. M. Sokolov, J. Klafter, and A. Blumen, "Fractional kinetics," *Physics Today*, vol. 55, no. 11, pp. 110 000–55, nov 2002.

[29] A. V. Chechkin, V. Y. Gonchar, R. Gorenflo, N. Korabel, and I. M. Sokolov, "Generalized fractional diffusion equations for accelerating subdiffusion and truncated Lévy flights," *Phys. Rev. E*, vol. 78, no. 2, p. 021111, Aug 2008.

[30] E. S. Sousa, "Performance of a spread spectrum packet radio network link in a Poisson field of interferers," *IEEE Transactions on Information Theory*, vol. 38, no. 6, pp. 1743–1754, nov 1992.

[31] S. M. Kogon and D. G. Manolakis, "Signal modeling with self-similar alpha-stable processes: the fractional Lévy stable motion model," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 1006–1010, apr 1996.

[32] G. A. Tsihrintzis and C. L. Nikias, "Performance of optimum and suboptimum receivers in the presence of impulsive noise modeled as an alpha-stable process," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 904–914, feb/mar/apr 1995.

[33] G. A. Tsihrintzis and C. L. Nikias, "Fast estimation of the parameters of alpha-stable impulsive interference," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1492–1503, jun 1996.

[34] B. Efron, *The jackknife, the bootstrap, and other resampling plans*. SIAM, 1982, vol. 38.

[35]

[36] P. Hall, *The bootstrap and edgeworth expansion*. New York: Springer-Verlag, 1992.

[37] J. Chambers, C. Mallows, and B. Stuck, "A method for simulating stable random variables," *J. of the American Stat. Assoc.*, vol. 171, no. 354, 1976.

[38] J. P. Nolan, STABLE *v5.3 for MatLab*. Robust Analysis, Inc., April 2011. [Online]. Available: http://www.robustanalysis.com

[39] S. Lang, *Real Analysis*, 2nd ed. New York: Addison-Wesley Pub. Co., 1969.

[40] V. M. Zolotarev, *One-dimensional stable distributions*, 1986.

[41] M. Matsui and A. Takemura, "Some improvements in numerical evaluation of symmetric stable density and its derivatives," *Communications in Statistics — Theory and Methods*, vol. 35, pp. 149 – 172, 2006.

Appendix: Proof of Theoretical Results

α-Stable Estimation Map Theorem. *Suppose X_{α_1} and X_{α_2} are two independent sequences of N i.i.d. SαS random variables with probability density functions $f_{\alpha_1}(x)$ and $f_{\alpha_2}(x)$. Suppose $\alpha_i \in (1, 2]$ with unit dispersion: $\gamma = 1$. Fix $p > 0$ and define $g_p(X)$ as*

$$g_p(X) = \frac{1}{N}\|X\|_p^p = \frac{1}{N}\sum_{k=1}^{N}|X_k|. \tag{A.1}$$

Define the maximum function $H(X)$ as

$$H(X) = \max_{1\le k\le N}|X_k|. \tag{A.2}$$

Then there exists an N_0 such that

$$E\left[g_p(X_{\alpha_1})|H(X_{\alpha_1}) = h\right] = E\left[g_p(X_{\alpha_2})|H(X_{\alpha_2}) = h\right] \tag{A.3}$$

for $h > 2$ and all sequences X_{α_1} and X_{α_2} with length $N \ge N_0$ only if $\alpha_1 = \alpha_2$.

Proof. Suppose $\alpha_1 \ne \alpha_2$. Suppose further that

$$E\left[g_p(X_{\alpha_1})|H(X_{\alpha_1}) = h\right] = E\left[g_p(X_{\alpha_2})|H(X_{\alpha_2}) = h\right]. \tag{A.4}$$

Denote the joint pdfs of X_{α_1} and X_{α_2} as

$$f_{\alpha_1}(x_1, \cdots, x_N) = f_{\alpha_1}(x_1)\cdots f_{\alpha_1}(x_N) = [f_{\alpha_1}(x)]^N \tag{A.5}$$

$$f_{\alpha_2}(x_1, \cdots, x_N) = f_{\alpha_2}(x_1)\cdots f_{\alpha_2}(x_N) = [f_{\alpha_2}(x)]^N \tag{A.6}$$

Then

$$E\left[g_p(X_{\alpha_1})|H(X_{\alpha_1}) = h\right] = \frac{E\left[g_p(X_{\alpha_1})\ I(H(X_{\alpha_1}) = h)\right]}{P\left[H(X_{\alpha_1}) = h\right]} \tag{A.7}$$

since

$$E[X|A] = \frac{E[X\ I_A]}{P[A]} \tag{A.8}$$

for a random variable X where I_A denotes the indicator function for the event A. Thus

$$E\left[g_p(X_{\alpha_1})|H(X_{\alpha_1}) = h\right] \tag{A.9}$$

$$= \frac{\int\cdots\int g_p(X_{\alpha_1})\, I\left(\max\left|X_{\alpha_1}\right| = h\right) f_{\alpha_1}(X_{\alpha_1})\ dx_1\cdots dx_N}{P\left[\max\left|X_{\alpha_1}\right| = h\right]} \tag{A.10}$$

$$= \frac{\int_{-H}^{H}\cdots\int_{H}^{H}\sum_{k=1}^{N}\left|x_{1,k}\right|^p f_{\alpha_1}(x_1)\cdots f_{\alpha_1}(x_N)\ dx_1\cdots dx_N}{P\left[\max\left|X_{\alpha_1}\right| = h\right]} \tag{A.11}$$

$$= \frac{\sum_{k=1}^{N}\int_{-H}^{H}|x|^p f_{\alpha_1}(x)\ dx}{P\left[\max\left|X_{\alpha_1}\right| = h\right]} \tag{A.12}$$

$$= \frac{N}{P\left[\max\left|X_{\alpha_1}\right| = h\right]}\int_{-H}^{H}|x|^p f_{\alpha_1}(x)\ dx \tag{A.13}$$

Similarly for X_{α_2}:

$$E\left[g_p(X_{\alpha_2})|H(X_{\alpha_2}) = h\right] \tag{A.14}$$

$$= \frac{N}{P\left[\max\left|X_{\alpha_2}\right| = h\right]}\int_{-H}^{H}|x|^p f_{\alpha_2}(x)\ dx. \tag{A.15}$$

Suppose $\alpha_1 < \alpha_2$. Lemma 1 below shows that there exists an N_0 such that $P\left[\max\left|X_{\alpha_1}\right| = h\right] < P\left[\max\left|X_{\alpha_2}\right| = h\right]$ for all $N \geq N_0$ and for all $h > 2$. Thus there exists N_0 such that

$$E\left[g_p(X_{\alpha_1})|H(X_{\alpha_1}) = h\right] = E\left[g_p(X_{\alpha_2})|H(X_{\alpha_2}) = h\right] \tag{A.16}$$

for random sequences with length $N \geq N_0$ only if

$$\int_{-H}^{H}|x|^p f_{\alpha_1}(x)\ dx < \int_{-H}^{H}|x|^p f_{\alpha_2}(x)\ dx. \tag{A.17}$$

This implies that $\alpha_1 > \alpha_2$ since the contrapositive of Lemma 2 states that $\int_{-H}^{H}|x|^p f_\alpha(x)\ dx$ decreases with α. Thus there is a contradiction since $\alpha_1 < \alpha_2$ by assumption. So $\alpha_1 \neq \alpha_2$. **QED**

Lemma 1. *Suppose X_{α_1} and X_{α_2} are two independent sequences of N i.i.d. SαS random variables with probability density functions $f_{\alpha_1}(x)$ and $f_{\alpha_2}(x)$ and cumulative distribution functions $F_{\alpha_1}(x)$ and $F_{\alpha_2}(x)$. Suppose $\alpha_i \in (1,2]$ with $\alpha_1 \neq \alpha_2$ and unit dispersion: $\gamma = 1$. Then there exists $N_0 < \infty$ such that $P\left[\max\left|X_{\alpha_1}\right| = h\right] < P\left[\max\left|X_{\alpha_2}\right| = h\right]$ for all $N \geq N_0$ and $h > 2$.*

Proof. Expanding the pdf of the maximum of a sequence of N i.i.d. random variables $X = (X_1, \cdots, X_N)$ gives

$$P[\max|X| = h] = \sum_{j=1}^{N} P\left[\left|X_j\right| = h\right]\left(\prod_{k=1,k\neq j}^{N} P\left[\left|X_j\right| \leq h\right]\right) \tag{A.18}$$

$$= \sum_{j=1}^{N} P\left[\left|X_j\right| = h\right](2F(h) - 1)^{N-1} \tag{A.19}$$

$$= N(f(h) + f(-h))(2F(h) - 1)^{N-1} \tag{A.20}$$

$$= 2Nf(h)(2F(h) - 1)^{N-1} \tag{A.21}$$

holds since X_k are i.i.d. and symmetric and also $P[|X_k| \leq x] = 2P[X_k \leq x] - 1 = 2F(x) - 1$ for $x \geq 0$.

Suppose $\alpha_1 < \alpha_2$. Suppose further that $h > 2$. Then $f_{\alpha_1}(h) > f_{\alpha_2}(h)$. So

$$R = \frac{f_{\alpha_1}(h)}{f_{\alpha_2}(h)} > 1. \tag{A.22}$$

Also $F_{\alpha_1}(x) < F_{\alpha_2}(x)$ since $f_{\alpha_1}(x) > f_{\alpha_2}(x)$ for all $x > 2$ and since $F(x) = 1 - \int_x^\infty f(x)\ dx$. Thus $2F_{\alpha_1}(x) - 1 < 2F_{\alpha_2}(x) - 1$. So

$$s = \frac{2F_{\alpha_1}(h) - 1}{2F_{\alpha_2}(h) - 1} < 1. \tag{A.23}$$

The ratio of the pdfs for $\max\left|X_{\alpha_1}\right|$ and $\max\left|X_{\alpha_2}\right|$ is

$$\frac{P\left[\max\left|X_{\alpha_1}\right| = h\right]}{P\left[\max\left|X_{\alpha_2}\right| = h\right]} = \frac{2Nf_{\alpha_1}(h)(2F_{\alpha_1}(h) - 1)^{N-1}}{2Nf_{\alpha_2}(h)(2F_{\alpha_2}(h) - 1)^{N-1}} \tag{A.24}$$

$$= \frac{f_{\alpha_1}(h)}{f_{\alpha_2}(h)}\left(\frac{2F_{\alpha_1}(h) - 1}{2F_{\alpha_2}(h) - 1}\right)^{N-1} \tag{A.25}$$

$$= Rs^{N-1} \tag{A.26}$$

Rs^{N-1} goes to zero as N increases since $s < 1$ and $R < \infty$. So there exists some N_0 such that $Rs^N < 1$ for all $N \geq N_0$ by definition of the limit. Thus $P\left[\max\left|X_{\alpha_1}\right| = h\right] < P\left[\max\left|X_{\alpha_2}\right| = h\right]$ for all sequences with length $N \geq N_0$. **QED**

Lemma 2. *The function $A(\alpha) = \int_{-H}^{H}|x|^p f_\alpha(x)\ dx$ strictly decreases on $\alpha \in (1,2]$ for $p > 0$ if $H > 2$.*

Proof. For $p > 0$ then

$$\frac{\partial}{\partial\alpha}A(\alpha) = \frac{\partial}{\partial\alpha}\int_{-H}^{H}|x|^p f_\alpha(x)\ dx \tag{A.27}$$

$$= 2\frac{\partial}{\partial\alpha}\int_{0}^{H}|x|^p f_\alpha(x)\ dx \tag{A.28}$$

$$= 2\frac{\partial}{\partial\alpha}\int_{0}^{H}x^p f_\alpha(x)\ dx. \tag{A.29}$$

(A.27) holds because $|x|^p f_\alpha(x)$ is an even function since $|x|^p$ is even and $f_\alpha(x)$ is a *symmetric* α-stable probability density function.

The Leibniz integral rule [39] states that the integral and derivative commute if the integrand satisfies two continuity requirements. Lemma 3 below shows that $A(\alpha)$ satisfies these conditions and so

$$\frac{\partial}{\partial\alpha}A(\alpha) = 2\int_{0}^{H}\frac{\partial}{\partial\alpha}\left[x^p f_\alpha(x)\right]\ dx \tag{A.30}$$

$$= 2\int_{0}^{H}x^p\frac{\partial}{\partial\alpha}\left[f_\alpha(x)\right]\ dx. \tag{A.31}$$

The Taylor series

$$f_\alpha(x) = \frac{1}{\pi\alpha}\sum_{k=1}^{\infty}\frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{(2k)!}(-1)^k x^{2k} \tag{A.32}$$

holds for $x \neq 0$ and $\alpha \in (1,2]$ [40], [41]. Substitution gives

$$\frac{\partial}{\partial\alpha}A(\alpha) = 2\int_{0}^{H}x^p\frac{\partial}{\partial\alpha}\left[\frac{1}{\pi\alpha}\sum_{k=1}^{\infty}\frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{(2k)!}(-1)^k x^{2k}\right]\ dx \tag{A.33}$$

$$= \frac{2}{\pi}\int_{0}^{H}x^p\sum_{k=1}^{\infty}\frac{\partial}{\partial\alpha}\left[\frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{\alpha}\right]\frac{(-1)^k x^{2k}}{(2k)!}\ dx \tag{A.34}$$

$$= -\frac{2}{\pi}\int_{0}^{H}x^p\sum_{k=1}^{\infty}\frac{1}{(2k)!}\frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{\alpha^2}\cdot \tag{A.35}$$

$$\left(1 + \frac{(2k+1)\psi^{(0)}\left(\frac{2k+1}{\alpha}\right)}{\alpha}\right)(-1)^k x^{2k}\ dx \tag{A.36}$$

where the digamma function

$$\psi^{(v)}(z) = \frac{d^{(v+1)}}{dx^{(v+1)}} \ln \Gamma(x) \tag{A.37}$$

is the $(v+1)^{\text{th}}$ derivative of the logarithm of the gamma function. The digamma function obeys

$$\psi^{(0)}(x) > 0 \tag{A.38}$$

for x greater than the positive local minima of the gamma function: $x \approx 1.47$. Thus

$$\psi^{(0)}\left(\frac{2k+1}{\alpha}\right) > 0 \tag{A.39}$$

since the ratio $\frac{2k+1}{\alpha} \geq 1.5 > 1.47$ because $2k+1 \geq 3$ and $\alpha \leq 2$. So

$$1 + \frac{(2k+1)\,\psi^{(0)}\left(\frac{2k+1}{\alpha}\right)}{\alpha} > 0 \tag{A.40}$$

since $\alpha > 1$ and $2k+1 \geq 3$. So

$$\frac{1}{(2k)!}\frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{\alpha^2}\left(1 + \frac{(2k+1)\,\psi^{(0)}\left(\frac{2k+1}{\alpha}\right)}{\alpha}\right) > 0. \tag{A.41}$$

The last equation holds since $\Gamma\left(\frac{2k+1}{\alpha}\right) > 0$. Thus

$$\frac{\partial}{\partial\alpha}A(\alpha) = -2\int_0^H x^p \sum_{k=1}^{\infty} C_k\left(-x^2\right)^k \, dx \tag{A.42}$$

where

$$C_k = \frac{1}{\pi}\frac{1}{(2k)!}\frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{\alpha^2}\left(1 + \frac{(2k+1)\,\psi^{(0)}\left(\frac{2k+1}{\alpha}\right)}{\alpha}\right) > 0 \tag{A.43}$$

The integral and summation commute because the power series converges absolutely on the interior of its region of convergence: $\{x \in \mathbb{R} : x \neq 0\}$. So

$$\begin{aligned}
\frac{\partial}{\partial\alpha}A(\alpha) &= -2\sum_{k=1}^{\infty} C_k \int_0^H x^p\left(-x^2\right)^k \, dx && \text{(A.44)}\\
&= -2\sum_{k=1}^{\infty} C_k \int_0^H x^p\left(x^{4k} - x^{2k}\right) \, dx && \text{(A.45)}\\
&= -2\sum_{k=1}^{\infty} C_k \int_0^H x^{4k+p} - x^{2k+p} \, dx && \text{(A.46)}\\
&= -2\sum_{k=1}^{\infty} C_k \left[\frac{x^{4k+p+1}}{4k+p+1} - \frac{x^{2k+p+1}}{2k+p+1}\right|_{x=0}^{x=H} && \text{(A.47)}\\
&= -2\sum_{k=1}^{\infty} C_k \left(\frac{H^{4k+p+1}}{4k+p+1} - \frac{H^{2k+p+1}}{2k+p+1}\right) && \text{(A.48)}
\end{aligned}$$

So $\alpha \in (1, 2]$ and

$$\frac{H^{4k+p+1}}{4k+p+1} > \frac{H^{2k+p+1}}{2k+p+1} \tag{A.49}$$

for all integers $k \geq 1$ implies that

$$A(\alpha) = \int_{-H}^{H} |x|^p \, f_\alpha(x) \, dx \tag{A.50}$$

strictly decreases because then

$$\frac{\partial}{\partial\alpha}A(\alpha) = \frac{\partial}{\partial\alpha}\int_{-H}^{H} |x|^p \, f_\alpha(x) \, dx < 0. \tag{A.51}$$

Suppose now that $H > 2$ and $k \geq 1$ is an integer. Then

$$\frac{H^{4k+p+1}}{4k+p+1} - \frac{H^{2k+p+1}}{2k+p+1} = H^{2k} - \frac{4k+p+1}{2k+p+1} > 4^k - 2 \geq 0 \tag{A.52}$$

(A.52) because $\frac{4k+p+1}{2k+p+1} < 2 < 4$ since $(4k+p+1) < 2(2k+p+1)$. **QED**

Lemma 3. *Suppose X_α is a SαS random variable with probability density function $f_\alpha(x)$ and characteristic function $\phi_\alpha(\omega)$. Then*

$$\frac{\partial}{\partial\alpha}\int_0^H x^p \, f_\alpha(x) \, dx = \int_0^H \frac{\partial}{\partial\alpha}\left[x^p \, f_\alpha(x)\right] \, dx \tag{A.53}$$

if $0 < H < \infty$.

Proof. The Leibniz integral rule states that

$$\frac{\partial}{\partial x}\int_{y_0}^{y_1} f(x,y) \, dy = \int_{y_0}^{y_1} \frac{\partial}{\partial x} f(x,y) \, dy \tag{A.54}$$

for $x \in [x_0, x_1]$ when $f(x,y)$ and $\frac{\partial}{\partial x} f(x,y)$ are continuous on $[x_0, x_1] \times [y_0, y_1]$.

We show that $\int_0^H x^p \, f_\alpha(x) \, dx$ satisfies these continuity conditions on $V = [1,2] \times [0, H] \subset \mathbb{R}^2$. $f_\alpha(x)$ is bounded since

$$\begin{aligned}
|f_\alpha(x)| &\leq \left|\int_{-\infty}^{\infty} e^{-|\omega|^\alpha} e^{-i\omega x} \, d\omega\right| = \int_{-\infty}^{\infty}\left|e^{-|\omega|^\alpha} e^{-i\omega x}\right| \, d\omega && \text{(A.55)}\\
&= \int_{-\infty}^{\infty} e^{-\omega^\alpha} \, d\omega \leq \left\lfloor 2 + \frac{1}{\alpha}\right\rfloor! < \infty. && \text{(A.56)}
\end{aligned}$$

So $f_\alpha(x)$ is continuous since it is the integral of a continuous function. Thus $x^p f_\alpha(x)$ is continuous on V because it is the product of two functions that are continuous on V.

The function $\frac{\partial}{\partial\alpha}\left[x^p f_\alpha(x)\right]$ is also bounded on V since

$$\begin{aligned}
\left|\frac{\partial}{\partial\alpha}\left[x^p f_\alpha(x)\right]\right| &= \left|x^p \frac{\partial}{\partial\alpha} f_\alpha(x)\right| && \text{(A.57)}\\
&\leq H^p\left|-2\int_0^H x^p \sum_{k=1}^{N} C_k\left(-x^2\right)^k \, dx\right| && \text{(A.58)}
\end{aligned}$$

by substitution from (A.42). So

$$\left|\frac{\partial}{\partial\alpha}\left[x^p f_\alpha(x)\right]\right| \leq 2H^p H^p H^{2k}\sum_{k=1}^{N} C_k \leq 2H^{2p+2k}M < \infty \tag{A.59}$$

since $M < \infty$ and because C_k is a power series and so it absolutely converges for $\alpha > 1$. Thus $\frac{\partial}{\partial\alpha}\left[x^p f_\alpha(x)\right]$ is also continuous on V. So

$$\frac{\partial}{\partial\alpha}\int_0^H x^p \, f_\alpha(x) \, dx = \int_0^H \frac{\partial}{\partial\alpha}\left[x^p \, f_\alpha(x)\right] \, dx \tag{A.60}$$

for $\alpha \in (1, 2]$. **QED**

τ-map Invertibility Corollary. *Define $g_p(X)$ by* (A.1) *and the maximum function $H(X)$ by* (A.2)*. Define X_{α_1} and X_{α_2} as in the α-Stable Estimate Map Theorem. Suppose that the conditions hold such that a finite N_0 exists. Suppose further that $h < \infty$. Then $\tau(X_\alpha) = \mathscr{G}(\alpha) = E\left[g_p(X_\alpha)\,|\,H(X_\alpha) = h\right]$ is a bijection from $\alpha \in (1, 2]$ onto $\tau(X_\alpha) \in [\mathscr{G}(1), \mathscr{G}(2)]$.*

Proof. The α-Stable Estimation Map Theorem shows that $\mathscr{G}$ is injective (1-to-1). $\mathscr{G}$ is also continuous for $\alpha > 1$. Thus the Intermediate Value Theorem shows that $\mathscr{G}$ is surjective onto $[\mathscr{G}(1), \mathscr{G}(2)]$. Therefore $\mathscr{G}$ is a bijection since it is 1-to-1 and onto. So $\mathscr{G}$ has an inverse function $\mathscr{G}^{-1}(\tau(X_\alpha)) = \widehat{\alpha}$. **QED**

Fuzzy Goal Programming Approach for Quadratic Fractional Bilevel Programming

Animesh Biswas and Koushik Bose
Department of Mathematics, University of Kalyani, Kalyani – 741235, INDIA

Abstract – *This paper presents a fuzzy goal programming methodology for solving bilevel programming problems having quadratic fractional form of objectives of the decision makers. In the proposed procedure, the membership functions for the defined fuzzy goals of objective of the decision makers at both levels are developed first. Then a fuzzy goal programming model is developed to minimize the group regret of degree of satisfactions of both the decision makers and thereby obtaining the most satisfactory solution in the decision making environment. In the solution process a two step linearization technique is adopted to break the fractional part at its first step and Taylor's series approximation technique is employed to linearize the quadratic membership goals in the next step. Then the problem is solved to reach a compromise decision by minimizing the deviational variables of the goals in the achievement function for overall benefit of the organization. An illustrative numerical example is solved to demonstrate the efficiency of the proposed procedure and the model solution is compared with the solutions obtained for the use of other existing techniques to expound the potential use of the proposed approach.*

Keywords: Fractional Programming, Quadratic Programming, Fuzzy programming, Fuzzy Goal Programming, Linear Approximation Technique.

1 Introduction

Candler and Townsley [1] first introduced the concept of Bilevel Programming Problem (BLPP). BLPP is considered as a hierarchical decision making problem with a structure of two levels in a highly conflicting decision making situation. The upper-level decision maker (DM) is termed as leader and the lower level DM as follower. Most of the developments on BLPPs are based on vertex enumeration method [1] and transformation approaches [2, 3] which are effective only for very simple types of problems. In these methods the DMs have no cooperating attitude with each other. So, decision deadlock arises frequently due to follower's dissatisfaction with the solution. Again if the parameter values involved are based on prediction of future conditions which inevitably contains some degree of uncertainty, the above approaches unable to give a satisfactory solution which would be acceptable to both the DMs.

To deal with such types of problems Zimmermann [4] first applied fuzzy set theory in decision making problems with several conflicting objectives. In 1996 Lai [5] introduced the concept of membership functions of fuzzy sets in BLPPs. Lai's solution concept was then extended by Shih et al. [6] and a supervised search procedure with the use of max-min operator of Bellman and Zadeh [7] was proposed. The basic concept of this procedure is that the follower optimizes his/her objective function, taking into consideration of leader's goal. The main difficulty of fuzzy programming (FP) approach is that the objectives of the DMs are conflicting. So there is possibility of rejecting the solution again and again by the DMs and the solution process is continued by redefining the membership functions repeatedly until a satisfactory solution is obtained. This make the solution process very lengthy and tedious one.

To overcome such difficulty fuzzy goal programming (FGP) procedure introduced by Mohamed [8] is applied in decision making problems. Here the fuzzy goals of the objectives are determined by using individual optimal solution. Then a feasible membership goal is constructed by introducing under- and over- deviational variables and assigning highest membership value (unity) as aspiration level to each of them. In the recent past FGP approaches have been discussed by Sinha and Biswal [9], Moitra and Pal [10], Pal and Moitra [11], Biswas and Pal [12], Pal and Biswas [13], Pramanik and Roy [14], Baky [15], Biswas and Bose [16], and others.

In a decision making situation management faces problems where the objectives of the DMs are fractional in nature. Such type of optimization problem is called fractional programming problem (FPP). Von Neumann [17] first used FPP in an equilibrium model for an expanding economy. In 1962, Charnes and Cooper [18] showed that a Linear FPP with one ratio can be reduced to a simple linear programming problem (LPP) using nonlinear variable transformations. FPPs have been studied extensively by many researchers like Swarup [19], Dinkelbach [20], Geoffrion [21], Bitran and Magnanti [22], Ibaraki et al. [23], Crouzeix and Ferland [24], Rodenas et al. [25], Bensen [26] and others. The FPPs have applications in various fields of game theory [27], network and flows [28], traffic planning [29] and many other areas of engineering, finance, corporate planning, business, bank balance sheet management, water resources, healthcare, etc. A review of various applications was given by Schaible [30].

In a BLPP, if the objective functions are linear fractional forms, then the problems are termed as linear fractional bilevel programming problems (LFBLPPs) and if they are of nonlinear fractional forms, they are termed as nonlinear fractional bilevel programming problems (NLFBLPPs). Quadratic FBLPP (QFBLPP) is one type of NLFBLPP.

In recent past LFBLPP was studied by Dutta et al. [31], Pal et al. [32], Pal and Moitra [33], Toksari [34], Ahlatcioglu and Tiryaki [35], Abo-sinna and Baky[36] and Mehrjerdi [37]. A solution approach for QFBLPP was studied by Alemayehu and Arora [38]. Also Mishra and Ghosh [39] used interactive fuzzy programming approach for solving QFBLPPs in the recent past. But FGP approaches to QFBLPP are yet to appear in literature.

In the present study, the FGP method is used to solve QFBLPPs. In the model formulation process, the membership functions defined for the fuzzy goals of the problem are transformed into flexible goals by assigning the highest degree (unity) of the membership functions as their aspiration level. A two step linearization technique is adopted to linearize the quadratic fractional goals and to arrive at the most satisfactory solution in the decision making context. The model formulation of the problem is presented in the next section.

2 Formulation of QFBLPP

In a hierarchical decision system, both the DMs are motivated to cooperate with each other and each DM tries to optimize his/her own benefit, paying serious attention to the interest of the other.

Let F_1 and F_2 be the objective functions of the leader and follower, respectively with respective controlling vector of decision variables X_1 and X_2.

In a QFBLPP the numerator and/or the denominator of the objective functions are represented by some quadratic functions.

The generic form of such QFBLPP can be represented as:

Find $X(X_1, X_2)$ so as to

$$\underset{X_1}{Min}\ F_1(X_1, X_2) = \frac{F_{11}(X_1, X_2)}{F_{12}(X_1, X_2)}$$

and for given X_1, X_2 solves

$$\underset{X_2}{Min}\ F_2(X_1, X_2) = \frac{F_{21}(X_1, X_2)}{F_{22}(X_1, X_2)}$$

subject to

$$(X_1, X_2) \in S = \left\{ (X_1, X_2) \mid A_1X_1 + A_2X_2 \begin{pmatrix} \le \\ = \\ \ge \end{pmatrix} b; X_1, X_2 \ge 0 \right\}. \quad (1)$$

Here $F_{ij}(X_1, X_2) = C_{ij}X + \frac{1}{2}X^T D_{ij} X, \quad i, j = 1,2.$

where X^T denotes transpose of decision vector X; $C_{ij}(i.j = 1,2)$ are constant vectors and $D_{ij}(i.j = 1,2)$ are constant symmetric matrices.

It is customary to assume that

$$F_{i2}(X_1, X_2) > 0, i = 1,2.$$

Also, it is assumed that $S(\neq \phi)$ is bounded.

3 FP Model Formulation

To formulate the FP model of the problem (1), both the objectives are required to be transformed into fuzzy goals by means of introducing an imprecise aspiration level to each of them. Then the goals are characterized by the membership functions to achieve their respective aspired levels.

3.1 Construction of Membership Functions

Since both the DMs are interested in minimizing their own objective values over the same feasible region S, the optimal solutions of both of them calculated in isolation can be taken as the aspiration levels of their associated fuzzy goals.

Let $\left(X_1^{b_l}, X_2^{b_l}; F_1^b\right)$ and $\left(X_1^{w_l}, X_2^{w_l}; F_1^w\right)$ be the independent best and worst solutions, respectively, of the leader. Similarly let $\left(X_1^{b_f}, X_2^{b_f}; F_2^b\right)$ and $\left(X_1^{w_f}, X_2^{w_f}; F_2^w\right)$ be the independent best and worst solutions, respectively, of the follower when calculated in isolation.

Here

$$F_i^b = \min_{(X_1, X_2) \in S} F_i(X_1, X_2), \quad i = 1,2.$$

and

$$F_j^w = \max_{(X_1, X_2) \in S} F_j(X_1, X_2), \quad j = 1,2.$$

The solutions $\left(X_1^{b_l}, X_2^{b_l}; F_1^b\right)$ and $\left(X_1^{b_f}, X_2^{b_f}; F_2^b\right)$ are obviously different due to conflicting nature of the objectives of leader and follower. Also these solutions are absolutely acceptable to the respective DMs. Similarly the solutions $\left(X_1^{w_l}, X_2^{w_l}; F_1^w\right)$ and $\left(X_1^{w_f}, X_2^{w_f}; F_2^w\right)$ are totally unacceptable to leader and follower, respectively, in decision making situation. So the respective fuzzy tolerance values of leader and follower appear as: $F_1 \lesssim F_1^b$ and $F_2 \lesssim F_2^b$.

Also the respective tolerance ranges of the decision vectors X_1 and X_2 of the leader and follower, are considered as

$$X_i^b = \min\left\{X_i^{b_l}, X_i^{w_l}, X_i^{b_f}, X_i^{w_f}\right\} \le X_i \le \max\left\{X_i^{b_l}, X_i^{w_l}, X_i^{b_f}, X_i^{w_f}\right\} = X_i^w, \quad i = 1,2.$$

Now the membership functions for the defined fuzzy tolerance values of the objectives of the leader and follower can algebraically be formulated as:

$$\mu_{F_i} = \begin{cases} 1 & if \quad F_i(X_1, X_2) < F_i^b \\ \dfrac{F_i^w - F_i(X_1, X_2)}{F_i^w - F_i^b} & if \quad F_i^b \le F_i(X_1, X_2) \le F_i^w \\ 0 & if \quad F_i(X_1, X_2) > F_i^w \end{cases} \quad i = 1,2 \qquad (2)$$

4 Formulation of Membership Goals

In FGP procedure the highest aspiration levels of the defined membership functions is considered as unity. Hence by introducing the under-and over-deviational variables, the membership functions are converted into fuzzy goals as [33]:

$$\frac{F_i^w - F_i(X_1, X_2)}{F_i^w - F_i^b} + d_i^- - d_i^+ = 1, \quad i = 1,2 \qquad (3)$$

where $d_i^-, d_i^+ \ge 0$ with $d_i^- . d_i^+ = 0$, $i = 1,2$ represents the under-and over-deviational variables, respectively.

Now it is to be noted that the membership goals in (3) are quadratic fractional in nature and the procedure for solving it is yet to appear in literature extensively.

The proposed two-step linearization technique is presented in the following section to find the compromise decision in the decision making context.

5 Two-Step Linearization Technique for Quadratic Fractional Membership Goals (QFMG)

A two step linearization method is adopted here to find the most satisfactory solution in the decision making environment. In the first step of the process, the quadratic fractional objectives are converted into quadratic forms by linearizing the fractional parts. Then Taylor's series approximation technique is used to linearize the quadratic part in the second step. The computational methodology of this process is described in the following sub sections.

5.1 Step-1

Considering the fractional part of the membership goals in (3), the model can be expressed for each $i = 1,2$ as:

$$m_i\left[F_i^w - \frac{F_{i1}(X_1, X_2)}{F_{i2}(X_1, X_2)}\right] + d_i^- - d_i^+ = 1$$

i.e. $$f_i(X_1, X_2) + D_i^- - D_i^+ = 1 \qquad (4)$$

where $m_i = \dfrac{1}{F_i^w - F_i^b}$ and

$$f_i(X_1, X_2) = 1 - m_i F_{i1}(X_1, X_2) - F_{i2}(X_1, X_2) + m_i F_i^w F_{i2}(X_1, X_2)$$

and $D_i^- = d_i^- F_{i2}(X_1, X_2)$, $D_i^+ = d_i^+ F_{i2}(X_1, X_2)$.

Here, clearly the above equation (4) contains only quadratic forms without any fractional part.

Now, in the next step linear approximation technique proposed by J. P. Ignizio [40] is used to linearize the reduced QFGP model.

5.2 Step-2

In 1976 J. P. Ignizio [40] proposed a methodology to solve nonlinear goal programming (NLGP).Thereafter a plenty of works has done [41- 43]. Here the concept of NLGP solution approach [40] is considered for the QFGP model (4) and Taylor's series linear approximation technique is used to linearize the quadratic forms.

In a decision making situation, let an approximate solution $X^0\left(X_1^0, X_2^0\right)$ determined by initial inspection is disclosed to both the DMs with the view of satisfying the objectives. Here the initial solution is chosen so that it lies in the tolerance ranges specified for decision vectors X_1 and X_2. Now it may happen that the DMs are dissatisfied with this solution. Then linear approximation technique is used to locate the satisfactory decision in the neighborhood of this initial solution $X^0\left(X_1^0, X_2^0\right)$.

Linear approximation to the goals in (4), using the Taylor's series, can be presented as

$$1 - D_i^- + D_i^+ = f_i(X) \cong f_i\left(X^0\right) + \sum_{k=1}^{2} \frac{\partial f_i\left(X^0\right)}{\partial X_k}\left(X_k - X_k^0\right), \quad i = 1,2 \qquad (5)$$

where

(i) $X_k^0 = k$ -th component of the present solution $X^0\left(X_1^0, X_2^0\right)$ for $k = 1,2$.

(ii) $X_k = k$ -th component of the new solution $X\left(X_1, X_2\right)$ for $k = 1,2$.

The linear approximation of the membership goals can be presented in vector form as

$$G_i : f_i\left(X^o\right)+\left[\nabla f_i\left(X^o\right)\right]^T\left(X-X^o\right)+D_i^- - D_i^+ = I \qquad (6)$$

where $\nabla f_i\left(X^o\right)$ is the gradient of $f_i\left(X^o\right)$, $\left[\nabla f_i\left(X^o\right)\right]^T$ is the transpose of $\left[\nabla f_i\left(X^o\right)\right]$ and I represents a column vector of order two having all the elements are 1.

Now further, the term $\left(X-X^o\right)$ may be replaced by a new vector Y given by $Y=X-X^0$. Consequently, Y represents the change from the current solution point X^0 to the next solution point X. Clearly, components of Y are unrestricted in sign. So a set of non-negative vectors P, Q are introduced by $Y = P-Q$, where $P,Q \geq 0$. Now by using these vectors in (6) the final form of approximation to the goals can be obtained as

$$G_i : f_i\left(X^o\right)+\left[\nabla f_i\left(X^o\right)\right]^T\left(P-Q\right)+D_i^- - D_i^+ = I \qquad (7)$$

In searching satisfactory decision with the use of linear approximation to f_i in the neighborhood of X_k^0, it is to be noted that X_k must not take any value lower than the corresponding lower tolerance limits X_k^b. So, in defining the neighborhood of X_k^0, the tolerance distance that X_k may move is found as

$$t_k = X_k^0 - X_k^b .$$

As a matter of fact, the tolerance range for $\left(X_k - X_k^0\right)$ is obtained as

$$-t_k \leq \left(X_k - X_k^0\right) \leq t_k .$$

Consequently, the range of $p_k - q_k$ of the expression in (7), can be defined as

$$-t_k \leq p_k - q_k \leq t_k$$

where $p_k - q_k$ is the k-th component of P-Q.

Then for $p_k \geq 0$ and $q_k \geq 0$, it follows that

$$0 \leq p_k \leq t_k, 0 \leq q_k \leq t_k, \quad k = 1,2. \qquad (8)$$

It is to be mentioned here that any value of X_k higher than its upper tolerance limit X_k^w is not acceptable to the DMs. So the tolerable range of X_k is found as $X_k^w - X_k^0$. Consequently, the resultant upper bound restrictions on p_k and q_k in (8) are obtained as

$$0 \leq p_k \leq \min\left\{t_k, X_k^w - X_k^0\right\}$$
$$0 \leq q_k \leq \min\left\{t_k, X_k^0\right\}$$
i.e., $0 \leq p_k \leq v_k$
$$0 \leq q_k \leq w_k, \quad k = 1,2. \qquad (9)$$

where $v_k = \min\left\{t_k, X_k^w - X_k^0\right\}$ and $w_k = \min\left\{t_k, X_k^0\right\}$ respectively.

6 Formulation of Linear GP Model

To formulate the GP model of QFGP, it is to be noted that achievement of the goals in (7) depends on the achieved values of p_k and q_k with their upper bound restrictions given in (9). So the expressions in (9) can also be considered as flexible goals with the view of optimizing them first by minimizing their over deviational variables. So the FGP problem can be formulated as

$$Minimize Z = \sum_i D_i^- + \sum_k \left(a_k^+ + b_k^+\right)$$

and satisfy

$$f_i\left(X^o\right)+\left[\nabla f_i\left(X^o\right)\right]^T\left(P-Q\right)+D_i^- - D_i^+ = I$$
$$p_k + a_k^- - a_k^+ = v_k$$
$$q_k + b_k^- - b_k^+ = w_k \qquad (10)$$

where $p_k, q_k \geq 0, D_i^-, D_i^+ \geq 0$ with $D_i^-.D_i^+ = 0$, and $a_k^-, a_k^+, b_k^-, b_k^+ \geq 0$ with $a_k^-.a_k^+ = 0$ and $b_k^-.b_k^+ = 0$ for $i,k = 1,2$.

a_k^-, a_k^+ and b_k^-, b_k^+ denote, respectively, the under- and over – deviational variables for the aspired goal levels of p_k and q_k.

The *minsum* GP [40] can be used to solve (10) and thereby the satisfactory decision for both the leader and the follower can be obtained in the decision situation.

7 Numerical Example

The following Quadratic Fractional Bilevel Programming Problem (QFBLPP) studied by Mishra and Ghosh [39] is considered and solve to illustrate the efficiency of the proposed methodology.

The QFBLPP is presented as

Find $X(x_1, x_2)$ so as to

$$\underset{x_1}{Min}\ F_1(x_1,x_2) = \frac{4x_1^2 + x_2^2 + 1}{2x_1^2 + 5x_2^2 + 1}$$

(Leader's Problem)

and for given X_1; X_2 solves

$$\underset{x_2}{Min}\ F_2(x_1,x_2) = \frac{3x_1^2 + 5x_2^2 + 1}{4x_1^2 + 3x_2^2 + x_2}$$

(Follower's Problem)

subject to

$$-5x_1 + 3x_2 \leq 15, 4x_1 + 3x_2 \leq 45, x_1, x_2 \geq 0. \qquad (11)$$

The leader's individual best and worst solutions are obtained as $(x_1, x_2; F_1^b) = (0.1, 5.12; 0.21)$ and $(x_1, x_2; F_1^w) = (11.25, 0; 2)$, respectively. Similarly the follower's individual best and worst solutions are found as $(x_1, x_2; F_2^b) = (11.16, 0.11; 0.75)$ and $(x_1, x_2; F_2^w) = (0.34, 5.57; 1.58)$, respectively.

Then the tolerance values for the respective fuzzy goals of the leader and follower are obtained as

$$F_1 \lesssim 0.21, \quad F_2 \lesssim 0.75.$$

Also the tolerance ranges of the decision variables are considered as

$0 \le x_1 \le 11.16$ and $0.11 \le x_2 \le 5.12$.

Now, using the above tolerance ranges the membership goals are obtained by using (4) as

$$0.56\left(2 - \frac{4x_1^2 + x_2^2 + 1}{2x_1^2 + 5x_2^2 + 1}\right) + d_1^- - d_1^+ = 1$$

$$1.2\left(1.58 - \frac{3x_1^2 + 5x_2^2 + 1}{4x_1^2 + 3x_2^2 + x_2}\right) + d_2^- - d_2^+ = 1$$

$d_i^-, d_i^+ \ge 0$ with $d_i^- . d_i^+ = 0$, $i = 1,2$. (12)

After linearizing the fractional goals in (12) the model takes the form as

$$-2x_1^2 + 0.04x_2^2 - 0.56 + D_1^- - D_1^+ = 1$$

$$-0.02x_1^2 - 3.31x_2^2 - 0.2 + D_2^- - D_2^+ = 1$$

where $D_1^- = d_1^-(2x_1^2 + 5x_2^2 + 1)$, $D_1^+ = d_1^+(2x_1^2 + 5x_2^2 + 1)$ and $D_2^- = d_2^-(4x_1^2 + 3x_2^2 + x_2)$, $D_2^+ = d_2^+(4x_1^2 + 3x_2^2 + x_2)$.

$D_i^-, D_i^+ \ge 0$ with $D_i^- . D_i^+ = 0$, $i = 1,2$ (13)

To approximate the quadratic goals in (13), the initial approximate solution is considered as $(x_1^0, x_2^0) = (2,2)$. Then applying the proposed approximation methodology, the resultant *minsum* GP model can be formulated as

$$Minimize\ Z = D_1^- + D_2^- + a_1^+ + a_2^+ + b_1^+ + b_2^+$$

so as to

$$-8(p_1 - q_1) + 0.16(p_2 - q_2) + D_1^- - D_1^+ = 1$$

$$-0.08(p_1 - q_1) - 13.24(p_2 - q_2) + D_2^- - D_2^+ = 1$$

$$p_1 + a_1^- - a_1^+ = 2$$

$$p_2 + a_2^- - a_2^+ = 1.89$$

$$q_1 + b_1^- - b_1^+ = 2$$

$$q_2 + b_2^- - b_2^+ = 1.89$$

where $p_k, q_k \ge 0, D_i^-, D_i^+ \ge 0$ with $D_i^- . D_i^+ = 0$, and $a_k^-, a_k^+, b_k^-, b_k^+ \ge 0$ with $a_k^- . a_k^+ = 0$ and $b_k^- b_k^+ = 0$.

for $i, k = 1,2$. (14)

The *software* LINGO (ver 6.0) is used to solve the problem.

7.1 Results and Discussions

Solving (14), the values of p_1, q_1, p_2 and q_2 are found as $p_1 = 0, q_1 = 0.126$, $p_2 = 0, q_2 = 0.1$.

The resulting solution to the original problem is obtained as $x_1 = 1.87, x_2 = 1.9$.

Hence the achieved objective function values of the leader and follower are calculated as F_1 = 0.71 and F_2 = 1.1, respectively.

The resulting membership values are given by

$$\mu_{F_1} = 0.72, \mu_{F_2} = 0.58.$$

In this connection it is to be noted that the solution obtained by Mishra and Ghosh [39] is $x_1 = 0.94$, $x_2 = 2.17$ with $F_1 = 0.35, F_2 = 1.37$. The resultant membership values are $\mu_{F_1} = 0.92$ and $\mu_{F_2} = 0.25$.

The solution shows that the follower's decision is fully dominated by the leader in their approach as like traditional BLPP. But in the proposed methodology, a compromised decision is achieved by maintaining hierarchy of decision powers of the DMs for overall benefit of the organization. The comparison reflects the superiority of the proposed approach over the other methodology.

8 Conclusions

In this paper, a QFBLPP is discussed in a hierarchical decision making environment for finding most satisfactory solution to the DMs for overall benefit of the organization. The proposed procedure can be extended to solve multilevel programming problems having quadratic fractional objectives. Also this methodology can be used to solve quadratic fractional decision making problems in a fully fuzzified domain. The proposed methodology can be applied to different real life problems for obtaining most satisfactory solution in a hierarchical decision making environment. However, the proposed procedure may open up new vistas into the way of making decision having nonlinear objectives of the DMs.

References

[1] Candler, W. and Townsley, R. (1982) 'A linear two-level programming problem'. *Computers and Operations Research*, Vol. 9, pp.59 – 76.

[2] Bialas, W.F. and Karwan, M.H. (1982) 'On two-level optimization'. *IEEE Transactions on Automatic Contro*l, Vol. 27,pp.211 – 214.

[3] Bialas, W.F. and Karwan, M.H. (1984) 'Two-level linear programming', *Management and Science*, Vol. 30, pp.1004 – 1020.
[4] Zimmermann, H.J. (1978) 'Fuzzy programming and linear programming with several objective functions', *Fuzzy Sets and Systems,* vol. 1, pp. 45 – 55.
[5] Lai, Y.J. and Hwang, C.L. (1996) *Fuzzy Mathematical Programming Methods and Applications*, Berlin: Springer.
[6] Shih, H.S., Lai, Y.J. and Lee, E.S. (1996) 'Fuzzy approach for multi-level programming problems', *Computers and Operations Research*, Vol. 23, pp.73-91.
[7] Bellman, R.E. and Zadeh, L.A. (1970) 'Decision-making in a fuzzy environment', *Management Sciences,* Vol. 17, pp. 141 – 164.
[8] Mohamed, R.H. (1997) 'The relationship between goal programming and fuzzy programming', *Fuzzy Sets and Systems,* Vol. 89, pp. 215 – 222.
[9] Sinha, S. and Biswal, M.P. (2000) 'Fuzzy programming approach to bi-level linear programming problems' *The Journal of Fuzzy mathematics,* Vol. 8, pp. 337 – 347.
[10] Moitra, B.N. and Pal, B.B. (2002) 'A fuzzy goal programming approach for solving bilevel programming problems', *Advances in Soft Computing-AFSS 2002*, Berlin: Springer, pp.91 – 98.
[11] Pal, B.B. and Moitra, B.N. (2003) 'A Fuzzy goal programming procedure for solving quadratic bilevel programming problems', *International Journal of Intelligent Systems,* Vol. 18, pp.529 – 540.
[12] Biswas, A. and Pal, B.B. (2007) 'Fuzzy goal programming procedure for multiobjective bilevel programming problems' *International Journal of Computer, Mathematical Sciences and Applications,* Vol. 1, pp. 87 – 98.
[13] Pal, B.B. and Biswas, A. (2007) 'Fuzzy goal programming approach for solving bilevel decentralized planning problems', *International Journal of Mathematical Sciences*, Vol. 6, pp.507 – 517.
[14] Pramanik, S. and Roy, T.K. (2007) 'Fuzzy goal programming approach to multilevel programming problems', *European Journal of Operational research,* Vol. 176, pp. 1151 – 1166.
[15] Baky, I.A. (2010) 'Solving multi-level multi-objective linear programming problems through fuzzy goal programming approach', *Applied Mathematical Modelling,* Vol. 34, pp. 2377 – 2387.
[16] Biswas, A. and Bose, K. (2010) 'A Fuzzy Goal Programming Technique for Solving Fully Fuzzified Quadratic Bilevel Constrained Programming Problems', In: Proceedings of the *International Conference on operations and Management Sciences ICOMS-10,* Excel India Publishers, New Delhi, pp. 144 – 152.
[17] Neumann, J.V. (1937) 'Uber ein es Gleichungssystem and eine Verallgemeinerung des Brouwerschen Fixpuntsatzes', *Ergebnisse eines mathematicschen Kolloquiums*, Leipzig and Wien, Vol. 8, pp. 245 – 267.
[18] Charnes, A. and Cooper, W.W. (1962) 'Programming with linear fractional, *Naval Research Logistics Quarterly,* Vol. 9, pp. 181 – 186.
[19] Swarup, K. (1965) 'Some aspects of linear fractional functional programming', *Australian Journal of Statistics*, Vol. 7, pp. 90 – 104.
[20] Dinkellbach, W. (1967) 'On nonlinear fractional programming', *Management Sciences,* Vol. 13, pp. 492 – 498.
[21] Geoffrion, A.M. (1967) 'Stochastic programming with aspiration or fractile criteria', *Management Sciences,*, Vol. 13, pp. 672 – 679.
[22] Bitran, G.R. and Magnanti, T.L. (1976) 'Duality and sensitivity analysis for fractional programs', *Operations Research,*, Vol. 24, pp. 675 – 699.
[23] Ibaraki, T., Ishii, H., Iwase, J., Hasegawa, T. and Mine, H. (1976) 'Algorithms for quadratic fractional programming problems', *Journal of Operational Research Society of Japan,* Vol. 19, pp. 174 – 191.
[24] Crouzeix, J.P. and Ferland, J.A. (1991) 'Algorithms for generalized functional programming' *Mathematical Programming,* Vol. 52, pp. 191 – 207.
[25] Rodenas, R.G., Lopez, M.L. and Verastegui, D. (1999) 'Extentions of Dinkelbach's algorithm for solving non-linear fractional programming problems', *Sociedad de Estadistica e Investigacion Operativa Top,* Vol. 7, pp. 33 – 70.
[26] Benson, H.P. (2006) 'Fractional programming with convex quadratic forms and functions, *European Journal of Operational Research,* Vol. 173, pp. 351 – 369.
[27] Isbell, J.R. and Marlow, W.H. (1956) 'Attrition Games', *Naval Research Logistics Quarterly,* Vol. 3, pp. 71 – 93.
[28] Arisawa, S. and Elmaghraby, S.E. (1972) '*Linear Programming and Network Flows, (2nd ed.),* John Wiley & Sons Inc., New York.
[29] Spiess, H. and Florian, M. (1989) 'Optimal strategies: a new assignment model for transit networks', *Transportation Research Part B: Methodological,* Vol. 23, pp. 83 – 102.
[30] Schaible, S. (1982) 'Bibliography in fractional programming', *Mathematical Methods of Operations Research,* Vol. 26, pp. 211 – 241.
[31] Dutta, D., Tiwary, R.N. and Rao, J.R. (1992) 'Multiple objective linear fractional programming – a fuzzy set theoretic approach', *Fuzzy Sets and Systems,* Vol. 52, pp. 39 – 45.
[32] Pal, B.B., Moitra, B.N. and Maulik, U. (2003) 'A goal programming procedure for fuzzy multiobjective linear fractional programming problem', *Fuzzy Sets and Systems,* Vol. 139, pp. 395 – 405.
[33] Pal, B.B. and Moitra, B.N. (2005) 'A fuzzy goal programming procedure for linear fractional bilevel

programming problems', *International journal of Management and Systems,* Vol. 21, pp. 35 – 52.

[34] Toksari, M.D. (2008) 'Taylor series approach to fuzzy multiobjective linear fractional programming', *Information Sciences,* Vol. 178, pp. 1189 – 1204.

[35] Ahlatcioglu, M. and Tiryaki, F. (2007) 'Interactive fuzzy programming for decentralized two-level linear fractional programming (DTLLFP) problems' *The International Journal of management science,* Vol. 35, pp. 432 – 450.

[36] Abo-Sinna, M.A. and Baky, I.A. (2010) 'Fuzzy goal programming procedure to bilevel multiobjective linear fractional programming problems', *International Journal of Mathematics and Mathematical Sciences,* Vol. 2010, pp. 1 – 15.

[37] Mehrjerdi, Y.Z. (2011) 'Solving fractional programming problem through fuzzy goal setting and approximation', *Applied Soft Computing,* Vol. 11, pp. 1735 – 1742.

[38] Alemayehu, G. and Arora, S.R. (2002) 'Bilevel quadratic fractional programming problem', *International Journal of Management and Systems*, Vol. 18, pp. 39 – 48.

[39] Mishra, S. and Ghosh, A. 'Interactive fuzzy programming approach to bi-level quadratic fractional programming problems', *Annals of Operations Research,* Vol. 143, pp. 251 – 263.

[40] Ignizio, J.P. (1976) *Goal Programming and Extentions*, Lexington, MA: Lexington Books.

[41] Inuiguchi, M., Ichihashi, H. and Kume, Y. (1990) 'A solution algorithm for fuzzy linear programming with piecewise linear membership functions'*, Fuzzy Sets and Systems*, Vol. 34,pp.15 – 31.

[42] Yang, T., Ignigio, J.P. and Kim, H.J. (1991) 'Fuzzy programming with nonlinear membership functions: Piecewise linear approximation', *Fuzzy Sets and Systems*, Vol. 41, pp.39 – 53.

[43] Saber, H.M. and Ravindran, A. (1993) 'Nonlinear goal programming theory and practice: A survey', *Computers and Operations Research*, Vol. 20, pp.275 – 291.

Study on a Smooth Preprocessing for Spectrum Including Outlier

Sunil Chon, Sukanya Sankarganesh, Hyouckmin Yoo, Dong Sun Park [1]
[1] School of Electronics Engineering, Chonbuk National University, Jeonju, Jeonbuk, South Korea

Abstract - *In the signal process part, the Smooth algorithm is the one of the most primitive and important job for attempts to capture important patterns in the data. This paper shows that we can get the result of more higher Signal to Noise Ratio and get more clear patterns of the data set of signals affected outlier noise where apply preprocessing of Smooth before when using Smooth algorithm.*

Keywords: Smooth, Signal Processing, Preprocessing

1 Introduction

In the statistics, image processing and signal processing, to smooth a data set is to create an approximating function that attempts to capture important patterns in the noise included data. Because Smooth algorithm is non-linear, it does not show that the data set is clear functional expression, but the data set is the Smooth processed result. And purpose of smooth algorithm is to get approximate data set where using the random noise included observation data set.[1]

On the other hands, the Curve fitting involves the use of an explicit function form for the result and concentrates on achieving as close match as possible. In this way, there is difference between Smooth and Curve fitting.

We frequently could see that noise and unexpected outlier included in second X-ray fluorescence energy spectrum when developing hand held X-ray fluorescence analyzer.[5]

Smooth signal processing algorithm is applied in addition to other cases occurring in the outlier considering the impact on the results of the process distorted the results is impossible to predict.

There are two methods to restore the signal with noise and outlier. The first is Curve fitting, and the second is Smooth algorithm. But the Curve fitting has big computing cost than smooth algorithm. So if we use the relatively low computational cost than Curve fitting, and the processing of the first signal processing sequence algorithm – Smooth is relatively Outlier removable and can overcome the signal after the end of the process by which we get fairly accurate results.

2 Preprocess Algorithms

2.1 Moving Average Filter

Moving average filter is a simple algorithm that decide smooth values in observed data at each location where the specified window is as much as the mean value.[3]

$$y_i' = \frac{1}{2m+1}\sum_{j=-m}^{+m} y_{i+j} \tag{1}$$

Where y_i' is the smoothed value and the window size is $2m+1$.

2.2 Savitzky Golay Smooth Filter

Savitzky Golay(SG) smooth filter take the polynomial regression value (actually partial polynomial regression) that applied the point and the surrounding values.[Fig. 1][3][4]

$$\mathbf{y} = \mathbf{wX} + \boldsymbol{\varepsilon} \tag{2}$$

$$\mathbf{w} = (X^T X)^{-1} X^T y \tag{3}$$

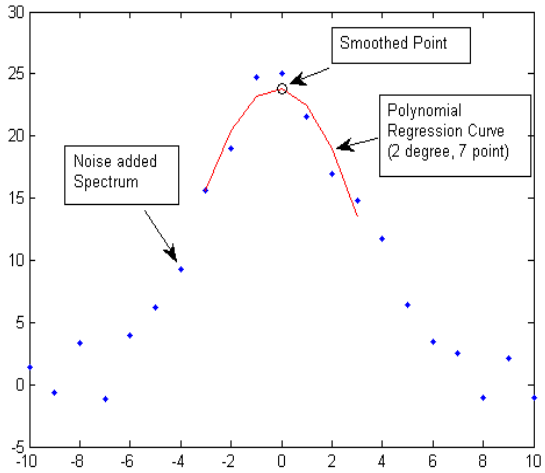


Figure 1. Example of smoothing point and apply partial polynomial regression(2degree, 7 point window)

Where **y** is observed values, **w** is polynomial coefficient matrix and **X** is the position value.

The observed values **y** is the formulation (2) that polynomial coefficient **w** and the position **X** with noise ε. We can compute the optimal polynomial coefficient **w** that satisfy the smallest Root Mean Square Error with source signal using Eq. (3).

We can get the SG filtered smooth results by apply Eq. (3) to all the points.

Because of SG Smooth filter apply partial polynomial regression to the data points that is strong about the noises. Moreover SG Smooth filter can get the optimal smooth values by applying this we can adjust the polynomial degree and the window size. These are the benefits of SG Smooth filter.

3 Proposal of Preprocessing Method

3.1 Explanation

This paper does not suggest Smooth algorithm. First identify the characteristics of the values to apply Smooth algorithm, and then propose the preprocess before apply Smooth filter that is already proposed to get Smooth result which is close to the original signal.

The Smooth algorithm has several characteristics when the data is applied to it. The values include white color noise with Gaussian distribution, and included Outliers in the signal.[Fig. 2 (a)] The purpose of Preprocessing of smooth is to remove the outliers and restore the signal which is more close to original signal.

The characteristic of observed values leads to the occurrence of Outlier in the signal, and we can find big impulse noises in that part.

To remove the effect of Outliers, we apply median filter that remove impulse noise in the steady signals.[Fig. 2 (b)][2]

Because of median filter has characteristics of obtaining the best results of values when applying first order polynomial or exponential function of the curved shape like a regular value, there occur problems that if we apply this median filter to the second order polynomial, sine, cosine wave form or Gaussian form then the peak part is flat.[Fig. 2 (c)]

To apply median filter to these wave form signals, first make a Model that the best express the signal and different with Curve fitting, second take the differences Model and observed signals. As a result, we can get the possible pattern to apply median filter.[Fig. 2 (d)]

Finally, add the median filtered value and Model, - although tiny noises remain - then we get the result of removed Outliers.

We get the restored signal result that is more precise and close to original signal that have high Signal to Noise Ratio(SNR) when applying preprocess to the observed signal a better result is obtained than just applying Smooth filters to the observed signal.

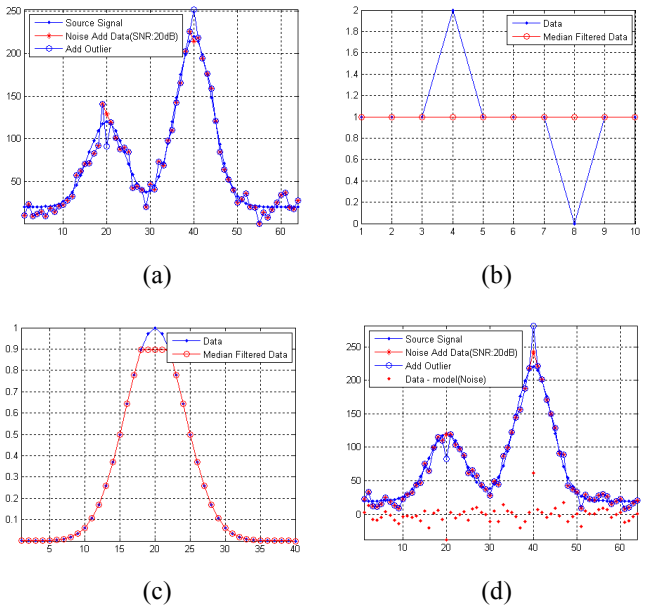


Figure 2. (a) Original signal + White Gaussian Noise + Outlier. (b) Median filter. (c) Apply median filter to gaussian curve. (d) Differences Model and observed signal

3.2 Summarized

The proposed algorithm consists of the following steps.

- Get the Model of observed signal
- Get the differences the Model and the observed signal
- Apply median filter to the differences
- Add the median filtered values and the Model
- Apply Smooth algorithm to the finally preprocessed result

4 Compare Simulation Result

Matlab is used for simulation and a 64 point spectrum is generated using the original signal that has two peaks with Gaussian form. With this 64 point spectrum we have generated 1000 data set that includes SNR for each data and to that we have added white Gaussian noise and outlier.[Fig. 3]

The process was first make a Model using SG filtered data, second apply preprocess and finally apply SG smooth filter(2 degree, various window size).

TABLE I. COMPARE THE SNR RESULT OF EACH SMOOTH PROCESS

SNR (dB)	Window Size	Proposed		SG Filter		Moving Average	
		With Outlier	*Without Outlier*	*With Outlier*	*Without Outlier*	*With Outlier*	*Without Outlier*
40	3	**37.73**	43.76	**21.19**	41.06	**26.07**	36.62
	7	**41.14**	44.46	**26.04**	45.02	**22.07**	21.88
	11	**35.69**	35.69	**28.17**	36.10	**14.74**	14.51
	15	**26.79**	26.75	**25.93**	26.84	**10.13**	10.00
	19	**20.44**	20.44	**20.50**	20.46	**7.20**	7.11
30	3	**32.51**	34.24	**20.81**	31.03	**25.68**	33.46
	7	**34.81**	35.85	**25.65**	35.76	**22.00**	21.81
	11	**33.71**	33.75	**27.79**	34.17	**14.73**	14.50
	15	**26.63**	26.51	**25.76**	26.64	**10.13**	10.00
	19	**20.44**	20.39	**20.46**	20.42	**7.20**	7.11
20	3	**23.65**	24.40	**18.21**	21.09	**23.00**	25.58
	7	**25.47**	26.13	**23.03**	25.93	**21.38**	21.26
	11	**26.76**	27.17	**25.17**	27.55	**14.64**	14.42
	15	**24.83**	24.82	**24.40**	25.08	**10.10**	9.97
	19	**20.08**	19.99	**20.10**	20.07	**7.18**	7.10
10	3	**14.23**	14.57	**11.01**	11.39	**15.56**	15.93
	7	**15.91**	16.25	**15.63**	16.02	**17.64**	17.73
	11	**17.43**	17.73	**17.73**	18.05	**13.85**	13.82
	15	**18.21**	18.45	**18.65**	18.86	**9.85**	9.80
	19	**17.17**	17.33	**17.50**	17.63	**7.07**	7.02

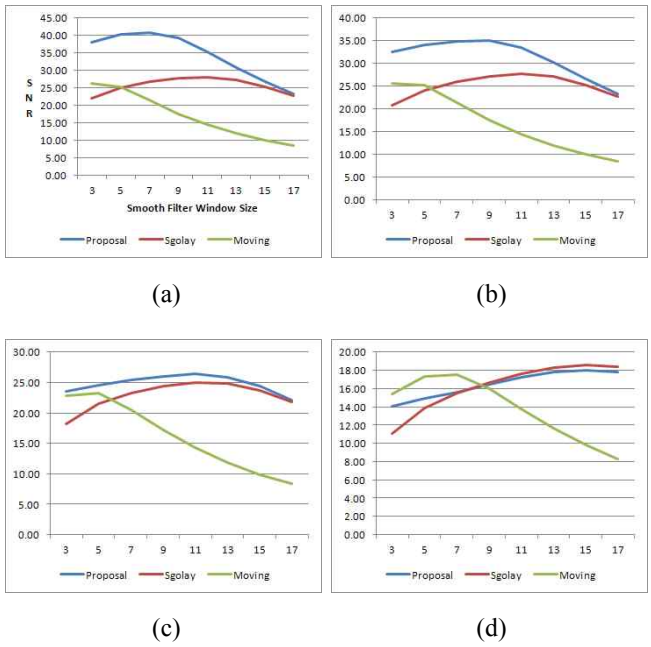


Figure 3. All signals inclue 40~50 Outliers like Fig. 2(a). (a) 40dB signal. (b) 30dB Signal. (c) 20dB signal. (d) 10dB signal

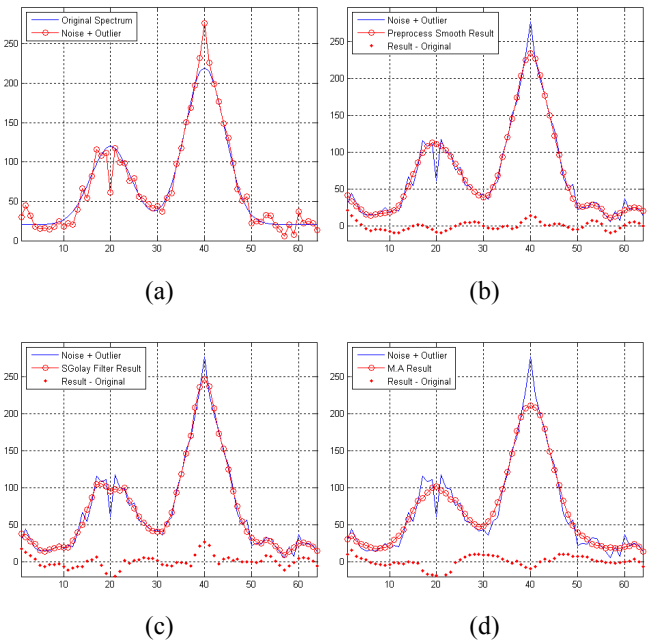


Figure 4. In the 20dB SNR using Smooth window size 7 (a) Original signal and Noise + Outlier. (b) Preprocess Smooth result and differences with Original. (c) Only SG Filter result and differences with Original. (d) Only Moving Average Filter result and differences with Original signal

Table 1 compares the SNR result of each Smooth process, that having two categories. The first one is when applying Outlier to the signal and the second one is without applying Outliers to the signal.

While comparing the SG filters and Preprocess without applying Outlier to the signal we can get a better result using Preprocess then when compared to SG filter in the small window size part. But it the big window size part the maximum SNR is good at SG filter than Preprocess.

In the 40, 30dB SNR result as show in Table 1, it is hard to say which is better. Because in the 40, 30 dB signals it has very low noise, so there are very low probability of this type of Outlier occurring in the signal. Compare the result of the 20, 10dB that has very low SNR, where include 40~50 Outliers, we can see 20dB results are good when Outliers removed. But 10dB results are most similar of SG Smooth filter. We can see that at the small window size the result is better than on applying SG Smooth filter to the noise with Outlier signals.

Figure 4 is the example of applying Smooth filters and applying Preprocess. Generated 20dB SNR signal and added 40~50 Outlier to the peak position. The o line is the results of each process of observed Outlier signal. And the dots are differences of results and the original signal. As we can see that the point of occurrence of Outlier, SG filter and Moving Average filter are influenced by the Outlier.

5 Discussion

The proposed method has two main points on discussion.

At first the model is created. SG Smooth result is obtained to create the model and the resulting algorithm is applied once the filters has been used as a model. As Curve fitting is computationally costly and in order to reduce the computational cost and better result this model is obtained. As can be seen in the proposed method, it is important to make the model well. Because depending on how well the model is drafted we may get different results on applying preprocessing.

The second one is to determine the size of median filter window. Since these are still some remaining signal component that resulted in the difference between the observed signal and the model we are using Median filters too large the remaining signal components are lost and we get a distorted result. In contrast upon using small window size it may affect the local noise and we can get a better result.

Best result were determined when the values of the window size was 3 or 5 in the experient.

6 Conclusion

In the second X-ray fluorescence energy spectrum, we can find that elements of the energy signal is a mixture of small parts and large parts.[6] If noise is added to a similar size - 0 mean, σ dispersion Gaussian distributed white noise - a small portion of the signal, the small signal to noise ratio is smaller and Outlier more receive higher.[Fig. 5 (d)]

In this paper, we propose a relatively simple Smooth preprocessing method that is robust to Outlier remove. Smooth as a way to suggest the results of applying the algorithm, we can get the closer result of original signal in Outlier included observed signal. In addition, because of the spectrum peaks has different SNR, we will get better Smooth result that depending on the detector's efficiency curve for Smooth progress, including preprocessing.

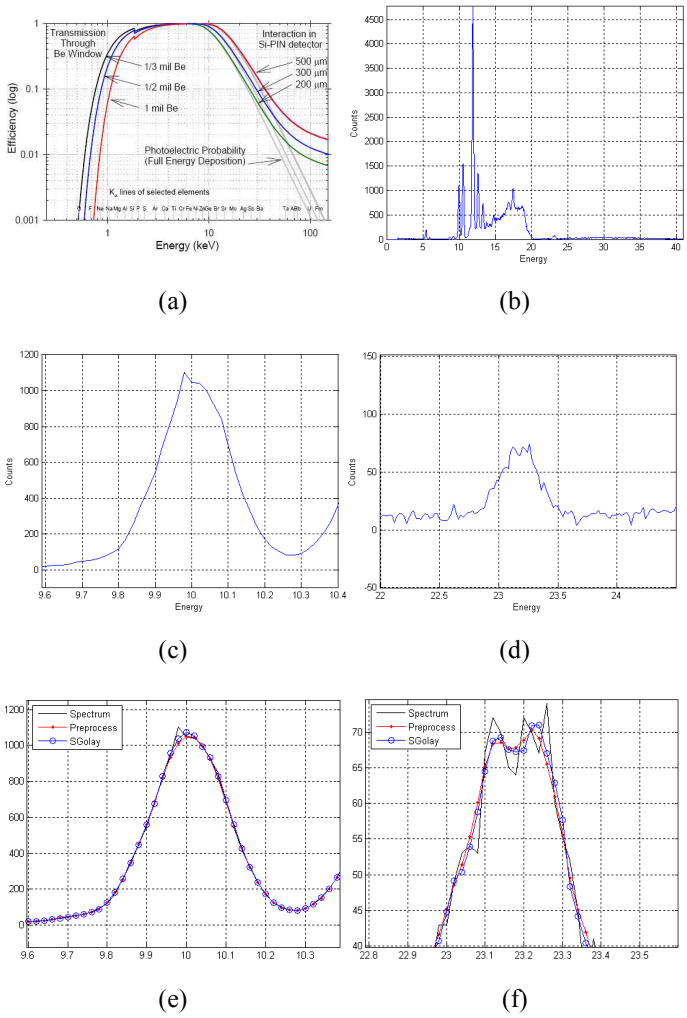


Figure 5. (a) XRF Detector's Efficiency Curve. (b) Plastic PE Spectrum(Cr 1000, Br 1100, Cd 330, Hg 1100, Pb 1200 ppm). (c) Enlargement of Energy 10keV Peak. (d) Enlargement of Energy 23.2keV Peak. (e) Apply Preprocess

and SG Filter at 10keV Energy. (f) Apply Preprocess and SG Filter at 23.2keV Energy

Acknowledgment

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation. And this work was supported by National Research Foundation of Korean Grant funded by the Korean Government (2009-0077772).

References

[1] http://en.wikipedia.org/wiki/Smoothing

[2] http://en.wikipedia.org/wiki/Median_filter

[3] P. Van Espen, "Handbook of X-ray Spectrometry 2nd Edition, Revised and Expanded", Ch4. Spectrum Evaluation, Marcell Dekker, pp. 249-252, 2002.

[4] Savitzky, A. Golay, M.J.E. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," Analytical Chemistry 36(8), pp. 1627-1639, 1964.

[5] M. Uda, O. Benka, K. Fuwa, K. Maeda, Y. Sasa, "Chemical effects in PIXE spectra", Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, Volume 22, Issues 1-3, Pages 5-12, 3 March 1987.

[6] Amptek Detector Efficiency Curves : http://amptek.com/xr100cr.html

Application of iSIGHT® (OLH & RBF Modules) to Optimal Design of a Dynamical System with High Speed Spindle considering Thermal Behavior and Natural Frequency

Su Seong Park[1], Won Jee Chung[1], Jin Su Ahn[1], Soo Tae Kim[2], Seog Jun Lee[2], and Dae Bong Choi[3]
[1]Dept. of Mechanical Design & Manufacturing Engineering, Changwon National University, Changwon-si, Gyeongsangnam-do, South Korea
[2]Control & instrumentation engineering, Changwon National University, Changwon-si, Gyeongsangnam-do, South Korea
[3]Dept. of Ultra Precision Machines and Systems, Korea Institute of Machinery & Materials / Changwon-si, Gyeongsangnam-do, South Korea

Abstract - *This paper presents the application of iSIGHT® 4.5 (OLH & RBF Modules) to optimal design of a dynamical system with high speed spindle considering thermal behavior and natural frequency. To analyze the change of thermal deformation according to spans of bearings, we will establish the FE (Finite Element) model of high-speed spindle system using ANSYS Workbench® 12.1. The change of natural frequency according to spans of bearings will be analyzed by constructing the FE model of high-speed spindle system using ARMD®, which is a special analysis software of rotating-shaft. Based on the two FE models, we will find out the optimal position of 4 bearings through predicting natural frequency and thermal deformation by applying iSIGHT® 4.5 (OLH (Optimal Latin Hypercube) & RBF (Radial basis Function)). Compared with the initial design based on the experience of an expert, thermal deformation using the optimized values will be shown to be decreased while natural frequency using the optimized values will be shown to be increased.*

Keywords: *High-Speed Spindle, Finite Element Modeling, Optimized Approximation Model, Optimal Latin Hypercube (OLH), Radial Basis Function (RBF)*

1 Introduction

This present, the demand for high-quality machining of parts and materials is increasing because of development of the material industry related to the high-tech industry such as aerospace, automobile, and cellular phone. To meet this demand, the high-speed and high-precision machining centers are necessary so that they are under development for Germany, Japan, Korea, and *etc*. The dynamical system with high-speed spindle which is a core part of machining centers needs to have powerful machining as well as high-precision cutting ability. In addition, large static and dynamic stiffness is required.

In existing spindles for machining centers, power is delivered from a motor to a spindle through coupling, belt, gear and *etc*. It can result in vibration and noise problems during operating machining centers. To cope with these problems and then to obtain satisfied performance, the construction of high-speed spindle should be in the form of built-in motor. The high-speed spindle with built-in motor has a simple construction to remove the problems caused by coupling, belt, gear and *etc*. But thermal deformation will occur by internal heating during operating a high-speed spindle. The magnitude in thermal deformation usually amounts to several tens㎛'s, in contrast to the static and dynamic deformation with a few ㎛'s. This is the issue that takes precedence over other problems.

In previous studies, Park *et al*. [1] have investigated on selecting the positions of bearings considering static and dynamic stiffness of high-speed spindle regardless of thermal deformation. Park *et al.* [2] has suggested a design and performance estimation technique of high-speed spindle using DOE (Design Of Experiment) which is based on Taguchi method by using Minitab®. But the number of experiments is so small that it is questionable whether an optimal result could have been found out.

Kwon *et al.* [3] has figured out the optimized design technique of tool holder spindle using iSIGHT® 4.5. The technique performed with ANSYS Workbench® is only focusing on stiffness with the exception of thermal deformation.

Therefore this paper aims at applying iSIGHT® 4.5 (OLH (Optimal Latin Hypercube) & RBF (Radial Basis Function)) to the optimal design of a dynamical system with high-speed spindle decreasing the thermal deformation in consideration of thermal behavior (using ANSYS Workbench®) and natural frequency (using ARMD®).

2 Design of Finite Element Modeling for Dynamical System with High Speed Spindle

Figure 1 shows the initial drawing (designed by the experts of KASWIN, Ltd., Korea) of the high-speed spindle with built-in motor.

The rotating-shaft of the spindle as shown in Fig. 2 is a structure supported by both 2 bearings in front of rotating-shaft and 2 bearing in the rear of rotating-shaft. Bearing type is angular contact ball bearing and oil & air system is applied for lubrication.

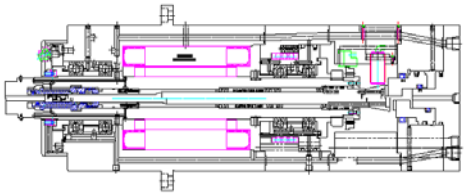


Fig. 1. Initial drawing of high-speed spindle

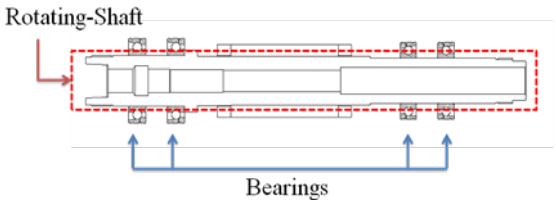


Fig. 2. Drawing of rotating-shaft of the spindle

The maximum allowed number of pages is seven for Regular Research Papers (RRP) and Regular Research Reports (RRR); four for Short Research Papers (SRP); and two for Posters (PST).

2.1 Finite element model of thermal deformation analysis

To analyze the change of *thermal deformation* according to spans of bearings, we have also established the FE model of high-speed spindle system using ANSYS Workbench® 12.1.[4] The spindle system has axial symmetry so that we have established the FE model of Fig. 3, which is a half of the total system. Parts which have no influence on results of analyses, such as assembly sections, supply lines of lubricating oil & air, and *etc*. are approximated simply as shown in Fig. 3. Figure 4 shows the spans of bearing to be used for natural frequency and thermal deformation analysis. FB and RB denote front bearings and rear bearings, respectively.

.

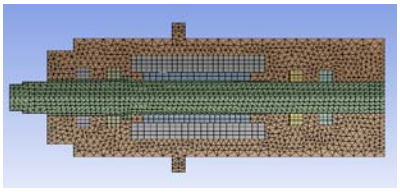


Fig. 3 FE model of spindle system using ANSYS workbench®12.1

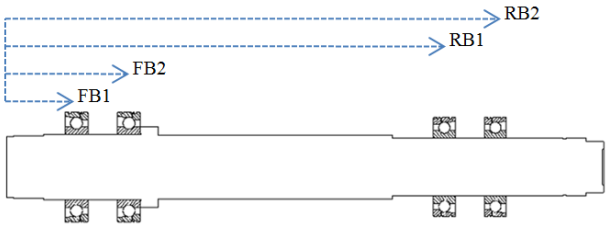


Fig. 4 Schematic diagram of bearing position

2.2 Finite element model of natural frequency analysis

To analyze the change of *natural frequency* according to spans of bearings, we have established the FE (Finite Element) model of high-speed spindle system using ARMD®[5] which is a special analysis software of rotating-shaft, as shown in Fig. 5.

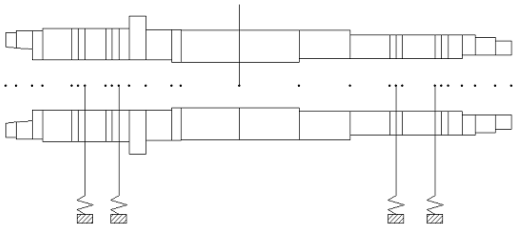


Fig. 5 FE model of spindle system using ARMD®

In general, front bearings results in high heat due to cutting power, loading condition and some additional causes, compared with the case of rear bearing. In this paper, the span of bearings of front and rear bearings has been assumed to be altered within the range of (±5mm + nominal (initial) position) considering structural constraints. The **p**roperties to be used for thermal deformation analysis are listed in Table 1. Table 2 shows initial values and upper/lower bounds of 4 bearing positions (see Fig. 4).

Table 1 Material Properties of Spindle System

	Density (kg/m3)	Specific Heat (J/kg•℃)	Thermal Conductivity (W/m•℃)
Housing	7817	446	52
Rotor	6250	590	90
Stator	8124	437	148
Axis	1.165	1006	0.026
Air	7769	473	43

Table 2 Initial Values and Lower/Upper Bounds of 4 Bearings (FB1/FB2/RB1/RB2)

(unit : mm)

	Initial Value	Lower Bound	Upper Bound
FB1	53	48	58
FB2	84	74	84
RB1	300.5	295.5	305.5
RB2	330.5	330.5	340.5

3 Application of iSIGHT® to Optimal Design

3.1 Simulation using DOE(Design Of Experiment) with OLH

It is well known that it is difficult to achieve the purpose of an experiment without sufficient plan of experimental factors. DOE *i.e.*, Design of Experiment will lead to efficient experimental planning [6].

DOE, as a statistical analysis methodology, is mainly used at the measurement step of 6 Sigma. Especially DOE can be used for CAE (Computer Aided Engineering) as well as actual experiments. An analysis with DOE needs less time to find out main effect variables, compared with the analysis without DOE.

According to previous studies[7][8], major factors having influence on natural frequency are diameter of rotating-shaft, stiffness, span of bearing, and elasticity of the material. In addition, a bearing has an effect on thermal deformation due to the internal heating of itself. In this paper, we will find out the optimal position of 4 bearings through predicting natural frequency and thermal deformation by applying iSIGHT® 4.5[9] (OLH & RBF).

In OLH method (as offered by one of modules in iSIGHT® 4.5), we generates the level values of design variables by using OLH method (as offered by one of modules in iSIGHT® 4.5). The OLH can sample the level values automatically under regular pattern instead of random one [3]. This results in one of the merits for OLH method. In specific, OLH offers ANOVA (Analysis of Variance) with high reliability and in turn can make the approximation model (given by RBF method) as suitable as the FE model. As shown in Table 3, we have obtained 100 sets of inputs (design variables) according to the DOE using OLH.

Figure 7 illustrates the procedure regarding how ANSYS Workbench ® 12.1 can be incorporated into iSIGHT® 4.5. First, the finite element modeling of high speed spindle is simulated for the base line values of 4 design variables (FB1, FB2 RB3, RB4) by using ANSYS Workbench ® 12.1. The initial simulation for the base line values aims at resulting in the LOG file of ANSYS Workbench ® 12.1 which can be imported into iSIGHT® 4.5. In iSIGHT® 4.5, 4 design variables are automatically input to the LOG file by using the OLH module. The OLH needs the lower bound, upper bound and increment for each design variable. The output of simulation using LOG file is thermal deformation of high speed spindle, as shown in the result (column) of Table 3.

Table 3 Level Values of Design Variables and Result of iSIGHT® 4.5 Simulation (incorporated in ANSYS workbench®)

(FB1, FB2, RB1, RB2 : mm Result : ㎛)

#	FB1	FB2	RB1	RB2	Result
1	55.37	77.03	297.22	331.21	26.967
2	55.27	82.89	297.82	333.93	27.3
3	57.6	74.4	300.85	336.36	26.567
4	50.83	78.04	305.4	334.34	28.667
5	57.39	77.74	297.72	336.76	27.567
6	54.26	74.71	304.49	337.27	27.533
7	51.33	78.44	304.99	333.33	27.167
8	56.28	75.92	298.63	335.35	28.367
9	48.51	82.69	302.07	333.53	27.567
10	53.15	75.62	298.13	339.69	26.633
11	54.87	80.36	305.5	334.64	27.633
12	57.29	79.25	302.97	334.94	27.733
⋮					
87	49.11	81.68	297.62	332.22	26.6
88	49.31	78.65	296.41	338.48	26.967
89	57.09	80.16	302.27	330.6	27.367
90	53.25	83.6	299.64	330.7	28.033
91	52.75	83.7	303.98	336.26	27.667
92	49.41	80.67	299.84	330.9	28.433
93	50.63	83.09	298.53	336.56	28.267
94	54.77	82.79	298.03	332.92	27.167
95	53.35	81.17	301.96	337.47	26.1
96	48.4	77.33	302.37	335.55	28.2
97	48.0	78.95	301.06	335.45	27.5
98	48.2	76.02	296.61	336.86	27.533
99	56.18	76.12	304.69	331.81	26.967
100	57.9	80.57	296.31	335.85	28.1

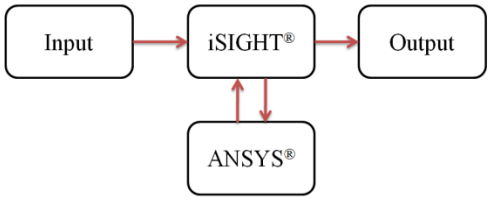


Fig. 6 Procedure of simulation using ANSYS workbench® and iSIGHT®

In order to proceed an optimal design using iSIGHT® 4.5, we need to incorporate iSIGHT® 4.5 into an analysis software in a similar manner to the case of ANSYS workbench®, as illustrated in Fig. 6. But ARMD® does not have the incorporation procedure of iSIGHT® 4.5 as shown in Fig. 7. Thus we have just performed the 100 simulations of ARMD® by using each set of level values of design variables according to the DOE using OLH in iSIGHT® 4.5, as shown in Table 4.

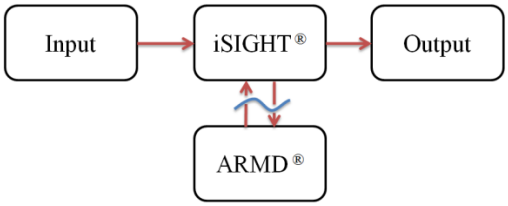


Fig. 7 Procedure of simulation using ARMD® and iSIGHT®

Table 4 Level Values of Design Variables and Result of iSIGHT® 4.5 Simulation
(not incorporated in ARMD®)
(FB1, FB2, RB1, RB2 : mm Result : Hz)

	FB1	FB2	RB1	RB1	Result
2	55.37	77.03	297.22	331.21	623.83
3	55.27	82.89	297.82	333.93	605.28
4	57.6	74.4	300.85	336.36	616.14
5	50.83	78.04	305.4	334.34	609.35
6	57.39	77.74	297.72	336.76	607.99
7	54.26	74.71	304.49	337.27	622.47
8	51.33	78.44	304.99	333.33	624.74
9	56.28	75.92	298.63	335.35	620.89
10	48.51	82.69	302.07	333.53	610.26
11	53.15	75.62	298.13	339.69	606.86
12	54.87	80.36	305.5	334.64	619.76
13	57.29	79.25	302.97	334.94	625.64
⋮					
87	48.3	75.01	304.39	336.36	618.85
88	49.11	81.68	297.62	332.22	621.12
89	49.31	78.65	296.41	338.48	623.38
90	57.09	80.16	302.27	330.6	622.02
91	53.25	83.6	299.64	330.7	615.23
92	52.75	83.7	303.98	336.26	610.48
93	49.41	80.67	299.84	330.9	623.61
94	50.63	83.09	298.53	336.56	626.32
95	54.77	82.79	298.03	332.92	625.87
96	53.35	81.17	301.96	337.47	606.64
97	48.4	77.33	302.37	335.55	605.73
98	48	78.95	301.06	335.45	618.63
99	48.2	76.02	296.61	336.86	611.16
100	56.18	76.12	304.69	331.81	612.29
101	57.9	80.57	296.31	335.85	619.99

3.2 Optimized Approximation

In general, it is accurate to employ FE models in all simulations using analysis software. But it has a demerit which usually needs a lot of computing time for optimal design problems of nonlinear analyse. Therefore, in this paper, we have performed the optimal design of high speed spindle by using the approximation technique of RBF (Radial Basis Function) module in iSIGHT® 4.5.

It is well known that RBF neural network is one of methods of curve fitting in multi-dimensional space. This means that training through RBF searches for one space from multi-dimensional spaces, which can be agreed best with data sets to be trained. Fig. 8 shows the structure of RBF. In specific, the data resulted from the previous subsection can be imported into the RBF module of iSIGHT® 4.5 so that two optimized approximation models (one model between 4 input variables (*i.e.*, FB1, FB2 RB3, RB4) and 1 output variable; another model between 4 input variables (*i.e.*, FB1, FB2 RB3, RB4) and 1 output variable (natural frequency) can be established.

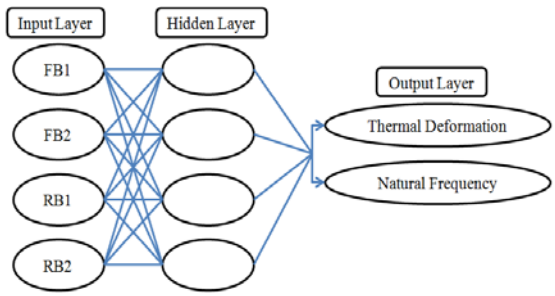
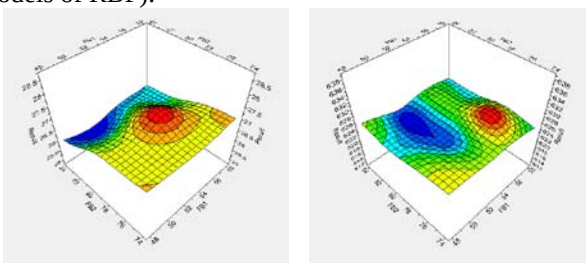


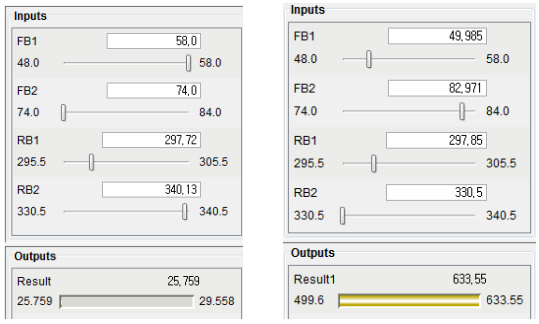
Fig. 8 Structure of RBF

As shown in Tables 3 and 4, we have obtained 100 sets of 4 inputs (design variables) and 2 outputs (thermal deformation and natural frequency) according to the DOE using OLH. These input and output data can be used for resulting in optimized approximation models based on the RBF module of iSIGHT® 4.5. The optimized approximation models resulted from the RBF module are used for finding out the optimal values of 4 design variables through 1,000 simulations. Figs. 9 and 10 shows optimized approximation models and optimized approximation values, respectively. Table 5 illustrates the values of optimal design variables which result in the minimum deformation and the maximum natural frequency for the rotating-shaft of high speed spindle. These optimal design variables are assigned to each FE modeling of ANSYS workbench® and ARMD®. Especially, the initial values of design variables are included for the comparison of optimized values (resulted from the optimized approximation models of RBF).



(a) Thermal deformation (b) Natural frequency

Fig. 9 Optimized approximation models



(a) Thermal deformation (b) Natural frequency

Fig. 10 Optimized approximation values

For the FE model of ANSYS workbench®12.1, the thermal deformation using the optimized values of 4 design variables has been decreased by 4.23%, compared with one using the initial values of those variables. In the meanwhile, for the FE model of ARMD®, the natural frequency using the optimized values has been increased by 16.89%, compared one using the initial values of those variables. These remarks are summarized in Table 6. In addition, Figs. 11 and 12 show the analysis results of ANSYS workbench® 12.1 (thermal deformation) and ARMD® (natural frequency) using the optimized values of 4 design variables, respectively.

Table 5 Initial Values of Design Variables *vs*. Optimized Values of Design Variables

(unit : mm)

	Initial Values of Design Variables	Optimized Values of Design Variables (Thermal Deformation)	Optimized Values of Design Variables (Natural Frequency)
FB1	53	58	49.99
FB2	84	74	82.97
RB1	300.5	297.72	297.85
RB2	330.5	340.13	330.5

Table 6 Comparison of Results using Initial Values and Optimized Values

	Result using Initial Values	Result using Optimized Values	Rate
Deformation (μm)	27.16	26.004	4.23% (decreased)
Natural Frequency (Hz)	520.18	625.88	16.89% (increased)

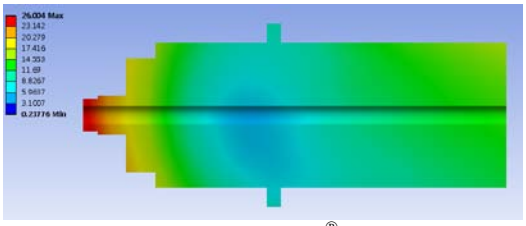


Fig. 11 Result of ANSYS workbench® using optimized values of design variables

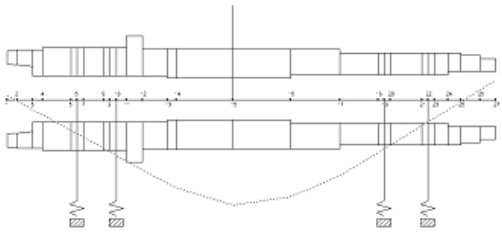


Fig. 12 Result of ARMD® using optimized values of design variables

The FE models of thermal deformation and natural frequency are almost the same as those of the optimized approximation models of RBF with 0.94% and 1.21% , respectively, as shown in Table 7.

Table 7 Comparison of FE Models with Optimized Approximation Models of RBF

	RBF Model	FE Model	Error
Deformation (μm)	25.759	26.004	0.94%
Natural Frequency (Hz)	633.55	625.88	1.21%

4 Final Optimal Design of High-speed Spindle (considering both thermal deformation and natural frequency)

In the previous section, we have obtained the optimal positions of bearings which have an effect on both thermal deformation and natural frequency using ANSYS Work bench®12.1 and ARMD®, based on OLH & RBF modules provided by iSIGHT® 4.5. This result means that the optimal position of bearings can be found out in consideration of both thermal deformation and natural frequency.

Under the advice of an experienced expert in the field of machining centers, it is assumed that the contrast between thermal deformation and natural frequency would be given in the sense of weighting factors, especially 0.8 and 0.2, respectively. Based on the weighting factors, we will obtain optimal position of bearings by using Minitab®[10] as follows. The objective function is concerned with the minimum thermal deformation and the maximum natural frequency. In order to make thermal deformation and natural frequency the same condition (*i.e.*, minimization), we have subtracted the values of natural frequency from 1000Hz. Thus, the major effect value for each design variable will be found using Minitab® to minimize both thermal deformation and natural frequency (subtracted from 1,000 Hz), This means that we need a multi-objective problem as follows:

$$Object = w_1 \times \frac{\text{Thermal Deformation}}{sf_1} + w_2 \times \frac{\text{Natural Frequency}}{sf_2} \quad (1)$$

Here *sf1* and *sf2* indicate scaling factors. Moreover, w_1 and w_2 are weighting factors. The scale factors are selected properly according to equation (1). Since the multi-objective function to be minimized has to be a linear combination function, 0.8 and 0.2 have been assigned to each w_1 and w_2 as mentioned before. In the meanwhile, the constants of *sf1* and *sf2* are given by the maximum values of thermal deformation and natural frequency, respectively, in order to impose the ceiling value of the maximum value of objective function on 1.

Before getting into Minitab®, we have sorted out 25 sets of 4 inputs (design variables) and 2 outputs (thermal deformation and natural frequency) from 1,000 simulations which are result of RBF as shown in Table 8. This is why Minitab® limits 25 sets of simulation cases. The criterion of this sorting is based on the ascending order of the values of objective

function from the minimum one among 1,000 simulation results using the RBF approximation models.

Table 8. 25 Sets sorted from 1,000 Simulation Cases from RBF

	FB1	FB2	RB1	RB2	Thermal Deformation	Natural Frequency
2	53.35	81.17	301.96	337.47	26.1	403.36
3	49.52	77.23	299.04	333.13	26.133	398.61
4	55.98	76.32	304.09	337.67	26.233	399.52
5	54.46	83.19	301.26	332.52	26.2	395.22
6	51.94	79.35	296.21	338.68	26.267	399.06
7	51.74	82.38	296.71	332.42	26.4	402.23
8	51.84	79.86	301.16	336.16	26.167	385.04
9	56.59	77.13	298.13	335.85	26.433	402.46
10	52.24	74.91	299.44	337.87	26.333	392.28
11	54.06	78.55	295.7	334.84	26.3	388.66
12	54.97	75.31	299.24	331.61	26.367	393.41
13	48.81	77.54	301.36	339.09	26.533	401.33
14	48.51	82.69	302.07	333.53	26.567	399.74
15	55.07	83.29	302.67	339.39	26.467	391.6
16	58	79.56	300.25	336.06	26.5	389.34
17	50.83	78.04	305.4	334.34	26.667	400.65
18	56.08	83.39	297.42	336.96	26.7	398.16
19	49.11	81.68	297.62	332.22	26.6	388.88
20	54.87	80.36	305.5	334.64	26.633	390.24
21	49.72	83.8	299.14	338.08	26.733	392.5
22	51.03	79.05	302.77	330.5	26.767	390.92
23	49.92	80.77	305.1	338.78	26.8	402.73
24	56.18	76.12	304.69	331.81	25.967	407.71
25	55.47	76.63	303.38	339.99	26.833	396.85
26	50.42	74.2	303.68	334.04	26.9	400.47

Figure 13 shows the main effects plot for SN (Signal-to-Noise) ratios obtained by using Minitab®. Now we can figure out the optimal design values (considering both thermal deformation and natural frequency) corresponding to the maximum peak values in Fig. 13, which are listed in Table 9. In considering both thermal deformation and natural frequency, Table 10 shows the comparison of results using initial values and optimized values. Compared with the result using initial values, thermal deformation using the optimized values has been decreased by 2.43% while natural frequency using the optimized values has been increased by 12.37%. Therefore it can be concluded that the application of iSIGHT® 4.5 (OLH & RBF) to the optimal design of a dynamical system with high-speed spindle can decrease the thermal deformation in consideration of thermal behavior (using ANSYS Workbench®12.1) and natural frequency (using ARMD®), compared with the initial design of bearing positions based on the experience of an expert in machine tool design.

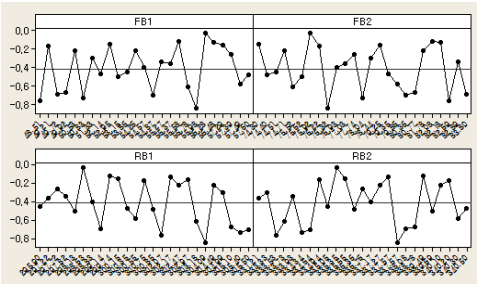


Fig. 13 Main effects plot for SN ratios

Table 9 Initial Values of Design Variables *vs*. Optimized Values of Design Variables
(considering both thermal deformation and natural frequency)

(unit : mm)

	Initial Values of Design Variables	Optimized Values of Design Variables
FB1	53	56.48
FB2	84	75.62
RB1	300.5	298.63
RB2	330.5	335.35

Table 10 Comparison of Results using Initial Values and Optimized Values
(considering both thermal deformation and natural frequency)

	Result using Initial Values	Result using Optimized Values	Rate
Deformation (㎛)	27.16	26.501	2.42% (decreased)
Natural Frequency (Hz)	520.18	593.6	12.37% (increased)

5 Conclusions

This paper aims at applying iSIGHT® 4.5 (OLH & RBF) to the optimal design of a dynamical system with high-speed spindle decreasing the thermal deformation in consideration of thermal behavior (using ANSYS Workbench®) and natural frequency (using ARMD®). Compared with the initial design of bearing positions based on the experience of an expert in machine tool design, thermal deformation using the optimized values has been decreased by 2.43% while natural frequency using the optimized values has been increased by 12.37%. In conclusion, the original work of this paper , *i.e.*, the application of iSIGHT® 4.5 (OLH & RBF) to the optimal design of a dynamical system with high-speed spindle in consideration of both thermal behavior and natural frequency can improve the thermal and dynamic characteristics.

6 Acknowledgement

This work was supported by grant No. RTI04-01-03 from the Regional Technology Innovation Program of the Ministry of Knowledge Economy (MKE)

The authors of this paper were partially supported by the Second Stage of Brain Korea21

7 References

[1] S. J. Park, D. H. Kim C. M. Lee, *A Study on the Determination of Bearing location for45,000rpm Spindle by Taguchi Method*, Journal of the Korean Society of Precision Engineering, spring Conference, 2009, pp. 479~480

[2] K. B. Park, W. J. Chung, C. M. Lee, Y. D. Cho, J. H. Kim, *Finite Element Modeling and Simulation for Shape Design of a High-speed Rotating-Shaft Using Design of Experiment*, MSV07, 2007, pp. 157~163

[3] K. H. Kwon, W. J. Chung, S. J. Lee, K. B. Park, J. H. Park, S. W. Kye, *Optimized Approximation of Finite Element Modeling for Complex Tool Holder Spindle using Optimal Latin Hypercube (OLH) method and Radial Basis Function (RBF) Neural Network*, CSC09, 2009, pp. 204~210

[4] ANSYS Workbench® 12.1 Manual, TSNE, 2008

[5] ARMD® Manual, Turbo Link, 2004

[6] S. H. Park, *Design of experiments*, 2005

[7] Report, *Study of the Clearance Control for High Speed Spindle Bearing and Optimization of Spindle Cooling System*, Korea Institute of Machine & Materials, 2004

[8] Choon Man Lee, Jeong Suk Lim, Won Jee Chung, 2009, *Selection of Bearing Position for Improving Static and Dynamic Stiffness of 40,000rpm High-speed Spindle*, KSMPE Vol.8 No.1, pp. 10~17

[9] iSIGHT® 4.5 Manual, Engineous Korea, 2010

[10] S. B. Lee, *application of Taguchi Method using Minitab®*, 2006

SESSION

THIRD WORKSHOP ON CELLULAR AUTOMATA, THEORY AND APPLICATIONS

Chair(s)

Lou D'Alotto
James F. Nystrom
William Spataro

Response Curves and Preimage Sequences of Two-Dimensional Cellular Automata

Henryk Fukś and Andrew Skelton
Department of Mathematics, Brock University, St. Catharines, Ontario, Canada.

Abstract— *We consider the problem of finding response curves for a class of binary two-dimensional cellular automata with L-shaped neighbourhood. We show that the dependence of the density of ones after an arbitrary number of iterations, on the initial density of ones, can be calculated for a fairly large number of rules by considering preimage sets. We provide several examples and a summary of all known results. We consider a special case of initial density equal to* 0.5 *for other rules and compute explicitly the density of ones after* n *iterations of the rule. This analysis includes surjective rules, which in the case of L-shaped neighbourhood are all found to be permutive. We conclude with the observation that all rules for which preimage curves can be computed explicitly are either finite or asymptotic emulators of identity or shift.*

Keywords: preimage, surjective, permutive, density, emulation

1. Introduction

Cellular automata (CA) can be viewed as computing devices, which take as an input some initial configuration. The CA rule is iterated a number of times, resulting in a final output configuration. In many practical problems, e.g., in mathematical modelling, one wants to know how a CA rule iterated over an initial configuration affects certain aggregate properties of the configuration, such as, for example, the density of ones. If we take a randomly generated initial configuration with a given density of ones, and iterate a given rule n times over this configuration, what is the density of ones in the resulting configuration? We want to know the "response curve", the density of the output as a function of the density of the input.

Response curves appear in computational problems, and a classical example of such a problem in CA theory is the so-called density classification problem (DCP). If we denote the density of ones in the configuration at time n by $P_n(1)$, the DCP asks us to find a rule for which $P_\infty(1) = 1$ if $P_0(1) > 1/2$ and $P_\infty(1) = 0$ if $P_0(1) < 1/2$. Since it is known that such a rule does not exist [1], once could ask which response curves are possible in CA rules? We propose to approach this problem from an opposite direction: given the CA rule, what can we say about its response curve? It turns out that in surprisingly many cases, the response curve can be calculated exactly, providing that preimage sets of finite strings under the CA rule exhibit recognizable patters.

2. Definitions

In what follows, we will be concerned with what we call two-dimensional elementary cellular automata, which have a local function depending on the central site, its right neighbour, and its top neighbour, and which allow two states only, 0 and 1. We will say that these are rules with "L-neighbourhood", since the neighbourhood has the shape of the letter L. Such three-input binary local rules can be considered the simplest "truly" two-dimensional CA rules, hence the name "elementary".

Before we define such rules formally, we will first introduce the concept of *triangular blocks*, defined as regions of 2D configurations in the shape of isosceles right triangles. The set of triangular blocks of size r, denoted $\mathcal{T}_r$, is the set consisting of elements

$$\begin{array}{l} b_{1,r} \\ \vdots \quad \ddots \\ b_{1,1} \ldots b_{r,1}, \end{array} \tag{1}$$

where each $b_{i,j} \in \mathcal{G}$. The set of eight blocks in T_2 will be called *basic blocks*.

We may define the *local mapping*, or *local rule*, of a 2D CA with L-neighbourhood as $g : \mathcal{T}_2 \to \mathcal{G}$. The local mapping g has a corresponding *global mapping*, $G : \mathcal{G}^{\mathbb{Z}^2} \to \mathcal{G}^{\mathbb{Z}^2}$ such that $(G(s))_{i,j} = g\left(\begin{smallmatrix} s_{i,j+1} & \\ s_{i,j} & s_{i+1,j} \end{smallmatrix}\right)$, for any $i, j \in \mathbb{Z}$, $s \in \mathcal{G}^{\mathbb{Z}^2}$.

The block evolution operator $\mathbf{g} : \mathcal{T}_r \to \mathcal{T}_{r-1}$ will be defined as a function which transforms triangular block (1) into another block, $c \in \mathcal{T}_{r-1}$, where $c_{i,j} = g\left(\begin{smallmatrix} b_{i,j+1} & \\ b_{i,j} & b_{i+1,j} \end{smallmatrix}\right) \in \mathcal{G}$ for $i \in \{1, \ldots, r-1\}$ and $j \in \{1, \ldots, r-i\}$. We denote $\mathbf{g}^n : \mathcal{T}_{r+n} \to \mathcal{T}_r$ to be the operator obtained by composing $\mathbf{g}$ with itself n-times.

Occasionally, we will need to define a distance between two configurations. One can show easily that for $s, t \in \mathcal{G}^{\mathbb{Z}^2}$ and $i, j \in \mathbb{Z}$, the following satisfies all axioms of a metric:

$$d(s,t) = \begin{cases} \dfrac{1}{1 + \min\limits_{i,j \in \mathbb{Z}} \left(\max\{|i|, |j|\} : s_{i,j} \neq t_{i,j}\right)} & \text{if } s \neq t \\ 0 & \text{if } s = t \end{cases}.$$

For 2D CA with L-neighbourhood, we adapt the numbering system used in [2]. A local rule g is assigned a *Wolfram number* W as follows

$$W(g) = \sum_{a_0, a_1, a_2 \in \{0,1\}} g\left(\begin{smallmatrix} a_0 & \\ a_1 & a_2 \end{smallmatrix}\right) 2^{4a_0 + 2a_1 + a_2}. \tag{2}$$

We note that, as in the case of radius-1 1D CA, there are 256 possible *elementary 2D CA*. Many of these rules are related to each other by the group of 4 transformations $D_1 \times S_2$, where D_1 is the dihedral group with a single reflectional symmetry and S_2 denotes all permutations of the elements in $\{0,1\}$. Among each class of four (not necessarily distinct) rules, we choose one representative with the smallest Wolfram number. We denote this rule to be a *minimal rule*. A list of all 88 minimal rules and their equivalences can be found in [3].

3. Densities of Blocks

We now consider the density response problem. Suppose that we start with an initial configurations in which a certain proportion of sites is in state 1. The simplest way to achieve this is to set each site to be in state 1 with probability ρ, and 0 with probability $1-\rho$, doing it independently for all sites. This means that the probability of randomly selected site to be in state 1 is ρ. Suppose that we apply n iterates of some CA rule to such configuration. What is the probability that in the resulting configuration, the state of a randomly selected site is 1?

In order to formulate this problem more precisely, we will use the concept of probability measure, similarly as done in [4], for one-dimensional CA.

Given a block $b \in \mathcal{T}_r$, we define a *cylinder set given by* b, $C_{i,j}(b)$, as the set of all configurations in which block b is fixed and placed at coordinate (i,j) aligned at the lower-left element of b. We define a *measure* of such as cylinder set, $\mu\left[C_{i,j}(b)\right]$, to be the probability of occurrence of block b placed as above. If the measure is translationally invariant we may drop the indices i,j. For $\rho \in [0,1]$, the Bernoulli measure is a measure where all sites are independently set to 1 with probability ρ, and to 0 with probability $1-\rho$. In such case, we have

$$\mu_\rho\left[C(b)\right] = \rho^j (1-\rho)^{(r^2+r)/2-j}, \tag{3}$$

where j is a number of cells in state 1 in b.

We now consider the action of the global mapping G on the measure of a cylinder set given by block b, which yields

$$\left(G\mu_\rho\right)\left[C(b)\right] = \mu_\rho\left[G^{-1}\left(C(b)\right)\right]. \tag{4}$$

Considering instead n iterations of G, we obtain

$$\left(G^n\mu_\rho\right)\left[C(b)\right] = \mu_\rho\left[G^{-n}\left(C(b)\right)\right]. \tag{5}$$

If we let $\mathbf{g}^{-n}(b)$ be the set of all *n-step preimages of block* b, that is, the set of all blocks a such that $\mathbf{g}^n(a) = b$, then we can write

$$\mu_\rho\left[G^{-n}\left(C(b)\right)\right] = \sum_{a\in\mathbf{g}^{-n}(b)} \mu_\rho\left[a\right]. \tag{6}$$

Using the notation $P_n(b) = \left(G^n\mu_\rho\right)\left[C(b)\right]$, we write (5) as

$$P_n(b) = \sum_{a\in\mathbf{g}^{-n}(b)} P_0(a). \tag{7}$$

If $b = 1$, and if the initial measure is Bernoulli, then in the above formula each $P_0(a)$ depends only on ρ, where $\rho = P_0(1)$. $P_n(1)$ can then be interpreted as the density of 1s in the configuration obtained by iterating the CA rule n times starting from disordered initial configurations with density of ones equal to ρ.

Plot of $P_n(1)$ versus ρ will be called a *response curve* for each elementary 2D CA. In the special case when $\rho = 1/2$, the probability of any block of a given size is equally likely and (7) can be expressed as

$$P_n(b) = 2^{-(r+n+1)(r+n)/2}\ \mathrm{card}\left[\mathbf{g}^{-n}(b)\right], \tag{8}$$

where $\mathrm{card}\left[\mathbf{g}^{-n}(b)\right]$ denotes the number of elements in the set $\mathbf{g}^{-n}(b)$. If we want to indicate that we consider the special case of $\rho = 1/2$, we will use the notation $P_n^{(s)}(b)$, at the sequence of $P_n^{(s)}(b)$ for $n = 0,1,2\ldots$ will be called a *response sequence*. Finally, we denote $P(b)$ to be the *asymptotic density of block* b, which we obtain by taking the limit of $P_n(b)$ as $n \to \infty$ (if the limit exists).

4. Theoretical Response Curves

For 26 minimal rules, we were able to determine an explicit response curve formula. In some cases, we found that the response curve was independent of n. In other cases, the response curve was dependent on n, and then a separate formula for the asymptotic density could be obtained. We present in detail three examples of each types. In each example, we describe the structure of the preimage sets but, due to space constraints, we omit direct proofs while noting that each case can be proved easily by induction.

4.1 Rules with Constant Density

In each of the following examples the formula for the response curve has no dependence on n. Therefore, the formula for the asymptotic density is the same as the response curve. We provide detailed analysis for Rules 0, 3 and 42 and the remaining results are presented in Table 1.

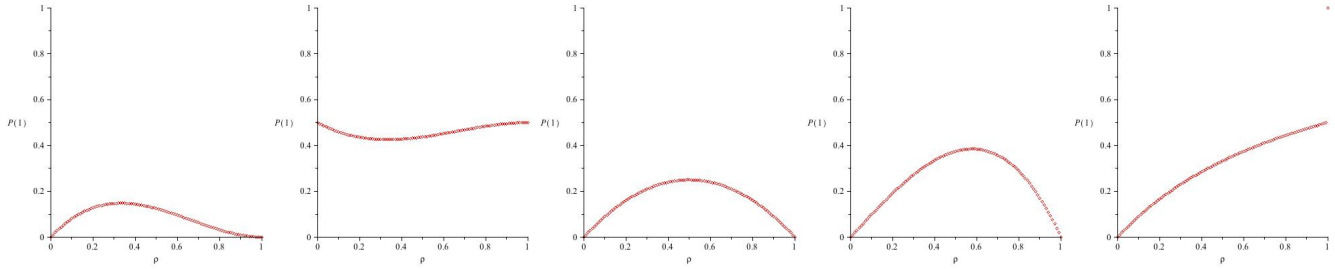
Proposition 1. *The response curve for Rule 0 is* $P_n(1) = 0$.

Proof: There are no triangular blocks of any size that can be mapped under $\mathbf{g}_0^n$ to single block 1. Therefore, $\mathrm{card}\left[\mathbf{g}_0^{-n}(b)\right] = 0$ and we apply (7) to obtain our result.

Proposition 2. *The response curve for Rule 42 is*

$$P_n(1) = \rho(1-\rho)(1+\rho).$$

Proof: It can be shown by induction that the only blocks that map to a single 1 under $\mathbf{g}_{42}^n$ are either blocks in $\mathcal{T}_n$ where $b_{n,1} = 1, b_{n-1,2} = 0$ and all other elements are arbitrary, or blocks in $\mathcal{T}_n$ where $b_{n,1} = b_{n-1,2} = 1, b_{n-1,1} = 0$ and all other elements are arbitrary. Using (3) and (7), we conclude that $P_n(1) = \rho(1-\rho)+\rho^2(1-\rho)$, which simplifies to the desired result. An experimental curve confirming this result is presented in Figure 1d.



(a) Rule 2 (b) Rule 3 (c) Rule 10 (d) Rule 42 (e) Rule 138

Fig. 1: Experimental Response Curves

Table 1: Constant Density Rules

Rules	$P_n(1)$	$P_n^{(s)}(1)$
0	0	0
2	$\rho(1-\rho)^2$	$1/8$
3	$(1-\rho)^2$ $\rho(1+\rho-\rho^2)$	$5/8$ (n odd) $1/4$ (n even)
4	$\rho(1-\rho)^2$	$1/8$
5	$(1-\rho)^2$ $\rho(1+\rho-\rho^2)$	$5/8$ (n odd) $1/4$ (n even)
10	$\rho(1-\rho)$	$1/4$
12	$\rho(1-\rho)$	$1/4$
51	$1-\rho$ (n odd) ρ (n even)	$1/2$
34	$\rho(1-\rho)$	$1/4$
42	$\rho(1-\rho)(1+\rho)$	$3/8$
51	$1-\rho$ (n odd) ρ (n even)	$1/2$
76	$\rho(1-\rho)(1+\rho)$	$3/8$
170	ρ	$1/2$
204	ρ	$1/2$

Proposition 3. *The response curve for Rule 3 is*

$$P_n(1) = \begin{cases} (1-\rho)^2 & \text{if } n \text{ even,} \\ \rho(1+\rho-\rho^2) & \text{if } n \text{ odd.} \end{cases}$$

Proof: Since Rule 3 has period-2 behaviour, we must consider cases when n is odd and when n is even. When n is odd, the only blocks that map to a single 1 under n-iterations of $\mathbf{g}_3$ are blocks in $\mathcal{T}_n$ where $b_{1,(n+1)/2} = b_{1,(n+3)/2} = 0$ and all other elements are arbitrary.

When n is odd, the only blocks that map to 1 under $\mathbf{g}_3^n$ are blocks in $\mathcal{T}_n$ where $b_{1,(n+2)/2} = 0$ and all other elements are arbitrary, or blocks in $\mathcal{T}_n$ where $b_{1,n/2} = b_{1,(n+4)/2} = 1, b_{1,(n+2)/2} = 0$ and all other elements are arbitrary.

Using (3) we can determine the initial probability of occurrence of blocks of either type and using (7) obtain a formula which simplifies to the desired result. Note that in the special case when $\rho = 1/2$, we conclude that $P_n^{(s)}(1)$ equals $1/4$ when n is even, and equals $5/8$ when n is odd. We averaged our experimental results over an even number of time steps, thus Figure 1b has the form

$$P_n(1) = \frac{1}{2}\left[(1-\rho)^2 + \rho(1+\rho-\rho^2)\right].$$

4.2 Rules with Decaying Density

In each of the following examples the formula for the response curve is dependent on n, and thus we can also determine an asymptotic density formula. We provide detailed analysis for Rules 32, 128 and 138, while the remaining results are presented in Table 2.

Proposition 4. *The response curve for Rule 128 is*

$$P_n(1) = \rho^{(n^2+3n+2)/2}.$$

Proof: The only block mapping to a single 1 under $\mathbf{g}_{128}^n$ is the block consisting entirely of ones. We use (3) to find the initial probability of this block and (7) produces our result. Thus, the asymptotic density under Rule 128 is

$$P(1) = \lim_{n\to\infty} P_n(1) = \begin{cases} 0 & \text{if } \rho \neq 1, \\ 1 & \text{if } \rho = 1. \end{cases}$$

Proposition 5. *The response curve for Rule 32 is*

$$P_n(1) = \rho^{n+1}(1-\rho)^n.$$

Proof: Under rule 32, the only blocks that map to a single 1 under $\mathbf{g}_{32}^n$ are of the form

$$n-1\left[\begin{array}{cccccc} & & & & & 1 \\ & & & & & 0 \\ & & & & \iddots & \star \\ & & & \iddots & \iddots & \vdots \\ & & \iddots & \iddots & & \vdots \\ \star & \cdots & \star & 0 & 1 & \star \end{array}\right].$$

Using (3), we can determine the initial probability of occurrence of blocks of this type and using (7), we obtain our result. The asymptotic density is thus $P(1) = 0$.

Proposition 6. *The response curve for Rule 138 is*

$$P_n(1) = \frac{\rho^{2n+2} + \rho}{\rho + 1}.$$

Proof: The only blocks that map to a single 1 under n-iterations of $\mathbf{g}_{138}$ are comprised entirely of arbitrary elements in the top $n-1$ rows, and have their lower two rows, where i ranges from 1 to $n+1$, of the form

$$\begin{array}{l} \star \cdots \star 0\, 1 \cdots 1\, 1 \\ \star \cdots \star \star \underbrace{1 \cdots 1\, 1\, 1}_{i} \end{array},$$

Table 2: Density Decaying Rules

Rules	$P_n(1)$	$P(1)$
8	$\rho^{n+1}(1-\rho)^n$	0
32	$\rho^{n+1}(1-\rho)^n$	0
40	$2^n\rho^{n+1}(1-\rho)^n$	0
72	$2^n\rho^{n+1}(1-\rho)^n$	0
128	$\rho^{(n^2+3n+2)/2}$	0 if $\rho \neq 1$ 1 if $\rho = 1$
130	see [5] for complete analysis	
132	can be derived from Rule 130	
136	ρ^{n+1}	0 if $\rho \neq 1$ 1 if $\rho = 1$
138	$\frac{\rho^{2n+2}+\rho}{\rho+1}$	$\frac{\rho}{1+\rho}$ if $\rho \neq 1$ 1 if $\rho = 1$
140	$\frac{\rho^{2n+2}+\rho}{\rho+1}$	$\frac{\rho}{1+\rho}$ if $\rho \neq 1$ 1 if $\rho = 1$
160	ρ^{n+1}	0 if $\rho \neq 1$ 1 if $\rho = 1$
162	$\frac{\rho^{2n+2}+\rho}{\rho+1}$	$\frac{\rho}{1+\rho}$ if $\rho \neq 1$ 1 if $\rho = 1$

Using (3) we can determine the initial probability of occurrence of blocks for each possible value of i. Summing over all i and using (7), we conclude that

$$P_n(1) = \rho^{2n+1} + \sum_{i=1}^{n} \rho^{2i-1}(1-\rho)$$
$$= \rho^{2n+1} + \frac{1-\rho}{\rho}\left(\frac{\rho^2\left(\rho^{2n}-1\right)}{\rho^2-1}\right),$$

which simplifies to our desired result.

Again, we can find the asymptotic density as

$$P(1) = \lim_{n\to\infty} P_n(1) = \begin{cases} \frac{\rho}{1+\rho} & \text{if } \rho \neq 1, \\ 1 & \text{if } \rho = 1. \end{cases}$$

This result is confirmed by the experimental curve in Figure 1e. Note that while the response curve is continuous, the asymptotic density has a discontinuity at $\rho = 1$, corresponding to an initial condition consisting entirely of ones.

5. Theoretical Response Sequences

In some cases we were unable to determine an explicit expression for the response curve of a given rule, but we were able to derive an explicit formula for card $[\mathbf{g}^{-n}(1)]$, and thus use (8) to obtain a response sequence. For 21 additional rules, we were able to either prove or conjecture a response sequence. We first consider the class of surjective rules.

5.1 Surjective Rules

Sites belonging to the L-shaped neighbourhood $\left(\begin{smallmatrix} a_{0,1} \\ a_{0,0}\ a_{1,0} \end{smallmatrix}\right)$ will be identified by their indices as $(0,1)$, $(0,0)$, and $(1,0)$. Similarly as in [6], a local function g will be called *permutive* with respect to the $(0,1)$ site if for any choice of $y, z \in \mathcal{G}$ the function $x \to g\left(\begin{smallmatrix} x \\ y\ z \end{smallmatrix}\right)$ is one-to-one. Permutivity with respect to the central site $(0,0)$ or the right neighbour $(0,1)$ is defined similarly. We now find a response sequence for rules permutive with respect to site $(0,0)$.

Proposition 7. *The response sequence for Rules 15, 30, 45, 51, 54, 57, 60, 90, 105, 106, 108, 150, 154, 156, 170 and 204 is* $P_n^{(s)}(1) = 1/2$.

Proof: There are 16 rules permutive with respect to the centre site, of which the following 9 are minimal: 51, 54, 57, 60, 105, 108, 150, 156 and 204. If a rule is permutive with respect to $(0,0)$, then there must exist numbers $x_0, \ldots, x_3 \in \{0,1\}$ such that the local function takes the form

$$g\left(\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right) = \begin{cases} 0 & \text{if } \left(\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right) \in \left\{ \begin{smallmatrix} \mathbf{0} \\ x_0\ \mathbf{0} \end{smallmatrix}, \begin{smallmatrix} \mathbf{0} \\ x_1\ \mathbf{1} \end{smallmatrix}, \begin{smallmatrix} \mathbf{1} \\ x_2\ \mathbf{0} \end{smallmatrix}, \begin{smallmatrix} \mathbf{1} \\ x_3\ \mathbf{1} \end{smallmatrix} \right\} \\ 1 & \text{if } \left(\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right) \in \left\{ \begin{smallmatrix} \mathbf{0} \\ \overline{x}_0\ \mathbf{0} \end{smallmatrix}, \begin{smallmatrix} \mathbf{0} \\ \overline{x}_1\ \mathbf{1} \end{smallmatrix}, \begin{smallmatrix} \mathbf{1} \\ \overline{x}_2\ \mathbf{0} \end{smallmatrix}, \begin{smallmatrix} \mathbf{1} \\ \overline{x}_3\ \mathbf{1} \end{smallmatrix} \right\} \end{cases}, \tag{9}$$

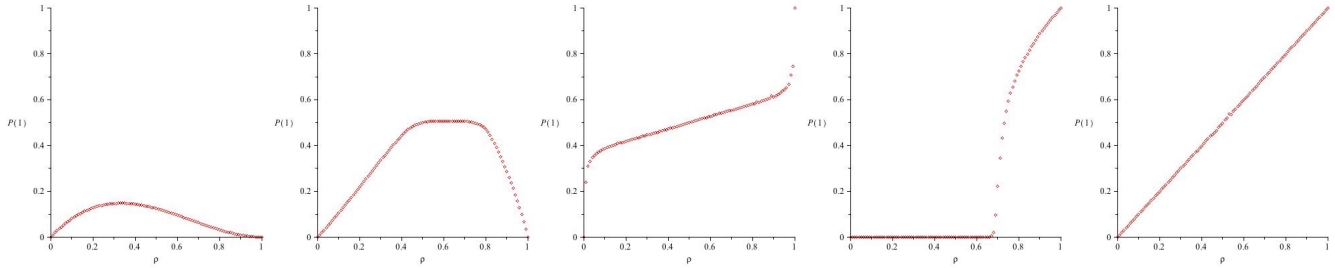
where $\overline{x}_i$ denotes $1 - x_i$. Assuming the above form of g, let us consider an arbitrary block $b \in \mathcal{T}^n$. We will now show how to construct all preimages of b under $\mathbf{g}$. First of all, we claim that blocks $c \in \mathcal{T}^{n+1}$ of the form

$$c = \begin{matrix} \alpha_1 & & & \\ c_{1,n} & \ddots & & \\ \vdots & \ddots & \ddots & \\ c_{1,1} & \ldots & c_{n,1} & \alpha_{n+1} \end{matrix}. \tag{10}$$

are the only preimages of b, where each α_i $(1 \leq i \leq n+1)$ is an arbitrary value in $\{0,1\}$, and values of $c_{i,j} \in \{0,1\}$ can be determined by an iterative algorithm.

To see that this is indeed true, we now present an algorithm with which we can construct all possible preimages:

1) Starting from $b_{1,n}$, we wish to find all neighbourhoods $\left\{\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right\}$ such that $g\left(\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right) = b_{1,n}$. The structure of the local mapping gives us four possible such neighbourhoods $\left\{\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right\} = \left\{\begin{smallmatrix} \alpha_0 \\ c_{1,n}\ \alpha_1 \end{smallmatrix}\right\}$, where
$$c_{1,n} = (1 - b_{1,n})x_{2\alpha_1+\alpha_2} + b_{1,n}(1 - x_{2\alpha_1+\alpha_2}),$$
and the values of α_1 and α_2 are arbitrarily selected. We now repeat step 2 for all values of $i \in \{2, \ldots, n\}$.
2) Since $b_{i,n-i+1}$ is given and α_i has been freely chosen in the previous iteration, we wish to know all neighbourhoods $\left\{\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right\}$, such that $g\left(\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right) = b_{i,n-i+1}$. The structure of the local mapping gives two possible neighbourhoods $\left\{\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right\} = \left\{\begin{smallmatrix} \alpha_i \\ c_{i,n-i+1}\ \alpha_{i+1} \end{smallmatrix}\right\}$, where
$$c_{i,n-i+1} = (1 - b_{i,n-i+1})x_{2\alpha_i+\alpha_{i+1}} + \\ + b_{i,n-i+1}(1 - x_{2\alpha_i+\alpha_{i+1}}),$$
and α_{i+1} is another arbitrarily selected value.
We now construct the rest of the preimage and show that all other values are uniquely determined based on each choice of the α values in the top diagonal. For all values of $j \in \{1, \ldots, n-i\}$ and then for all $i \in \{1, \ldots, n-j\}$, we repeat step 3 as follows.
3) Since $b_{i,n-i-j+1}$ is fixed, we wish to know all neighbourhoods $\left\{\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right\}$, such that $g\left(\begin{smallmatrix} a_0 \\ a_1\ a_2 \end{smallmatrix}\right) = b_{i,n-i-j+1}$. Since $c_{i,n-i-j+2}$ and $c_{i+1,n-i-j+1}$ were fixed in a previous iteration, the structure of the local



(a) Rule 2 (b) Rule 108 (c) Rule 154 (d) Rule 168 (e) Rule 172

Fig. 2: Experimental Response Curves

mapping tells us that our neighbourhood must have the form $\left\{ \begin{smallmatrix} a_0 \\ a_1 \ a_2 \end{smallmatrix} \right\} = \left\{ \begin{smallmatrix} c_{i,n-i-j+2} \\ c_{i,n-i-j+1} \ c_{i+1,n-i-j+1} \end{smallmatrix} \right\}$, where

$$c_{i,n-i-j+1} = (1-b_{i,n-i-j+1})x_{i'} + b_{i,n-i-j+1}(1-x_{i'}),$$

and $i' = 2c_{i,n-i-j+2} + c_{i+1,n-i-j+1}$. Note that no new arbitrary parameter appears here, thus the neighbourhood is determined uniquely.

The only arbitrary values in the preimage are the $(n+1)$ values of α_i on the main diagonal. Therefore, we know that there are exactly 2^{n+1} preimages for a given $b \in \mathcal{T}^n$. Therefore, we can see that card $[\mathbf{g}^{-n}(1)] = 2^{(n^2+3n+2)/2}$. Now, using (8), we conclude that $P_n^{(s)}(1) = 1/2$ for all n. Considering rules permutive with respect to the other two sites, we conclude that also Rules 15, 30, 45, 90, 106, 154 and 170 possess a response sequence $P_n^{(s)}(1) = 1/2$.

It turns out that these rules are the class of surjective 2D CA with L-neighbourhoods. In one dimension, it is known that rules permutive with respect to one of the variables located at the left or the right end of the neighbourhood are surjective, as proved in [6]. Recently, this result has been generalized to two dimensions by Dennunzio and Formenti [7], who demonstrated that any rule with Moore neighbourhood (of any radius) which is permutive with respect to one of the corner sites is surjective. We now show how one can prove a similar result specifically for the L-shaped neighbourhood, adapting the idea in [8] to 2D CA.

Proposition 8. *If the local mapping of an elementary 2D CA with L-neighbourhood is permutive with respect to any site, then the corresponding global mapping is surjective.*

Proof: From Proposition 7, we know that for any permutive rule and all $b \in \mathcal{T}^n, (n \geq 1)$, card$[\mathbf{g}^{-1}(b)] = 2^{n+1}$. Consider any infinite configuration, $t \in \mathcal{G}^{\mathbb{Z}^2}$. Define for all $n \geq 1$, the set, $S_n = \{s \in \mathcal{G}^{\mathbb{Z}^2} : \mathbf{g}(s_{[n+1]}) = t_{[n]}\}$, where $s_{[n+1]}$ denotes a block of size n contained in an infinite configuration $s \in \mathcal{G}^{\mathcal{Z}^2}$ and placed at $(0,0)$. Our assumption guarantees that all S_n are non-empty for $n \geq 1$. We also know that $S_{n+1} \subseteq S_n$. We consider the complement of S_n, the set $\overline{S_n} = \{s \in \mathcal{G}^{\mathbb{Z}^2} : \mathbf{g}(s_{[n+1]}) \neq t_{[n]}\}$, to show that S_n is a clopen set.

We first show that $\overline{S_n}$ is open. Let $s \in \overline{S_n} \subset \{0,1\}^{\mathbb{Z}^2}$ be an arbitrary configuration. For all $\epsilon > 0$, we choose $k \in \mathbb{Z}$, where $k > n$, such that $\frac{1}{k+1} < \epsilon$. We now pick an infinite configuration $s' \in \{0,1\}^{\mathbb{Z}^2}$ such that $d(s,s') = \frac{1}{k^*+1}$, where $k^* > k$. Since $s \in \overline{S_n}$, we know that $s' \in \overline{S_n}$, and

$$d(s,s') = \frac{1}{k^*+1} < \frac{1}{k+1} < \epsilon.$$

Thus, $\overline{S_n}$ is open. Similar analysis shows that S_n must also be open, and thus S_n is a clopen set. By the Nested Set Theorem [9], there must exist $s \in \mathcal{G}^{\mathbb{Z}^2}$, such that $F(s) = t$.

To conclude that these are the only surjective rules, we use the reverse direction of the Balance Theorem.

Proposition 9. *If a elementary 2D CA with L-neighbourhood is surjective, then for all $n \geq 1$ and all blocks $b \in \mathcal{T}^n$, card$[\mathbf{g}^{-1}(b)] = 2^{n+1}$.*

Proof: The Balance Theorem was proved in 1D in [6] and in 2D in [10]. A version of the proof specifically tailored for the L-neighbourhood is to be reported elsewhere [3].

For all other elementary rules, we performed a computerized search and found blocks for which no preimages exist. By Proposition 9, these rules must be non-surjective.

5.2 Conjectured Response Sequences

To find response sequences for the remaining rules, we performed an exhaustive search through all potential preimages for each rule. For the L-neighbourhood, the number of potential preimages is $2^{(n^2+3n+2)/2}$, which makes searches for large n impossible. We performed our searches using the Shared Hierarchical Academic Research Computing Network (SHARCNET) and we were able to obtain cardinalities of preimage sets to level $n = 7$. We then attempted to conjecture a formula for the sequence using the first six terms, and checked the conjecture with the seventh term.

Rules 23, 27, 29, 43, 46, 58, 77, 78, 142, 172, 178, 184 each shared the first seven terms of the preimage sequence with the surjective rules above, so that for these rules we conjecture that $P_n^{(s)}(1) = 1/2$. For all remaining rules, a list of the first seven preimage cardinalities is available upon request.

6. Experimental Response Curves

For those rules for which an explicit response curve formula could not be derived, we were able to perform computer simulations to obtain experimental response curves. We start with a square configuration of 250000 elements and we iterate $1000/\rho(1-\rho)$ times when $\rho \in (0,1)$ and 100000 times otherwise, with periodic boundary conditions and averaging density over the last 10 time steps and over 10 iterations from different initial conditions. Examples of such experimental curves are presented in Figures 1 and 2. We note in passing that one of the examples shown in Figure 2, namely Rule 168, exhibits response curve resembling "phase transition", that is, discontinuity of the derivative. None of 1D elementary rules exhibits such behaviour.

7. Rule emulation

We will now briefly turn our attention to dynamics of 2D rules. When one prints sample spatiotemporal diagrams of 2D rules with L-shaped neighbourhood (not shown here for the lack of space), one can easily observe that all rules for which density response curves can be calculated theoretically exhibit somewhat "simple" dynamics. A convenient was to describe this "simplicity" is to say that after a few iterations these rules essentially behave like identity or shift. In order to formalize this statement, we need to introduce the concept of emulation, first finite and then asymptotic.

7.1 Finite Rule Emulation

We say that *Rule X emulates Rule Y at level* n if,

$$\mathbf{g}_X^{n+1}(b) = g_Y\left(\mathbf{g}_X^n(b)\right). \tag{11}$$

for any block $b \in \mathcal{B}^{n+2}$. We will demonstrate this with an example. Consider Rule 76, with a local rule given by

$$g_{76}\left(\begin{smallmatrix} x \\ y\ z \end{smallmatrix}\right) = (1-x)y(1-z)+(1-x)yz+xy(1-z) = y(1-xz).$$

We now compose g_{76} with itself as follows

$$\begin{aligned}\mathbf{g}_{76}^2(b) &= g_{76}\left(\mathbf{g}_{76}\left(\begin{smallmatrix} x_0 \\ x_1\ x_2 \\ x_3\ x_4\ x_5 \end{smallmatrix}\right)\right)\\ &= x_3(1-x_1x_4)\left(1-x_1(1-x_0x_2)x_4(1-x_2x_5)\right)\\ &= x_3(1-x_1x_4) = g_{204}\left(\mathbf{g}_{76}(b)\right),\end{aligned}$$

where we have used the fact that when $x \in \{0,1\}$, we know that $x^2 = x$. We therefore conclude that Rule 76 emulates identity at level 1. We checked all 88×87 pairs of distinct elementary rules for finite rule emulation. In Figure 3, we show all level 1 emulation relations between all minimal elementary 2D rules with L-neighbourhood as directed graphs in which an arrow travels from X to Y if and only if Rule X emulates Rule Y at level 1. In Figure 3a are all rules which finitely emulate the identity Rule 204. In Figure 3b are all rules which finitely emulate the left shift Rule 170. Finally, in Figure 3c are another class of interrelated emulation rules. In addition to the rules in the graph, we also discovered that rules 6, 14, 18 and 50 emulate rules 134, 142, 146 and 178 respectively.

7.2 Asymptotic Rule Emulation

In [11], the author defined the following metric to describe the distance between two elementary 1D cellular automata rules. We adapt this and define the following metric to describe the distance between two elementary 2D cellular automata rules with L-neighbourhood

$$d(f,g) = 2^{(-k^2-3k-2)/2} \sum_{b \in \mathcal{B}_k} |f(b) - g(b)|. \tag{12}$$

We say that *Rule f asymptotically emulates Rule g* if

$$\lim_{n\to\infty} d\left(f^{n+1}, f^n \circ g\right) = 0. \tag{13}$$

We now derive a useful equation with which we can calculate the distance between two rules at a given level-n. First, we define the following function for any block $b \in \mathcal{B}$,

$$(f \oplus g)(b) = f(b) + g(b) \mod 2,$$

which outputs 1 if and only if $f(b) \neq g(b)$. Thus, we can use this function to count the number of blocks on which local mappings f and g differ. Adapting Proposition 3 from [11], we obtain the following proposition (proof in [11]).

Proposition 10. *If f, g are 2D local L-neighbourhood mappings, $A_0 = (f \oplus g)^{-1}(1)$, and $A_n = \mathbf{f}^{-n}(A_0)$, then*

$$d\left(f^{n+1}, f^n \circ g\right) = \frac{\operatorname{card}[A_n]}{2^{(n^2+5n+6)/2}}. \tag{14}$$

We demonstrate this procedure with an example.

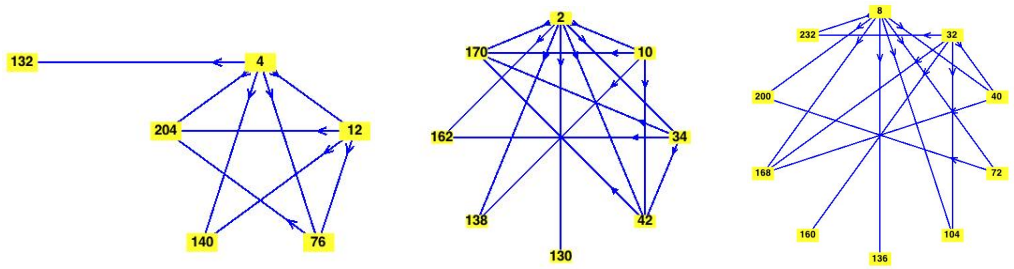
Proposition 11. *2D CA Rule 160 asymptotically emulates the identity rule.*

Proof: If we consider the local mappings for both Rules 160 and 204 we see that the set of blocks on which the rules output differ is $A_0 = \left\{ \begin{smallmatrix} 0 \\ 1\ 0 \end{smallmatrix}, \begin{smallmatrix} 0 \\ 1\ 1 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 0\ 1 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 1\ 0 \end{smallmatrix} \right\}$ To use Proposition 10, we must find the set A_n in general and thus we must know the sequence of preimages for these particular four basic blocks. We found the first five terms of these sequences and conjectured patterns are $B\left(2^{n+2}-3\right), B, B\left(2^{n+1}-1\right)$ and B, respectively, where $B = 2^{(n^2+n)/2}$. From equation (14), we determine that

$$d\left(\mathbf{g}_{160}^{n+1}, \mathbf{g}_{160}^n \circ \mathbf{g}_{204}\right) = 3 \cdot 2^{-n-2} - 4^{-n-1}.$$

Therefore, since the limit of this expression goes to 0, we conclude that Rule 160 emulates identity asymptotically.

Table 3 shows all known results of rules emulating shift or identity. We can now state our observation expressed at the beginning of this section using the concept of emulation: *all rules included in Tables 1 and 2 emulate identity or shift either in a final number of steps or asymptotically.*



(a) Rules Emulating Identity (b) Rules Emulating Shift (c) Other Finite Emulations

Fig. 3: Finite Emulation Relations

Table 3: Asymptotic Emulation

Rule f	$d(f^{n+1}, f_{204} \circ f^n)$	$P^{(s)}(1)$
8	$3 \cdot 2^{-2n-3}$	0
32	$5 \cdot 2^{-2n-3}$	0
40	2^{-n-1}	0
72	4^{-n-2}	0
128	$2^{(-n^2-3n-2)/2} - 2^{(-n^2-5n-6)/2}$	0
132	$2^{(-n^2-3n-4)/2}$	$\simeq 0.179$
136	2^{-n-2}	0
140	2^{-2n-3}	$1/3$
160	$3 \cdot 2^{-n-2} - 4^{-n-1}$	0
Rule f	$d(f^{n+1}, f_{170} \circ f^n)$	$P^{(s)}(1)$
130	$2^{(-n^2-3n-4)/2}$	$\simeq 0.179$
138	2^{-2n-3}	$1/3$
162	2^{-2n-3}	$1/3$

8. Further Results: Basic Blocks

We also note that if $\rho = 1/2$, it is often possible to compute the number of preimages of other blocks. For example, for 40 of the 88 minimal rules, we were able to find preimage sequences for all eight basic blocks, that is, blocks in $\mathcal{T}_2$. In each case, it is only necessary to determine preimage sequences for 5 of the 8 blocks, then we may use Kolmogorov consistency conditions [12] to determine the remaining three. In some cases, these formulas are rather striking, such as in the case of rule 130, reported in detail in [5], or rule 172, for which we make the following conjecture.

Conjecture 1. *Under 2D CA Rule 172 the preimage sequences of basic blocks are given by*

$$\operatorname{card}\left[\mathbf{g}^{-n}(b)\right] = \begin{cases} 2^{(n^2+5n)/2} \sum_{k=0}^{n} \frac{C_k}{4^k} & \text{if } b \in B_1, \\ 2^{(n^2+5n)/2} \left(2 - \sum_{k=0}^{n} \frac{C_k}{4^k}\right) & \text{if } b \in B_2, \end{cases}$$

where C_k denotes the k-th Catalan number and

$$B_1 = \left\{ \begin{smallmatrix} 0 \\ 0\,0 \end{smallmatrix}, \begin{smallmatrix} 0 \\ 1\,1 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 0\,0 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 1\,1 \end{smallmatrix} \right\}, \quad B_2 = \left\{ \begin{smallmatrix} 0 \\ 0\,1 \end{smallmatrix}, \begin{smallmatrix} 0 \\ 1\,0 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 0\,1 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 1\,0 \end{smallmatrix} \right\}.$$

Work on a proof of this result is ongoing and will be reported elsewhere.

9. Conclusions and future work

We demonstrated that response curves are calculable for simple rules that emulate shift or identity. Response curves clearly deserve further study and it is worthwhile to systematically study them for other CA rules. However, due to rapidly increasing preimage size, this won't be an easy task for larger neighbourhoods. One would need a more efficient way to construct the set of preimages of a given block, as simple brute force search becomes computationally too expensive. We also hope that rigorous results can be obtained for rules with somewhat more complicated dynamics.

Acknowledgements: One of the authors (HF) acknowledges financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) in the form of a Discovery Grant. We also wish to thank Shared Hierarchical Academic Research Computing Network (SHARCNET).

References

[1] M. Land and R. K. Belew, "No perfect two-state cellular automata for density classification exists," *Phys. Rev. Lett.*, vol. 74, pp. 5148–5150, 1995.

[2] S. Wolfram, *Cellular Automata and Complexity: Collected Papers*. Addison-Wesley, 1994.

[3] A. Skelton, "Response curves of deterministic and probabilistic cellular automata in one and two dimensions," Master's thesis, Brock University, St. Catharines, Canada, 2011.

[4] H. Fukś, "Probabilistic initial value problem for cellular automaton rule 172," *DMTCS proc.*, vol. AL, pp. 31–44, 2010.

[5] H. Fukś and A. Skelton, "Response curves for cellular automata in one and two dimensions - an example of rigorous calculations," *Journal of Natural Computing Research*, vol. 1, pp. 85–99, 2010.

[6] G. Hedlund, "Endomorphisms and automorphisms of shift dynamical systems," *Mathematical Systems Theory*, vol. 3, pp. 320–375, 1969.

[7] A. Dennunzio and E. Formenti, "Decidable properties of 2d cellular automata," *Lecture Notes in Computer Science*, vol. 5257, pp. 264–275, 2008.

[8] K. Sutner, *Linear Cellular Automata and de Bruijn Automata*. Kluwer, 1999, ch. 5, pp. 189–228.

[9] J. E. Marsden and M. J. Hoffman, *Elementary Classical Analysis*, 2nd ed. W.H.Freeman and Company, 1993.

[10] A. Maruoka and M. Kimura, "Condition for injectivity of global maps for tessellation automata," *Information and Control*, vol. 32, pp. 158–162, 1976.

[11] H. Fukś, "Sequences of preimages in elementrary cellular automata," *Complex Systems*, pp. 29–43, 2002.

[12] E. B. Dynkin, *Markov Processes-Theorems and Problems*. Plenum Press, 1969.

Decontamination with Temporal Immunity by Mobile Cellular Automata

Yassine Daadaa [1], Paola Flocchini [1], and Nejib Zaguia [1]

SITE, University of Ottawa.
Ottawa, ON K1N 6N5, Canada.
{ydaadaa,flocchin,zaguia}@site.uottawa.ca

***Abstract*—In this paper we consider the network decontamination problem in a mobile cellular automata (MCA). The system consists of a two-dimensional lattice that evolves like a cellular automata, where however some cells are in a special *active* state. Such a state indicates the presence of an *agent* which also follows a local transition rule to move from cell to neighbouring cell. A dynamic contamination process causes the spread of a virus (or a fault), and the presence of an agent on a cell guarantees local disinfection (or decontamination). Once disinfected, a cell stays immune to recontamination for a predetermined amount of time. The goal is to design the local rules for the agents and their initial placement so that the agents can decontaminate the entire system without allowing any cell to be re-contaminated. To be efficient, the decontamination should employ as few agents as possible. We design several strategies depending on the type of neighborhood, and on the ability of the agents to clone themselves.**

***Index Terms*—Cellular Automata, Network Decontamination, Mobile Agents, Mobile Cellular Automata.**

I. Introduction

Faults and viruses often spread in networked environments by propagating from site to neighboring site. The process due to this spread is called *network contamination*. The propagation patterns can follow different dynamics, depending on the behavior of the affected site. At one extreme we have a full spread behavior: when a site is affected by a virus or any other malfunction, such a malfunction can propagate to all its neighbors Other times, faults propagate only to sites that are susceptible to be affected; the definition of susceptibility depends on the application but oftentimes it is based on local conditions, for example, a node could be vulnerable to contamination if the majority of its neighbours is faulty, immune otherwise (e.g., see [11], [12], [15]); or it could be immune to contamination for a certain amount of time after being repaired (e.g., see [5], [8]).

Given a network where there is possibly a contamination process, it is crucial to have a mechanism in place to decontaminate the affected sites and to stop the spread.

Among the various possible techniques, two types of decontamination problems have been identified in the literature (for a survey see [6]): internal and external decontamination.

In *internal decontamination* a site can decontaminate itself (i.e., it can activate an antiviral software) when a certain condition on the neighborhood is verified; a clean site, however gets re-contaminated when some other condition on the neighborhing states is verified. This approach has been followed, for example, in [16], where a node becomes clean when the majority of its neighbours is clean, and a clean node becomes contaminated if any of its neighbours is. A similar approach has been taken in [14], [15], where a decontaminated node is immune to recontamination if at least the majority of its neighbours is clean. Internal decontamination essentially describes the dynamics of cellular automata, and it has been studied especially in the context of fault-tolerance to describe mechanisms of spread of faults and of auto-correction (e.g., see [17] for a survey). In [5] specific local rules have been designed for the internal cleaning to take place, to force patterns that minimize the number of simultaneous disinfecting sites. In all these studies the main objective is typically to determine the minimum size of a set of faulty nodes which completely disrupts the system under given contamination/decontamination dynamics or, equivalently, the minimum size of a set of decontaminating nodes that can decontaminate the whole network under the same circumstances.

On the other hand, *external decontamination* is performed by *mobile agents* moving on the network. There is an extensive literature on external decontamination either in specific topologies (e.g., see [1], [7]), under various assumption on the capabilities of the agents, or in arbitrary topologies (e.g., see [3]). Typically agents have memory, distinct identifiers, can communicate with other agents when they meet or can exchange information writing on whiteboards (storage area located at the nodes). In all models investigated, agents can move from node to node (usually asynchronously and independently) decontaminating the sites the pass through and a clean site becomes contaminated if at least one of its neighbors is contaminated. External decontamination has been studied especially in the context of intruder capture to design algorithms to neutralize a virus in a network, or in graph search (e.g., [1], [2], [3], [7], [9]). The main goal of decontamination in these settings is usually to design a strategy that employs the minimum possible size of the team of cleaning agents.

In this paper we are interested in *Mobile Cellular Automata* (MCA), a model that has some components of both internal and external models for decontamination and presents some advantages over both. The environment is a lattice that evolves in discrete time steps where, like in cellular automata, cells change their state according to the state of their neighbours. Some cells are in a special *active* state which indicates the presence of an agent. An agent can move from node to neighbouring node (like in the external decontamination)

but it does so on the basis of the state of its neighbours. Mobile Cellular Automata is the term used in [19] to indicate such a model, but with a single active cell. We assume that cells are possibly contaminated and we want to devise local transition rules for the MCA so to have all cells simultaneously decontaminated. We assume that decontaminated cells have an *immunity time* T during which they are immune to recontamination, regardless of the state of their neighbours. Once this time is expired, however, they become contaminated again as soon as at least one neighbour is contaminated. While the general decontamination problem has been investigated quite extensively, very few results exist assuming some type of immunity for the decontaminated cells: majority immunity is considered in [14], [15] for internal decontamination, time immunity has been studied in trees [8] in a more powerful mobile agents model, and in [5] for internal decontamination. No results exist so far in the model considered in this paper. In this model, we want to devise local rules to obtain complete simultaneous decontamination and we want to do so by never allowing a cell to be re-contaminated, and by employing the minimum possible number of active cells at each time. As mentioned, such a measure is the classical parameter studied in the decontamination literature.

Due to lack of space some proofs are sketched and some omitted.

II. Model

In Mobile Cellular Automata (MCA), a team of active mobile entities (here called *agents*) move on a two-dimensional $n \times n$ cellular space. Like in the case of Cellular Automata, a cell of the space is in a state belonging to a finite set and a cell changes its state according to the states of its neighbours. Unlike Cellular Automata, however, in MCA there is a special *active* state which corresponds to the presence of an agent; an active cell can change, not only its own state, but also the ones of its neighbours. In the following we will be considering finite cellular automata (with backgrounds of clean cells) with both *von Neumann* and *Moore* neighborhoods at distance one.

This model is very similar to the one of ants and turmites (e.g., see [4], [10], [13]) where active elements also move from node to node following local rules. The models differ in the action taken when two active elements collide, which is usually probabilistic in the case of ants/turmites, while it is always deterministic in this setting. A general formalism that encompasses all these models describing general multi-agents systems as discrete dynamical systems can be found in [4] and [18].

Let $x^t_{i,j}$ denote the state of cell (i,j) at time t. State $x^t_{i,j} = 0$ correspond to a *decontaminated* cell, state $x^t_{i,j} = 1$ to a *contaminated* cell and $x^t_{i,j} = \bullet$ to an *active* cell containing an agent.

The system is updated synchronously at discrete time steps. A *decontaminated* cell stays so for T time units (the *immunity time* of the system). After that time, it will become *contaminated* if at least one of its neighbours is. Let s be the immunity time of a cell i in *decontaminated* state at time t: we shall indicate such a time in parenthesis as follows: $x^t_{i,j} = 0(s)$.

An *active* cell at time t can change its own state at time $t+1$ as well as the state of one or more of its neighbours. More precisely, an active cell applies a local transition rule that returns a new state for the cell and one or more movement directions indicating to which neighbours the agent is "moving". If the direction is unique the action triggered by the rule corresponds to a simple movement; if more directions are indicated, we say that the agent is *cloning* (or duplicating itself) to the corresponding neighbours. We will denote the transition rule by f: the rule takes as input a set of states and returns a pair containing its own state and a set of directions. For example the rule applied to cell $(i.j)$ is indicated as follows: $f(N^t(i,j)) = (x^{t+1}_{i,j}, mov)$ where $N^t(i,j)$ indicates the states of the neighbouring cells of (i,j) and that of its own at time t ($N^t(i,j) = x^t_{i,j}, x^t_{i-1,j}, x^t_{i,j+1}, x^t_{i+1,j}, x^t_{i.j-1}$, in the case of Von Neumann neighbourhood), and mov is a set of directions from $\{\Uparrow, \Downarrow, \Leftarrow, \Rightarrow\}$. Let $f_1()$ denote the first component $x_{i,j}$ of the output of the transition rule f, and $f_2()$ the second component mov. In the rest of the paper the transition rule for an active cell is indicated in a table where, for a given configuration we show the next state of the cell and the direction of the agent's movement. A configuration always indicate the state of the cell itself, and its neighborhood from the left neighbour in clock-wise order (e.g., in the case of Von Neumann neighborhood: $x^t_{i,j}, x^t_{i-1,j}, x^t_{i,j+1}, x^t_{i+1,j}, x^t_{i.j-1}$).

The behavior of the system from time t to time $t+1$ can be then described by the following:

1) For all cells (i,j), regardless of their state $x^t_{i,j}$.
   $x^{t+1}_{i,j}$ becomes *active* if $((f_2(N^t(i-1,j)) = \Rightarrow) \vee (f_2(N^t(i,j+1)) = \Downarrow) \vee (f_2(N^t(i+1,j)) = \Leftarrow) \vee (f_2(N^t(i,j+1)) = \Uparrow))$.
2) If 1) does not apply and $x^t_{i,j}$ is *active*.
   $x^{t+1}_{i,j} = f_1(N^t(i,j))$
3) If 1) does not apply and $x^t_{i,j} = 0(s)$.
   If $s > 0$ then: $x^{t+1}_{i,j} = 0(s-1)$.
   If $s = 0$ and $((x^t_{i-1,j} = 1) \vee (x^t_{i+1,j} = 1) \vee (x^t_{i,j-1} = 1) \vee (x^t_{i,+1j} = 1))$ then: $x^{t+1}_{i,j} = 1$.

An important observation has to be made regarding the behavior of the system when two or more agents move into the same cell. The rules described above transform into *active* a cell that receives at least one agents, which means that if more than one agents move into the same cell they are fused into one. The number of agents in the system might then decrease in time. As already mentioned, we will consider two situations: 1) agents that can clone (or duplicate themselves), in which case a cell in state $\bullet$ could propagate in several direction, and 2) agents that cannot clone, in which it must move (like a physical agent) in a single direction.

Given a MCA of size $n \times n$ and an immunity time T, our goal is to choose the initial location of the *active* cells and the local transition rule f that achieve global decontamination in such a way that the number of active cells at a given time is as low as possible.

III. Temporal Decontamination without Cloning

In this Section we consider the case when the transition function returns a new state for the cell and a single direction for the agent. Our solutions are designed in such a way that two agents never move into the same cell. In doing so we insure a constant number of agents is present in the system.

A. Basic Decontamination

We first briefly look at the case when the decontaminated cells have no immunity (*basic decontamination*). Notice that with basic decontamination, a single agent would obviously not be sufficient because immediately after decontaminating a cell it would inevitably be exposed to a contaminated cell as at most one of its neighbours can become *active*. The question is what is the minimum number of agents which could guarantee decontamination without recontamination. Similarly to basic decontamination in cellular automata [5], we can prove that n agents are necessary and sufficient. The strategy is simple and consists of placing the n active cells in the first column and let the active cells propagate from one column to the next following the rules of Table I for Von Neumann neighborhood.

Theorem 3.1: Optimal basic decontamination can be achieved in a Mobile Cellular Automata with Von Neumann neighbourhood using n agents.

TABLE I
Rules for basic decontamination in a MCA with Von Neumann neighbourhood

Configuration	Next State	Agents Movement
$\{\bullet, 0, 0, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, \bullet, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, \bullet, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, \bullet\}$	0	*Terminate*
$\{\bullet, 0, \bullet, 0, \bullet\}$	0	*Terminate*
$\{\bullet, 0, \bullet, 0, 0\}$	0	*Terminate*

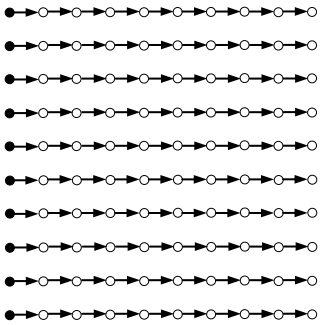


Fig. 1. Basic decontamination in a MCA with Von Neumann neighbourhood

B. Temporal Decontamination with Von Neumann Neighbourhood

We now turn to the general case of a given system immunity time $T > 1$, and we describe two different initial placements of the agents and two sets of rules that achieve decontamination. Depending on the relationship between n and the immunity time T of the system one could choose the best of the two strategies, which are however not proven to be optimal.

Notice that our strategies with Von Neumann neighborhood do require the initially active cells to be placed equidistant on the first column. With such a constrain, the number of agents that we have to employ for a given immunity time T clearly depends on the relationship between n and T

Let t_1 be the largest integer smaller than or equal to T such that $(t_1 + 1)$ divides n. Depending on whether $(t_1 + 1)$ is odd or even we could use Strategy 1 or Strategy 2.

Strategy 1: siblings at odd distance. This strategy applies when t_1 is odd (i.e., $(t_1 + 1)$ is even).

Initial placement. We place one agent at the top-left corner, one at the bottom-left corner and the other agents in groups of two at distance t_1 from each other. In other words, we divide n in equal groups of $t_1 + 1$ consecutive cells and employ 2 agents in each group, one on top, and one on the bottom (let us call such a pair *sibling agents*). In this way the two siblings are separated by an even number $(t_1 - 1)$ of cells (see Figure 2).

Fig. 2. Initial Configuration: sibling at odd distance

Fig. 3. Initial Configuration: sibling at even distance

TABLE II
MCA with Von Neumann neighbourhood: odd distance between siblings.

Configuration	Next State	Agents Movement
$\{\bullet, 0, 0, 1, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 1, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, \bullet, 1, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 1, 0\}$	0	$\Uparrow$
$\{\bullet, 0, \bullet, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 0, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, \bullet, 0, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 0, 0\}$	0	$\Uparrow$
$\{\bullet, 0, \bullet, 0, 0\}$	0	*Terminate*
$\{\bullet, 0, 0, 0, \bullet\}$	0	*Terminate*
$\{\bullet, 0, 0, 0, 0\}$	0	*Terminate*

Pattern of Movement. The set of rules is given in Table II where all missing combinations of states for the neighbourhood of cell (i, j) do leave $x_{i,j}$ unchanged. These rules

corresponds to the movement of each group of siblings in opposite directions on the column, turning right as soon as they become adjacent. Note that adjacent agents on a column can recognize whether they have to turn right (because they have their other neighbour on the same column *decontaminated*) or whether they have to move on the column in opposite direction (because they have their other neighbour on the same column *contaminated*). Also note that this technique is possible only because the two siblings do become adjacent but never move on the same cell, hence the need for placing them at odd distance (i.e., of separating them with an even number of cells).

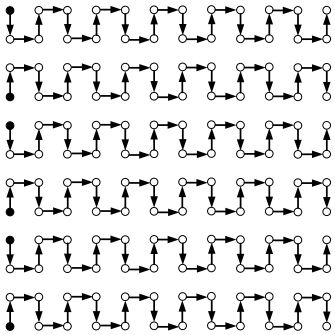


Fig. 4. Pattern of Movement with Strategy 1.

Theorem 3.2: Let t_1 be the largest odd integer smaller than or equal to T such that t_1+1 divides n. Temporal decontamination can be achieved in a Mobile Cellular Automata with Von Neumann neighbourhood and immunity time T using $\frac{2n}{(t_1+1)}$ agents

Proof: We want to show that, following the rules of Table II, decontamination is achieved monotonically. That is, that once *decontaminated*, every cell stays *decontaminated* until the end of the decontamination process, when all cells are *decontaminated*. We now prove by induction on the number of columns that, for each column i there is a time t when: (i) all cells in column i are either *decontaminated* or *active*; (ii) all cells in column $i-1$ (for $i>0$) are *decontaminated*; (iii) by time $t+t_1$ all right neighbours of a *decontaminated* cell of column i are either *decontaminated* or *active*; (iv) the distance between any two siblings in column $i+1$ (for $i<n-1$) is smaller than t_1+1.

1. Base - column 0: According to the set of rules given in Table II, agents move vertically in both directions on the first column (by rules $f(\bullet,0,0,1,1)=(0,\Downarrow)$, $f(\bullet,0,\bullet,1,1)=(0,\Downarrow)$, $f(\bullet,0,1,1,\bullet)=(0,\Uparrow)$ and $f(\bullet,0,1,1,0)=(0,\Uparrow)$), and since the maximal distance between initially consecutive agents is t_1 by construction, within $\lfloor\frac{t_1-1}{2}\rfloor$ time units all the cells on column 0 are either *active* or *decontaminated*. According to the given rules, agents then moves to column 1 (by rules $f(\bullet,0,0,1,\bullet)=(0,\Rightarrow)$ and $f(\bullet,0,\bullet,1,0)=(0,\Rightarrow)$). Since the movement to column 1 happened for all agents within $\lfloor\frac{t_1-1}{2}\rfloor$ time units from the beginning, the distance between any two siblings in column 1 cannot be greater than T. By a similar argument as for column 0, all cells in column 1 will then become *decontaminated* or *active* within other $\lfloor\frac{t_1-1}{2}\rfloor$ time units. Thus, within at most t_1 time units, all the cells in column 0 and in column 1will be either *decontaminated* or *active*.

2. Induction hypothesis: At some point during the decontamination, assume all cells of column i $(0<i<n-1)$ and all their right neighbours in column $i+1$ are either in *active* or *decontaminated* state, their left neighbours in column $i-1$ are *decontaminated*, and siblings in column $i+1$ are at distance at most t_1.

3. Induction Step: Consider column $i+1$: By induction hypothesis we know that there is a time t when all cells in column $i-1$, are *decontaminated*, the ones in columns i, and $i+1$ are either *active* or *decontaminated* and the siblings in column $i+1$ are at maximum distance T from each other. It follows that agents move vertically in column $i+1$ within $\lfloor\frac{t_1-1}{2}\rfloor$ time units from time t, thus leaving all cells in column i *decontaminated*. Agents then move to column $i+2$ and within other $\lfloor\frac{t_1-1}{2}\rfloor$ time units all the cells in the column $i+2$ become either *decontaminated* or *active*. We can conclude that, by time $t+t_1$ all cells in column $i+1$ together with all their right neighbours are either *decontaminated* or *active* and the cells in column i are *decontaminated*, thus concluding the proof. ■

Strategy 2: siblings at even distance. This strategy is applied when t_1 is even (i.e., (t_1+1) is odd).

Initial placement. The idea is to place two siblings at even distance t_1 (i.e., separated by t_1-1 cells) and employ a third agent in each interval between siblings to be placed in the central cell. The third agent (called the *delimiter*) has the only role of keeping a separation between the two intervals.

Pattern of Movement. The siblings move like before, but they turn to the right to move to the next column when they become in contact with the delimiter agent. The delimiter, on the other hand, moves to the next column when it sees its two neighbours on the same column occupied by one agent each. The set of rules corresponding to this case is given in Table III. All missing combinations of states for the neighbourhood of cell (i,j) do leave $x_{i,j}$ unchanged.

Note that the delimiter agent moves only to synchronize the two adjacent cleaners and the transition function that corresponds to waiting (not indicated in the Table because it does not change its state) is the following: $f(\bullet,0,1,1,1)=(\bullet,\{\})$.

Following a reasoning similar to the one of the proof of Theorem 3.2, we have:

Theorem 3.3: Let t_1 be the largest even integer smaller than or equal to T such that (t_1+1) divides n. Temporal decontamination can be achieved in a Mobile Cellular Automata with Von Neumann neighbourhood and immunity time T using $\frac{3n}{(t_1+1)}$ agents.

Each of the two proposed set of rules can be employed under specific circumstances. In order to take advantage of the largest possible immunity while minimizing the number of agents, we can choose the best of the two sets depending

TABLE III
MCA WITH VON NEUMANN NEIGHBOURHOOD: EVEN DISTANCE BETWEEN SIBLINGS

Configuration	Next State	Agents Movement
$\{\bullet, 0, 0, 1, 1\}$	0	$\Downarrow$
$\{\bullet, 0, \bullet, 1, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 1, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, 1, 1, 0\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, \bullet, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, \bullet, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, \bullet, 0, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 0, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 0, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 0, 0\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 0, \bullet\}$	0	*Terminate*
$\{\bullet, 0, \bullet, 0, \bullet\}$	0	*Terminate*
$\{\bullet, 0, \bullet, 0, 0\}$	0	*Terminate*
$\{\bullet, 0, 0, 0, 0\}$	0	*Terminate*

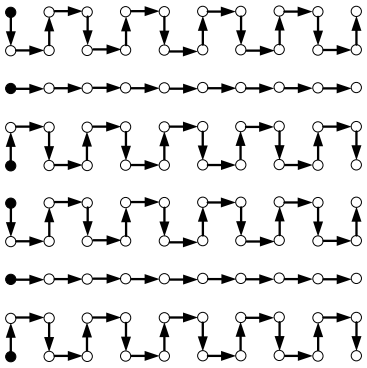


Fig. 5. Pattern of Movement with Strategy 2.

on the size n and the time immunity T of the system. In fact, let t_1 be the largest odd integer smaller than or equal to T such that $(t_1 + 1)$ divides n, and let t_2 be the largest even integer smaller than or equal to T such that $(t_2 + 1)$ divides n. By using the best of the two sets of rules, the system can be decontaminated by $k = \min\{\frac{2n}{t_1+1}, \frac{3n}{t_2+1}\}$ agents.

Theorem 3.4: Temporal decontamination can be achieved in a Mobile Cellular Automata with Von Neumann neighbourhood and immunity time T using $k = \min\{\frac{2n}{t_1+1}, \frac{3n}{t_2+1}\}$ agents, where $t_1 = \max\{\ odd\ t < T : (t_1 + 1)|n$, and $t_2 = \max\{\ even\ t < T : (t_2 + 1)|n\}$.

C. Temporal Decontamination with Moore Neighbourhood

In the case of Moore neighbourhood, we can exploit the enlarged visibility of the agents to design a strategy that decontaminates the network starting from a less restricted initial placement of the agents.

Initial Placement. There are two types of agents: the *cleaners* and the *delimiters*. Two cleaners are placed in the top-most and bottom-most cells of the first column. The other agents are placed arbitrarily in the first column alternating a cleaner and a delimiter in such a way that the distance between a cleaner and a delimiter is smaller than or equal to $\lfloor\frac{(T+1)}{2}\rfloor$.

TABLE IV
MCA WITH MOORE NEIGHBOURHOOD.

Configuration	Next State	Agents Movement
$\{\bullet, 0, 0, 0, 0, 1, 1, 1, 0\}$	0	$\Downarrow$
$\{\bullet, 0, 0, \bullet, 1, 1, 1, 1, 0\}$	0	$\Downarrow$
$\{\bullet, 0, 0, 0, 1, 1, 1, 1, 0\}$	0	$\Downarrow$
$\{\bullet, 0, \bullet, 1, 1, 1, 1, 1, 0\}$	0	$\Downarrow$
$\{\bullet, 0, \bullet, 1, 0, 0, 0, 1, 0\}$	0	$\Downarrow$
$\{\bullet, 0, 0, 0, 0, 0, 0, 1, 0\}$	0	$\Downarrow$
$\{\bullet, 0, 0, 0, 1, 1, 1, \bullet, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, \bullet, 1, 0, 0, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, \bullet, 1, 1, 1, 1, 0, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, 0, 1, 1, 0, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, 0, 1, \bullet, 0, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, 1, 1, 1, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 1, 1, 1, 1, 1, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 1, 1, 1, 1, 0, 0\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 1, 0, 0, 0, 1, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 1, 0, 0, 0, 0, 0\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 0, 0, 0, 0, 1, \bullet\}$	0	*Terminate*
$\{\bullet, 0, \bullet, 1, 0, 0, 0, 0, 0\}$	0	*Terminate*
$\{\bullet, 0, 0, 0, 0, 0, 0, 0, 0\}$	0	*Terminate*

Pattern of Movement. The rules are defined in Table **??**. As already mentioned, all missing combinations of states for the neighbourhood of cell (i, j) do leave $x_{i,j}$ unchanged.

The dynamics of movement is quite different from the case of Von Neumann neighbourhood and the pattern of movement is shown in Figure 6. A cleaner move down towards its delimiter; when the two agents are adjacent, cleaners move to the next column (right) while the delimiter moves to the next column if and only if it sees all its other neighbour except the right neighbour *decontaminated*. Obviously the agents are unaware of their role, but they understand it from the states of the neighbouring cells. In this way, the delimiter enforces synchronicity between agents while moving from column to column thus allowing the intervals to be of different size. With Moore neighbourhood, agents have enlarged visibility, which is first crucial for the cleaner when entering a new column to decide whether to go up or down (left visible neighbours help choose the good direction), and then allows the delimiter to decide when to move to the next column (when all neighbour are *decontaminated*). This type of movement would not be possible with Von Neumann neighbourhood, because the cleaner could not see diagonal neighbours *decontaminated* by another cleaner; moreover, the delimiter could not distinguish the cases when it should move to next column or wait (such an ambiguity derives again from the lack of diagonal visibility).

Assume, for simplicity, that $\frac{T}{2} + 1$ divides n. In this case we can organize the agents so that the distance between the delimiter and the cleaner is precisely $\lfloor\frac{(T+1)}{2}\rfloor$ and obtain the following.

Theorem 3.5: Temporal decontamination can be achieved in a Mobile Cellular Automata with Moore neighbourhood and immunity time T using $\lfloor\frac{4n}{T+2}\rfloor + 1$ agents.

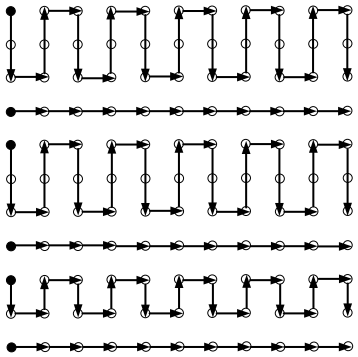


Fig. 6. Temporal decontamination in a MCA with Moore neighborhood without cloning

IV. Temporal Decontamination with Cloning

In this Section we consider the more general case when the local transition function can return several directions. It is easy to see that in the case of basic decontamination the result is the same as in the case of basic decontamination without cloning: the agents can be placed on the first column and let move to the next to clean sequentially all the columns. We now consider the case of temporal decontamination with $T > 1$ and Von Neumann neighborhood. With Moore neighborood we cannot improve the result of Section III-B.

Initial Placement. We divide the n cells of the first column in groups of *at most* $T + 1$ consecutive cells and employ 2 agents (the siblings) in each group, one on top, and one on the bottom. The two siblings can be separated by an arbitrary distance smaller than T.

TABLE V
MCA with cloning: Von Neumann neighbourhood.

Configuration	Next State	Agents Movement
$\{\bullet, 0, 0, 1, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 1, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, \bullet, 1, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 1, 0\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 1, 1, 1\}$	0	$\Downarrow\Uparrow$
$\{\bullet, 0, \bullet, 1, 0\}$	0	$\Rightarrow$
$\{\bullet, 0, 1, 0, 0\}$	0	$\Uparrow$
$\{\bullet, 0, \bullet, 0, 1\}$	0	$\Downarrow$
$\{\bullet, 0, 1, 0, \bullet\}$	0	$\Uparrow$
$\{\bullet, 0, 0, 1, \bullet\}$	0	$\Rightarrow$
$\{\bullet, 0, 0, 0, 1\}$	0	$\Downarrow$
$\{\bullet, 0, \bullet, 0, 0\}$	0	*Terminate*
$\{\bullet, 0, 0, 0, \bullet\}$	0	*Terminate*
$\{\bullet, 0, 0, 0, 0\}$	0	*Terminate*

Pattern of Movement. The set of rules is given in Table V and it corresponds to the movement of each group of siblings in opposite directions on the column, agents turn right once they become adjacent (e.g., see cells b and c in Figure 7) or when they meet in the same cell (e.g., see cell a in Figure 7), and in this case they will fuse and act as a single agents. If they fuse, when the agent is in the next column and realizes that its *up* and *down* neighbours are contaminated, it will duplicate and propagate in both directions.

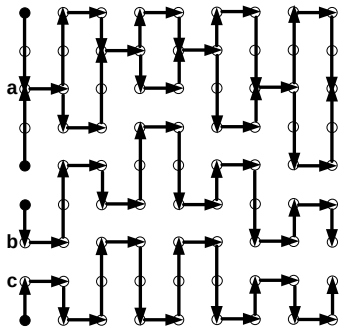


Fig. 7. Temporal decontamination with cloning in a MCA with VonNeumann neighborhood.

Theorem 4.1: Temporal decontamination can be achieved in a Mobile Cellular Automata with Von Neumann neighbourhood and immunity time T using $\lceil \frac{2n}{(T+1)} \rceil$ agents that can clone.

Proof: We now show that, by following the rules of Table V, decontamination is achieved monotonically. That is, that once *decontaminated*, every cell stays clean until the end of the process, when all cells are clean. Let us call *entry points* of decontamination in a column, the cells in such a column that becomes *active* due to a left *active* neighbour (horizontal propagation). We now prove by induction on the number of columns that, for each column i there is a time t when: (i) all cells in column i are either *decontaminated* or *active*; (ii) all cells in column $i-1$ (for $i > 0$) are *decontaminated*; (iii) by time $t+T$ all right neighbours of a *decontaminated* cell of column i are either *decontaminated* or *active*; (iv) the distance between any two consecutive entry points in column $i+1$ (for $i < n-1$) is smaller than T.

1. Base - column 0: According to the set of rules the *active* state propagates vertically in both directions on the first column (see rules $f(\{\bullet, 0, 0, 1, 1\}) = (0, \Downarrow)$, $f(\{\bullet, 0, \bullet, 1, 1\}) = (0, \Downarrow)$, $f(\{\bullet, 0, 1, 1, \bullet\}) = (0, \Uparrow)$ and $f(\{\bullet, 0, 1, 1, 0\}) = (0, \Uparrow)$), and since the maximal distance between initially consecutive active cells is T by construction, within $\lfloor \frac{T-1}{2} \rfloor$ time units all the cells on column 0 are either *active* or *decontaminated*. According to the local rules, decontamination then propagates to column 1. In fact, by rule $f(\{\bullet, 0, 1, 1, 1\}) = (0, \Uparrow\Downarrow)$ if there is a single *active* cell with up and down *contaminated* neighbours, it will duplicate and start propagating in both direction; if instead there is a pair of *active* cells, each of them starts propagating in the direction of their *contaminated* neighbour on the column (rules: $f(\bullet, 0, \bullet, 1, 1) = (0, \Downarrow)$ and $f(\bullet, 0, 1, 1, \bullet) = (0, \Uparrow)$). Since the propagation to column 1 happened for all *active* cells within $\lfloor \frac{T-1}{2} \rfloor$ time units from the beginning, the distance between any two consecutive entry points in column 1 cannot be greater than T. By a similar

argument as for column 0, all cells in column 1 will then become *decontaminated* or *active* within other $\lfloor \frac{T-1}{2} \rfloor$ time units. Thus, within at most T time units, all the cells in column 0 and in column 1 will be either *decontaminated* or *active*.

2. Induction hypothesis: Assume that there is a time when all cells of column i ($0 < i < n-1$) and all their right neighbours in column $i+1$ are either in *active* or *decontaminated* state, their left neighbours in column $i-1$ are clean, and the entry points in column $i+1$ are at distance at most T.

3. Induction Step: Consider column $i+1$. By induction hypothesis we know that there is a time t when all cells in column $i-1$, are clean, the ones in columns i, and $i+1$ are either *active* or *decontaminated* and the entry points in column $i+1$ are at maximum distance T from each other. It follows that the decontamination propagates vertically in column $i+1$ within $\lfloor \frac{T-1}{2} \rfloor$ time units from time t, thus leaving all cells in column i clean. Decontamination propagates then to column $i+2$ and within other $\lfloor \frac{T-1}{2} \rfloor$ time units all the cells in the column $i+2$ become either *decontaminated* or *active*. We can conclude that, by time $t+T$ all cells in column $i+1$ together with all their right neighbours are either *decontaminated* or *active* and the cells in column i are clean.

■

V. Conclusions

In this paper we have continued the line of investigation started in [5] to study the network decontamination problem in cellular systems where the spread of faults, as well as the decontamination process, are regulated by local rules. In [5] we have looked at cellular automata (CA), where *decontaminating* cells can spread following classical cellular automata local rules. In this paper we have focused on mobile cellular automata to look at the impact that *active* cells, in an otherwise reactive environment, have on the decontamination problem. We can observe that, if cloning is not allowed, with Von Neunmann neighborhood we obtain a more general solution than in the case of CA with the same neighborhood. The efficiency of our solution depends on the relationship between n and T and is always better than the one devised for CA. Note that the solutions for CA could be "simulated" also by MCA, but it would be efficient only for large immunity times ($T \geq n-2$). On the other hand, with Moore neighborhood the solution obtained with CA is better than the one devised for MCA, possibly because not having cloning capabilities for the agents is a bigger constraint than the lack of active cells. When the agents can clone, the Von Neumann neighborhood allows to obtain the same results as the ones obtained for CA with Moore neighborhood, thus showing that the extra power of the active cells is traded off with the enlarged neighborhood. A summary of the results is shown in Tables VI and VII.

TABLE VI
Decontamination in MCA with immunity $T > 1$, where $t_1 = \max\{\, odd\ t < T : (t+1) | n\}$ and $t_2 = \max\{\, even\ t < T : (t+2) | n\}$

Neigh.	Simultaneously active cells
No Cloning	
VN	$\min\{\frac{2n}{t_1+1}, \frac{3n}{t_2+1}\}$
Moore	$\lfloor \frac{4n}{T+3} \rfloor + 1$
Cloning	
VN	$\lceil \frac{2n}{(T+1)} \rceil$
Moore	$\lceil \frac{2n}{(T+1)} \rceil$

TABLE VII
Decontamination in CA with immunity T [5].

Neigh.	Simultaneously decontaminating cells
VN	n, for $T < n-2$ 4, for $n-2 \leq T < 2n-3$ 2, for $2n-3 \leq T < 4n-5$ 1, for $T \geq 4n-5$
Moore	$\lceil \frac{2n}{T+1} \rceil$

References

[1] L. Barrière, P. Flocchini, P. Fraigniaud, and N. Santoro. Capture of an intruder by mobile agents. In *14th annual ACM symposium on Parallel algorithms and architectures (SPAA)*, pages 200–209, 2002.

[2] L. Barrière, P. Fraigniaud, N. Santoro, and D.M. Thilikos. Searching is not jumping. In *29th Int. Workshop on Graph Theoretic Concepts in Computer Science (WG)*, pages 34–45, 2003.

[3] L. Blin, P. Fraigniaud, N. Nisse, and S. Vial. Distributed chasing of network intruders. *Theor. Comput. Sci.*, 399(1-2):12–37, 2008.

[4] V. Chevrier and N. Fatès. How important are updating schemes in multi-agent systems? An illustration on a multi-turmite model. In *9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 533–540, 2010.

[5] Y. Daadaa, P. Flocchini, and N. Zaguia. Network decontamination with temporal immunity by cellular automata. In *9th Int. Conf. on Cellular Automata for Research and Industry (ACRI)*, pages 287–299, 2010.

[6] P. Flocchini. Contamination and decontamination in majority-based systems. *Journal of Cellular Automata*, 4(3):183–200, 2009.

[7] P. Flocchini, M. J. Huang, and F. L. Luccio. Decontamination of hypercubes by mobile agents. *Networks*, 52(3):167–178, 2008.

[8] P. Flocchini, B. Mans, and N. Santoro. Tree decontamination with temporary immunity. In *19th Int. Symposium on Algorithms and Computation (ISAAC)*, pages 330–341, 2008.

[9] P. Fraigniaud and N. Nisse. Connected treewidth and connected graph searching. In *7th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 479–490, 2006.

[10] A. Gajardo and E. Goles. Dynamics of a class of ants on a one-dimensional lattice. *Theor. Comput. Sci.*, 322(2):267–283, 2004.

[11] S. Kutten and D. Peleg. Fault-local distributed mending. *Journal of Algorithms*, 30:263–273, 1999.

[12] S. Kutten and D. Peleg. Tight fault locality. *SIAM Journal on Computing*, 30:247–268, 2000.

[13] C. G. Langton. Studying artificial life with cellular automata. *Phys. D*, 2(1-3):120–149, 1986.

[14] F. Luccio and L. Pagli. A general approach to toroidal mesh decontamination with local immunity. In *23th Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2009.

[15] F. Luccio, L. Pagli, and N. Santoro. Network decontamination with local immunization. In *20th Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2006.

[16] D. Peleg. Size bounds for dynamic monopolies. *Discrete Applied Mathematics*, 86(2-3):263–273, 1998.

[17] D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282(2):231–257, 2002.

[18] A. Spicher, N. Fatès, and O. Simonin. From reactive multi-agents models to cellular automata - Illustration on a diffusion-limited aggregation model. In *Int. Conf. on Agents and Artificial Intelligence (ICAART)*, pages 422–429, 2009.

[19] S. Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

Neuronal CDMA and Neural Spread Spectrum Multi-Access: Biologically Plausible Computing, Communication and Coordination in Brain Circuits and Microcircuits

John-Thones Amenyo [1]
[1]Department of Mathematics & Computer Science
York College, CUNY, New York, NY, USA

Abstract - *Spread spectrum (SS) and CDMA schemes may be possible in biological neurons. If so, even single neurons have the ability to switch and route multiple concurrent flows across themselves, as an alternative to the conventional view of neurons as integrate-and-fire systems, or variants and extensions thereof. The paper presents a biologically plausible non-algebraic scheme for achieving Neuronal CDMA and SS capabilities. The availability of Neuronal CDMA allows one to regard neurons and neuron assemblies as versatile computation, communication and coordination container or bag devices, and also to bring to bear sophisticated modern notions of data structures and their manipulations via function arrays, combinatory, operator arrays and patterns as well as algorithmic skeletons, in creating neuron, brain and brain function models.*

Keywords: Neuronal CDMA, Neuronal Spread Spectrum, Neuromone, Codes, Algorithmic Skeletons, Cellular Automata, Dataflow Process Networks

1 Introduction

The brain, as part of the central nervous system (CNS), [17,37], is conventionally considered as neuron collections or assemblies organized into circuits and microcircuits [1,13,35,40], a system-of-systems viewpoint. The brain circuits are typically multi-scale structures, encompassing brain regions, functional cortical columns, layers and nuclei; synaptic circuits, dendritic circuits [10,36], as well sub-cellular (cytoskeletal) circuits coupled to neuron membranes structures. Brain circuits and brain micro-circuits, and other neuron assemblies, can be considered as rich sources of examples of "computation" (information processing) by Cellular Automata (CA). This is so provided one forgoes, or more fruitfully, extends and generalizes the attributes, properties conventionally used to characterize CA and related computational formalisms, such as distributed process networks [16, 24, 48] and cellular neural networks (CNN), [8, 9]. Specifically, one has to switch from concepts of uniformity, homogeneity, regularity and isotropy in topological connectivity, neighborhood relations, transition rules and functions. In addition, there is an absence of a central clock for global synchronization. Brain circuits and microcircuits are not non-synchronous or asynchronous. Instead, they are likely to be inter-connected islands of unison and wave synchrony, due to the presence of sequential or temporal composition and parallel or spatial composition of activity patterns, and thus can be described as being poly-synchronous or multi-synchronous.

A crucial part in the CA modeling approach is the determination, characterization and specification of the heterogeneous, multiple transition rules useful for brain "computation". This leads to the fundamental question of what exactly do brain circuits and microcircuits compute? [14,19,20]. From the top down, at a very general level, brain circuits are there to support an organism's survival during its individual lifetime, as well as supporting reproductive success in passing the organism's genes on to the posterity. But what are the details (bottom up view) of such survival-driven "computations" (control, coordination, etc)?

This paper advances and analyzes the following integrated set of hypotheses about plausible information processing and biological computation in the brain and the central nervous system (CNS).

The biological neuron is A) a versatile communication device that is organized as a shared multi-access medium, and supports multiple, logically separate concurrent flows across it. B) To do so, the neuron uses multi-access schemes that can be modeled or described as being based on CDMA (code division multi-access) or spread spectrum techniques, [47, 50]. C) The users or endpoints of the concurrent flows are associated with synapses, specifically, dendritic post-synaptic bodies on one neuron as senders, and dendritic post-synaptic bodies on directly connecting downstream neurons (and/or glandular cells or motor cells). D) Similarly, the gap junctions (or electrical synapses) of a neuron, as well as neuro-glial interconnection bodies between neurons and glia cells, can also serve as senders and receivers (transceivers) of logical flows. E) The action potential (or neuron spiking, firing or pulse) zone serves as a kind of communications modulator; the axon and axonal arboration serve as communications transmissions and distributors; and the axonal pre-synaptic bodies serve as flow stream modulators/encoders and demodulators/decoders; (it is also plausible that pre-synaptic endpoints also participate in flow stream communication processing). F) In most cases, the post-synaptic bodies of a neuron act as add-drop transceivers, in that each taps and extracts information from the flow stream it receives from upstream, and then injects or insert new data, (for example,

from memory and synaptic plasticity structures), into the flow, for onward transmission downstream. G) The dendritic arboration and the cell body or soma serve as the shared medium interconnection fabric, physically integrating the multiple logical flows via cell membrane processes. H) The dendritic arboration and cell soma also perform roles of inter-neuron and intra-neuronal control, coordination, synchronization, and computation (creation, selection, manipulations and transformations of logical data structures and patterns).

The rest of the paper is organized as follows. The logical architectural details are presented about a non-algebraic, syntax-directed, multi-access resource sharing of single neurons, using spreading codes called *neuromones*. Next, there is a discussion of leveraging the Neuronal CMDA / SS that shows the possibilities of biological neurons and neuron assemblies to be considered truly versatile communication, computation and coordination devices and systems. Then, there is a discussion regarding the Neuronal CDMA scheme as an alternative single neuron model to the dominant conventional one of neurons being integrate-and-fire systems, as well as variants and non-linear extensions, thereof. Several directions of research to elaborate the Neuronal CDMA theory are then presented.

2 Neuronal CDMA and Spread Spectrum Multi-Access Processes in the Brain and CNS

The conventional (algebraic) model and characterization of Direct Sequence Spread Spectrum (DSSS) or Direct Sequence CDMA (DS-CDMA), [30,47,50], states that in order for multiple (sender, receiver) or flow pairs to be able to use a common shared medium, without undue multi-access interference (MAI), each flow pair is allocated a unique spreading code or codeword, known to both the sender(s) and receiver(s) of each logical flow. The sender of a flow takes portions of a message data to be sent as part of the flow, transforms (encodes) it according the spreading code, by spreading the data into a higher bandwidth than the data originally has. The encoded data is then combined (mixed, superimposed, integrated) with encoded data from other flows, modulated onto a carrier medium, if necessary; or otherwise transmitted across the shared medium or communication channel. The receiver of a particular (logical) flow, on the receiving the integrated flow, uses its spreading code to de-spread and decode the flow, and is thus able to extract the information sent in the flow intended or targeted to it. In order for such a spread spectrum scheme to work, it is generally understood that the spreading codes in the code set for all the supported flows should meet several "correlation" requirements [12,15,47]. Namely, a) the cross-correlation between any two distinct spreading codes should be as low as possible; b) the (in-phase) auto-correlation of any spreading code, with itself, should be as high as possible; c) if multi-path propagation effects are possible, the (out of phase) auto-correlation of a spreading code with shifted versions of itself, (self-interference), should be as low as possible; (multi-path propagation is probably not an important first order effect across a single biological neuron; although it may be important in logical flows and multi-access across neuron assemblies).

In order to keep the encoding and decoding processes and structures very simple in neurons, it is hypothesized that the Brain / CNS (BCNS) spread spectrum techniques are based largely on non-algebraic approaches. One such approach, based on multi-tagged brackets (or the Bra-Ket SS/CDMA) is detailed here.

Let A = {a0, a1, …, aQ}, Q = (q – 1), q > 1, |A| = q, be the common alphabet set used for communication in all flows across a neuron (or inter-neuron assembly), as a shared medium. It is unlikely that a binary scheme (q = 2) is used in the brain / CNS (BCNS). The minimum q-value is likely to be 11 or 13, according to the following reasoning. Use the subset A_m = {0,1,2, 9} of A to support metadata subsequences such as blanks, spaces, whitespaces, escape symbols; use the subset A_c = {3,4,5,10,12} for coding the (minimal) spreading codes or neuromones, and use the subset A_d = {6,7,8,11} for coding actual data transfer. Let $|A_c|$ = p. (According to (Galois) finite field theory, q or p have to be either a prime integer or a positive power of a prime.) Nevertheless, with the technique of dynamic code switching, as explained below, the BCNS could use several values of q, even concurrently in different flows, over an organism's lifetime.

Let U = {0, 1, …, N}, N = n – 1, |U| = n, be the number of logical flows, data streams or (sender, receiver)-pairs supported for a shared medium, such as a neuron. The size of n depends on the biological species, and can be estimated according to the number of synapses incident on a neuron. In lower animals, the neuronal incidence degree is estimated to be roughly 50 – 200. In the human brain, a common value is n = 10,000, but values as high as n = 200,000 have been quoted [17, 37].

Each flow, or (sender, receiver)-pair is associated with two types of spreading codes, a header code or Bra code, and a trailer code or Ket code. Let B = { $[_0$, $[_1$, …, $[_M$}, = { <0|, <1|, …, <M| } be the set of Bra codes, and C = { $]_0$, $]_1$, …, $]_M$ } = { |0>, |1>, …, |M> } be the set of Ket codes, M = (m-1), n < m = p^k. Thus, 7 < k < 19. The notations <a| and |a>, inspired by Dirac's bracket notation in physics, can also be seen to be XML-like in the use of labeled or annotated tags. The Bra and Ket spreading codes are termed *neuromones*, in analogy to pheromones.

Let there be two distinct codes, symbolized by (and), to be used in forming (encoded) data molecules. Being a data molecule means all the symbols in a molecule, (whether arranged as a sequence, array, nested array, graph, hyper-graph or polytope), are strongly bonded or inter-linked, and always go together, a form of transactional atomicity, and for short time durations can be distinctly recognized as a distinct bouquet, blend or cocktail pattern.

If Sender i has a message data $d \in A_d$ to send, it uses an operation $\otimes$ together with its specific neuromone spreading codes to encode the message to be transmitted as a data molecule: $(\otimes [i \otimes d \otimes]i \otimes) = (<i| d |i>)$. At any moment, relying on the combination operator, $\oplus$, the combined message being transmitted over the shared medium has the pattern or form:

$$\ldots\oplus (\otimes[_0\otimes d0\otimes]_0\otimes) \oplus \ldots \oplus (\otimes[_i\otimes di\otimes]_i\otimes) \oplus \ldots \oplus(\otimes[_j\otimes dj\otimes]_i\otimes) \oplus \ldots \oplus(\otimes[_N\otimes dN\otimes]_N\otimes) \oplus \ldots \quad (1)$$

$$\ldots\oplus (\otimes<0|d0|0>) \oplus \ldots \oplus (<i|di|i>) \oplus \ldots \oplus(<j|dj|j>) \oplus \ldots \oplus(<N|dN|N>) \oplus \ldots, \quad (2)$$

$d0, \ldots, di, \ldots dj, \ldots dN \in A.$

The Receiver i is able to use the $\oplus$ operator to isolate each encoded data molecule, $(\otimes [j \otimes dj \otimes]j \otimes) = (<j| dj |j>)$, and then use its neuromone codes to construct the de-spreading operator $\odot_i$ to pattern match its own spreading codes and then deconstruct a matching molecule, and extract the data message d_i sent by the Sender i:

$$\odot_i(\otimes [_j \otimes dj \otimes]_j \otimes)\odot_i = \odot_i(<j| dj |j>)\odot_i \quad (3)$$

$$(\otimes\odot_i[_j\odot_i\otimes dj \otimes\odot_i]_j\otimes\odot_i) = (\odot_i<j|\odot_i dj \odot_i|j>\odot_i)$$
$$= di, \text{ if } i = j \quad (4)$$
$$= (<j|dj|j>), \text{ if } i \neq j \quad (5)$$

The non-algebraic scheme can also be termed the grammar-theoretic or syntax-directed DS-CDMA or SSMA. In this scheme, the essential fundamental building blocks needed include various kinds of parsing, or de-structuring or isolations (P1: $x\oplus y \rightarrow \{x, y\}$, P2: $x\otimes y \rightarrow \{x, y\}$), as well as pattern matching and pattern recognition operations (in particular, $\odot_i[_j\odot_i$, $\odot_i[_i\odot_i$, $\odot_i]_j\odot_i$, & $\odot_i]_i\odot_i$). The $\otimes$ operations will correspond to adjoining or concatenation (attaching, appending, pre-pending), much like using bar codes and RFID tags as metadata to augment and annotate physical entities. It is also akin to the key and plaintext transformations and recombination performed in stream ciphers. The $\oplus$ operations are likely to be interleaving, interlacing, inter-mixing, scrambling superimpositions, governed eventually by the changes in voltage differences caused by differences in ion populations across the neuron cell membrane sections and regions.

A crucial part of the scheme is the identification of the code sets that can serve neuromone spreading codes. In the alternative algebraic approach, (discussed briefly below), there are several code schemes derived from (pseudo-noise) pseudorandom sequences that are required to meet several suitable low cross-correlation and low out-of-phase auto-correlation properties. These sequences include m-sequences, de Bruijn sequences, Gold sequences, Kasami sequences, optical orthogonal codes, and combinations, concatenations and generalizations thereof, [12,15,50]. In the syntax-directed non-algebraic scheme discussed here, the main requirement is that the codes in a neuromone code set should be sufficiently distinguishable from each other, (unique ids, uid), so that they cannot be confused by pattern matching , (for example, NOT XOR), mechanisms. This can be achieved by choosing the neuromone spreading codes from an extension field $GF(q^L)$ of $GF(q) = GF(p^k)$. Given n, the number of flows to be supported, choose k such that q^k is just greater than n ($n < q^k$). Now choose L such that $q^L = q^{ak}$. Divide all the possible q^L codes into q^k bins, each with q^{L-k} codes. Now choose 1 code from each bin to serve as a neuromone for a particular flow in the flow set U. One could also obtain the uid by means of base sequence concatenation, decimation, products or interleaving.

Alternative approaches can be related to using pseudorandom sequences to generate encryption keys, built-in test patterns and collision free hash functions. Again the exact choices of the code sets are biologically adaptable, because the BCNS can employ code switching. Code switching means that for any flow, the specific SS/CDMA coding scheme used can be changed dynamically, on the fly, in real time, after a suitable notification ("hand-shake") to the other side by the sender (or receiver) making change. The switching will be a special designated code in the existing coding scheme, for example using subsequences from the A_m alphabet. Thus, for code switching to work, there must be an initial code scheme (Garden of Eden code), a designation of when the switching is occurring in an existing scheme, and either a pre-arranged or a hastily formed capability to be able to perform encoding and decoding in the new code scheme.

3 Neural Correlates and Biological Embodiment of Neuronal CDMA

The primary time period for the generation of neuromones is during BCNS development (neurogenesis, as part of morphogenesis). Code generation may also occur in the case of neuronal stem cell differentiation, as well as dynamically during learning (for example, synaptic plasticity). At the cellular level, it is hypothesized that neuronal CDMA occurs at the synapses, with each individual neuron behaving like a (multi-access) shared medium, on par with a local area network (LAN). That is, the logical flow streams occur across a neuron, from the dendritic post-synaptic bodies (local senders) through the axonal pre-synaptic bodies to dendritic post-synaptic bodies on downstream neurons (local receivers), (see Figure 1). The formal apparatus used is likely based on (parallel implementations) of Shift Registers (SR), Feedback Shift Registers (FSR), and (digital) filters, [11]. Post-synaptic bodies will include structures that support both decoders and encoders, so that these bodies can behave like add-drop

"multiplexers". Local patches, caches or buffers of ions, (Na^+, K^+, Ca^{2+}, Cl^-, A^-), serve as storage registers. For each ion type, the local ion caches are split to occur both outside the neuron membrane and inside the neuron, under the membrane. The state of the registers or ion caches are changed by means of the activity of (local) trans-membrane ion channels. (Neuro-) transmitter-gated ion channels, and ligand-gated ion channels [2] serve as inputs to the SR / FSR or digital filters. The shift or delay operators are embodied using voltage-gated ion channels and ion-gated ion channels. Flows of data or information are embodied both as propagation of changes in voltage differences, as well as actual ion flows. Both dendritic graded potentials and action potentials can be supported in the model. As noted below, antidromic feedback flows have also been observed in neurons, using modern experimental techniques, [44]. The communication substructures can be distinguished for the synapse (dendritic spine), dendritic shaft segment without spines of synapses, and dendritic arbor joint points. Also, in this respect, the array layout properties of microtubules make them suitable candidates for playing roles as additional embodiments of storage registers. Look up tables (LUT) may be in use in simpler BCNS. The overall sub-cellular structures involved in neuronal CDMA may thus include gating of ion conductance channels, pathways linking receptors to sub-cellular structures such as actin filaments and microtubule cytoskeleton structures, via second messengers, microtubule associated proteins (MAP), as well as calcium ion (Ca^{2+}) concentration management at various regions inside and proximate to a neuron. The determination of the exact mechanisms in use will require future collaborations with neurophysiologists and other neuro-experimentalists.

4 Leveraging Neuronal CDMA in the Brain & Central Nervous System

The availability of CDMA/SS capability in a neuron means that a neuron is a true (versatile) communication device, capable of playing the multi-faceted roles of relay, transponder, bridge, switch, concentrator, multiplexer, router, gateway, and intermediator adaptor. Thus a neuron will be capable of bundling multiple logical flows of data streams or sequences across itself, considered much like a local area network or variants. It also allows one to model and attempt to discover the use of various modern communication paradigms in BCNS. These includes, resource-sharing, access to services, grid computing, client-server, mash-ups, cloud computing, virtualization, hastily formed networks, ad hoc computing, inter-operability, collaboration, cooperation, self-organization, peer-2-peer, friend-to-friend comm., master-worker, leader-follower, data parallelism, algorithmic skeletons; distributed self-assembly, reconfigurable architectures, wireless sensor networks, dense RFID networks, etc. A neuron may also support flows classified according to communication (U-plane / D-plane), control and coordination (C-plane), and meta, management, self-star or autonomic management (M-plane / K-plane).

Neuronal CDMA provides the opportunity for modeling both intra-neuron and inter-neuron dendritic and synaptic processing, to serve various purposes [3,7,10,26,27,28,29,34,38,39,42,43,45,49]. There is also the possibility of specifying communication based architectures for distributed pattern generation, such for central pattern generators (CPG), fixed action pattern (FAP) generators, and modular action pattern (MAP) generators, for sensory patterns, sensory integration patterns, motor patterns and sensory-motor integration patterns. These architectures can be multi-scale, multi-level and hierarchical. The Neuronal CDMA also allows the possibility of modeling and investigating remote or distal associations and correspondences, beyond conventionally accepted topographic mappings, such as retinotopy, tonotopy, and somatotopy.

In fact, with the Neuronal CDMA capability, from a communications viewpoint, a neuron can be regarded as having an architecture similar to that of a high-end network processor (core switch or core router), with identifiable functional elements of input (axo-dendritic, axo-somatic, axo-axonal), input-output (dendro-dendritic, electrical, neuro-glial junctions) and output (axonal) line cards (synapses); interconnection fabric; and supporting parallel computational architectures for computation, coordination, life cycle support (LCS) management, "ILities" and Self-* autonomic management.

5 Discussion and Related Work

An alternative approach is to hypothesize that the BCNS uses conventional algebraic CDMA/SS techniques. In that case the alphabet A will involve number representations or symbolizations, and to be able to decode and recover data messages, without undue MAI (multi-access interference), the commutative, associative and distributive requirements on the $\oplus$ and $\otimes$ will mean $<A, \oplus, \otimes>$ is a finite field, also called a Galois field, GF(q), with q being a prime number. The $\oplus$ operation will be ordinary arithmetic addition (mod q) or (mod q^L), and the $\otimes$ and $\odot$ operations will be arithmetic multiplication (mod q) or (mod q^L), for some L>0. In the binary case of q = 2, the operations reduce to XOR and AND Boolean logic operations, respectively. For field extensions GF(q^L), the elements as well as the $\otimes$ multiplication rules can be obtained by means of primitive elements satisfying irreducible primitive polynomials of the field. However, the code generation and decoding machinery will be more involved and complicated, being based fundamentally on $x \otimes y$ multiplication operations. It is not clear how correlation based machinery can be viable in simple organisms such as C. elegans. The (numeric) neuromone spreading codes will be required to have suitable orthogonality and correlation properties, for example, Golomb's PN properties, [12,15,33]. The correlation properties are the replacement for the pattern matching and pattern recognition capabilities needed in the

non-algebraic approach. Also it is not clear that modulation of carriers actually occurs in the biological situation.

Several neuromorphic circuits (analog, VLSI, mixed analog digital), which have been designed to mimic the dominant integrate-and-fire model of neurons, use a special form of time division multi-access (TDMA) scheme, called Address Event Representation (AER) to support multi-access of shared channels of communications between neuromorphic chips [5, 6, 41]. The use of conventional CDMA, LFSR, m-sequences and Gold sequences for neuron memory circuit loops was mentioned briefly in [46].

The Neuronal CDMA / SS approach can also be regarded as advocating an alternative or at least supplementary single neuron model to the currently dominant integrate-and-fire model and its variants. Ever since the neuron doctrine was formulated, based on the work of such researchers as Cajal, Sherrington and others [17, 37], the conventional view of a biological neuron is that the inputs to its synaptic terminals in (mainly) the dendrites and soma are integrated via additive summation, and if the sum is high enough above a threshold, it results in the non-linear (active) generation of pulse trains or spikes, also called action potential. There are several forms and variants of this model, including Hodgkin-Huxley's very influential model, [17, 37]. It has also permeated theoretical models, such as artificial neural networks (ANN), both the formulations based on the original McCulloch-Pitts model, and modern Hopfield revival. Later research has reveal that neurons have more complex, active and non-linear information processing capabilities, in the dendrites, for example, dendritic spines, including several forms of arithmetic and (Boolean) logic operations, [4,10,14,18,21,22,23,25,28]. Research has also shown that electricity based information flow is not merely directional, from the dendritic arbor towards the soma and axonal arboration, but also that there is a soma-dendritic backward propagation of AP-induced electric waves or signals from the AP trigger zone (axon-hillock) towards the cell body, soma and dendritic nets, spines, synapses, a form of antidromic electrotonic spread [44]. Such flows can be leveraged for feedback signaling. Nevertheless, there is a fundamental issue with the integrate-and-fire model and various attempts to update it, [31,32]. Namely, according to these models, each of the individual input flows into sensory cells or motor neurons, but also into inter-neurons, that may have 10,000 or even up to 200,000 synapses! Distinct input sequences get "lost" and "swallowed up", becoming part of the "statistic", once it reaches a neuron.There is no way in these models to route or switch a particular input flow (logical data stream or sequence) distinctly across a neuron so that there is some kind of correlation between axonal pre-synaptic (input) and dendritic post-synaptic (output) flow processes of the same neuron. The Neuronal CDMA scheme advocates that such flow routing is possible even across single neurons.

The biological plausibility and realism of the specific details advanced here are empirical issues that can only be resolved via future collaborations on neurological experiments. A future direction of research is to determine how digital filter, SR, FSR, NLFSR, LFSR circuits are embodied and incarnated (at the sub-cellular level) to support CDMA/SS processing at the synaptic level. Of particular interest are biological architectures that support parallel generation of neuromone spreading codes and their uses for CDMA/SS decoding and encoding. Another future direction of research is the determination of the embodiment of the CDMA/SS capabilities at the neuron assembly (inter-neuron architecture) level.

6 Summary and Conclusions

It is biologically plausible that spread spectrum (SS) and CDMA occur in the BCNS, even at the single neuron level, so that neurons (and neuron assemblies) are able to route and switch differentiated logical flows of information across themselves, acting as shared multi-access media. The capability would allow the BCNS to create multiple, concurrent and complex, distributed patterns in spatial, temporal and spatio-temporal dimensions and scales to support survival needs. A non-algebraic scheme for Neuro CDMA is described, with the important property that the attendant encoding and decoding embodiments can be particularly simple in a biological setting.

7 References

[1] Abeles, M., Corticonics: Neural Circuits of the Cerebral Cortex. Cambridge University Press. Cambridge, UK (1991).

[2] Alberts, B., Johnson, A., Lewis, J. et al., Molecular Biology of the Cell. 4e. Garland Science. New York, NY (2002).

[3] Barlow, H.B., Intraneuronal Information Processing, Directional Selectivity and Memory for Spatio-temporal Sequences. Network. Volume 7, (1996), pp. 251 – 259.

[4] Bloomfield, S., Arithmetical Operations Performed by Nerve Cells. Brain Research. Volume 69, (1974), pp. 115 - 124.

[5] Boahen, K., Neuromorphic Microchips. Scientific American. (May 2005), pp. 56 -63.

[6] Boahen, K., Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events. IEEE Transactions on Circuits and Systems II, Analog Digital Signal Processing. Volume 47, No. 5, (May 2000), pp. 416 – 434.

[7] Carr, C.E., Processing of Temporal Information in the Brain. Annual Reviews of Neuroscience. Volume 16, (1993), pp. 223 – 243.

[8] Chua, L.O. & Roska, T., The CNN Paradigm. IEEE Transactions on Circuits and Systems. Volume 40, No. 3, (1993), pp. 147 – 156.

[9] Chua, L.O. & Yang, I., Cellular Neural Networks: Theory, Cellular Neural Networks: Applications, IEEE Transactions on Circuits and Systems. Volume 35, (October 1988), pp. 1257 – 1295.

[10] Euler, T. & Denk, W., Dendritic Processing. Current Opinions in Neurobiology. Volume 11, (2001), pp. 415 – 422.

[11] Golomb, S., Shift Register Sequences. Aegean Park Press. Laguna Hills, CA (1981).

[12] Golomb, S. & Gong, G., Signal Design for Good Correlation Cambridge Univ. Press, New York, 2005.

[13] (eds.) Grillner, S. & Graybiel, A.M., Microcircuits: The Interface Between Neurons and Global Brain Function. MIT Press. Cambridge, MA (2006).

[14] Hausser, M. & Mel, B., Dendrites: Bug or Feature? Current Opinions in Neurobiology. Volume 13, (2003), pp. 372 – 383.

[15] Helleseth, T. & Kumar, P.V., Sequences with Low Correlation. In (eds.) Pless, V. & Huffman, W., Handbook of Coding Theory. North-Holland-Elsevier, Amsterdam, (1998).

[16] Kahn, G., The Semantics of a Simple Language for Parallel Programming. Proceedings of IFIP '74 Congress (1974), pp. 471 – 475.

[17] (eds.) Kandel, E.R., Schwartz, J.H. & Jessell, T.M., Principles of Neural Science, 4e. McGraw-Hill, New York, NY (2000).

[18] Koch, C., Biophysics of Computation. Oxford University Press. New York, NY (1999).

[19] Koch, C., Computation and the Single Neuron. Nature. Volume 385, (1997), pp. 207 – 210.

[20] Koch, C. & Poggio, T., Biophysics of Computation: Neurons, Synapses and Membranes. In (eds.) Edelman, G.M., Gall, W.E. & Cowan, W.M., Synaptic Function, Wiley. New York, NY (1987), pp. 637 – 697.

[21] Koch, C., Poggio, T. & Torre, V., Nonlinear Interactions in a Dendritic Tree: Localization, Timing, and Role of Information Processing. Proceedings of the National Academy of Sciences. USA. Volume 80, (1983), pp. 2799 – 2802.

[22] Koch, C. & Segev, I., The Role of Single Neurons in Information Processing. Nature Neuroscience. Volume 3 (Supplement), (2000), pp. 1171 – 1177.

[23] (eds.) Koch, C & Segev, I., Methods in Neuronal Modeling. MIT Press. Cambridge, MA (1993).

[24] Lee, E.A. & Parks, T.M., "Dataflow Process Networks", *Proceedings of the IEEE*, Volume. 83, (May 1995), pp. 773-801.

[25] London, M. & Hausser, M., Dendritic Computation. Annual Reviews in Neuroscience. Volume 28, (2005), pp. 503 – 532.

[26] Magee, J.C., Dendritic Integration of Excitatory Synaptic Inputs. Nature Reviews Neuroscience. Volume 1, (2000), pp. 181 – 190.

[27] Mainen, Z.F., Functional Plasticity at Dendritic Synapses. (eds.) Stuart, G., et al., Dendrites. Oxford University Press. New York, NY (1999), pp. 310 – 318.

[28] Mel, B. W., Why Have Dendrites? A Computational Perspective. In (eds.) Stuart, G., Spruston, N. & Hausser, M., Dendrites. Oxford University Press. New York, NY (2003), pp. 271 – 289.

[29] Mel, B. W., Synaptic Integration in Excitable Dendritic Trees. Journal of Neurophysiology. Volume 70, (1993), pp. 1086 – 1101.

[30] Pickholtz, R., Schilling, D. & Milstein, L.B., Spread Spectrum Communications – A Tutorial. IEEE Transactions on Communications. Volume COM-30, No. 5 (May 1982), pp. 855 – 884.

[31] Poirazi, P., Brannon, T. & Mel, B.W., Pyramidal Neuron as a 2-layer Neural Network. Neuron. Volume 37, (2003), pp 989 – 999.

[32] Poirazi, P., Brannon, T. & Mel, B.W., Arithmetic of Subthreshold Synaptic Summation in a Model of a CA1 Pyramidal Cell. Neuron. Volume 37, (2003), pp. 977 – 987.

[33] Sarwate, D. & Pursley ,M., Crosscorrelation properties of pseudorandom and related sequences.
Proc. IEEE 68 (1980), 593–619.

[34] Segev, I. & London, M., A Theoretical View of Passive and Active Dendrites. (eds.) Stuart, G., Spruston, N. & Hausser, M., Dendrites. Oxford University Press. New York, NY (1999), pp. 205 – 230.

[35] (ed.) Shepherd, G.M., The Synaptic Organization of the Brain. 5e. Oxford University Press, New York, NY (2004).

[36] Shepherd, G.M., Information Processing in Complex Dendrites. In (eds.) Squire, L.R. et al. Fundamental Neuroscience. Academic Press. New York, NY (2003), pp. 319 – 338.

[37] Shepherd, G.M., Neurobiology, 2e. Oxford University Press. New York, NY (1988).

[38] Shepherd, G.M. & Brayton, R.K., Logic Operations are Properties of Computer-Simulated Interactions Between

Excitable Dendritic Spines. Neuroscience. Volume 21, (1987), pp. 151 – 166.

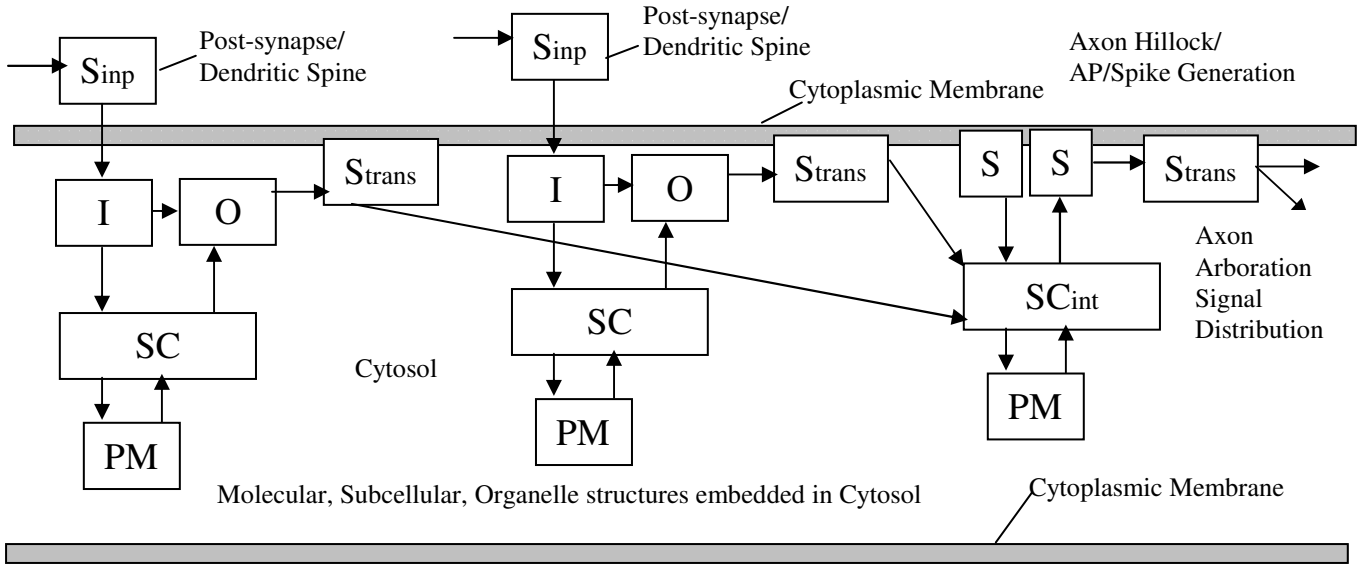
[39] Shepherd, G.M., Carnevale, N.T., & Woolf, T.B., Comparisons Between Active Properties of Distal Dendritic Branches and Spines: Implications for Neuronal Computations. Journal of Cognitive Neuroscience. Volume 1, (1989), pp. 273 – 286.

[40] (eds.) Shepherd, G.M. & Grillner, S., Handbook of Brain Microcircuits. Oxford University Press. New York, NY (2010).

[41] Sivilotti, M., Wiring Considerations in Analog VLSI Systems with Application to Field-programmable Networks. Ph.D. dissertation, Comp. Sci. Div., California Institute of Technology, Pasadena, CA, (1991).

[42] Single, S. & Borst, A., Dendritic Integration and its Role in Computing Image Velocity. Science. Volume 281, (1998), pp. 1848 – 1850.

[43] Softky, W., Sub-millisecond Coincidence Detection in Active Dendritic Trees. Neuroscience. Volume 58, (1994), pp. 13 – 41.



Sinp : Synaptic Input : Signal Reception, Transceiver
Strans : Membrane based Signal Transfer, Transmission, Comm.
SCint : Signal Flow Integration, Multi-Access Superimposition
I : Input Decoding (for Drop) ; O : Output Encoding (Add)
SC : Signal Switching, Routing
PM : Signal Sequence Experience Storage & Manipulation

Figure [1] : Sub-cellular & Molecular Scale Implementation of Neuronal CDMA

[44] Stuart, G., Spruston, N., Sakmann, B. & Hausser, M., Action Potential Initiation and Back Propagation in Central Neurons. Trends in Neuroscience. Volume 20, (1997), pp. 125 – 131.

[45] Stuart, G. & Hausser, M., Dendritic Coincidence Detection of EPSPs and Action Potentials. Nature Neuroscience. Volume 4, (2001), pp. 63 – 71.

[46] Tamura, S., Mizuno-Matsumoto, Y., Chen, Y-W. & Nakamura, K., Association and Abstraction on Neural Circuit Loop and Coding. 2009 IEEE Fifth Conference on Intelligent Information Hiding and Multimedia Signal Processing. (2009), pp. 414 – 417.

[47] Viterbi, A.J., CDMA Principles of Spread Spectrum Communications. Addison-Wesley. Reading, MA (1995).

[48] Webb, D., Wendelborn, A.L. & Maciunas, K.J., Process Networks as a High-Level
Notation for Metacomputing. In (eds.) Rolim, J. et al., Parallel and Distributed Processing. Lecture Notes in Computer Science. Volume LCNS 1586 (Springer, 1999), pp. 797-812.

[49] Williams, S.R. & Stuart, G., Role of Dendritic Synapse Location in the Control of Action Potential Output. Trends in Neuroscience. Volume 26, (2003), pp. 147 – 154.

[50] Zigangirov, K. SH., Theory of Code Division Multiple Access Communication. IEEE Press. Piscataway, NJ (2004).

Cellular Automata Based Cryptographic Hash Function*

Jun-Cheol Jeon
Department of Information Security, Woosuk university, Jeonbuk, Korea

Abstract - *Recently, in many applications where speed is important and very large amounts of data have to be authenticated, hardware implementation is demanded as a natural solution. A cellular automata (CA) is one of the best solutions for the hardware structure because of its parallelism and bit-wise operations. This paper proposes a secure and efficient cryptographic hash function based on a linear group and nonlinear non-group CA. Our algorithm satisfies the secure properties and forms remarkably simple structure. We show that our hash function produces an excellent quality of hash result having a low construction cost.*

Keywords: Cryptography, Hash Function, Cellular Automata, Hardware Implementation

1 Introduction

Cryptographic hash functions play an important role in modern communication technology. The basic idea of cryptographic hash functions is that a hash-value serves as a compact representative image(sometimes called an imprint, digital fingerprint, or massage digest) of an input string, and can be used as if it was uniquely identifiable with that string. Many cryptographic hash functions, all based on the so called MD4 initially proposed in [9], have received the greatest attention. However, in applications where speed is important and very large amounts of data have to be authenticated (e.g., electronic financial transactions, software integrity), hardware implementations are the natural solution. Thus dedicated cryptographic hash functions based on cellular automata are strongly recommended [3][7].

The Second Cryptographic Hash Workshop was held on Aug. 24-25, 2006, at University of California, Santa Barbara, in conjunction with Crypto 2006. The workshop program consisted of a day and a half of presentations of papers that were submitted to the workshop and panel discussion sessions. The main topics of the workshop included survey of hash function applications, new structures and designs of hash functions, cryptanalysis and attack tools, and development strategy of new hash functions [8].

Thomas Ristenpart of the University of California, San Diego Security and Cryptography Laboratory proposed replacing the Merkle-Damgard transform with a multi-property-preserving domain extension transform. The goal is to build hash functions to be secure for as many applications as possible. He gives an example of a multi-property-preserving transform, called Enveloped Merkle-Damgard(EMD). This is shown to have some provable security properties [1].

However, some works have been suggested based on cellular automata (CA); In [3], Daemen et al. have persisted in vulnerability of scheme from [4] together with a new CA based hash function. Another research on CA based hash function has been reported by Mihaljevic et al.[7] based on their previous report. They have proposed a family of fast dedicated one-way hash functions based on linear CA over GF(q) in 1999. In a CA viewpoint, the above mentioned schemes are not fully CA based hash functions since they did not provide any specific neighborhood and rules. Moreover, a compression function in [7] has only two times linear CA operations and other nonlinear functions are from HAVAL [12]. Though the previous papers have persisted in their security and advantages, they did not provide enough comprehension on security and experimental results. Moreover the previous works did not use specific rules so that it is hard to determine the characteristics of their schemes. Therefore, well-defined and designed CA based hash function has been so required.

The remainder of this paper is organized as follows. Section 2 introduces some CA knowledge upon which the subject matter of this paper is based. In Section 3, we propose a secure cryptographic hash function based on a CA which is highly optimized and is suitable for hardware implementation. In Section 4, we analyze the security of our scheme proposed in this paper. We show that the proposed scheme is secure, and produces a good quality of message digest. Then, our scheme is compared with previous well-known hash schemes in term of a hardware construction cost. Finally, Chapter 5 gives our conclusions.

2 Cellular automata

A CA is a collection of simple cells connected in a regular fashion. A CA was originally proposed by John von

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0014977).

Neumann as formal models of self-reproducing organisms. Wolfram [11] pioneered the investigation of CA as mathematical models for self-organizing statistical systems and suggested the use of a simple two-state, three-neighborhood (left, self and right) CA with cells arranged linearly in one dimension. The CA structure investigated by Wolfram can be viewed as a discrete lattice of cells where each cell can assume either the value 0 or 1. The next state of a cell is assumed to depend on itself and on its two neighbors (three-neighborhood dependency). The cells evolve in discrete time steps according to some deterministic rule that depends only on local neighbors. In effect, each cell consists of a storage element (D flip-flop) and a combinational logic implementing the next-state function [2].

In an *m*-state, *k*-neighborhood CA, each cell can exist in *m* different states and the next state of any particular cell depends on the present states of *k* of its neighbors. In this thesis, we use a simple 2-state 3-neighborhood CA with the cells in one dimension. Mathematically, the next state transition of the *i*th cell can be represented as a function of the present sates of the *i*th, (*i*+1)th and (*i*−1)th cells: $s_i = f(s_{i-1}, s_i, s_{i+1})$, where *f* is known as the *rule* of the CA denoting the combinational logic.

For a 2-state 3-cell neighborhood CA, there can be a total of 2^3 distinct neighborhood configurations. For such a CA with cells having only 2 states there can be a total of $2^{2\times2\times2}$(=256) distinct mappings from all these neighborhood configurations to the next state. If the next-state function of a cell is expressed in the form of a truth table, then the decimal equivalent of the output is conventionally called the *rule number* for the cell [10].

If the rule of a CA cell involves only XOR logic, then it is called a *linear rule*. A CA with all the cells having linear rules is called a *linear CA*, whereas a CA with AND-OR logic is a *nonlinear CA*. If a state transition of a CA contains only cyclic states, then the CA is called a *group CA*; otherwise it is a *nongroup CA*. The rule applied on a uniform group CA is called a *group rule*; otherwise it is a *nongroup rule* [2].

3 Proposed hash function (CAH-256)

The total configuration of our scheme is similar to the EMD structure mentioned in Section 1 while the core algorithm of compression function is based on a CA. Given a message *M* to be compressed, CAH-256 pads *M* first. The length of (i.e., the number of bits in) the message after padding is a multiple of 256, and padding is always applied even when the length of *M* is already a multiple of 256. The last block of the padded message contains the number of bits in the unpadded message. Now suppose that the padded message is $M^{(0)}\ldots M^{(n-2)}M^{(n-1)}$, where each $M^{(j)}$ is a 256-bit block. CAH-256 starts from the block $M^{(0)}$ and a 256-bit string initial value *IV*, and processes the message $M^{(0)}\ldots M^{(n-2)}M^{(n-1)}$ in a block-block way. At the final stage, our algorithm uses different parameters with another supplement value, *SV*. More precisely, it compresses the message by repeatedly calculating

$$H^{(0)} = IV_1,$$
$$H^{(j+1)} = H_{CAH}(H^{(j)} \oplus M^{(j)}),\ 0 \le j \le n\text{-}2,$$
$$H^{(n)} = H_{CAH}(H^{(n-1)} \oplus M^{(n-1)} \oplus IV_2),$$

where *j* ranges from 0 to *n*−2 and H_{CAH} is called the updating algorithm of CAH-256. Finally $H^{(n)}$ is the hash result.

The main purpose of padding is for security reason, as used on the MD structure. The other purposes of padding are two-fold: to make the length of a message be a multiple of 256 and to let the message indicate the length of the original message. CAH-256 uses a 64-bit field to specify the length of an unpadded message. Thus messages of up to $(2^{64}-1)$ bits are accepted which is long enough for practical applications.

CAH-256 pads a message by appending a single bit '1' followed by the necessary number of 0-bits until the length of the message is 192 modulo 256. Then it appends to the message the 64-bit field. Three constant vectors, IV, SV, and K which are 256-bit each, are considered. IV and SV are two different fixed bit-strings and K is the first thirty-two bits of the fractional parts of the cube roots of the first sixteen prime numbers such as SHA-256.

The heart of algorithm is a module that consists of processing of 64 rounds. All rounds have the same structure which is composed of XOR operations with the constant K, two CA rule functions and 3-bit shift operation.

In order to design a concrete hash function, we use combinations of a linear group rule and nonlinear non-group rule. A linear group rule provides a collision resistance from present states to next states and a nonlinear non-group rule provides one-way property and nonlinearity. Rule 150 based on periodic boundary condition is only a linear group rule for a message with 256-bit length, and it has a highest dependency from neighborhood in the middle of the whole linear rules. Meanwhile, we choose rule 23 for a nonlinear non-group CA operation since rule 23 provides not only a high nonlinearity but also a special transition form.

Theorem 1 Rule 150 ($s_{i-1} \oplus s_i \oplus s_{i+1}$) based on periodic boundary condition forms group rules for length l where l mod 3 □ 0.

Proof. Let TR, l represent the T matrix for a CA of length l with rule R. Then TR, l for a three-neighborhood periodic boundary CA with a rule R of the form $a_1 s_{i-1}(t) + a_2 s_i(t) + a_3 s_{i+1}(t)$ where $a_1, a_2, a_3 \in \{0, 1\}$, can be written as:

$$T_{R,I} = \begin{bmatrix} a_2 & a_3 & 0 & 0 & 0 & \cdots & \cdots & 0 & a_1 \\ a_1 & a_2 & a_3 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & a_1 & a_2 & a_3 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_3 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 \end{bmatrix}$$

The determinant of this matrix can be recursively expressed as

$$det\ T_{R,\,l} = a_2\, det\ T_{R,\,(l-1)} + a_3 a_1\, det\ T_{R,\,(l-2)}$$

Based on this recursive relation on the characteristic matrix for any CA, the length for which the CA becomes a group CA can be established. By T150, l, a1 = a2 = a3 = 1, the recursive form of the determinant can be found out from the relation as follows:

$$det\ T_{150,\,l} = det\ T_{150,\,(l-1)} + det\ T_{150,\,(l-2)} = det\ T_{150,\,(l-3)}$$

Further, from limiting conditions,

$$det\ T_{150,\,4} = det\ T_{150,\,5} = 1;$$

$$\Rightarrow det\ T_{150,\,6} = 0$$

Extending the recursive relation for higher values of l, the result follows. □

Theorem 2 Rule 23 ($s_{i\,-1} \oplus ((s_{i\,-1} \oplus (\neg s_i)) \vee (s_i \oplus s_{i+1}))$) updates the next states having the same transition probability, 1/2.

Proof. Let the state transition for rule 23 with three neighbors, s_{i-1}, s_i, and s_{i+1} be $f_{23}(s_{i-1}, s_i, s_{i+1})$, then the result from the possible states combinations, 111 to 000, is {00010111}. Suppose that a state of one of neighborhoods is complemented then we obtain the following results: $f_{23}(\neg s_{i-1}, s_i, s_{i+1})$ = {01110001}, $f_{23}(s_{i-1}, \neg s_i, s_{i+1})$ = {01001101}, $f_{23}(s_{i-1}, s_i, \neg s_{i+1})$ = {00101011}. Now we find some specific property among the results that Hamming distances among four strings are exactly 4 of 8-bit string by pairs. It guarantees that the present states via rule 23 would be updated with the same transition probability, 1/2 so that a changed input state makes the next states with a half difference. This property also makes it impossible to find previous bit values from attackers.

Theorem 3 Rule 23 updates the next states having closely zero-one balanced strings.

Proof. Let 1^n0^m be the three neighborhood states, and n and m are the number of 1s and 0s, respectively, where $n + m = 3$ and $0 \le n, m \le 3$, then there exist four different types such as 1^30^0, 1^20^1, 1^10^2 and 1^00^3. If $n \ge 2$ or $m \le 1$, the next state becomes 0, otherwise 1.

The computation can be considered as a 4-step transformation of H_{CAH}. The calculations in each step are done simultaneously on all bits of H. Let $m_0m_1\ldots m_{255}$ denote the bits of $M^{(j)}$ and $h_0h_1\ldots h_{255}$ denote the bits of $H^{(j)}$, an intermediate message value during a round, and $k_0k_1\ldots k_{255}$ denote the bits of constant K and d denotes the number of round. Before starting every rounds, the computation, $h_i = h_i \oplus m_i$ ($0 \le i \le 255$) is preprocessed. The following steps illustrate a single step of the updating function, where $\oplus$ and $\neg$ are a bit-wised XOR operation and NOT operation, respectively.

Step 1. $h_i = h_i \oplus k_i$ ($0 \le i \le 255$)

Step 2. $h_i = h_{i-1} \oplus h_i \oplus h_{i+1}$ ($0 \le i \le 255$)

Step 3. $h_{4i+j} = h_{4i+j-1} \oplus ((h_{4i+j-1} \oplus (\neg h_{4i+j})) \vee (h_{4i+j} \oplus h_{4i+j+1}))$ ($0 \le i \le 63$, $j = d \bmod 4$)

Step 4. h_i = 3bits-circular-left-shift(h_i) ($0 \le i \le 255$)

In Step 1, a 256-bit output result in each round is xored with the constant K. It can disperse the input message even though an input value is repeated characters. The result of Step 1 is computed by linear group rule 150 in Step 2. A linear group CA operation gives the diffusion of the message and blocks the primary collision. In Step 3, a nonlinear non-group rule 23 is applied to a quarter of 256 bits according to the number of round as described in the above algorithm. The nonlinear CA operation is applied to the selected 64 bits for supporting the nonlinearity and zero-one balance. Finally in Step 4, 3-bit circularly left shift operation is applied to every bit of the result of the previous step. Our CA operation is performed based on periodic boundary condition.

4 Security and efficiency evaluation

One thing we know is that the security of hash functions is indeed based on confusion and diffusion. However, it is quite hard to explain a level of confusion and diffusion so that we provide randomness and area-time complexity, and compare with the previous well-known hash functions.

The computation in Step 1 blocks that the CA operation generates repeated patterns on the characteristics of CA operation. The linear group CA in Step 2 generates a distinctive output result according to a different input based on group property so that it blocks a primary collision in the updating function. Our function employed the rule 23 as a nonlinear rule which guarantees a high nonlinearity. We have

Table 1. Comparison of quality tests according to the number of input-bit and the number of random data between the proposed function and SHA-256

# of input-bit	# of random data	Hash scheme	Frequency test	Runup-down test	Max-run test	Diffusion test
64-bit	1.0×10^4	CAH-256	128.09 (8.07)	127.56 (8.03)	8.32 (1.82)	126.19 (8.23)
		SHA-256	127.97 (8.01)	127.45 (8.10)	8.33 (1.85)	127.95 (7.87)
	1.0×10^5	CAH-256	127.99 (7.98)	127.51 (7.99)	8.32 (1.83)	127.31 (8.20)
		SHA-256	127.95 (8.03)	127.47 (7.96)	8.34 (1.84)	128.01 (8.00)
	1.0×10^6	CAH-256	128.01 (7.99)	127.52 (7.98)	8.33 (1.83)	128.30 (8.18)
		SHA-256	128.00 (8.02)	127.50 (7.98)	8.33 (1.83)	128.00 (8.00)
512-bit	1.0×10^4	CAH-256	128.12 (8.04)	127.51 (7.98)	8.31 (1.81)	127.98 (7.96)
		SHA-256	127.96 (8.02)	127.47 (8.03)	8.31 (1.83)	128.01 (8.02)
	1.0×10^5	CAH-256	128.02 (7.99)	127.49 (7.97)	8.34 (1.83)	127.97 (8.01)
		SHA-256	128.00 (7.99)	127.52 (7.97)	8.33 (1.84)	128.00 (7.98)
	1.0×10^6	CAH-256	128.00 (8.00)	127.50 (7.99)	8.33 (1.83)	128.01 (8.00)
		SHA-256	128.03 (8.00)	127.47 (8.00)	8.33 (1.84)	127.98 (7.99)

Frequency test: Return the number of 1's, *Run-up and down test*: Return the number of run-up and run-down
Max-run test: Return the maximum length of run, *Diffusion test*: Return the number of changed bits in output according to changing 1-bit input
The numbers in ( ) describe their standard deviations.

applied rule 23 to a quarter of a block (64 bits) in different bit positions at each round. Thus every bit position is equivalently applied to the nonlinear operation 16 times.
Confusion is caused by the high nonlinearity and constant K based on the repeated structure of the cellular automata mechanism in Step 1 and 3. The nonlinear CA operation in Step 3 can generate 1's from a zero background. On the input of the next iteration, these would give rise to characteristics with high confusion effect. Hence simple difference patterns in $H^{(j)}$ gives rise to a vast amount of possible difference patterns in $H^{(j+1)}$. Each bit of $H^{(j)}$ depends on nearly 4 bits of the previous round by two CA functions and XOR operations after 3-bit shift operation so that the influence increases by 4-bit a round during the continuous 64 rounds. Thus the influence of the first injection of a message bit has spread out over all bits of $H^{(j+1)}$ with same transition probability by the time of the last injection. Hence it satisfies the diffusion property. The actual message bits injection in H is realized to be diffused and confused in subsequent rounds.

In order to compare the efficiency among the schemes, we have chosen SHA-1 and SHA-256 which are known as the best quality hash functions and the current U.S. federal standard. In order to compare the quality of hash function, we made an experiment on several points of view as shown in Table 1. The specified test methods based on randomness test in [6] is suitably determined to examine and compare the quality of hash functions. The results show that both functions have produced good results.

Table 2. Comparison of area and time complexity for proceeding 512-bit message

	SHA-1	SHA-256	CAH-256
Area complexity (tr)	80,592	127,096	31,360
Time complexity (ns)	11,128	10,789	5,146
AT^2 value	$9{,}980\times10^9$	$14{,}794\times10^9$	830×10^9

We are usually trying to find the design that will best satisfy a given set of design requirements when we implement arithmetic unit design. We consider construction simplicity, defined by the number of transistors needed for its construction and the time needed for the signal change to propagate through gates [5].

In terms of area and time complexity in gate level, the best method, as noted in [2], is to evaluate the AT^2 value for each scheme. Area is assumed to be totally contributed by the number of transistors in gates and registers required to compute a find hash result. The cost due to time consists of the delay time of the gates and registers for proceeding a 512-bit input message block. As shown in Table 2, our scheme based on a CA has outstanding complexity compared to the other well-known schemes. Consequently, the proposed

CAH-256 has roughly 12 times and 18 times less AT^2 value than SHA-1 and SHA-256 for proceeding 512-bit message, respectively. The description allows a straightforward chip implementation. Based on parallelism and logical bitwise operation of a CA, our scheme makes extremely high speed possible.

5 Conclusion

This paper has proposed a hardware oriented cryptographic hash function. We conclude that the proposed cryptographic hash function based on a CA has satisfied confusion and diffusion properties and a high randomness that it has produced an excellent quality of message digest in spite of having an exceedingly low construction cost. Therefore, we expect that the proposed function will be efficiently used for preserving the integrity of a potentially large message on hardware implementation.

6 References

[1] M. Bellare and T. Ristenpart, Multi-Property-Preserving Hash Domain Extension and the EMD Transform, Asiacrypt, LNCS 4248 (2006) 299-314.

[2] P. P. Chaudhuri, D. R. Choudhury, S. Nandi and S. Chattopadhyay, Additive Cellular Automata Theory and Applications: Volumn 1, IEEE Computer Society Press (1997).

[3] J. Daemen, R. Govaerts and J. Vandewalle, A Framework for the Design of One-Way Hash Functions Including Cryptanalysis of Damgard's One-Way Function Based on a Cellular Automaton, proceeding of Asiacrypto'91, LNCS 739 (1993) 82-96.

[4] I. B. Damgarrd, A Design Principle for Hash Functions, Crypto, LNCS 435 (1989) 416-442.

[5] D. D. Gajski, Principles of Digital Design, Prentice-Hall International Inc., (1997).

[6] D. E. Knuth, The Art of Computer Programming, Seminumerical Algorithms, Vol. 2 3rd Edition, Addison-Wesley Longman Publishing Co., Inc., (1997).

[7] M. Mihaljecvic, Y. Zheng and H. Imai, A Family of Fast Dedicated One-Way Hash Functions Based on Linear Cellular Automata over GF(q), IEICE Transactions on Fundamentals, E82-A, (1) (1999) 40-47.

[8] J. Nechvatal and S. J. Chang, Workshop Report, The Second Cryptographic Hash Workshop, University of California, Santa Barbara (August, 2006).

[9] R. L. Rivest, The MD4 message-digest algorithm, Crypto, LNCS 537 (1991) 303-311.

[10] J. Touch, "Performance Analysis of MD5, ACM Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication (1995) 77-86.

[11] S. Wolfram, Statistical Mechanics of Cellular Automata, Review of Modern Physics, Vol. 55, (1983) 601-644.

[12] Y. Zheng, J. Pieprzyk and J. Sebery, HAVAL - A One-Way Hashing Algorithm with Variable Length of Output, Auscrypt, LNCS 718 (1993) 83-104.

Wildfire Hazard Mapping Using Cellular Automata

Maria Vittoria Avolio
William Spataro
and Salvatore Di Gregorio
Department of Mathematics
University of Calabria
87036 Rende (CS), Italy
Email:{avoliomv,spataro,dig}@unical.it

Giuseppe A. Trunfio
Department of Architecture, Planning and Design
University of Sassari
P.zza Duomo 6, 07041, Alghero, Italy
Email: trunfio@uniss.it

***Abstract*—Since fuel load is a major factor influencing wildfire risk, the standard approach to build related hazard maps is mainly grounded on land-cover data. However, the risk level is also influenced by other factors interacting nonlinearly, such as wind, fuel moisture, ignition sources and topography. For these reasons, an increasingly used approach for the computation of hazard maps involves the explicit simulation of the fire dynamics. This paper exploits a novel CA model for wildfire simulation to evaluate fire risk within a Monte Carlo approach. The adopted CA model has the ability to provide accurate burned areas, taking much less computing time than a typical vector approach for wildfire simulations. The improved accuracy and efficiency were obtained: (i) relaxing the restriction to a few pre-defined directions of spread, which characterizes most of the techniques for simulating wildfires on a raster space; (ii) using an adaptive time-step duration, which allows for avoiding unnecessary computation. The preliminary tests presented in this paper indicate that the model under study can be a suitable component of a tool for wildfire risk assessment.**

I. Introduction

Every year forest fires cause significant ecological and economic damages and, in many cases, may represent a serious risk to people. For this reason, fire-risk evaluation and in particular fire-risk maps have become widely used in many countries.

Wildfire risk assessment is traditionally conducted on the basis of the fuel load on the area under study. However, the many factors that determine the fire behavior interact nonlinearly to determine the hazard level. For this reason software tools for simulating the wildfire spread are increasingly being used to assess the fire risk [1], [2], [3]. The typical approach consists of carrying out a high number of simulations, under different weather scenarios and ignition locations [1], to generate burn probability and fire intensity maps. Clearly, this require efficient and accurate simulation models.

Most wildfire spread simulators are based on the Rothermel fire model [4], [5], which provides the heading rate and direction of spread given the local landscape and wind characteristics. An additional constituent is usually represented by an elliptical description of the spread under homogeneous conditions (i.e. spatially and temporally constant fuels, wind and topography) [6], [7]. The spread simulation in heterogeneous condition is then given by suitable expansion algorithms, based on some form of space-time discretization and local homogeneity assumption, to automate the application of the elliptical model.

There are two approaches commonly used to simulate the fire propagation across a non-homogeneous landscape.

The first, adopted in many wildfire simulators like the well known FARSITE [8] and PROMETHEUS [9], is a vector approach inspired by the Huygens' wavelet principle in which the fire front is represented as a polygon expanding at specified time steps [10], [11]. Although this approach has proved to be very accurate, it is not well suited in the context of fire risk assessment because of the computational requirements. In fact, it requires computationally expensive de-looping heuristics able to cope with the topological complications which, at each time-step, may affect the fire front [8], [9].

The second type of commonly adopted fire spread algorithm, which can be described in terms of Cellular Automata (CA) [12], expands the burned area directly on a raster space representing the landscape, through a discrete sequence of cells' ignitions driven by the proximity between cells [13], [14], [15], [16], [17], [18]. Cell-based methods do not need problematic de-looping processes and can be highly optimized. As a result, they can perform the same simulations in a fraction of the run time taken by their vector-based counterpart [18]. For this reason cell-based methods represent the ideal approach when it comes to running thousands of simulations to build a risk map.

Nevertheless, a well-known problem associated with the raster approaches is the distortion that may affect the produced fire shape. For example, under homogeneous conditions and in presence of wind, the shape of the heading portion of the fire is often angular rather than rounded as in the expected ellipse [19], [20]. Many authors found that such distortions, which obviously correspond to simulation errors also in operational contexts, can be reduced by increasing the number of neighboring cells influenced by each cell [21], [20], [22]. However, increasing the size of the neighborhood obviously corresponds to higher computational times.

Given the mentioned importance of the availability of accurate and efficient simulators in the context of fire risk assessment, we have designed a new wildfire CA model, which is characterized by low distortion and small run time.

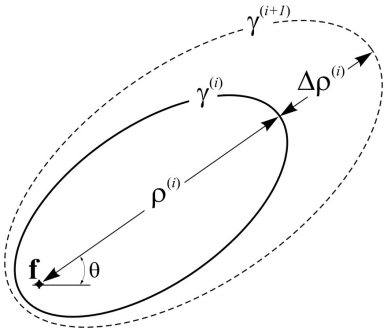


Fig. 1. Growth of the ellipse γ locally representing the fire front. The symbol ρ denotes the forward spread which is incremented by $\Delta\rho$ at the i-th time step.

The improved accuracy was achieved by devising a spreading algorithm in which the fire does not travel only along the few fixed directions imposed by both the lattice and neighborhood. Moreover, the run time efficiency of the model is significantly high thanks to an adaptive time step strategy, which simulates the progression of the fire by avoiding unnecessary computation. In this paper we present a software tool which includes such CA model in order to allow for the fast production of fire risk maps on heterogeneous landscapes.

This paper is organized as follows. The next section outlines the main components of the proposed software tool. In section III we illustrate a preliminary application to the production of a fire risk map. The paper concludes with section IV in which we draw some conclusions and outline future work.

II. An improved CA model for wildfire simulations

In the model object of this study, the two-dimensional fire propagation is simulated through a growing ellipse having the semi-major axis along the direction of maximum spread, the eccentricity related to the intensity of the so-called *effective wind* and one focus acting as a 'fire source'.

At each time step the ellipse's size is incremented according to both the duration of the time step and maximum rate of spread (see Figure 1). Afterwards, a neighboring cell invaded by the growing ellipse is considered a candidate to be ignited by the spreading fire. In case of ignition, a new ellipse is generated according to a heuristics described later.

To ensure the meaningfulness of the simulated fire front, the local ellipse is not allowed to go beyond the nearest cells in a single step. This is accomplished by automatically adapting the size of the time step according to both the size of the cells and maximum rate of fire spread in the whole automaton.

More formally, the model is a two-dimensional CA with square cells defined as:

$$CA = \langle \mathcal{K}, \mathcal{N}, \mathcal{S}, \mathcal{P}, \omega, \Psi \rangle \quad (1)$$

where:

- $\mathcal{K}$ is the set of points with integer co-ordinates in the finite region where the phenomenon evolves. Each point identifies a square cell;
- $\mathcal{N}$ is the set that identifies the pattern of cells influencing the cell state change (i.e the neighborhood);
- $\mathcal{S}$ is the finite set of the states of the cell, defined as the Cartesian product of the sets of definition of all the cell's substates;
- $\mathcal{P}$ is the finite set of global parameters, which affect the transition function and are constant in the overall cellular space. Some relevant parameters in set $\mathcal{P}$ are the current time p_t, the size of the cell's side p_e, the time corresponding to a single CA step $p_{\Delta t}$ and a threshold p_r for the maximum spread below which the ignition of the cell is not activated by the transition function. Additional parameters in $\mathcal{P}$ define the reference values of weather conditions and the fuel models (fuel bed characteristics are specified according to the fuel models used in BEHAVE [23], [24], [25], [26], [27]);
- $\omega : \mathcal{S}^{|\mathcal{N}|} \rightarrow \mathcal{S}$ is the transition function accounting for the fire ignition, spread and extinction mechanisms. It is described in detail in section II-A;
- Ψ is the set of global functions, activated at each step before the application of the transition function ω to modify either the values of model parameters in $\mathcal{P}$ or the cells' substates. Among these, the function ϕ_τ adapts the size $p_{\Delta t}$ of the time step according to both the size of the cells p_e and current maximum spread rate in the whole automaton. The value of $p_{\Delta t}$ is then used by another function, ϕ_t, for keeping the current time p_t up to date. Additional global functions can account for fire fighting interventions or changing of meteorological conditions.

The cell's substates include all the local quantities used by the transition function for modeling the local interactions between the cells (i.e. the fire propagation to neighboring cells) as well as its internal dynamics (i.e. the fire ignition and growth). In particular, among the substates that define the state of each cell, there are:

- the altitude $z \in S_z$ of the cell;
- the fuel model $\mu \in S_\mu$, which is an index referring to one of the mentioned fuel models that specifies the characteristics of vegetation relevant to Rothermel's equations;
- the combustion state $\sigma \in S_\sigma$, which takes one of the values *unburnable*, *not ignited*, *ignited* and *burnt*.
- the rear focus $\mathbf{f} \in S_\mathbf{f}$ of the ellipse locally representing the fire front (see Figure 1); it can be virtually considered the local source of the fire expansion;
- the accumulated forward spread $\rho \in S_\rho$, that is the current distance between the focus $\mathbf{f}$ of the local ellipse and the farthest point on the semi-major axis (see Figure 1);
- the angle $\theta \in S_\theta$ (see Figure 1), giving the direction of the maximum rate of spread. In the context of the semi-empirical Rothermel's approach, such an angle is obtained through the composition of two vectors, namely the so-called *wind effect* and *slope effect* [4], both obtained on the basis of the local wind vector, local terrain slope and fuel model;
- the maximum rate of spread $r \in S_r$, also provided by

```
1   Update the fuel moisture according to the current weather conditions;
2   if ( the current substate σ is ignited ) {
3       if ( c no longer belongs to the fire front ) {
4           Set σ to burnt;
5           return;
6       }
7       Update the local ellipse γ using the current p_Δt;
8   }
9   else
10  if ( the current substate σ is not ignited )
11      for each cell c_i in the neighborhood
12          if ( the substate σ_i of c_i is ignited )
13              if ( the cell is reached by γ_i ) {
14                  Compute r and θ;
15                  if ( r ≥ p_r ) {
16                      Set σ to ignited;
17                      Compute ε;
18                      Compute the focus f and the current local spread ρ;
19                      return;
20                  }
21              }
```

Fig. 2. Outline of the cell's transition function ω. The index i refers to the i-th cell of the neighborhood.

Rothermel's equations on the basis of the relevant local characteristics [4];

– the eccentricity $\varepsilon \in S_\varepsilon$ of the ellipse γ representing the local fire front, which is obtained as a function of both the wind and terrain slope through the empirical relation proposed in [28], [8].

Among the remaining substates are the local wind vector and the relative humidity value of the cell, both provided as external input to the model.

The simulation runs until a predetermined termination criterion is met (e.g. based on the final simulation time). Therefore, since the time duration corresponding to a CA step is dynamically adapted, the number of steps actually required depends on both the cells' size p_e and fire's behavior. In brief, the scheduling of each CA step is organized as follows:

1) first, the global functions in Ψ are executed. In particular, function ϕ_τ computes the current duration of the time step $p_{\Delta t}$ while function ϕ_t updates the current time p_t;
2) afterwards, the transition function ω is executed for each cell of the automaton. This implies computing the fire-front expansion during the time interval p_τ, according to the algorithm described below;
3) finally, if the termination criterion is not met, a new step is executed, otherwise the simulation ends.

A. The transition function

The transition function ω of the current cell $\mathbf{c}$ is outlined in Figure 2. In case of an ignited cell, ω first checks if the fire can continue burning in the cell (line 4). In particular, the condition that triggers the transition to the *burnt* state is verified when all eight cells of the Moore's neighborhood are in the *ignited* state or in the *unburnable* state. Clearly, in this case the cell can be considered as burnt because its contribution is no longer necessary to the fire spread mechanism.

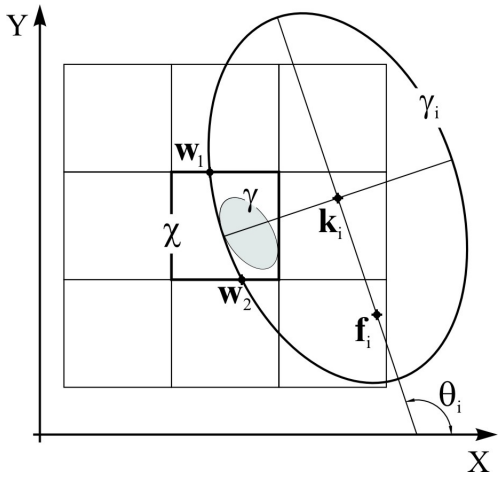


Fig. 3. The central cell intersected by a neighboring ellipse γ_i locally representing the fire front.

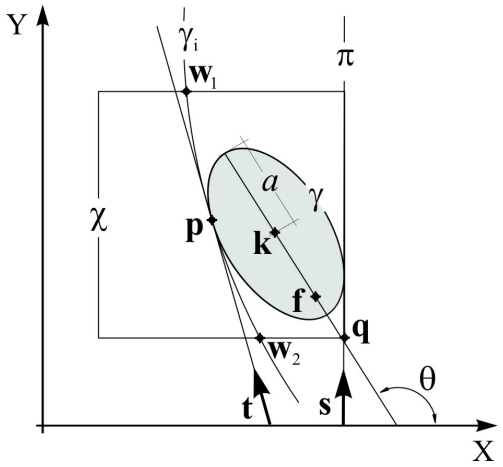


Fig. 4. Entities involved in the computation of the initial ellipse γ on a newly ignited cell according to the proposed heuristics.

The subsequent and final step of the transition function in case of ignited cell (see line 8 of the pseudo-code in Figure 2), and consists of updating the size of the local ellipse γ. This is accomplished by adding the incremental spread $r\,p_{\Delta t}$ to the accumulated forward spread ρ, being the current size of the CA step $p_{\Delta t}$ provided by the global function ϕ_τ.

When the cell $\mathbf{c}$ is in the *not ignited* state, the transition function tests if the fire is spreading towards it from other cells $\mathbf{c}_i$ of the neighborhood that currently are ignited (see Figure 2, line $10-12$). Since the latter state corresponds to a neighboring local ellipse γ_i held by $\mathbf{c}_i$, such a spread test is carried out through the computation of the set $\mathcal{W} = \gamma_i \cap \chi$, where χ is the boundary of $\mathbf{c}$ (see Figure 3). If $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2\}$, with $\mathbf{w}_1 \neq \mathbf{w}_2$, then it is assumed that the fire can spread to $\mathbf{c}$ and the proper Rothermel's equations [4] are used for the computation of both the intensity r of the maximum spread rate vector and its inclination θ (see line 14 in Figure 2).

Then, when the value of r is not below the threshold

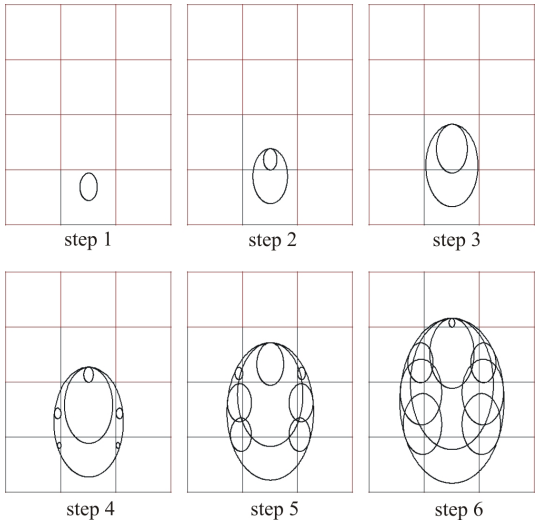


Fig. 5. First steps of fire spreading using the described algorithm in case of $\theta = 0$

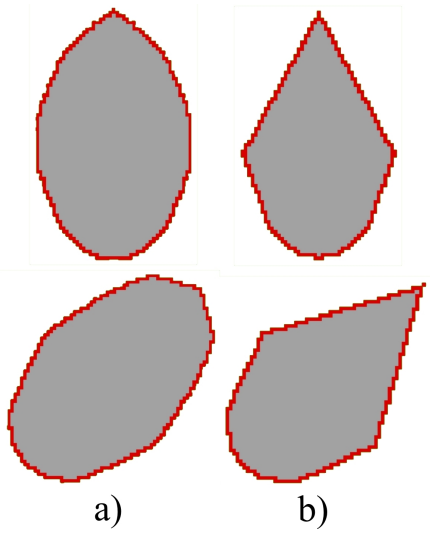


Fig. 6. A comparison between the improved CA (case a) and a typical CA for fire spread simulation (case b). In both cases the standard Moore's neighborhood was used.

p_r, the transition function sets the cell combustion state σ to *ignited* and computes a suitable local ellipse γ locally describing the new fire front (see Figure 2, line $16-18$). One of the characteristics of γ, namely its orientation given by the angle θ, is already known. In addition, its eccentricity ε can be easily computed using one of the empirical formulas that have been developed for relating the eccentricity of the elliptical fire spread to both the wind speed and terrain slope [28], [7], [8]. In particular, in the current implementation the value of ε is computed according to the formula proposed in [28] as modified in [8], which accounts for both the effect of wind and topography through the previously mentioned *effective wind speed* [4]. Thus, two further characteristics are required in order to identify the new ellipse, that is the current local spread ρ, which gives its size, and the focus $\mathbf{f}$, which defines its position (see Figure 1). A suitable heuristics for determining both ρ and $\mathbf{f}$, leading to accurate results at reasonable computational costs, is adopted as follows:

1) the point $\mathbf{p} \in \gamma_i$ located in the middle of the elliptic arc $\widehat{\mathbf{w}_1\mathbf{w}_2} \in \gamma_i$ belonging to the current cell $\mathbf{c}$ (i.e. $|\widehat{\mathbf{w}_1\mathbf{p}}| = |\widehat{\mathbf{p}\mathbf{w}_2}|$) is computed;
2) the line π, containing the cell's side intersected by the line $\mathbf{p}\mathbf{k}_i$, is determined;
3) to describe the local fire front, the ellipse γ tangent to both ellipse γ_i in $\mathbf{p}$ and line π, is computed in terms of the accumulated local spread ρ and focus $\mathbf{f}$;

More details about the algorithm described in this paper can be found in [29].

An example of a sequence of local ellipses generated according to the algorithm described above is shown in Figure 5. As can be seen, when a cell is reached by an ellipse from a neighboring burning cells it is ignited. Then, a new ellipse is generated inside the newly ignited cell.

Note that, although the algorithm illustrated above was originally designed for simulating surface fires, a similar procedure is adopted in this study for simulating crown fires spread following the approach of [8].

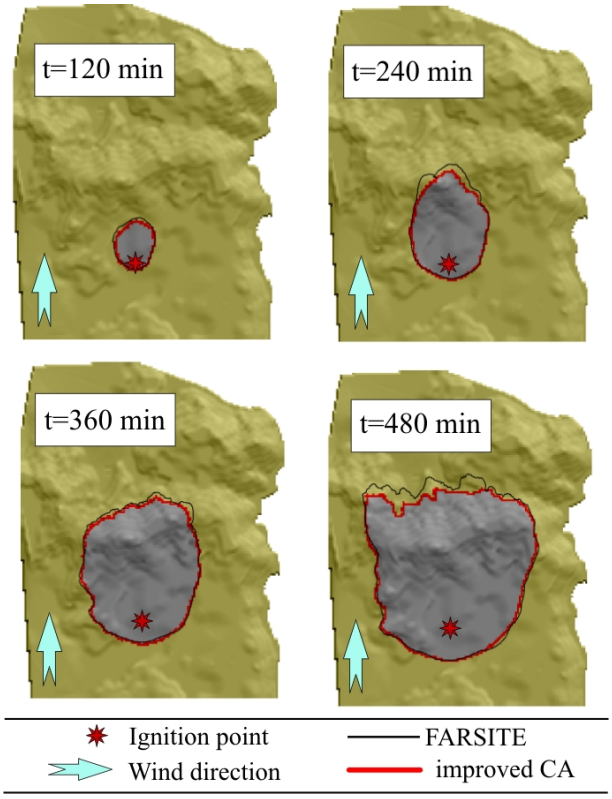


Fig. 7. A comparison between the improved CA and FARSITE (i.e., a widely used simulation tool based on the vector approach) on real landscape. The standard Moore's neighborhood was used for the improved CA.

The main advantage of the approach described above, when compared with a typical CA algorithm for wildfire simulation,

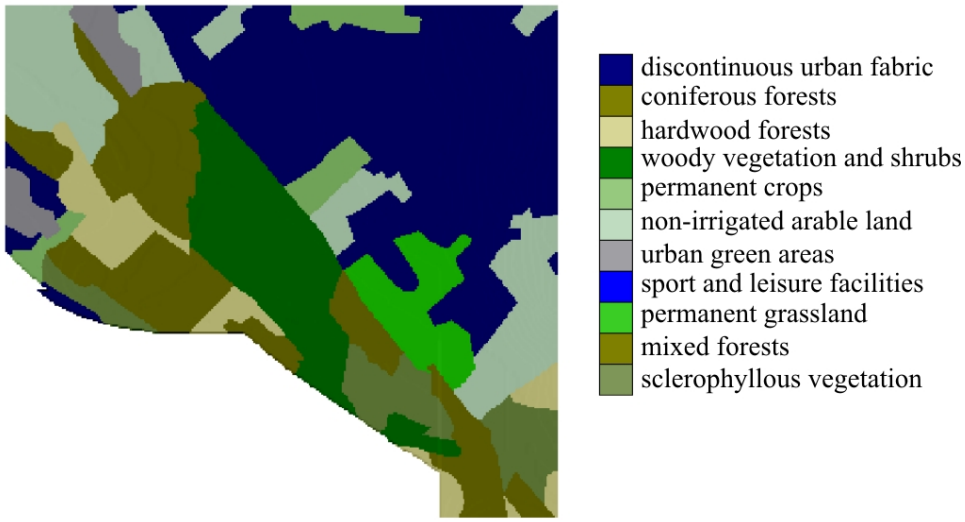


Fig. 8. The area under study for the first example according to the CORINE land-cover data.

lies in its ability to increase the directions of spread. In fact, the latter in case of standard point-to-point fire propagation are restricted to few fixed angles and this causes distortion of simulated fire shapes. This can be seen in Figure 6, where the fire shape given by the algorithm described above is compared with the corresponding burned area simulated using a typical CA. While the standard CA leads to a triangular head of the fire shape, the improved algorithm is able to better reproduce the expected ellipse. This superior accuracy, together with the high run-time efficiency provided in general by the cell-based methods [18], suggests the suitability of the algorithm described above for the risk assessment application object of the next section.

The improved CA was validated by comparison with FARSITE [8] (i.e., a widely used simulation tool based on the vector approach) in a variety of real landscapes under different wind conditions. As can be seen in the example of comparison depicted in Figure 7, it provides burned regions which are equivalent, for practical purposes, to those given by FARSITE. In addition, the regions affected by fire are obtained at a far more convenient computational cost by cell-based methods as shown in [18].

III. An application to wildfire risk mapping

Since fuel load is a major component of fire risk, related hazard maps are traditionally constructed using only classified vegetation cover. However, the risk level is also influenced by other factors interacting nonlinearly, such as wind, fuel moisture, ignition sources and topography. Also, most of these factors are random variables and this further affects the fire risk. For these reasons, an increasingly used approach for the computation of hazard maps involves the explicit simulation of the fire dynamics [1], [2], [3].

In brief, the method consists of a Monte Carlo approach in which a high number of different fire spread simulations are carried out, sampling from suitable statistical distributions the random variables relevant to the fire behavior. At the end, the local risk is computed on the basis of the frequency of burning. Clearly, the simulation model underlying such an approach should be fast and accurate. This suggests to adopt the improved CA illustrated above as the simulation engine of a software tool for wildfire risk mapping. The latter has been named SVAMPAU, which is the acronym of 'Simulating through a VAlidated Model fire Propagation by cellular AUtomata" and means *blazed* in Sicilian.

The technique for computing hazard maps adopted in this study is based on a prefixed number n of simulation runs, where each run represents a single simulated fire. Ignition locations, wind direction and the durations of the fire are selected randomly from uniform distributions. In particular, the wind direction is selected in a range corresponding to the typical directions of severe wind for the area. Also the fire duration is selected considering the duration of historical fires in the regions under study. All the other relevant characteristics are kept constant during the simulations.

Once the latter have been carried out, the resulting n maps of burned areas are overlaid and cells' fire frequency are used for the computation of the fire risk. In particular, a *burn probability* $p_b(\mathbf{c})$ for each cell $\mathbf{c}$ is computed as:

$$p_b(\mathbf{c}) = \frac{f(\mathbf{c})}{n}; \tag{2}$$

where $f(\mathbf{c})$ is the number of times the cell $\mathbf{c}$ is ignited during the n simulated fires. The burn probability for a given cell is an estimate of the likelihood that a cell will burn given a single random ignition within the study area and given the assumed burn conditions.

In the preliminary application presented here two test cases are discussed. The first example concerns the area of the urban park *Pineta Castel Fusano*, a protected area near Rome, Italy. It includes a pinewood which covers an area of 916 hectares

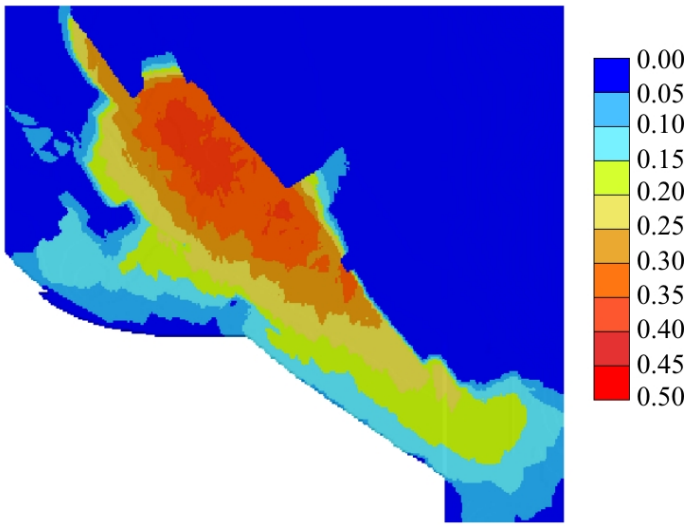


Fig. 9. The risk map obtained for the area depicted in Figure 8

and it is the largest green area of Rome. The pine forest is often affected by fires, some of which have caused relevant environmental damages. For example, on July 4, 2000, 300-350 hectares of pine forest and Mediterranean evergreen were hit by fire, of which 280 hectares were completely destroyed. Other serious fires that have decimated hectares of *Pinus pinea* occurred on 9 July 2002, from June to September 2003, on 11 July 2004 and on July 1, 2005. Also, in July 2008, at least another 80 hectares of pine forest were destroyed by several wildfires.

The landscape was modeled through a Digital Elevation Model composed of 313×288 square cells with side of $20\,m$. In the area, the terrain is relatively flat with an altitude above sea level ranging from 2 to 20 m. The heterogeneous fuel bed, depicted in Figure 8, was based on the use of the 1:25000 land cover map from the CORINE project (EEA 2002). The CORINE land-cover codes were mapped on the standard fuel models used by the CA model (i.e., the substate μ). Plausible values of fuel moisture content were obtained from literature data. Also, a domain-averaged open-wind vector for 10 different directions in the range South-West - South-East, having an intensity of $20\,km\,h^{-1}$, was used for producing time-constant gridded winds through WindNinja [30], a computer program that simulates the effect of terrain on the wind flow. The range of fire durations were randomly sampled between $1\,h$ and $6\,h$.

To produce the hazard map represented in Figure 9, $n = 500$ simulations were carried out. The task took about $2.6\,h$ using a laptop equipped with a 1.83GHz Core Duo T2400 processor. As can be seen in Figure 9, interestingly the higher fire risk was obtained in the coniferous forest contiguous to the area with woody vegetation and shrubs. This can be explained considering that shrub vegetation is usually more flammable than other vegetation types and that the wind was blowing from the shrubland towards the coniferous forest.

To give an example of the influence that the topography may have on the final risk map, a second example was considered. In particular, the Digital Elevation Model represented in Figure 10, composed of 131×167 square cells with side of $20\,m$, was used as input landscape for the risk assessment tool. In this case, the altitude above sea level ranging from 20 to 220 m. The fuel bed of the whole landscape was modeled as uniform and corresponding to the standard fuel model 2 (i.e., timber grass and understory) [25]. Also, a wind vector having an intensity of $20\,km\,h^{-1}$ was used. The wind directions were sampled randomly in the interval between South - East and North - East and for each direction a time-constant gridded wind was produced through WindNinja [30]. The range of fire durations were randomly sampled between $1\,h$ and $8\,h$.

Also in this case, to produce the hazard map $n = 500$ simulations were carried out. As can be seen in Figure 10, the topography greatly influenced the risk map. In particular, it is important to point out that the terrain slope and aspect significantly shaped the wind field. Then, the variable wind vector, again with the terrain slope and aspect, significantly affected the behavior of each simulated fire. It is also worth to note that using only the fuel load to quantify the hazard would have led to a substantially uniform fire risk throughout the whole area. This highlights the effectiveness of the production of risk maps based on the fire dynamics simulation.

IV. Conclusion and Future Work

We have illustrated some characteristics of SVAMPAU, a software tool which exploits a new CA simulation model to assess fire hazard within a Monte Carlo approach. The adopted CA model has the ability to provide accurate burned areas, taking much less computing time than a typical vector approach for wildfire simulations. The improved accuracy and efficiency were obtained: (i) relaxing the restriction to a few pre-defined directions of spread, which characterizes most of the techniques for simulating wildfires on a raster space;

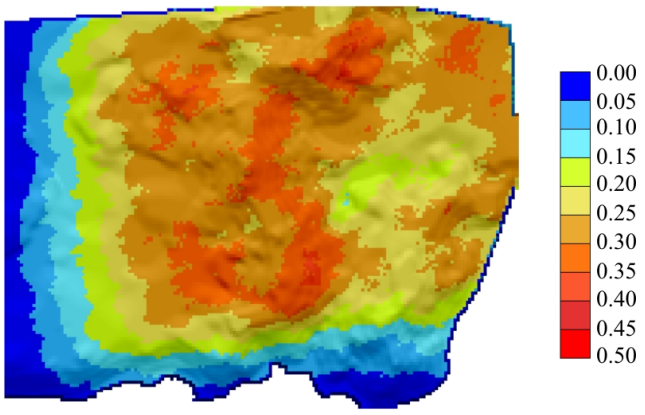


Fig. 10. The risk map obtained for the area depicted in Figure 7

(ii) using an adaptive time-step duration, which allows for avoiding unnecessary computation (i.e., at each step at least one new cell is ignited).

The preliminary tests presented in this paper suggest that the model under study can be a suitable component of a tool for assessing the wildfire risk.

Future work will focus on parallelizing the Monte Carlo phase according to a simple master-slave approach, in order to allow much shorter run time and the opportunity to study areas of greatest extension, which require a higher number of simulations.

ACKNOWLEDGMENT

This research was partly funded by ASI (Italian Space Agency) project 'SARFIRE: Spaceborn SAR imagery and environmental data fusion for the dynamical evaluation of land regions susceptibility to fire' - Agreement: I/049/09/00 - Project ID: 2288.

REFERENCES

[1] Y. Carmel, S. Paz, F. Jahashan, and M. Shoshany, "Assessing fire risk using monte carlo simulations of fire spread," *Forest Ecology and Management*, vol. 257, no. 1, pp. 370 – 377, 2009.

[2] A. A. Ager, N. M. Vaillant, and M. A. Finney, "A comparison of landscape fuel treatment strategies to mitigate wildland fire risk in the urban interface and preserve old forest structure," *Forest Ecology and Management*, vol. 259, no. 8, pp. 1556 – 1570, 2010.

[3] A. Ager and M. Finney, "Application of wildfire simulation models for risk analysis," in *Geophysical Research Abstracts, Vol. 11, EGU2009-5489, EGU General Assembly*, 2009.

[4] R. C. Rothermel, "A mathematical model for predicting fire spread in wildland fuels," U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT, Tech. Rep. INT-115, 1972.

[5] ——, "How to predict the spread and intensity of forest and range fires," U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT, Tech. Rep. INT-143, 1983.

[6] C. Van Wagner, "A simple fire growth model," *Forestry Chron.*, vol. 45, pp. 103–104, 1969.

[7] M. Alexander, "Estimating the length-to-breadth ratio of elliptical forest fire patterns," in *Proc. 8th Conf. Fire and Forest Meteorology*, 1985, pp. 287–304.

[8] M. A. Finney, "FARSITE: fire area simulator-model development and evaluation," U.S Department of Agriculture, Forest Service, Tech. Rep. RMRS-RP-4, 2004 February 2004.

[9] C. Tymstra, R. Bryce, B. Wotton, S. Taylor, and O. Armitage, "Development and structure of Prometheus: the canadian wildland fire growth simulation model," Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Edmonton, Alberta, Tech. Rep. NOR-X-417, 2010.

[10] J. Sanderlin and J. Sunderson, *A simulation for wildland fire management planning support (FIREMAN). Vol. 2. Prototype models for FIREMAN (Part II): Campaign fire evaluation. Mission Research Corp. Contract 21-343, Spec. 222.*, 1975.

[11] D. Anderson, E. Catchpole, N. DeMestre, and T. Parkes, "Modeling the spread of grass fires," *J. Aust. Math. Soc. (B)*, vol. 23, pp. 451–466, 1982.

[12] J. von Neumann, *Theory of self reproducing automata*. University of Illinois Press, Urbana, 1966.

[13] P. H. Kourtz and W. G. O'Regan, "A model for a small forest fire to simulate burned and burning areas for use in a detection model," *Forest Science*, vol. 17, no. 7, pp. 163–169, 1971.

[14] D. G. Green, "Shapes of simulated fires in discrete fuels," *Ecological Modelling*, vol. 20, no. 1, pp. 21 – 32, 1983.

[15] A. M. G. Lopes, M. G. Cruz, and D. X. Viegas, "Firestation - an integrated software system for the numerical simulation of fire spread on complex topography," *Environmental Modelling and Software*, vol. 17, no. 3, pp. 269–285, 2002.

[16] G. A. Trunfio, "Predicting wildfire spreading through a hexagonal cellular automata model," in *ACRI*, ser. LNCS, P. M. A. Sloot, B. Chopard, and A. G. Hoekstra, Eds., vol. 3305. Springer, 2004, pp. 385–394.

[17] S. Yassemi, S. Dragicevic, and M. Schmidt, "Design and implementation of an integrated GIS-based cellular automata model to characterize forest fire behaviour," *Ecological Modelling*, vol. 210, no. 1-2, pp. 71–84, 2008.

[18] S. H. Peterson, M. E. Morais, J. M. Carlson, P. E. Dennison, D. A. Roberts, M. A. Moritz, and D. R. Weise, "Using HFIRE for spatial modeling of fire in shrublands," U.S. Department of Agriculture, Forest Service, Pacific Southwest Research Station, Albany, CA, Tech. Rep. PSW-RP-259, 2009.

[19] I. French, D. Anderson, and E. Catchpole, "Graphical simulation of bushfire spread," *Mathematical Computer Modelling*, vol. 13, pp. 67–71, 1990.

[20] I. French, "Visualisation techniques for the computer simulation of bushfires in two dimensions," Master's thesis, University of New South Wales - Australian Defence Force Academy. Dept. of Computer Science, 1992.

[21] W. G. O'Regan, "Bias in the contagion analog to fire spread," *Forest Science*, vol. 22, 1976.

[22] W. Cui and A. H. Perera, "A study of simulation errors caused by algorithms of forest fire growth models," Ontario Forest Research Institute, Tech. Rep. 167, 2008.

[23] F. A. Albini, "Estimating wildfire behavior and effects," U.S Department of Agriculture, Forest Service, Tech. Rep. INT-30, 1976.

[24] H. E. Anderson, "Aids to determining fuel models for estimating fire behavior," U.S. Department of Agriculture, Forest Service, Tech. Rep. INT-122, 1982.

[25] P. Andrews, "BEHAVE: fire behavior prediction and fuel modeling system - burn subsystem, part 1," U.S Department of Agriculture, Forest Service, Tech. Rep. INT-194, 1986.

[26] R. E. Burgan, "Concepts and interpreted examples in advanced fuel modeling," U.S. Department of Agriculture, Forest Service, Tech. Rep. INT-238, 1987.

[27] R. E. Burgan and R. Rothermel, "BEHAVE: Fire behavior prediction and fuel modeling system - fuel subsystem," U.S. Department of Agriculture, Forest Service, Tech. Rep. INT-167, 1984.

[28] H. Anderson, "Predicting wind-driven wildland fire size and shape," U.S Department of Agriculture, Forest Service, Tech. Rep. INT-305, 1983.

[29] G. A. Trunfio, D. D'Ambrosio, R. Rongo, W. Spataro, and S. Di Gregorio, "A new algorithm for simulating wildfire spread through cellular automata," *Submitted*.

[30] J. Forthofer, K. Shannon, and B. Butler, "Simulating diurnally driven slope winds with windninja," in *Proceedings of 8th Symposium on Fire and Forest Meteorological Society - Kalispell, MT*, 2009.

FPGA Implementation of a Bioinspired Model for Real-Time Traffic Signals Control

Georgios Kalogeropoulos, Georgios Ch. Sirakoulis* and Ioannis Karafyllidis
Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, GREECE

Abstract - *During the last decades, traffic congestion in urban networks is getting worse affecting many aspects of the residents lives to an increasing extent. Traffic lights play a decisive role in the aforementioned traffic networks of modern metropolises, and the existing conditions of the corresponding vehicular traffic flows. In order to develop an efficient system dedicated to the real-time traffic signals control for which the hardware implementation will be straightforward, Cellular Automata (CAs) were chosen as the simulation and implementation method. Despite its ease of implementation and simplicity, CAs is a powerful tool that can generate realistic traffic models. In this paper, a Cellular Automaton (CA) model was implemented on a FPGA to take full advantage of the inherent parallelism of CAs and provide real-time traffic signals control in accordance with vehicular traffic flow. The proposed hardware was optimized and the resulting single FPGA processor can be finally considered as basic component of an advanced electronic system able to provide real time information concerning the traffic conditions in the under study intersections and thus to efficiently handle-control the traffic signals in real conditions.*

Keywords: Traffic Signals; Cellular Automata; FPGA; Real-time control.

1 Introduction

Metropolitan centers everywhere are battling an increase in demand and an inability to build sufficient infrastructure to cope with the huge traffic congestion increment. Consequently, in recent decades the traffic flow problem of urban networks has drawn the attention of specialists in various fields including physics, mechanics and mathematics. However, improving transportation systems is about more than just adding road lanes, transit routes, sidewalks and bike lanes. It is also and mostly about operating those systems efficiently. Not only does congestion cause slow speeds, it also decreases the traffic volume that can use the roadway; stop-and-go roads only carry half to two-thirds of the vehicles as a smoothly flowing road [1]. On the other hand, it is well known that in urban street networks, the flow of vehicles is almost entirely controlled by traffic lights and traffic engineers are often forced to question if the capacity of the network is exploited by the chosen control strategy. Consequently by choosing signal control schemes one has a large impact on average fuel consumption and travel times. Respectively, the development of efficient traffic signals control models may be very beneficial when studying various kinds of city networks, even those with a more sophisticated topology.

Regarding the traffic models so far, for almost half a century, there were strong attempts to develop a theoretical framework of traffic science. It is stated that the movement of vehicles can be considered as an example of a self-driven many-particle system driven far from equilibrium [2]. The resulting traffic model can be analyzed both macroscopically and microscopically [3]. In the former, attention is paid among others to fluid dynamical models such as kinematic waves, incompressible Navier-Stokes-like momentum equations as well as to Gas-kinetic models (Boltzmann equations). These models are often suitable for analytical investigation, ensure simple treatment of inflows and enable simulations of several lanes by effective one-lane models with certain probabilities of overtaking. On the other hand, in microscopic models, which are more usually used, each individual vehicle is represented by a particle and the interactions among the particles depend on the way the vehicles influence each other. It should be also noticed that the microscopic models, at least in theory, can be used to study macroscopic properties of traffic streams [4]. Furthermore, in the macroscopic models, traffic flow is viewed as a compressible fluid formed by vehicles that do not appear explicitly in the theory [5]. In contrast, in the microscopic models, traffic is treated as a system of interacting particles where attention is explicitly focused on individual vehicles and the interactions among them. These models are therefore much better suited for the investigation of the under study urban traffic networks.

Microscopic models include follow-the-leader car models in which it is assumed that the acceleration is determined be vehicles in front of the driver, and Cellular automata (CAs) models in which each vehicle is represented by an occupied cell in a CA model. The main advantage of the aforementioned models based on CA programming paradigm and being developed for the last two decades, was an efficient and fast performance when used in computer simulations. This alternative arrives from the fact that in general, CAs are very effective in simulating systems and solving scientific problems, because they can capture the essential features of systems where global behavior arises

from the collective effect of simple components which interact locally [6]. As a result, in 1992, Nagel and Schreckenberg proposed a CA model, the well-known NaSch model of road traffic, that was able to reproduce several characteristics of real-life traffic flows [7]. Some more refined models are derived from NaSch model, such as Takayasu and Takayasu deterministic model [8], based on the CA-184 Wolfram rule [9], the Benjamin, Johnson, and Hui (BJH) model [10], as well as the Nishinari et al. model [11] all using slow-to-start rules. Urban network traffic flow theory is represented by the Biham-Middleton-Levine (BML) [12] model of city traffic, which in opposition to the previous models for one-dimensional highway traffic was introduced as a simple two-dimensional square lattice CA model. Furthermore, because of the effect of traffic lights, urban roads as well as the corresponding CA models have special characteristics [3]. Guoging et al. extended the BML model by revisiting the regulation on traffic lights [13] and rearranging the BML rules [14]. The study and improvement of network efficiency has aroused till today much concern [15] while modern optimization techniques have been used for the parameterization of CA rules [16]. For an extended and detailed review on CAs for road traffic models see also [17].

Among several CA-based mobility models for urban traffic, the model of Wei et al. [18] is the most similar to our work. The most intriguing part of the aforementioned model is the fact that used the original characteristics of the CA microscopic model to successfully describe and handle macroscopic properties of traffic streams resulting in an almost macroscopic CA model. However, the proposed here model provides some new features to cope better with some of the previous model limitations; for example, simulating traffic over a period of time with random values of a distribution simulating incoming boundary traffic does not guarantee that every possible traffic scenario will be tested, or the original implementation uses a Poisson distribution for the incoming boundary directions flow pressure values although sounds reasonable do not provide realistic situations of vehicular traffic flows near the studied intersections. On the other hand, the presented CA model is characterized by as much as low complexity as possible so that the computational recourses are kept low while its computation speed is kept high without losing any of the requested essence of complexity regarding the real-time signals control of urban networks intersections. Furthermore, one of the most pronounced features of the introduced model is because of the inherent parallelism of CAs, the proposed model is hardware implemented with the help of Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) synthesizable code in order to speed up the application of CAs to the real-time traffic signals control. It should be mentioned that CAs are one of the computational structures best suited for hardware realization. The CAs architecture offers a number of advantages and beneficial features such as simplicity, regularity, ease of mask generation, silicon-area utilization, and locality of interconnections. In this paper, the design processing of the finally produced VHDL code, i.e. analysis, elaboration and simulation, has been checked out with the help of the Quartus II, v. 9.0® design software of the ALTERA® Corporation. The proposed hardware was optimized and the resulting single FPGA processor can be considered as basic component of an advanced electronic system able to provide real time information concerning the traffic conditions in the under study intersections and thus to handle-control the traffic signals in real conditions. As a result, the proposed FPGA design could serve as the basis of a support decision system for monitoring train movement in real-time, providing valuable near optimum control services.

In the length of this paper, details about the preliminaries of the CAs function and the proposed CA model are found in Section II. Details about the FPGA architecture of the presentation model and its automation design procedure as well as the corresponding hardware simulation results are discussed in section III. Finally, the conclusions are drawn in section VI.

2 CA model

2.1 Cellular Automata Preliminaries

Cellular Automata (CAs) are models of physical systems, where space and time are discrete and interactions are local [9]. Prior and more recent works proved that CAs are very effective in simulating physical systems and solving scientific problems and they can easily handle complex boundary and initial conditions, inhomogeneities and anisotropies [6]. Moreover, the CA approach is consistent with the modern notion of unified space–time. In computer science, space corresponds to memory and time to processing unit. In CA, memory (CA cell state) and processing unit (CA local rule) are inseparably related to a CA cell. As a result, CAs have also been successfully used as a VLSI architecture [19]. Furthermore, for readability reasons, in this section a more formal definition of a CA will be presented in order to help the reader to follow up with the presented approach. In general, a CA requires:

A regular lattice of cells covering a portion of a d-dimensional space;

A set $\boldsymbol{C}\left(\vec{r},t\right)=\left\{C_1\left(\vec{r},t\right),C_2\left(\vec{r},t\right),\ldots,C_m\left(\vec{r},t\right)\right\}$ of variables attached to each site $\vec{r}$ of the lattice giving the local state of each cell at the time $t = 0, 1, \ldots$;

A rule $R=\{R_1, R_2, \ldots, R_m\}$ which specifies the time evolution of the states $\boldsymbol{C}\left(\vec{r},t\right)$ in the following way: $C_j\left(\vec{r},t+1\right)=R_j\left(\boldsymbol{C}\left(\vec{r},t\right),\boldsymbol{C}\left(\vec{r}+\vec{\delta}_1,t\right),\boldsymbol{C}\left(\vec{r}+\vec{\delta}_2,t\right),\ldots,\boldsymbol{C}\left(\vec{r}+\vec{\delta}_q,t\right)\right)$ where $\vec{r}+\vec{\delta}_k$ designates the cells belonging to a given neighborhood of cell $\vec{r}$.

2.2 CA cell model and time step evolution

In the proposed CA traffic model each traffic intersection is regarded as a cell in an urban traffic signal network. As shown in Fig. 1, each intersection is modeled as a cell with a Von-Neumann neighborhood domain. Each approach is appointed a number from 1 to 4 in a clockwise order, with 1 being the west approach of the intersection. All intersections can then be put in a two-dimensional array with size *N*x*M*, where *N* is the number of rows with each row representing an intersection and *M* the number of columns where each column represents the corresponding approach and their value of the resulting flow pressure (FP), respectively.

Regarding the time evolution of the CA cell to its next state it must first gather information about the state of its neighbor cells; in this case the flow pressure of each direction. This phase is called the state perception phase. At this point, it should be made clear that in the considered network, all streets are equal in respect to the processes at intersection, in other words no streets or directions are dominant. Furthermore, it should be also noticed that the free flow phase of the proposed method is an artifact of the periodic boundary conditions and of the fact that no vehicle turns. In more realistic situations, if an intersection is blocked by e.g. an accident, the method would not allow the blockage to spread to other intersections by blocking flow into the affected intersection. Finally, pedestrians are not considered in the presented model and the critical time for safe crossing of the under study intersection, at least a period of 20 seconds, should be accordingly implemented in the model manually by the user by fine tuning the corresponding parameter. As a result, traffic light periods for all streets (intersections) are assumed to be equal in the following.

Assuming sensors measure the flow pressure taking continuous values that range from 0 to 1, according to the following expression:

$$FP = \frac{N \times S}{L \times K} \tag{1}$$

where *N* is the cars number, *S* the average length of the car, *L* the measurable road length and *K* the number of lanes of each approach. The flow pressure readings are then inserted into the intersection array so that each cell has access to the states of its neighbors. As a result, the state of an intersection would be finally given by the following equation:

$$P_i^{t+1} = R\left(P_i^t, P_{i1}^t, P_{i2}^t, P_{i3}^t, P_{i4}^t\right) \tag{2}$$

where *t* stands for the time step evolution and *R* denotes the state CA rule. In general, the same rule with corresponding directions can be applied to every intersection. More

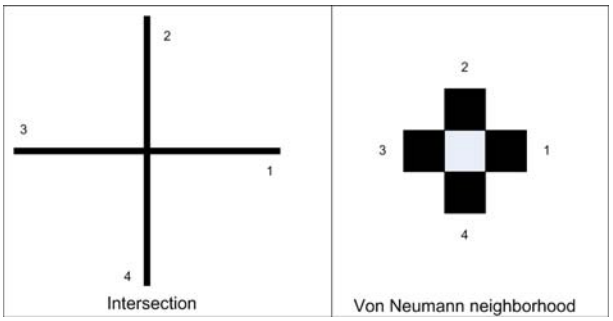


Figure 1. Von Neumann neighborhood cell

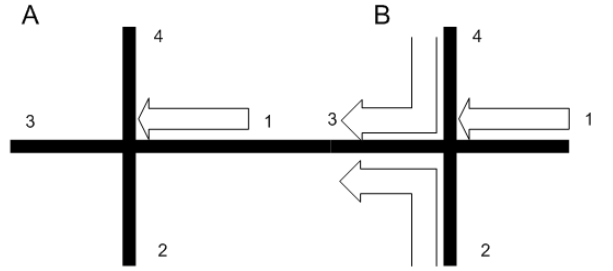


Figure 2. Intersection Flow Pressure

specifically, the outline of the proposed rule can be summarized to the following.

- *If* the normalized sum of the rest three neighbors and potential directions [taking into account for each of them, their impact to the final decision (through, left and right, respectively)] is greater that the sum of the specific direction flow of the under study intersection with the minimum flow increment, i.e. 0.1
- *Then* this direction flow will be raised by the aforementioned minimum flow increment, namely 0.1
- *Else* it will not change during this time step.

In the above rule description, let us name W_k the total impact normalization coefficient to the local intersection after *k* normalizations that have passed according to the function:

$$W_k = (10 - k) \times \left(\frac{0.5}{10}\right) + 0.5 \tag{3}$$

As a result, the state perception phase results then from different rounds of normalization in order to proceed to next time step and evolve each CA cell state. At this point it should be also mentioned that each of the three possible neighbor directions contributes by the corresponding coefficients, respectively. For example, as shown in Fig. 2, in order the intersection A cell to update its state all eastern flow pressure from intersection B must be taken into account and in this case, *w*0, *w*1 and *w*2 would be the resulting impact coefficients of each direction (straight, left and right turn, respectively) with their sum always equal to 1.

3 CA hardware implementation and simulation results

3.1 VHDL implementation

As mentioned before, in terms of circuit design and layout, ease of mask generation, silicon-area utilization, and maximization of clock speed, CAs are perhaps one of the most suitable computational structures for VLSI realization [19]. More specifically, from circuit designing point of view, there are four main factors that determine the cost/performance ratio of an integrated circuit, namely, circuit design and layout, ease of mask generation, silicon-area utilization, and maximization achievable clock speed; for a given technology, the latter is inversely proportional to the maximum length of the signal paths. CA circuit design reduces to the design of a single, relatively simple cell and layout is uniform. The whole mask for a large CA array (the cells with their internal connections as well as the interconnection between cells) can be generated by a repetitive procedure so no circuit area is wasted on long interconnection lines and because of the locality of processing, the length of critical paths is minimal and independent of the number of cells.

The proposed algorithm consists of two stages as shown in the diagram (Fig.3). The first one is the state perception phase which was described earlier. After this phase is complete, the signaling phase begins. Certain rules are applied for the traffic signal control. In order of descending priority these are:

- Change the signal to green if red is shown to this direction for RED_M consecutive times.
- Change the signal to green to the direction in which the previous signal was red and has the highest flow pressure.
    - The flow pressure must be at least Red_t.
- If the signal is green for the Green_Max consecutive time in the same direction then change the signal to red.
- If none of the above applies then change the signal to green to the approach with the highest flow pressure.

All variables used (RED_M, Red_t, Green_Max) are preset. In order to increase the functionality of the presented CA model two more stages to the original diagram were added. The first one refers to the initialization of the CA model in

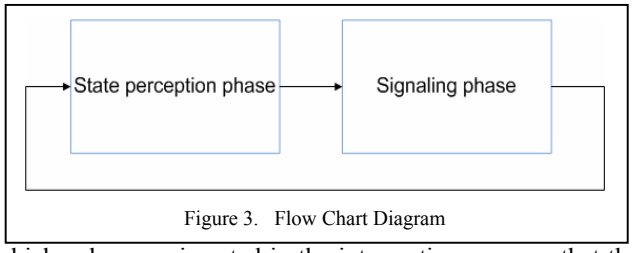


Figure 3. Flow Chart Diagram

which values are inserted in the intersection array so that the simulation does not begin with an empty array, which would mean no traffic at all in every intersection. The second one is the usage of a linear feedback shift register (LFSR) which is responsible for the inbound traffic simulation coming through the boundary directions, based on pseudo-random values in order to represent the flow pressure of the boundary directions. In general, some of improvements found in the proposed model can be summarized as follows. The signaling computations were done according to intervals, where each interval equals the time needed for calculating the next CA cells states and the appropriate signaling values without constraining them in time. If needed a maximum time constraint can be easily added. Furthermore, the original implementation uses a Poisson distribution for the incoming boundary directions flow pressure values whereas in the FPGA a Linear Feedback Shift Register is used with pseudorandom values to simulate the incoming traffic and in order to provide realistic situations of vehicular traffic flows near the studied intersections. More over, simulating traffic over a period of time with random values of a distribution emulating incoming boundary traffic does not guarantee that every possible traffic scenario will be tested. So, in order to check every possible signaling rule, signals were carefully selected so that every rule and possible traffic situation can be tested and handled appropriately. Initially, the width of the road and car volume was passed to the software to calculate the flow pressure values. However, in the presented FPGA implementation the used sensors automatically calculate the flow pressure (FP) values and reduce it accordingly. Finally, the flow pressure takes continuous values that range from 0 to 1, according to eq. (1).

3.2 VHDL implementation

The implementation of the algorithm was developed with VHDL using the Altera Quartus II software. The schematic design of the corresponding CA cell is depictured in Fig. 5. The parameters used in the development and simulations of the code were in accordance with the ones found in [17]. More specifically, the number of intersections was set to 6 and arranged as shown in Fig. 6. The initial flow

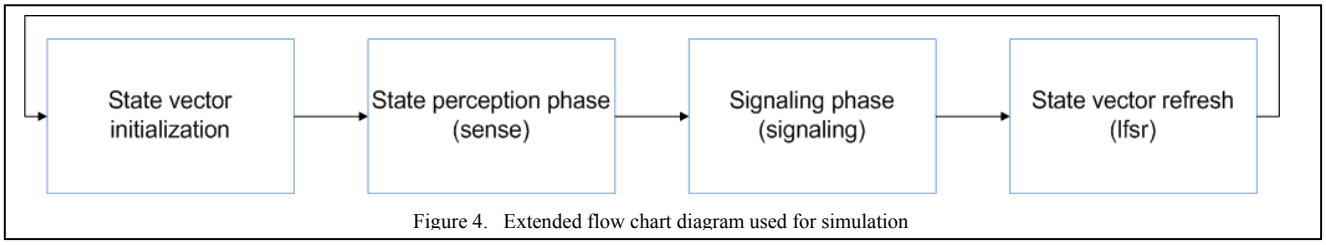


Figure 4. Extended flow chart diagram used for simulation

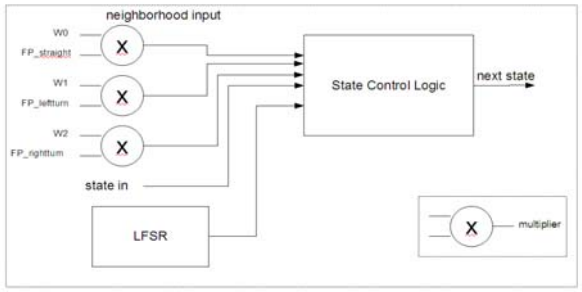


Figure 5. Block diagram of the CA cell architecture

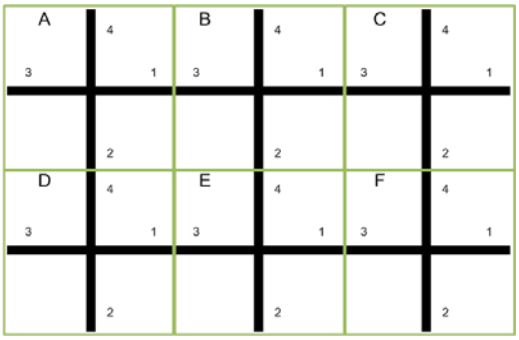


Figure 6. Intersection Layout

pressure values in the intersection array were chosen randomly on a scale from 0 to 100. The boundary approaches flow pressures are attributed through a 4-bit linear feedback shift register (LFSR), while the values of signals RED_M, Red_t and Green_Max are set to 8, 60 and 2 respectively. Furthermore, highest priority is given to the east priority with descending priority given to the remaining directions in a clockwise order. As a result, if the rule for Red_t flow pressure applies then the order is counterclockwise. Finally, the cell state evolving generations is set to 10 and the values selected for impact coefficients *w*0 is 0.8 and for *w*1 and *w*2, respectively.

The state perception phase results after simulation in Quartus II are presented in Fig. 7. The red squares indicate the changed values from one step to the next one. It is noted that through generations 5 to 10 no changes to the flow pressure are made due to the relatively small number of intersections which causes the vectors to stabilize to their values earlier. More specifically, based on the described CA rules, the Quartus II simulation for intersections A and B is shown in Fig.8. (Ax_IN and Axse represent the inputs and outputs respectively). After all the normalizations of the state perception phase are complete a signal is raised in order for the traffic signaling phase to begin (output "signaling" in Fig. 8). The simulation results of the signaling rules in intersection A are described below. The Ax_IN inputs represent the intersection's directions flow pressures, the Ax_IN_R and Ax_IN_G equal to the number of consecutive red and green signals respectively and the Ax_R and Ax_G the red and green signals on the intersection. We can observe that the input A3_IN_R equals 8 which is the threshold for the

Generation 1	1	2	3	4	Generation 2	1	2	3	4
A	10	20	50	0	A	20	30	50	0
B	30	40	10	0	B	30	40	20	0
C	10	50	10	20	C	10	50	10	20
D	30	50	10	10	D	30	50	10	10
E	20	20	40	40	E	20	20	40	40
F	0	20	40	50	F	0	20	40	50

Generation 3	1	2	3	4	Generation 4	1	2	3	4
A	20	30	50	0	A	20	30	50	0
B	30	40	30	0	B	30	40	30	0
C	10	50	10	20	C	10	50	20	20
D	30	50	10	10	D	30	50	10	10
E	20	20	40	40	E	20	20	40	40
F	0	20	40	50	F	0	20	40	50

Generation 5	1	2	3	4	Generation 10	1	2	3	4
A	20	30	50	0	A	20	30	50	0
B	30	40	30	0	B	30	40	30	0
C	10	50	20	20	C	10	50	20	20
D	30	50	10	10	D	30	50	10	10
E	20	20	40	40	E	20	20	40	40
F	0	20	40	50	F	0	20	40	50

Figure 7. State perception phase simulation results

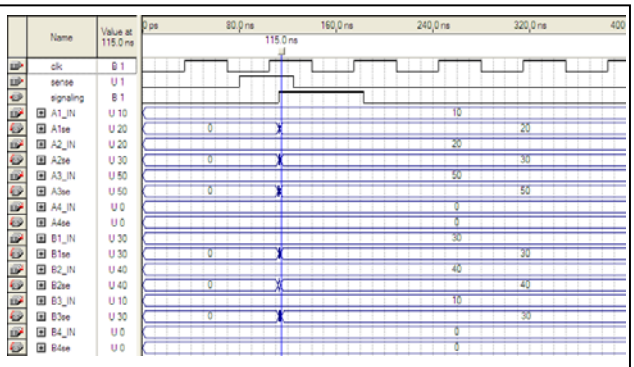


Figure 8. State perception phase simulation results

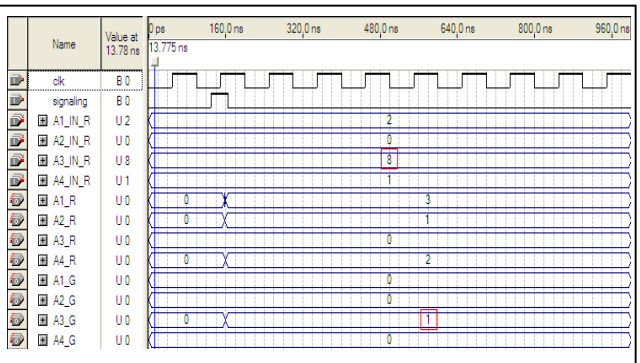


Figure 9. Consecutive red signaling rule

maximum number of consecutive red signals. As expected the green signal is shown to the western approach [Fig. 9 (A3_G)].

In the next case the western approach's flow pressure is 60 which is greater than the threshold set which triggers the

Red_t signaling rule. The next rule in descending priority is the Green_Max rule when 2 consecutive green signals are given in the same direction. Then a red signal must be given to this direction and decide according to the maximum flow pressure of the other directions. We can see in Fig. 10 that the red signal is given to the southern approach (number 2) because of the Green_Max rule and the green signal is decided by the flow pressure values of the rest approaches. Thus the green signal is given to the eastern approach with a FP value of 40. Finally when none of the above rules is valid the green signal is decided by the maximum flow pressure.

The exact model of the FPGA used for the implementation of the system is Altera Stratix, device EP1S60F1020C5. This FPGA uses 130nm technology and it allows the use of up to 57,120 logic elements. Table I indicates the use of the resources provided, for the implementation described.

TABLE I. THE RESOURCES OF FPGA CIRCUIT

Stratix II Compilation Report	
Total logic elements	52,535 / 57,120 (92%)
Total pins	214 / 782 (27%)
Total virtual pins	0
Total memory bits	0 / 5,215,104 (0%)
DSP block 9-bit elements	0 / 144 (0%)
Total PLLs	0 / 12 (0%)
Total DLLs	0 / 2 (0%)

In addition to the VHDL implementation, Matlab code was also developed in order to compare with the FPGA performance. The CPU used for the simulations is an AMD Phenom II 920 Quad-Core CPU clocked at 2.8GHz. Software based simulations are limited in sequential machines, since the inherent parallelism of the CA has to be emulated. Typically, this is achieved by calculating the time evolution of each cell separately and using double buffers to simulate the parallel nature of CA, thus leading to a considerable slowing down of the simulations [18]. The CA cell rules in this simulation are preset so the average number of instructions which have to be executed can be calculated. In average 2,550 instructions (adding, comparing, subtracting, multiplying etc.) were required for calculating the next cell state and signaling. This means that even with a super scalar CPU architecture the necessary clock cycles are 159 whereas the FPGA only requires 10 cycles for the output signals to be calculated. In case of Matlab simulation the mean completion time was 280ns while 240ns were required for the FPGA to complete the necessary calculations. The FPGA's native parallelism in executing commands in conjunction with the inherent parallelism of the CAs allows us to presume that in larger systems which include a vast number of intersections the FPGA's ability for parallel processing can give a significant advantage over a general purpose CPU which makes it suitable for this controller.

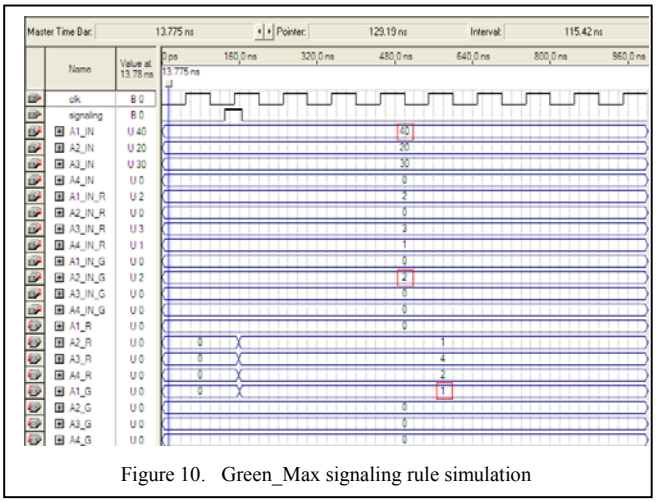


Figure 10. Green_Max signaling rule simulation

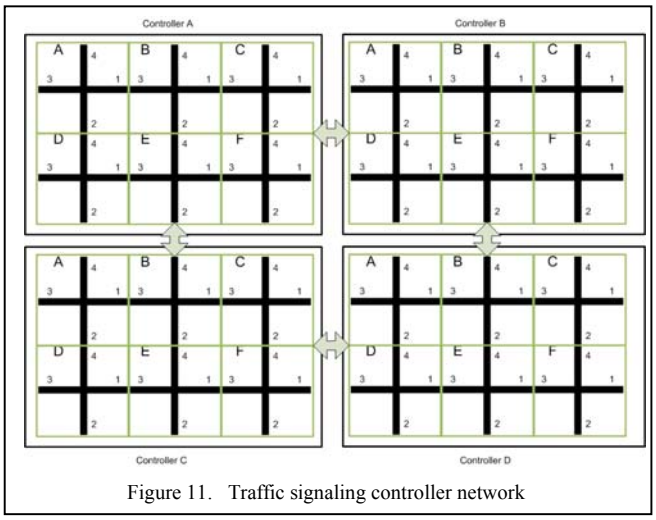


Figure 11. Traffic signaling controller network

Finally, it should be noticed that the code can be easily adopted for various traffic applications and all three components (sense, signaling and LFSR) can be changed to suit various urban network traffic conditions. All variables used can be automatically parameterized through the architecture section so they can be easily changed to the desired-requested values. Furthermore, what is of great importance, the number of the under study intersections that can be modeled and simulated can be adjusted by the flow pressure and signaling arrays' bounds. The only change needed is that the CA rules for both the state perception and signaling phase must be adjusted to the new intersections boundary approaches as depicted in Fig. 11. As a result, CA traffic several controllers, as the one proposed here, can be connected, each one controlling a number of intersections and feed each other with traffic information to form a larger, more complex and more versatile traffic signaling network.

4 Conclusions and future work

In this paper, a Cellular Automaton (CA) model was implemented on a FPGA to take full advantage of the inherent parallelism of CAs and provide real-time traffic signals control in accordance with vehicular traffic flow. The

presented CA model is characterized by as much as low complexity as possible so that the computational recourses are kept low while its computation speed is kept high without losing any of the requested essence of complexity regarding the real-time signals control of urban networks intersections. The proposed CA model implemented in hardware, while it presents two different phases of evolution, namely state perception phase and signaling phase, respectively, succeeds to control efficiently complicated traffic intersections with the help of the traffic lights as imposed by the initial different traffic conditions. As future work concerns, the expansion of the CA model for different types of neighborhoods, i.e. Moore neighborhood, more delicate model of signal synchronization, more detailed interesting characteristics of the network should be also considered. Finally, it would be interesting to compare the predictions of the CA model proposed in this article with empirical urban traffic traces and real-world traffic flow data selected by traffic cameras. Consequently, the single FPGA processor can be finally considered as basic component of an advanced electronic system able to provide real time information concerning the traffic conditions in the under study intersections and thus to efficiently handle-control the traffic signals in real conditions. More over, this, in turn, can verify the validity of the mobility model used in this article or provide valuable feedback on how to further refine it. As a result, at this point, some real case experiments are going to take place in the Xanthi's city vehicular traffic network of Greece and the experimental results will be used to validate the calibration of the CA model FPGA implementation parameters.

5 References

[1] Texas Transporation Institute. 2010 Urban Mobility Report. http://mobility.tamu.edu/ums/report/

[2] D. Helbing, "Traffic and related self-driven many-particle systems", Rev. Mod. Phys., vol. 73, no. 4, pp. 1067, 2001.

[3] J. Szklarski, "Cellular automata model of self-organizing traffic control in urban networks", Bulletin of the Polish Academy of Sciences Technical Sciences, vol. 58, no. 3, pp. 435-441, 2010.

[4] P. Chakroborty, "Models of vehicular traffic: An engineering perspective," Physica A, vol. 372, 2006, pp. 151–161.

[5] E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg, "Optimizing traffic lights in a cellular automaton model for city traffic," Phys. Rev. E, vol. 64, 2001, 056132 1-12.

[6] G. Ch. Sirakoulis, I. Karafyllidis, and A. Thanailakis, "A Cellular Automaton for the propagation of circular fronts and its applications," Engineering Applications of Artificial Intelligence, vol. 18, no. 6, 2005, pp. 731-744.

[7] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic", J. de Physique I, vol. 2, no 12, 1992, pp. 2221–2229.

[8] M. Takayasu and H. Takayasu, "1/f noise in a traffic model", Fractals, vol. 1, no. 4, 1993, pp. 860–866.

[9] S.Wolfram, "Statistical mechanics of cellular automata," Rev. Mod. Phys., vol. 55, 1983, pp. 601–644.

[10] S.C. Benjamin, N.F. Johnson, and P. Hui, "Cellular automata models of traffic flow along a highway containing a junction", J. Phys. A: Math. Gen., vol. 29, 1996, pp. 3119–3127.

[11] K. Nishinari, M. Fukui, and A. Schadschneider, "A stochastic cellular automaton model for traffic flow with multiple metastable states", J. Phys. A: Math. Gen., vol 37, 2004, pp. 3101–3110.

[12] O. Biham, A.A. Middleton, and D. Levine, Self-organization and a dynamical transition in traffic-flow models, Phys. Rev. A, vol. 46, no. 10, 1992, pp. R6124–R6217.

[13] Gu Guoqing, Hui Poming, Wang Binghong and Dai Shiqiang, "Two-dimensional cellular automaton traffic model with randomly switching traffic lights", Applied Mathematics and Mechanics, vol. 19, no. 9, 1998, pp. 807-813.

[14] Suwei Feng, Guoqing Gu, and Shiqiang Dai, "Effects of traffic lights on CA traffic model", Commun. Nonlinear Sci. Numer. Simul., vol. 2, no. 2, May 1997, pp. 70-74.

[15] A. Varas, M. D. Cornejo, B. A. Toledo, V. Muñoz, J. Rogan, R. Zarama, and J. A. Valdivia, "Resonance, criticality, and emergence in city traffic investigated in cellular automaton models," Phys. Rev. E, vol. 80, 2009, pp. 056108.

[16] S. Maerivoet and B. De Moor, "Cellular automata models of road traffic", Physics Reports, vol. 419, no. 1, November 2005, pp. 1-64

[17] J. Sanchez, M. Galan, and E. Rubio, "Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: "Las Ramblas" Area in Santa Cruz de Tenerife", IEEE Trans. Evol. Comput., vol. 12, no. 1, Feb. 2008, pp. 25-40.

[18] J. Wei, A. Wang, and N. Du, "Study of Self-Organizing Control of Traffic Signals in an Urban Network Based on Cellular Automata," IEEE Trans. Veh. Technol., vol. 54, no. 2, March 2005, pp. 744-748.

[19] G. Ch. Sirakoulis, I. Karafyllidis, and W. Spataro, "A computational intelligent oxidation process model and its VLSI implementation," in Proceedings of "2009 International Conference on Scientific Computing (CSC'09)", pp.329-335, Las Vegas, USA, July 13-19, 2009.

Lava Flow Simulation with Cellular Automata: Applications for Civil Defense and Land Use Planning

W. Spataro[1], M.V. Avolio[1], D. D'Ambrosio[1], V. Lupiano[2], R. Rongo[2], G.A. Trunfio[3]
[1] Department of Mathematics and High Performance Computing Center, University of Calabria, Italy
[2] Department of Earth Sciences and High Performance Computing Center, University of Calabria, Italy
[3] DADU, University of Sassari, Italy

Abstract –*The determination of areas exposed to new eruptive events in volcanic regions is crucial for diminishing consequences in terms of human causalities and damages of material properties. In this paper, we illustrate a methodology for defining flexible high-detailed lava invasion hazard maps which is based on an robust and efficient Cellular Automata model for simulating lava flows. We also present some applications for land use planning and civil defense to some inhabited areas of Mt Etna (South Italy), Europe's most active volcano, showing the methodology's appropriateness.*

Keywords: Cellular Automata, Lava flows simulation, Hazard Maps, Land Use Planning, Mt Etna.

1. Introduction

The use of thematic maps of volcanic hazard is of fundamental relevance to support policy managers and administrators in taking the most correct land use planning and proper actions that are required during an emergency phase. In particular, hazard maps are a key tool for emergency management, describing the threat that can be expected at a certain location for future eruptions.

At Mt. Etna (South Italy), the most active volcano in Europe, the majority of events occurred in the last four centuries report damage to human properties in numerous towns on the volcano flanks [6]. Notwithstanding, the susceptibility of the Etnean area to lava invasion has increased in last decades due to continued urbanization [18], with the inevitable consequence that new eruptions may involve even greater risks. During past eruptive episodes, different countermeasures based on embankments or channels have been adopted to halt or deflect lava [3][4]. Nevertheless, such kinds of interventions are generally performed while the eruption is in progress, inevitably putting into danger the safety of involved persons. Current efforts for hazard evaluation and contingency planning in volcanic areas draw heavily on hazard maps and numerical simulations (e.g. [21] [17], [1]), for the purpose of individuating affected areas in advance. For instance, in 2001 the path of the eruption that threatened the town of Nicolosi on Mt Etna was correctly predicted by means of a lava flows simulation model [13], providing at that time useful information to local Civil Defense authorities. However, in order to be efficiently and correctly applied, the above approaches require an a priori knowledge of the degree of exposure of the volcano surrounding areas, to allow both the realization of preventive countermeasures, and for a more rational land use planning.

In the following, we illustrate a methodology for the definition of flexible high-resolution lava invasion hazard maps, based on an improved version of SCIARA, a reliable and efficient Cellular Automata lava flow model, and show some specific applications related to inhabited areas of Mt Etna, which demonstrate the validity of the application for civil defense purposes and land use planning.

2. Lava flow modeling

The behavior of lava flows is difficult to be dealt with using traditional methods based on differential equation systems (e.g., cf. [20][14][22]). In fact, due to the complexities of its rheology, lava can range from fluids approximating Newtonian liquids to brittle solids while cooling, and thus it is difficult to solve the differential equations without making some simplifications. Nevertheless, many attempts of modelling real cases can be found in literature. However, since lava flows movement can be conveniently described in terms of ''local interactions'', Cellular Automata (CA) may represent a suitable solution. Regarding Cellular Automata-like models, Crisci and co-workers were the first to adopt Cellular Automata (CA) for modelling Etnean lava flows through the numerical simulation code SCIARA, initially fully three-dimensional [12] and successively reduced to a two-dimensional CA [5]. Ishihara et al. [21] were the first to adopt a Binghamian rheology in a CA numerical code, with good results on the simulation of some lava flows in Japan. Subsequently, Miyamoto and Sasaki [24] proposed a non-deterministic CA model that, thanks to a Monte Carlo approach, did not present the anisotropic problem due to the discretization of the considered (square) cellular space. Afterwards, a similar - non deterministic - approach was adopted by Vicari et al. [32] by the CA model MAGFLOW with good results on the simulation of Etnean lava flows.

2.1 Cellular Automata

The Cellular Automata (CA) computational paradigm was introduced in 1947 by John von Neumann [25], quickly gaining the attention of the Scientific Community both as powerful parallel computational models and as a convenient apparatus for modeling and simulating several types of complex physical phenomena [9]. Besides theoretical studies [33], CA have been applied to a variety of fields such as pattern recognition [23], image processing [28] and cryptography [31]. However, major interest for CA regard their use in Complex Systems modelling in various fields like Physics, Biology, Earth Sciences and Engineering (e.g., see [9],[16],[30]). Classical Cellular Automata can be viewed as an n-dimensional space, R, subdivided in cells of uniforms shape and size. Each cell embeds an identical finite automaton (*fa*), whose state accounts for the temporary features of the cell; Q is the finite set of states. The *fa* input is given by the states of a set of neighbouring cells, including the central cell itself. The neighbourhood conditions are determined by a geometrical pattern, X, which is invariant in time and space. The *fa* have an identical state transition function $\tau : Q^{\sharp X} \rightarrow Q$, where $\sharp X$ is the cardinality of the set of neighbouring cells, which is simultaneously applied to each cell. At step $t = 0$, *fa* are in arbitrary states and the CA evolves by changing the state of all *fa* simultaneously at discrete times, according to τ.

While Cellular Automata represents a powerful tool for simulating complex systems at a microscopic level of description, Macroscopic Cellular Automata (MCA) [19] can represent a valid alternative when the main features of the phenomena of interest can be directly described at a macroscopic level (e.g. in the case of a lava flow model: average amount of lava, temperature, etc), thus disregarding microscopic aspects. MCA introduce some extensions to the classical CA formal definition. In particular, the finite set of states Q of the cell is decomposed in r substates, Q_1, Q_2,…, Q_r, each one representing a particular feature of the phenomenon to be modelled (e.g. for lava flow models: cell temperature, lava content, outflows, etc.). The overall state of the cell is thus obtained as the Cartesian product of the considered substates: $Q = Q_1 \times Q_2 \times ... \times Q_r$. A set of parameters, $P=\{p_1, p_2, ..., p_p\}$, is furthermore considered, which allow to "tune" the model for reproducing different dynamical behaviours of the phenomenon of interest (e.g. for lava flow models, the Stephan-Boltzmann constant, lava density, lava solidification temperature, etc.). As the set of state is split in substates, also the state transition function τ is split in elementary processes, τ_1, τ_2, ..., τ_s, each one describing a particular aspect that rules the dynamic of the considered phenomenon. Eventually, $L \subset R$ is a subset of the cellular space that is subject to external influences (e.g. for lava flow models, the crater cells), specified by a supplementary function γ. External influences are introduced in order to model features which are not easy to be described in terms of local interactions. In the MCA approach, by opportunely discretizing the surface on which the phenomenon evolves, the dynamics of the system can be described in terms of flows of some quantity from one cell to the neighbouring ones. Moreover, as the cell dimension is a constant value throughout the cellular space, it is possible to consider characteristics of the cell (i.e. substates), typically expressed in terms of volume (e.g. lava volume), in terms of thickness. Still, owing to their intrinsic parallelism, both CA and MCA models implementation on modern parallel computers is straightforward, and the simulation duration can be reduced almost proportionally to the number of available processors ([15], [34]).

2.2 The SCIARA CA lava flow model

The methodology presented here heavily relies on the application of a computational model for simulating lava flows. In order to be applied for land use planning and civil defense purposes in volcanic regions, the model should be well calibrated and validated against test cases to assess its reliability (e.g., cf. [17], [27], [32]). Another required characteristic is the model's efficiency since, depending on the extent of the considered area, a great number of simulations could be required [10]. All these requirements are met by the latest release [29] of the SCIARA Cellular Automata model for simulating lava flow, adopted in this work, where a Bingham-like rheology has been introduced for the first time as part of the Minimization Algorithm of the Differences [19], which is applied for computing lava outflows from the generic cell towards its neighbors. In addition, the hexagonal cellular space adopted in the previous releases [13] of the model for mitigating the anisotropic flow direction problem has been replaced by a square one, nevertheless by producing an even better solution for the anisotropic effect. The model has been calibrated by considering three important real cases of studies, the 1981, 2001 and 2006 lava flows at Mt Etna (Italy), and on ideal surfaces in order to evaluate the magnitude of anisotropic effects. We briefly outline to model's main specifications in the following.

In formal terms, the SCIARA MCA model is defined as:

$$SCIARA = < R, L, X, Q, P, \tau, \gamma > \quad (1)$$

where:

– R is the set of square cells covering the bidimensional finite region where the phenomenon evolves;

– $L \subset R$ specifies the lava source cells (i.e. craters);

– $X = \{(0,0), (0,1), (-1,0), (1,0), (0,-1), (-1,1), (-1,-1), (1,-1), (1,1)\}$ identifies the pattern of cells (Moore neighbourhood) that influence the cell state change, referred to cells by indexes 0 (for the central cell) through 8;

– $Q = Q_z \times Q_h \times Q_T \times Q^8_f$ is the finite set of states, considered as Cartesian product of "substates". Their meanings are: cell elevation a.s.l. (above sea level), cell lava thickness, cell lava temperature, and lava thickness outflows (from the central cell toward the eight adjacent cells), respectively;

– $P = \{w, t, T_{sol}, T_{vent}, r_{Tsol}, r_{Tvent}, hc_{Tsol}, hc_{Tvent}, \delta, \rho, \varepsilon, \sigma, c_v\}$ is the finite set of parameters (invariant in time and space) which affect the transition function (please refer to [29] for their specifications);
– $\tau : Q^9 \rightarrow Q$ is the cell deterministic transition function, applied to each cell at each time step, which describes the dynamics of lava flows, such as cooling, solidification and lava outflows from the central cell towards neighbouring ones function (please refer to [29] for major specifications);
– $\gamma : Q_h \times N \rightarrow Q_h$ specifies the emitted lava thickness, h, from the source cells at each step $k \in N$ (N is the set of natural numbers).

As stated before, the new SCIARA model introduces a rheology inspired by the Bingham model and therefore the concepts of critical height and viscosity are explicitly considered (cf. [26], [20]). In particular, lava can flow out from a cell towards its neighbours if and only if its thickness overcomes a critical value (i.e. the critical height), so that the basal stress exceeds the yield strength. Moreover, viscosity is accounted in terms of flow relaxation rate, r, a parameter of the distribution algorithm that influences the amount of lava that outflows the cell, according to a power law of the kind:

$$\log r = a + bT \tag{2}$$

where T is the lava temperature and a and b coefficients determined by solving the system:

$$\begin{cases} \log r_{Tsol} = a + bT_{sol} \\ \log r_{Tvent} = a + bT_{vent} \end{cases}$$

where T_{sol} and T_{vent} are the lava temperature at solidification and at the vents, respectively. Similarly, the critical height, hc, mainly depends on lava temperature according to a power law of the kind:

$$\log hc = c + dT \tag{3}$$

whose coefficients c and d are obtained by solving the system:

$$\begin{cases} \log hc_{Tsol} = c + dT_{sol} \\ \log hc_{Tvent} = c + dT_{vent} \end{cases}$$

Please refer to [29] for further details on the model, where an experiment in order to evaluate the magnitude of the anisotropic effect on an ideal surface is also reported.

3. The methodology for defining hazard maps

Volcanic hazard maps are fundamental for determining locations that are subject to eruptions and their related risk. Typically, a volcanic hazard map divides the volcanic area into a certain number of zones that are classified differently on the basis of the probability of being exposed to a specific volcanic event in future. Mapping both the physical threat and the exposure and vulnerability of people and material properties to volcanic hazards can help local authorities to guide decisions about where to locate critical infrastructures (e.g. hospitals, power plants, railroads, etc) and human settlements and to devise mitigation measures that might be appropriate. This could be useful for avoiding the development of inhabited areas in high risk areas, thus controlling land use planning decisions.

A lava flow simulation model can represent an effective instrument for analyzing volcanic risk in a certain area by simulating possible *single* episodes with different characteristics (e.g. vent locations, effusion rates, cf. [11]). However, the methodology for defining high detailed hazard maps presented here is based on the application of the SCIARA lava flows computational model for simulating an *elevated* number of events on present topographic data. In particular, the methodology requires the analysis of the past behavior of the volcano, for the purpose of classifying the events that historically affected the region. In such a way, a meaningful database of plausible simulated lava flows can be obtained, by characterizing the study area both in terms of areal coverage, and lava flows typologies. Data is subsequently processed by considering a proper criterion of evaluation. A first solution could simply consist in considering lava flows overlapping, assigning a greater hazard to those areas affected by a higher number of simulations. However, a similar choice could be misleading since, depending on the event's volcanological characteristics (e.g., location of the main crater, duration and amount of emitted lava, or effusion rate trend), different events can occur with different probabilities, which should be taken into account in evaluating the actual contribution of performed simulations with respect to the definition of the overall hazard of the study area. In most cases, such probabilities can be properly inferred from the statistical analysis of past eruptions, allowing for the definition of a more refined evaluation criterion. Accordingly, in spite of a simple hitting frequency, a measure of lava invasion hazard can be obtained in probabilistic terms. In the following, we show how such approach was applied to Mt Etna.

3.1 Mt. Etna volcano: A case study

By adopting a technique well described in [10] and [2], which referred to the Eastern sector of Mt. Etna and which was applied by employing a previous version of the SCIARA CA model, we here show the application to the entire area of the volcano using the new SCIARA model briefly described in Section 2.2. Firstly, based on documented past behavior of the volcano, the probability of new vents forming was determined, resulting in a characterization (a probability density function map - pdf) of the study region into areas, that represent different probabilities of new vents opening [8].

Then, flank eruptions of Etna since 1600 AD were classified according to duration and lava volume [10] and a representative effusion rate trend considered to characterize lava temporal distribution for the considered representative

eruptions, reflecting the effusive mean behavior of Etnean lava flows [7]. An overall probability of occurrence, p_e, was thus defined, by considering the product of the individual probabilities of its main parameters:

$$p_e = p_s \cdot p_c \cdot p_t \quad (4)$$

where p_s denotes the probability of eruption from a given location (i.e., based on the pdf map), p_c the probability related to the event's membership class (i.e., emitted lava and duration), and p_t the probability related to its effusion rate trend. Once representative lava flows were devised as above, a set of simulations were planned to be executed in the study area by means of the SCIARA lava flows simulation model. At this purpose, a grid composed by 4290 craters, equally spaced by 500m, was defined as a covering for Mt Etna, from where the simulations have been carried out. This choice allowed to both adequately and uniformly cover the study area, besides considering a relatively small number of craters. Specifically, a subset of event classes which define 6 different effusion rates probabilities, derived from historical events considered in [10], were taken into account for each crater, thus resulting in a total of 25740 different simulations to be carried out. Owing to the elevated number of SCIARA simulations to be executed, thanks to the adoption of Parallel Computing each scenario was simulated for each of the vents of the grid. Simulations were performed on an 80-node Apple Xserve Xeon-based cluster and were performed in ca. 10 days. Lava flow hazard was then punctually (i.e. for each cell) evaluated by considering the contributions of all the simulations which affected a generic cell in terms of their probability of occurrence.

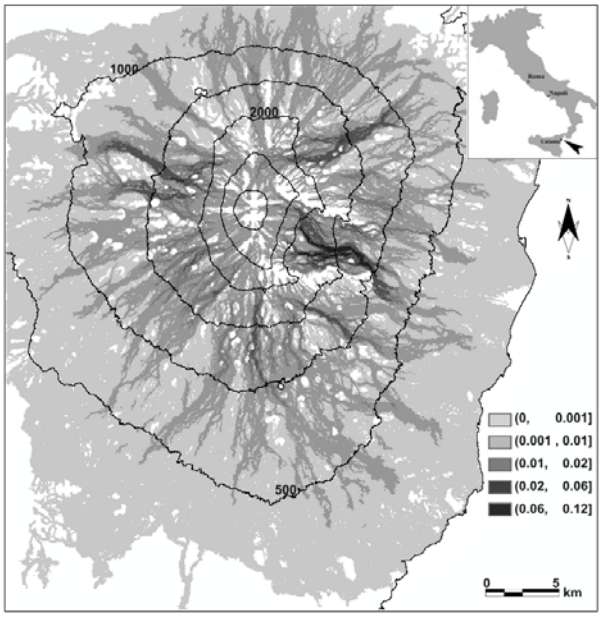


Figure 1: Hazard map of the study area based on the 25740 simulations. As a compromise between map readability and accuracy, 5 classes are reported (grey colouring), in increasing order of susceptibility (probability of lava invasion).

The obtained lava flow hazard map resulting from these simulations is presented in Figure 1, and represents the probability that future eruptions will affect the entire Etnean area. Here, as in all following applications, as a compromise between map readability and reliability 5 classes are reported (grey colouring), in increasing order of susceptibility (probability of lava invasion).

Importantly, the methodology for the compilation of lava flows invasion hazard maps proposed here provides for, as integrant part, a process for the verification of results. A validation procedure was thus contemplated for the produced hazard map, consisting in a technique which produces statistical indicators on which one can quantify the reliability of the results. Refer to [10] for major details on the methodology validation process.

4. Applications for Civil Defense and Land Use Planning

As shown previously, the described methodology permits the definition of general hazard maps, as the one reported in Figure 1, which can give valuable information to Civil Defense responsible authorities. However, further, more specialized applications can be devised by considering that the SCIARA simulation model is integrated in a GIS (Geographic Information System) application that permits, besides other features, to take also into account the effects of "virtual" embankments, channels, barriers, etc. In addition, the fact that a large number of lava flows of different eruption types, magnitudes and locations are stored in the database, a rapid extraction of various scenarios is possible.

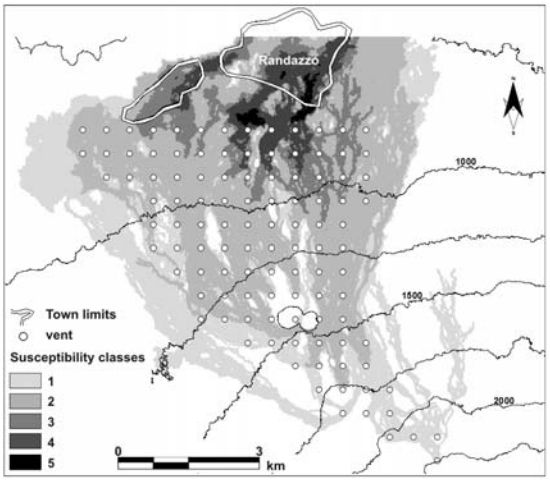


Figure 2: Map showing vents, belonging to the simulation grid, which can produce eruptions capable of affecting the town of Randazzo, together with the resulting susceptibility scenario, allowing to immediately assess the threat posed by an eruption exclusively on the basis of its source location.

A first fundamental Civil Defense oriented application regards the possibility to identify all source areas of lava

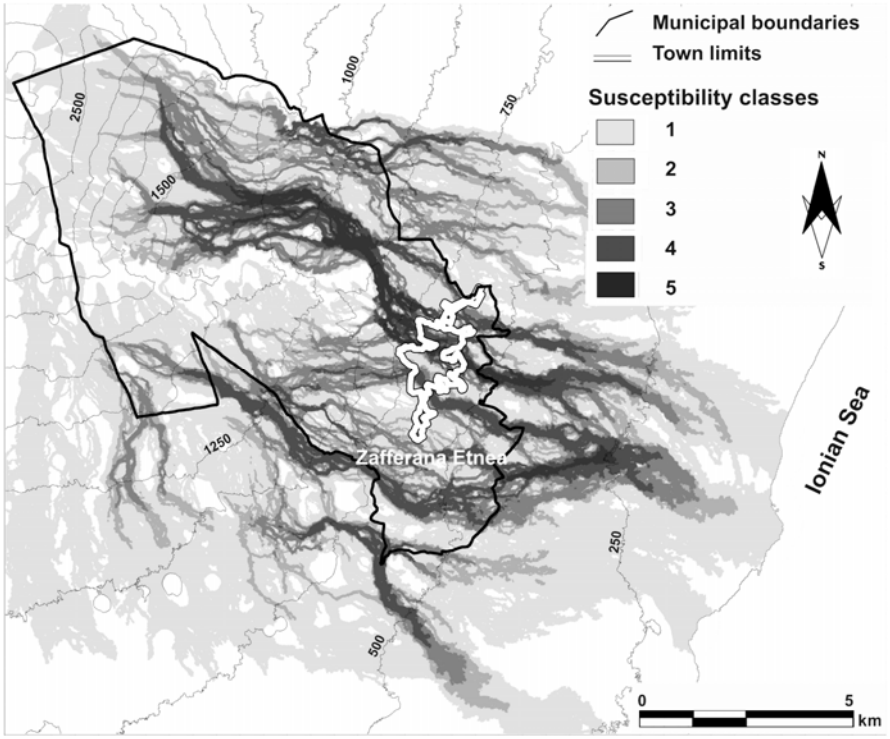


Figure 3: A second example of application of hazard zonation referred to the entire town district of Zafferana Etnea. The town district boundaries are indicated by the black line, while the present inhabited area with white line. As shown, the majority of the municipal area is at risk.

flows that are capable of affecting a given area of interest, such as a town or a major infrastructure (e.g., hospitals, power plants, etc.). In this case, this application is rapidly accomplished by querying the simulation database, selecting the lava flows that affect the area of interest and by circumscribing their sources. For this application we have chosen the town of Randazzo, an important historical and cultural site of the Etnean area. Figure 2 shows vents which can originate eruptions capable of affecting the urban area of Randazzo, together with the resulting hazard scenario, allowing to immediately assess the threat posed by an eruption exclusively on the basis of its source location.

While the previous application localizes craters that can originate events that may interest an inhabited area, the one reported in Figure 3 can have even more impact in land use planning, referred for the *entire* town district of Zafferana Etnea, another important inhabited area of the volcano. This application is fundamental in understanding how local authorities can plan the future development of the city, avoiding it in elevated risk areas. Specifically, the figure shows how several areas of the entire municipality are at risk, especially to the North-West and South.

A further specific category of simulation regards the assessment of protective measures, such as earth barriers or channel digging, for mitigating lava invasion susceptibility in

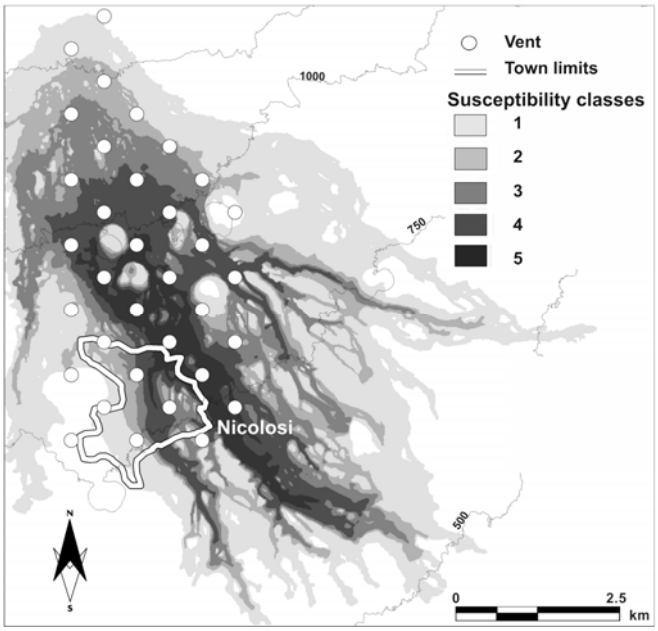


Figure 4: Map showing the location of a set of vents (white dots) which originate lava flows intersecting a hypothetical 2 km long and 20 m tall earth barrier (cf. Figure 5) to protect the centre of Nicolosi (white line).

given areas. To illustrate this kind of application, a northwest-southeast trending barrier, 2 km long and 20 m high, was considered along the northern margin of Nicolosi, an urban area with many administrative buildings and tourist facilities. for diverting lava flows into a valley at the eastern margin of the town without, however, considering the legal and ethical aspects of such an operation. By querying the simulation database, all the lava flows that affected the barrier were selected and thus re-simulated on the modified topography which embeds the presence of the barrier. Similarly to the case of the applications shown in Figures 2 and 3, an ad hoc susceptibility scenario was extracted by considering these new simulations (Figure 4).

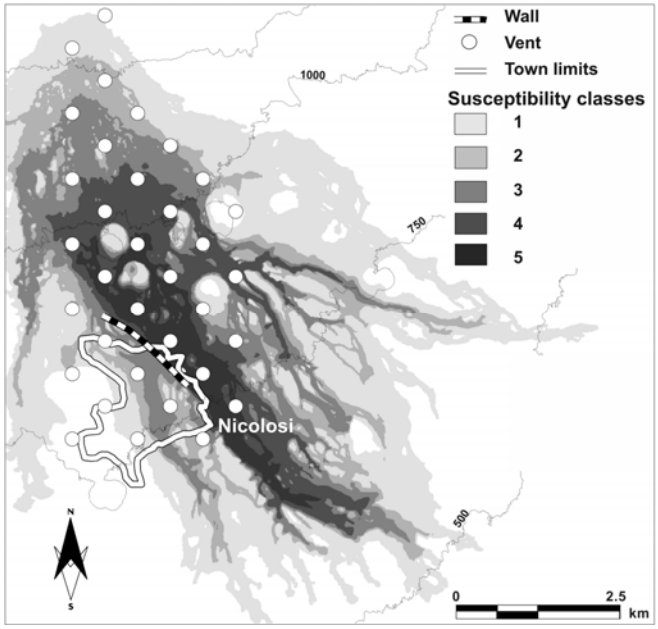


Figure 5: The same area considered in Figure 4, together with the scenario resulting from lava flows intersecting the barrier, which are re-simulated on a modified topography that embeds the presence of the barrier. As shown, the hazard decreases by two classes within the town limits (white line).

Results show that the barrier would be necessary to effectively protect the town centre. The susceptibility here decreases by two classes (Figure 5) and, at the same time, the areas invaded by diverted flows prove characterised by only a slightly higher susceptibility degree. In this specific case, the protective measure has a substantially positive effect. If this was not the case, further experiments with barriers of different positions and dimensions will reveal to what degree damage from lava flow invasion can be minimized, or whether it would be preferable to abandon any prospects of this kind of protective measure.

5. Conclusions

The fundamental problem of assessing the impact of future eruptions in a volcanic region lies mostly in the uncertainty concerning their duration, effusion rate, and location. A valid assessment tool relies on the adoption of volcanic hazard maps which, however, are usually based on the sole analysis of past events. Conversely, maps should represent the full range of probable hazards that can be expected to an agreed probability, considering thus all potential future scenarios. As a consequence, probabilistic hazard maps can provide a better base for planning mitigation strategies. We tackled this issue by an elaborate approach in the numerical simulation of a wide variety of lava flows, which are typical of Etna for duration and effusion rate, on a dense grid of vents, by attributing them a statistical likelihood. Regarding the adopted new SCIARA Cellular Automata computational model at the basis of the methodology, it re-introduces a square tessellation of the cellular space instead of the previously adopted hexagonal one, considered in the earlier versions to limit the effect of the anisotropic flow direction problem. It is worth noting that the main advantage of the presented methodology, besides the possibility of assessing the efficiency of protective measures for inhabited areas and/or major infrastructures, is that the simulation data permits to produce general susceptibility maps in unprecedented detail, and contains each single scenario out of a total of over thousands of simulated cases. As a consequence, the methodology described here can represent a substantial advance in the field of lava flow impact prediction and can also have immediate, far reaching implications both in land use and civil defense planning.

6 Acknowledgments

A special thanks goes to Prof. Salvatore "Toti" Di Gregorio and Prof. Gino Mirocle Crisci for the common researches. Authors thank Dr. B. Behncke and Dr. M. Neri from the Istituto Nazionale di Geofisica e Vulcanologia of Catania (Sicily, Italy), who provided volcanological data.

References

[1] M.V. Avolio, G.M. Crisci, S. Di Gregorio, R. Rongo, W. Spataro, and D. D'Ambrosio. "Pyroclastic flows modelling using cellular automata"; Comp. Geosc., 32, 897–911, 2006.

[2] M.V. Avolio, D. D'Ambrosio, S. Di Gregorio, V. Lupiano, R. Rongo, and W. Spataro. "Evaluating lava flow hazard at Mount Etna (Italy) by a cellular automata based methodology"; LNCS 6068, 495–504, 2010.

[3] F. Barberi, F. Brondi, M.L. Carapezza, L. Cavarra, and C. Murgia. "Earthen barriers to control lava flows in the 2001 eruption of Mt. Etna"; J. Volcanol. Geotherm. Res., 123, 231–243, 2003.

[4] F. Barberi, M. Carapezza, M. Valenza, and L. Villari. "The control of lava flow during the 1991-1992 eruption of Mt. Etna"; J. Volcanol. Geotherm. Res., 56, 1–34, 1993.

[5] D. Barca, G.M. Crisci, S. Di Gregorio, S. Marabini, and F.P Nicoletta. "Lava flow simulation by cellular automata and

Pantelleria's example"; Proceedings of the Kagoshima International Conference on Volcanoes, (1988) 475–478.

[6] B. Behncke, and M. Neri. "Cycles and trends in the recent eruptive behaviour of Mount Etna (Italy)"; Can. J. Earth Sci., 40, 1405–1411, 2003.

[7] B. Behncke, M. Neri, and A. Nagay. "New data from a GIS-based study, kinematics and dynamics of lava flows"; Geol. Soc. Am. Spec. Pap., 396, 189–208, 2005.

[8] A. Cappello, A. Vicari, and C. Del Negro. "A retrospective validation of lava flow hazard map at Etna volcano"; Spec. Issue of Annals of Geophy., To Appear, 2011.

[9] B. Chopard, and M. Droz. Cellular Automata Modeling of Physical Systems. Cambridge University Press, UK, 1998.

[10] G.M. Crisci, M.V. Avolio, B. Behncke, D. D'Ambrosio, S. Di Gregorio, V. Lupiano, M. Neri, R. Rongo, and W. Spataro. "Predicting the impact of lava flows at Mount Etna"; J. Geophy. Res., 115(B0420):1–14, 2010.

[11] G.M. Crisci, S. Di Gregorio, F. Nicoletta, R. Rongo, and W. Spataro. "Analysing lava risk for the Etnean area: Simulation by cellular automata methods". Natural Hazards, 20, 215–229, 1999.

[12] G.M. Crisci, S. Di Gregorio, and G. Ranieri. "A cellular space model of basaltic lava flow"; In Proceedings International AMSE Conference Modelling & Simulation, 1982.

[13] G.M. Crisci, R. Rongo, S. Di Gregorio, and W. Spataro. "The simulation model SCIARA: The 1991 and 2001 lava flows at Mount Etna"; J. Volc. Geoth. Res., 132, 253–267, 2004.

[14] J.A. Crisp, and S.M. Baloga. "A model for lava flows with two thermal components"; Journal of Geophysical Research, 95, pp. 1255-1270, 1990.

[15] D. D'Ambrosio, and W. Spataro. "Parallel evolutionary modelling of geological processes"; Paral. Comp., 33(3), 186–212, 2007.

[16] D. D'Ambrosio, W. Spataro, G. Iovine, H. Miyamoto. "A macroscopic collisional model for debris flows simulation"; Environ. Model. Soft. 22 (2007) 1417–1436.

[17] C. Del Negro, L. Fortuna, A. Herault, and A. Vicari. "Simulations of the 2004 lava flow at Etna volcano using the magflow cellular automata model"; Bull. Volcanol., 70, 805–812, 2008.

[18] C. Dibben. "Leaving the city for the suburbs – the dominance of 'ordinary' decision making over volcanic risk perception in the production of volcanic risk on Mt Etna, Sicily"; J. Volcanol. Geotherm. Res., 172, 288–299, 2008.

[19] S. Di Gregorio, and R. Serra. "An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata". Fut. Gen. Comp. Sys., 16, 259–271, 1999.

[20] M. Dragoni, M. Bonafede, and E. Boschi. "Downslope flow models of a Bingham liquid: Implications for lava flows"; J. Volc. Geoth. Res., 30(3-4), 305–325, 1986.

[21] K. Ishihara, M. Iguchi, and K. Kamo. "Numerical simulation of lava flows on some volcanoes in Japan"; In IAVCEI Proceedings in Volcanology, 174–207. Springer, Berlin Heidelberg New York, 1990.

[22] A. Longo, and G. Macedonio. "Lava flow in a channel with a bifurcation"; Physics and Chemistry of the Earth Part A – Solid Earth and Geodesy, V24 N11-12, 953-956, 1999.

[23] P. Maji, C. Shaw, N. Ganguly, B.K. Sikdar, P.P. Chaudhuri. "Theory and Application of Cellular Automata For Pattern Classification"; Fund. Inform. 58, 321–354, 2003.

[24] H. Miyamoto, and S. Sasaki. "Simulating lava flows by an improved cellular automata method"; Comp. Geosci., 23, 283–292, 1997.

[25] J. von Neumann. "Theory of self-reproducing automata"; Univ. Illinois Press, Urbana, 1966.

[26] S. Park, and J. Iversen. "Dynamics of lava flow: Thickness growth characteristics of steady 2-dimensional flow"; Geophys. Res. Lett., 11, 641–644, 1984.

[27] R. Rongo, W. Spataro, D. D'Ambrosio, M.V. Avolio, G. Trunfio, and S. Di Gregorio. "Lava flow hazard evaluation through cellular automata and genetic algorithms: An application to Mt Etna volcano"; Fund. Inform., 8, 247–268, 2008.

[28] P.L. Rosin. "Training Cellular Automata for Image Processing"; IEEE Trans. In. Proc. 15 (2006) 2076–2087.

[29] W. Spataro, M.V. Avolio, V. Lupiano, G. Trunfio, R. Rocco, and D. D'Ambrosio. "The latest release of the lava flows simulation model SCIARA: First application to Mt Etna (Italy) and solution of the anisotropic flow direction problem on an ideal surface"; In Proceedings of the International Conference on Computational Science. ICCS 2010, 1, 17–26. Procedia Computer Science, 2010.

[30] S. Succi. "The Lattice Boltzmann Equation for Fluid Dynamics and Beyond"; Oxford University Press, 2004.

[31] M. Tomassini and M. Perrenoud. "Cryptography with cellular automata"; Appl. Soft Comp. 1 (2001) 151-160.

[32] A. Vicari, A. Herault, C. Del Negro, M. Coltelli, M. Marsella, and C. Proietti. "Modelling of the 2001 lava flow at Etna volcano by a cellular automata approach"; Environ. Model. Soft., 22, 1465–1471, 2007

[33] S. Wolfram. "A new kind of Science"; Wolfram Media Inc., Champaign, USA, 2002.

[34] G. Zito, D. D'Ambrosio, W. Spataro, R. Rongo, G. Spingola, M.V. Avolio. "A Dynamically Load Balanced Cellular Automata Library for Scientific Computing"; Proceedings of The 2009 International Conference on Scientific Computing, 322-328, 2009.

The Number of DFAs Produced by a Given Spanning Tree

Parisa Babaali, Edoardo Carta-Gerardino and Christopher Knaplund
Department of Mathematics and Computer Science
York College, CUNY
Jamaica, NY 11451

***Abstract*—In the last few decades, several techniques to randomly generate a deterministic finite automaton have been developed. These techniques have implications in the enumeration of automata of size n. One of the ways to generate a finite automaton is to generate a random tree and to complete it to a deterministic finite automaton, assuming that the tree will be the automaton's breadth-first spanning tree. In this paper we explore some ideas related to this method. We introduce the notions of tail characteristic and characteristic of a tree, and use it to define the weight of a tree. It turns out that the weight of a tree can be used to count the number of automata having this tree as their spanning tree. We also present a recursive formula for this quantity in terms of the "derivative" of a tree. Finally, we analyze the implications of this formula in terms of the distribution of the number of automata with a given spanning tree with n nodes.**

I. Introduction

Enumeration and generation of Accessible Deterministic Finite Automata have been of interest since late 1950s. There is a number of ways to generate an accessible deterministic finite automaton (ADFA) with n states. However, the question of generating a minimal automaton with n states remains open. One way to generate a minimal automaton is to generate a DFA at random and use a rejection algorithm, such as Hopcroft-Ullman's algorithm [9], to decide if it is minimal assuming that the asymptotic density of minimal automata is constant. This question is very important in algorithmic analysis, in calculating the average case complexity of algorithms, and in analyzing certain properties of formal languages. Harary and Palmer [7] in 1973 enumerate isomorphic automata with output functions as certain ordered pairs of functions. Harrison [8] considered the enumeration of non-isomorphic DFAs and connected DFAs up to a permutation of the alphabet symbols. With the same criteria, Narushima [11] enumerated minimal DFAs. Domaratzki et al. [6] have proposed a lower bound for the number of accessible deterministic finite automata (ADFAs) over an alphabet of size k. Also Nicaud [12], and Champarnaud and Paranthoën [5] presented a method for randomly generating ADFAs. Bassino and Nicaud [4] showed that the number of ADFAs is $\Theta(n2^nS(kn,n))$, where $S(kn,n)$ is the Stirling number of the second kind. Almeida and Moreira [1] have also proposed efficient algorithms to generate ADFAs at random and confirmed the previous result.

One of the methods to generate a random automaton, presented in [1], and [2] is to generate a random k-ary tree and complete it to a DFA by assigning the missing transitions at random. This method considers a canonical string representation for an automaton and a tree, and using this representation every automaton will be generated uniquely. In our paper we explore this method in more detail and find an exact formula to calculate the number of automata produced by a given tree, i.e. the automata having the given tree as a breadth-first spanning subtree.

We organize our paper as follows. In Section II we start by recalling some basic definitions and results. In particular, we recall that an automaton can be generated at random using breadth-first spanning trees. Then, in Section III, we introduce two definitions that will prove to be very useful, tail characteristic and the characteristic of a tree. We use these ideas to define a weight function on the set of trees in Section IV. This weight function will be used to count the number of automata with n states having a given tree as their spanning tree. We then present a recursive formula for this quantity. Even though our method does not count the total number of automata of size n, we do integrate some of the methods used in the generation algorithms in [1], [4]. Finally, we look at the implications that our formula has on the distribution of the number of automata with a given spanning tree.

II. Preliminaries

A **deterministic finite automaton (DFA)** $\mathcal{A}$ is a 5-tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of

states, Σ is a finite *input alphabet*, $q_0 \in Q$ is the *initial state*, $F \subset Q$ is the set of *final states*, and δ is the *transition function* mapping $Q \times \Sigma$ to Q. We can extend δ by defining $\delta(q, aw) = \delta(\delta(q, a), w)$. We will define a **transition structure** to be an automaton (Q, Σ, δ, q_0) with no final states. Finally, if an initially connected DFA has the property that it contains a directed path from q_0 to every other state, then we say that it is an **accessible deterministic finite automaton (ADFA)**. For the remainder of the paper we assume $\Sigma = \{a, b\}$.

The **language** accepted by a DFA $\mathcal{A}$ is defined as $L(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. We will say that two DFAs are equivalent if they accept the same language. Furthermore, an automaton is said to be **minimal** if it is the automaton with the smallest number of states accepting a given language.

Any ADFA, $\mathcal{A}$ with n states, can be decomposed into its breadth-first spanning tree with n nodes and the remaining transitions. For a detailed argument of this decomposition, the reader is referred to [3]. This decomposition will lead to an ordering of states of A, and hence a numbering of states, from 1 to n.

A tree T with n nodes can be presented by a binary sequence β_T of length $2n + 1$, where $\beta_T(i) \in \{0, 1\}$. For $1 \leq k \leq n$, define the **binary representation β_T of a tree** T by defining $\beta_T(0) = 1$ and

$$\beta_T(2k-1) = \begin{cases} 1 & \text{if there is an edge leaving} \\ & \text{state } k \text{ with label } a, \\ 0 & \text{otherwise,} \end{cases}$$

$$\beta_T(2k) = \begin{cases} 1 & \text{if there is an edge leaving} \\ & \text{state } k \text{ with label } b, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $\beta_T(i)$ represents an edge leaving state $\left\lceil \frac{i}{2} \right\rceil$, labeled by alphabet letters a or b. This representation of trees has been explored in detail in [2]. Similar representations have been used to generate random binary trees, as shown in the survey by Mäkinen in [10]. It can be shown that each β_T is a string of length $2n+1$, with n 1s, and $n+1$ 0s. This representation also has the property that each prefix w of β_T satisfies $|w|_0 < 1 + |w|_1$ where $|w|_\sigma$ is the number of occurrences of σ in w. In other words, each 1 in β_T represents a node, and each 0 presents a missing transition to turn the tree into an ADFA. An example of such a representation is shown in the following Example.

Example 1: Consider the automaton $\mathcal{A}_T$ and its breadth-first spanning tree T shown below. Using the definition of the binary representation of a tree, it is easy to see that $\beta_T = 101110000$.

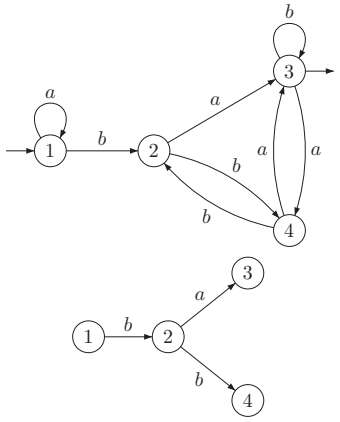


Fig. 1: An automaton $\mathcal{A}_T$ and its spanning tree T. Here, $\beta_T = 101110000$.

It is worth mentioning that this decomposition and representation of trees is very useful in enumerative and combinatorial arguments used in the analysis of ADFAs.

III. The Characteristic Sequence of a Tree

We start with an important definition. In the remainder of this paper, let $\mathcal{T}_n$ denote the set of edge labeled binary trees with n nodes. The edges will be labeled by the alphabet $\Sigma = \{a, b\}$.

Definition 1: Given a tree $T \in \mathcal{T}_n$ ($n \geq 1$), with binary representation β_T, the **tail characteristic** $\chi_t(T)$ of T is the number, r of 0s followed by the last 1 in β. In other words $\chi_t(T)$ is the nonnegative integer r satisfying the following three conditions:

1) $\beta_T(2n - r) = 1$,
2) $|\beta_T(2n - (r-1)) \dots \beta_T(2n)|_1 = 0$,
3) $|\beta_T(2n - (r-1)) \dots \beta_T(2n)|_0 = r$,

Example 2: Consider the tree $T \in \mathcal{T}_5$ shown below. It is easy to see that T has binary representation $\beta_T = 11011010000$ and tail characteristic $\chi_t(T) = 4$.

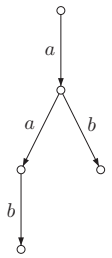


Fig. 2: A tree T with 5 nodes, and tail characteristic $\chi_t(T) = 4$.

Using the tail characteristic, we will define the characteristic of a tree. But before, we introduce the notion of a tree derivative. Suppose that $T \in \mathcal{T}_n$, where $n \geq 2$. Then the **tree derivative** of T is the tree $T' \in \mathcal{T}_{n-1}$ obtained by removing the *last* node in T. Note that if β_T is the binary representation of T, then the binary representation of T' can be obtained from β_T by removing the last two 0s and replacing the last 1 with a 0.

In general, the kth tree derivative of T ($1 \leq k \leq n-1$) is the tree $T^{(k)} \in \mathcal{T}_{n-k}$ obtained by removing the last k nodes of T. It is easy to see that if T has n nodes, then T^{n-1}, the $(n-1)$th derivative of T, is the tree with one node. For such a tree, the tail characteristic is 2.

Definition 2: Given a tree $T \in \mathcal{T}_n$ ($n \geq 1$), the **characteristic** $\chi(T)$ of T is a sequence of n nonnegative integers $(r_1, r_2, \ldots, r_{n-1}, r_n)$ such that r_n is the tail characteristic of T, r_{n-1} is the tail characteristic of T', $\ldots$, r_2 is the tail characteristic of $T^{(n-2)}$, and $r_1 = \chi_t(T^{n-1}) = 2$.

Example 3: Consider the tree $T \in \mathcal{T}_5$ from Example 2. Then it is not difficult to see that the characteristic of T is $\chi(T) = (2, 3, 3, 4, 4)$.

The characteristic of a tree is unique for a given tree, and can be used to count the number of transition structures having a tree as a breadth-first spanning subtree.

Lemma 1: Let $T \in \mathcal{T}_n$ be a tree with n nodes, and let $\chi(T) = (r_1, r_2, \ldots, r_n)$ be the characteristic of T. Then the following conditions hold

1) $2 \leq r_i \leq i+1$ for $1 \leq i \leq n$
2) $r_{i+1} \leq r_i + 1$ for $1 \leq i \leq n$

Conversely, if χ is a sequence of n nonnegative integers, satisfying conditions 1 and 2, then there is a tree $T \in \mathcal{T}_n$, such that $\chi(T) = \chi$.

Proof It is clear that $r_i \geq 2$, since β_T always has a 00 at the end. Let $T \in \mathcal{T}_n$ be a tree with $\chi(T) = (r_1, r_2, \ldots, r_n)$, and let β_T be the string representation of T. First we show that the tail characteristic of T is at most $n+1$. Assume $\chi_t(T) > n+1$, this is a contradiction since the total number of 0s in β_T is $n+1$. Since each r_i is the tail characteristic of a tree with i nodes, we have $r_i \leq i+1$ and the first part is proven.

To show that the second condition holds, suppose that r_n is the tail characteristic of T. To obtain the tree derivative of T we need to remove the last two 0s of β_T and replace the last 1 with a 0, a condition similar to the following equations

$$\beta_T = 1011\ldots1\underbrace{00\ldots0000}_{r_n} \quad \text{and}$$

$$\beta_{T'} = 1011\ldots0\underbrace{00\ldots00}_{r_n-2}.$$

Notice that the tail characteristic of T', r_{n-1} is the number of 0s following the last one , and hence $r_{n-1} \geq r_n - 2 + 1 = r_n - 1$ and we have $r_n \leq r_{n-1} + 1$. Using the same argument for r_{n-1}, r_{n-2}, up to r_1, we can show $r_{i+1} \leq r_i + 1$, since there are only finitely many r_is .

Now suppose $\chi = (a_1, a_2, \ldots, a_n)$ is a sequence satisfying conditions 1 and 2. We will construct a tree T, such that $\chi(T) = \chi = (a_1, a_2, \ldots, a_n)$. Let $k_i = a_i - a_{i+1} + 1$ for $1 \leq i \leq n-1$ and define $k_n = a_n$. We claim that $K_T = (k_1, k_2, k_3, \ldots, k_{n-1}, k_n)$ is a sequence of nonnegative integers, having the following properties:

i. $0 \leq \sum_{j=1}^{i} k_j \leq i$ for $1 \leq i \leq n$
ii. $\sum_{i=1}^{n} k_i = n+1$

Note that $\sum_{j=1}^{i} k_j = a_1 - a_{i+1} + i = 2 - a_{i+1} + i$. Since by condition (1) of Lemma 1, $2 \leq a_{i+1} \leq i+2$, we have $0 \leq i - a_{i+1} \leq i$. The second condition holds by a simple calculation of the alternating series $\sum_{i=1}^{n} k_i$ with $a_1 = 2$.

$$\sum_{i=1}^{n} k_i = \sum_{i=1}^{n-1}(a_i - a_{i+1} + 1) + a_n =$$
$$a_1 + n - 1 = n + 1$$

We use K_T to construct our tree, by defining $\beta_T = 10^{k_1}10^{k_2}10^{k_3}\ldots10^{k_{n-1}}10^{k_n}$. β_T is a string of length $2n+1$ with n 1s and $n+1$ 0s, since $\sum_{i=1}^{n} k_i = n+1$. To show that β_T is a string representation of a tree, we must show that any prefix, w, of β_T has the property that $|w|_0 < 1 + |w|_1$.

First suppose w has the form $w = 10^{k_1}10^{k_2}\ldots10^{k_i}$. In this case $|w|_1 = i$ and $|w|_0 = \sum_{j=1}^{i} k_j \leq i$. So we have $|w|_0 \leq i < 1 + i = 1 + |w|_1$. Suppose now that $w = 10^{k_1}10^{k_2}\ldots10^{l}$ where $l < k_i$, in this case $\sum_{j=1}^{i-1} k_j + l < \sum_{j=1}^{i} k_j \leq i$ and the inequality $|w|_0 < 1 + |w|_1$ holds. Finally if $w = 10^{k_1}10^{k_2}\ldots10^{k_i}1$ then $|w|_1 = i+1$ and $|w|_0 = \sum_{j=1}^{i} k_j \leq i$ and we have $|w|_0 < 1 + |w|_1$.

To calculate $\chi(T)$, we know by definition that $r_n = k_n = a_n$. To find r_{n-1}, the reader should observe that the derivative of T will have the form

$$\beta_{T'} = 10^{k_1}10^{k_2}10^{k_3}\ldots10^{k_{n-1}}00^{r_n-2}$$

which implies $r_{n-1} = r_n + 1 + k_{n-1} - 2 = r_n + 1 + r_{n-1} - r_n + 1 - 2 = r_{n-1}$. Using a similar argument for each r_i, one can see that $\chi(T) = \chi$ and the proof is complete. ■

This lemma shows that there is a one to one correspondence between the set of string representations of binary trees and the set of sequences of non-negative integers of length n, satisfying conditions 1 - 2 of Lemma 1. There

is also a one to one correnspondence between these sets and the set of sequences K_T, satisfying conditions i, ii.

IV. The Number of Transition Structures Generated by a Tree

Using the characteristic of a tree, we can define a weight function on the class of binary trees with the property that heavier trees will generate more transition structures. Using this weight we will show that the number of deterministic transition structures, having a tree as a breadth-first spanning tree is a constant multiple of the tree's weight, where the number of states in n.

Definition 3: Let $T \in \mathcal{T}_n$ be a tree with characteristic $\chi(T) = (r_1, r_2, \ldots, r_{n-1}, r_n)$, for $n \geq 2$. Then the **weight** of T, denoted by $W(T)$, is defined by

$$W(T) = \left(\frac{2}{1}\right)^{r_2} \cdots \left(\frac{n-1}{n-2}\right)^{r_{n-1}} \left(\frac{n}{n-1}\right)^{r_n} = \prod_{k=2}^{n} \left(\frac{k}{k-1}\right)^{r_k}$$

Alternatively, we can define $W(T)$ recursively as

$$W(T) = W(T') \left(\frac{n}{n-1}\right)^{r_n}$$

where T' is the tree derivative of T and $r_n = \chi_t(T)$. If $T \in \mathcal{T}_1$, we let $W(T) = 1$.

Our goal now is to use the weight of a tree T to compute the number of automata having T as their spanning tree. Since trees are not endowed with the notion of final states (or final nodes,) our computations will be concerned with the number of *transition structures*, and not the number of *automata*, having a given tree $T \in \mathcal{T}_n$ as their spanning tree. Denote the former by C_n^T. Then it is easy to see that given C_n^T, the total number of automata having T as their spanning tree is $2^n C_n^T$. This is because, given a transition structure with n nodes/states, there are 2^n choices for the set of final states.

Theorem 4.1: Given a tree $T \in \mathcal{T}_n$, let C_n^T be the number of accessible transition structures with n states that have T as their breadth-first spanning tree. Then

$$C_n^T = (n-1)!W(T).$$

Using the sequence of difference values, $K = (k_1, k_2, \ldots, k_n)$ the equation above can be written as

$$C_n^T = \prod_{i=1}^{n} i^{k_i}$$

Proof We prove this statement by induction on n. For the case $n = 1$, note there is only one tree T with one node and there is only one transition structure having T as a spanning tree. Hence $C_1^T = 1$.

Now assume that $T \in \mathcal{T}_n, n \geq 2$ is a tree with binary representation β_T. Let T' be the tree derivative of T. Then there are $C_{n-1}^{T'}$ transition structures having T' as a spanning tree. In order to find C_n^T, recall from [3] or [1] that

$$C_n^T = \prod_{i=1}^{2n} n_i \text{ with}$$

$$n_i = \begin{cases} \sum_{j<i} \beta_T(j) & \text{if } \beta_T(i) = 0, \\ 1 & \text{if } \beta_T(i) = 1 \end{cases}$$

In other words, n_i counts the number of nodes generated prior to reaching the missing transition i. It is easy to see that the first position where $\beta_T(i)$ and $\beta_{T'}(i)$ differ is at position $2n - r_n$, where $r_n = \chi_t(T)$, specifically, $\beta_T(2n - r_n) = 1$ while $\beta_{T'}(2n - r_n) = 0$, and β_T has an additional 00 at the end. For instance, β_T and $\beta_{T'}$ may be $\beta_T = 1011\ldots1\underbrace{00\ldots0000}_{r_n}$ and $\beta_{T'} = 1011\ldots0\underbrace{00\ldots00}_{r_n-2}$. It is important to notice that the number of transition structures that can be generated from β_T is the same as the number that can be generated from $\beta_{T'}$, except for

1) An over-count of $n-1$ at position $2n - r_n$;
2) For each 0 in positions $2n+1-r_n$ to $2n-2$, $C_{n-1}^{T'}$ has a factor of $n-1$ but C_n^T has a factor of n which can be fixed by multiplication by $\left(\frac{n}{n-1}\right)^{r_n-2}$;
3) An additional factor of n^2 in C_n^T accounting for the last two 0s of β_T.

Hence

$$C_n^T = \left(\frac{n^2}{n-1}\right)\left(\frac{n}{n-1}\right)^{r_n-2} C_{n-1}^{T'} = (n-1)\left(\frac{n}{n-1}\right)^{r_n} C_{n-1}^{T'}.$$

Finally using the previous equation and the inductive hypothesis of $C_{n-1}^{T'} = (n-1)(n-1)!W(T')$ we prove the assertion.

$$C_n^T = (n-1)\left(\frac{n}{n-1}\right)^{r_n} (n-2)!W(T') = (n-1)!W(T)$$

Rewriting the previous equation using the difference terms

$$C_n^T = (n-1)! \left(\frac{2}{1}\right)^{r_2} \cdots \left(\frac{n-1}{n-2}\right)^{r_{n-1}} \left(\frac{n}{n-1}\right)^{r_n}$$
$$= 2^{k_2} 3^{k_3} \cdots n^{k_n} = \prod_{i=1}^{n} i^{k_i}$$

■

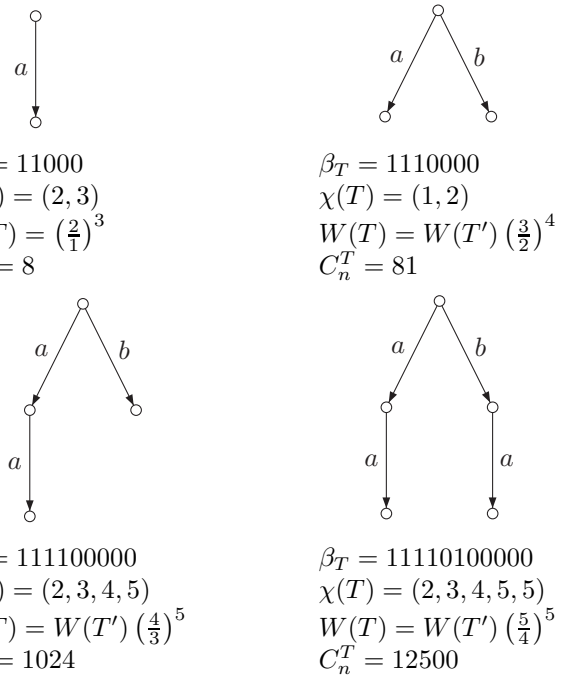
Corollary 4.2: Let $T \in \mathcal{T}_n$, where $n \geq 2$. Then $C_n^T = (n-1)\left(\frac{n}{n-1}\right)^{r_n} C_{n-1}^{T'}$, where T' is the tree derivative of T and $r_n = \chi_t(T)$.

One of the implications that the formulas in Theorem 4.1 and Corollary 4.2 have is the significant difference in the number of automata produced by diffrerent trees.

Lemma 2: For every $n \geq 2$, the tree P producing the smallest number of automata has the form $\chi(P) = (2, 2, \ldots, 2)$ and the tree producing the largest number of automata has the form $\chi(Q) = (2, 3, 4, \ldots, n+1)$. Additionally $C_n^P = n(n!)$ and $C_n^Q = n^{n+1}$.

Proof Through constructing the sequence of difference values for P and Q , notice that $K_Q = (1, 1, \cdots, 2)$ and $K_P = (0, 0, \cdots, 0, n+1)$. Using Theorem 4.1 we can calculate $C_n^P = n^2(n-1)! = n(n!)$ and $C_n^P = n^{n+1}$. Hence the number of transition structures (and hence, the number of automata) having different trees as their spanning tree (even with the same number of nodes) can be asymptotically different.■

Example 4: Using Theorem 4.1 and Corollary 4.2, we can compute C_n^T for different trees T.



$\beta_T = 11000$
$\chi(T) = (2, 3)$
$W(T) = \left(\frac{2}{1}\right)^3$
$C_n^T = 8$

$\beta_T = 1110000$
$\chi(T) = (1, 2)$
$W(T) = W(T')\left(\frac{3}{2}\right)^4$
$C_n^T = 81$

$\beta_T = 111100000$
$\chi(T) = (2, 3, 4, 5)$
$W(T) = W(T')\left(\frac{4}{3}\right)^5$
$C_n^T = 1024$

$\beta_T = 11110100000$
$\chi(T) = (2, 3, 4, 5, 5)$
$W(T) = W(T')\left(\frac{5}{4}\right)^5$
$C_n^T = 12500$

Fig. 3: Calculation of the number of transition structures, C_n^T, having a given T as their spanning tree.

V. Conclusion

We have presented a method, and a formula to calculate the number of automata having a given tree as a spanning subtree. One of the main consequences of this theorem is that the number of transition structures (and thus, the number of automata) generated from different trees, even with the same number of nodes, is not of the same order for different trees. Equipped with the information provided by Theorem 4.1 , and Corollary 4.2, we can estimate quantities like the probability of generating a specific type of automaton given its breadth-first spanning tree. We can also impose a distribution on the set $\mathcal{T}_n$ where trees are generated with modified probabilities such that each automaton can be generated uniformly.

References

[1] M. Almeida, N. Moreira, and R. Reis. Enumeration and generation with a string automata representation. *Theoretical Computer Science*, 387(2):93–102, 2007.

[2] P. Babaali. Generic and structural properties of random automata. *PhD Thesis*. Stevens Institute of Technology, 2007.

[3] P. Babaali. Generating random automata. *Proceedings of The 2009 International Conference on Scientific Computing*, pages 85–89, 2009.

[4] F. Bassino and C. Nicaud. Enumeration and random generation of accessible automata. *Theoretical Computer Science*, 381(1-3):86–104, 2007.

[5] J.-M. Champarnaud and T. Paranthoën. Random generation of DFAs. *Theoretical Computer Science*, 330:221–235, 2005.

[6] M. Domaratzki, D. Kisman, and J. Shallit. On the number of distinct languages accepted by finite automata with n states. *J. Automata Lang. Combin.*, 7:67–78, 2002.

[7] F. Harary and E.M. Palmer. *Graphical enumeration*. Academic Press New York, 1973.

[8] M.A. Harrison. A census of finite automata. In *Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design*, pages 44–46, 1964.

[9] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[10] E. Mäkinen. Generating random binary trees, A survey. *Information Sciences: an International Journal*, 115(1-4):123–136, 1999.

[11] H. Narushima. Principle of inclusion-exclusion on partially ordered sets. *Discrete Mathematics*, 42(2-3):243–250, 1982.

[12] C. Nicaud. Étude du compartement en moyenne des automates finis et des langages rationnels. *PhD Thesis*. University Paris 7, 2000.

Remarks on the Application of the Infinite Unit Axiom to Cellular Automata

Louis D'Alotto
Department of Mathematics and Computer Science
York College/City University of New York
Jamaica, New York 11451
and
The Doctoral Program in Computer Science
CUNY Graduate Center
ldalotto@york.cuny.edu

***Abstract*— In [6], The Infinite Unit Axiom (see [1] - [4]) is applied to the development of one-dimensional cellular automata. This application allows the establishment of a new and more accurate metric on the space of definition for one-dimensional cellular automata. In this paper, the new metric is discussed and shown to increase accuracy of computations.**

keywords: Cellular automata; Infinite Unit Axiom; Grossone; Metric; Classes of Cellular Automata.

I. Introduction

Cellular automata are discrete dynamical systems that are known for their strong modeling and self-organizational properties. Defined on an infinite lattice (in the usual one-dimensional case, an infinite sequence configuration is defined on the integers), even starting with complete disorder, evolution of cellular automata maps can generate organized structure. Originally developed by Von Neuman in the 1940's to model biological self-reproduction, cellular automata have long been used in computational, physical, and biological applications. For a more complete description of applications of cellular automata, see [5], [10], [13], [14], [15], and [19]. Cellular automata can be defined for any dimension greater than or equal to one. This paper is concerned with one-dimensional or linear cellular automata defined on the integers (when no confusion arises, we will refer to one-dimensional cellular automata as simply cellular automata). As with all dynamical systems, it is interesting to understand the long term behavior under forward time evolution and achieve an understanding or classification of the system.

The concept of classifying cellular automata was initiated by Stephen Wolfram, see [17]. Through numerous computer simulations, Wolfram noticed that if an initial configuration was chosen at random the probability is high that a cellular automaton rule will fall within one of four classes. In [17], one-dimensional cellular automata are partitioned into four classes depending on their dynamical behavior. A later (more rigorous) classification scheme, see [12], was developed by Robert Gilman. Here a probabilistic/measure theoretic classification scheme was developed based on the probability of choosing a sequence that will stay arbitrary close to a given initial sequence under forward evolution (iteration). Gilman uses a metric that considers the central window where two sequences agree and continue to agree upon forward iterations of a cellular automata map. However, in the development, this metric is limited because it doesn't take into account sequences that agree on an infinite interval to the right (or respectively to the left). Indeed, the metric considers the absolute value of the first integral place where sequences disagree and uses that as their distance apart. For example, if sequences agree on the right hand side out to infinity their distance apart is determined by where they disagree on the left. In this paper, the definition of cellular automata and the metric involved are extended to include sequences that do not necessarily agree on a finite central window, symmetric around 0, but which can agree on, not necessarily symmetric infinite intervals.

The classical concept of infinity has presented limitations in computations. Indeed, metrics used on infinite sequences, and hence cellular automata, either do not allow us to observe minute differences or can lead to calculations beyond finite computations. Analogous to the Hamming distance for finite sequences, the following metric is used to compute distances between infinite sequences.

$$d(x,y)=\sum_{i=-\infty}^{\infty}\frac{|x(i)-y(i)|}{2^{|i|}}$$

Here the differences in the respective sequence values are computed and divided by $2^{|i|}$ to assure convergence. However, this procedure can lead to a calculation beyond finite computation and to possible inaccuracies. For instance, using the binary alphabet S=$\{0,1\}$, suppose two sequences agree completely on the left of 0 and at 0. That is, they disagree on the right of 0 or for integral values $i>-1$. Applying the traditional well known formula

$$\sum_{i=0}^{k}\frac{1}{2^{|i|}}=2-\frac{1}{2^k}$$

and taking limits as k approaches infinity, results in a value of 2. By using the *infinite unit axiom*, see [1], [2], [3], [4], and $|\mathbb{N}|=①$, the computational limitations caused by sequences

that agree out to one-sided infinity or that are subject to infinite computations are overcome. As shown for infinite k, that is for $k = ①$

$$\sum_{i=0}^{①} \frac{1}{2^{|i|}} = 2 - \frac{1}{2^{①}} \tag{1}$$

and $\frac{1}{2^{①}}$ is infinitesimal. Hence the classical computation presents inaccuracies and yields to the more accurate computation above in 1.

Before defining cellular automata with the infinite metric a few notational preliminaries are necessary. The set of integers is denoted by $\mathbb{Z}$; $\mathbb{N}$ is the set of natural numbers and let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Given a finite alphabet S with two or more symbols, i.e. $|S| \geq 2$, consider the space of all functions from the integers to the finite alphabet, i.e. $S^{\mathbb{Z}}$. This space may also be considered as the space of all bi-infinite sequences, defined on the integers, with values taken from the alphabet S. In [12] the following metric was used on the space $S^{\mathbb{Z}}$. Let

$$d(x, y) = 2^{-n}, \ \ where\ n = \inf\{|i| \ : x(i) \neq y(i)\} \tag{2}$$

It is noted that this metric satisfies the ultrametric property. A metric $d(x, y)$ satisfies the ultrametric property iff it is a metric and obeys the inequality:

$$d(x, y) \leq \max\left[d(x, z), d(z, y)\right]$$

A space that satisfies the ultrametric inequality is also called a nonarchimedean space. It is obvious that the triangle inequality is implied by the ultrameric inequality. As can be seen, the metric defined in (2) above considers only the symmetric window around $i = 0$. Two sequences may agree in more integral values on one side but this information is not communicated by the metric. In this paper this inaccuracy is overcome by using the Infinite Unit Axiom, $|\mathbb{N}| = ①$, and developing new machinery to work with infinite sequences.

Let S be a finite alphabet of size $s \geq 2$ and let $X = (S \cup \{*\})^{\mathbf{Z}}$. X is the set of all maps from the integers to $S \cup \{*\}$. That is, for $x \in X$, $x : \mathbb{Z} \to S \cup \{*\}$. It is noted that the set $S \cup \{*\}$ is compact and hence the product space X is also compact. The Infinite Unit Axiom is applied and used in the construction of the metric for computations with infinite configurations. It is shown in [6] that the following metric is an ultrametric and the space is nonarchimedean.

Definition 1: Let

$$x \wedge y = \begin{cases} x & if\ \ x = y \\ * & if\ \ x(0) \neq y(0)\ or\ x(0) = * \\ x(m)...x(0)...x(n) & if\ \ x(i) = y(i) \forall i \in [m, n] \\ & \quad and\ * outside \end{cases}$$

Note: $m \leq 0$ and can equal $-①$, similarly $n \geq 0$ can equal $①$. Hence computations on infinite configurations are allowed. Thus, $x \wedge y$ is the place where two configurations agree on the largest stretch around 0 and is $*$ valued outside.

Definition 2:

$$F(x \wedge y) = \begin{cases} 1 & if\ x \wedge y = * \\ 2^{-(n+1-m)} & if\ x \wedge y = \\ & \quad ... * * * x(m)...x(0)...x(n) * * * ... \end{cases}$$

We form the following metric on the space of bi-infinite configurations:

$$d(x, y) = \begin{cases} 0 & if\ \ x = y \\ F(x \wedge y) & otherwise \end{cases}$$

The restriction of $x \in \mathbf{X}$ to a non-empty interval $[i, j]$ of $\mathbb{Z}$, where $-① \leq i \leq j \leq ①$ is called a *word*. Words are written $x[i, j]$. The length of a word $w = x[i, j]$ is $|w| = j - i + 1$. It is important to note that, using ①, words (or the length of a word) can be infinite, however cannot have an endpoint greater than ① (nor less than $-①$). Also, for any $a \in S$, define $x_a \in X$ by $x_a(i) = a$, for $i \in \mathbb{Z}$.

Example 1 shows how sequences can agree on infinite words and their distance computed.

Example 1: Given $S = \{0, 1\}$, let $x = ...111\langle 1\rangle 111...$ and $y = ...00011\langle 1\rangle 111...$ In the examples, when not explicitly denoted, we will use the symbol $\langle\ \rangle$ to denote the zeroth place. x and y agree completely on the right hand side, and at integral values $0, -1$, and -2.

$$x \wedge y = ... * * * x(-2)x(-1)x(0)....x(n)....x(①)$$

$$F(x \wedge y) = 2^{-(①+1-(-2))}$$

and

$$d(x, y) = \frac{1}{2^{①+3}}$$

Hence, the distance between the two points x and y is infinitesimal. As Example 2 shows, the above construction easily covers the finite word case.

Example 2: Again, using the binary alphabet $S = \{0, 1\}$, let $x = ...1110\langle 1\rangle 0111...$ and $y = ...1110\langle 1\rangle 0101...$

$$x \wedge y = ... * * * x(-1)x(0)x(1)x(2) * * * ...$$

That is, the sequences differ in the -2 and 3^{rd} integral positions. Hence,

$$F(x \wedge y) = 2^{-(2+1-(-1))} = 2^{-4}$$

and

$$d(x, y) = \frac{1}{2^4}$$

II. Cellular Automata

As before, S is an alphabet of size s such that $s \geq 2$ and let $X = S^{\mathbb{Z}}$, i.e. the set of all maps from the lattice $\mathbb{Z}$ to the set S. That is, for $x \in X$, $x : \mathbb{Z} \to S$. Cellular automata are induced by arbitrary (local) maps:

$$F : S^{(2r+1)} \longrightarrow S$$

These are usually called local rules or block maps in the literature, see [9] and [12]. The value $r \in \mathbb{N}_0$ is called the range of the map. The automaton map f induced by F is defined by $f(x) = y$ with

$$y(i) = F[x(i-r), ..., x(i+r)]$$

To illustrate the importance of discrete time steps in the forward evolution of the automaton, we will use the following formula where t represents time.

$$y(i)_{t+1} = F[x(i-r)_t, ..., x(i+r)_t]$$

The following is a simple, but important, example of a cellular automaton of range $r = 1$. The evolutionary behavior of this automaton is clearly exhibited.

Example 3: Let $S = \{0, 1\}$ and let f be the automaton induced by the local rule $F : S^3 \to S$ by $F(1, 1, 1) = 1$ and $F(a, b, c) = 0$ otherwise. If we apply forward iterations of the induced automaton map f, all sequences eventually go to the quiescent state of x_0, except for the initial sequence x_1 which remains constant. In Example 3, given any finite or infinite word $x[i, j]$ with at least one element in the word not equal to 1, the configuration will eventually evolve, under forward iterations, to the quiescent state of x_0. There are numerous other examples of cellular automata maps. A more chaotic rule can be seen via the following example.

Example 4: Let $S = \{0, 1\}$ and let f be the automaton induced by the local rule $F : S^3 \to S$ by $F(a, b, c) = (a + c) mod\ 2$. Applying forward iterations of the induced automaton map f yields no particular pattern. Beginning with an initial random configuration in $S = \{0, 1\}^{\mathbb{Z}}$ can yield many different configuration sequences.

The following theorem shows that the number of configurations in the definition space of cellular automata can now be determined. The proof is given in [6].

Theorem 1: Given the space $S^{\mathbb{Z}}$ of bi-infinite sequences, the number of elements $x \in S^{\mathbb{Z}}$ is equal to $|S|^{2①+1}$.

III. Conclusion

In this paper, the framework for defining and working with cellular automata has been extended by applying the Infinite Unit Axiom, $|\mathbb{N}| = ①$. In the classical sense, the space $S^{\mathbb{Z}}$ is considered uncountable and beyond our computational abilities. Usual metrics on the space $S^{\mathbb{Z}}$ (the space of definition for cellular automata) are limited in accuracy. Indeed, configurations can agree on infinite intervals and not have this information communicated by the metric. This loss of information has also been overcome by applying the Infinite Unit Axiom to the development of a new metric.

References

[1] Sergeyev, Y.D., *Arithmetic of Infinity*, Edizioni Orizzonti Meridionali, Italy, 2003.

[2] Sergeyev, Y.D. (2009) Numerical Computations and Mathematical Modelling with Infinite and Infinitesimal Numbers, *Journal of Applied Mathematics and Computing*, 29, 177-195.

[3] Sergeyev, Y.D. (2009) Numerical Point of View on Calculus for Functions Assuming Finite, Infinite, and Infinitesimal Values Over Finite, Infinite, and Infinitesimal Domains, *Nonlinear Analysis Series A: Theory, Methods & Applications*, 71(12), e1688-e1707.

[4] Sergeyev, Y.D. (2008) A New Applied Approach for Executing Computations with Infinite and Infinitesimal Quantities, *Informatica*, 19(4), 567-596.

[5] P. Manneville, N. Boccara, G.Y. Vichniac, and R. Bidaux, editors, *Cellular Automata and Modeling of Complex Physical Systems*, Springer-Verlag, Berlin, 1989.

[6] L. D'Alotto (2010), Cellular Automata Using Infinite Computations, preprint.

[7] L. D'Alotto, C. Giardina (2003), A Metric Generated by a Lower Tree Semi-Lattice, International Conference-Artificial Intelligence 2003 Proceedings, Vol. 2, 938-940, CSREA Press.

[8] L. D'Alotto, C. Giardina (1994), The Kolmogorov Metric and a Generalization on a Classification of Cellular Automata, *International Journal on Artificial Intelligence Tools*, Vol.3 No. 3 (1994) 311-326.

[9] Hedlund, G.A.,"Edomorphisms and Automorphisms of the Shift Dynamical System", Math. Sys. Theory **3** (1969), 51-59.

[10] McIntosh, H.V., *One Dimensional Cellular Automata*, Luniver Press, United Kingdom, 2009.

[11] L. Narici, E. Beckenstein, G. Bachman, *Functional Analysis and Valuation Theory*, Marcel Dekker, New York, 1971.

[12] R. Gilman (1987), Classes of Linear Automata, *Ergodic Theory and Dynamical Systems*, 7, 105-118.

[13] A. Ilachinski, *Cellular Automata, A Discrete Universe*, World Scientific Publishers, Singapore, (2001).

[14] G. Zito, D. D'Ambrosio, W. Spataro, et al. (2009), A Dynamically Load Balanced Cellular Automata Library for Scientific Computing, International Conference on Scientific Computing Proceedings, 322-328, CSREA Press.

[15] G. Ch. Sirakoulis, I. Krafyllidis, W. Spataro (2009), A Computational Intelligent Oxidation Process Model and its VLSI Implementation, International Conference on Scientific Computing Proceedings, 329-335.

[16] Wolfram, S. (1983), *Statistical Mechanics of Cellular Automata*. Reviews of Modern Physics. Vol. 55, No. 3, 601-644.

[17] S. Wolfram (1984), *Universality and Complexity in Cellular Automata*. Physica 10D 1-35.

[18] S. Wolfram (1984), *Computation Theory of Cellular Automata* . Communications in Mathematical Physics, 96, 15-57.

[19] Wolfram, S., *A New Kind of Science*, Wolfram Media, Inc., IL, (2002).

Dynamics of Wolfram's Class III Cellular Automaton Rule 73

Jing Chen, Fangyue Chen, Yunfeng Bian, and Wei Chen
School of Science, Hangzhou Dianzi University, Hangzhou, Zhejiang, P. R. China

Abstract— *In this paper, the dynamics of elementary cellular automaton (ECA) rule 73 is investigated under the framework of the bi-infinite symbolic sequence space. This paper provides a rigorous mathematical analysis for the evolution of symbol sequences in some subsystems of rule 73. ECA rule 73, a member of Wolfram's class III and Chua's complex Bernoulli-shift rules, defines many more subsystems with rich and complicated dynamical properties such as topologically mixing, topologically transitivity and positive topological entropy, and henceforth the dynamical system generated by the global map of the rule is chaotic in the sense of both Devaney and Li-Yorke.*

Keywords: cellular automata; complex Bernoulli-shift CA rule; symbolic dynamics; topologically mixing; chaos.

1. Introduction

Cellular automata (CA) was introduced by J. von Neumann and S. Ulam in the 1940s to 1950s [1]. In the late 1960s, J. Conway proposed his now-famous Game of Life, which shows the great potential of CA in the simulation of complex systems [2]. The topological dynamics of CA began in 1969 with G. Hedlund who viewed one-dimensional CA in the context of symbolic dynamics as endomorphisms of the shift dynamical systems. His main results are the characterization of surjective and open CA [3]. Since the 1980s, S. Wolfram focused on the analysis of dynamical systems and studied CA in detail [4-6], and in 2002, he introduced his monumental work *A New kind of Science* [7]. Wolfram classified CA into 4 classes based on extensive computer simulations: **(I)** CA evolving to a homogeneous state; **(II)** CA evolving periodically; **(III)** CA evolving chaotically and **(IV)** also known as a class of complex rules with rich behaviors, including all previous cases.

Since 2002, L. O. Chua *et al.* provided a nonlinear dynamics perspective to Wolfram's empirical observations from the viewpoint of mathematical analysis via the concepts like characteristic function, forward time-τ map, basin tree diagram, Isle-of-Eden digraph and so on [8-12]. It was known that there are 256 elementary cellular automata (ECA) rules, only 88 rules are globally independent from each other [9-10, 13]. These 88 global independent ECA rules are also organized into 4 groups with distinct qualitative dynamics: 40 period-k ($k = 1, 2, 3, 6$), 30 topologically distinct Bernoulli shift rules, 10 complex Bernoulli shift rules and 8 hyper Bernoulli shift rules [9-10].

This paper considers the dynamics of rule 73 under the framework of bi-infinite symbolic sequence space. The rule, a member of Wolfram's class III and Chua's complex Bernoulli-shift rules, defines many more subsystems with rich and complicated dynamical properties such as topologically mixing, topologically transitivity and positive topological entropy. This means its global map is chaotic in the sense of both Devaney and Li-Yorke.

The structure of this paper is organized as follows: Section 2 presents the basic concepts of symbolic dynamical systems and CA, and obtains many subsystems of rule 73. Section 3 explores the topological dynamics of rule 73. Finally, Section 4 concludes the paper and prospects for future studies.

2. Preliminaries and subsystems

2.1 Preliminaries of symbolic dynamical systems and CA

The bi-infinite binary symbols sequence space is a configuration set on $S = \{0, 1\}$:

$$\Sigma_2 = \{x = (\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) \mid x_i \in S = \{0, 1\}, i \in Z\}$$

and the metric "d" on Σ_2 defined as $d(x, y) = \max\{\frac{\rho(x_i, y_i)}{2^{|i|}}\}$ for any x, $y \in \Sigma_2$, where $\rho(\cdot, \cdot)$ is the metric on S defined as

$$\rho(x_i, y_i) = \begin{cases} 0, & if \quad x_i = y_i \\ 1, & if \quad x_i \neq y_i. \end{cases}$$

It is known that Σ_2 is a compact, perfect and totally disconnected metric space.

If $x \in \Sigma_2$ and $I = [i, j]$ is an interval of integers, put $x_{[i,j]} = (x_i, x_{i+1}, \cdots, x_j)$ $(i < j)$, $x_{[i,j)} = (x_i, \cdots, x_{j-1})$. Let $x_{(-\infty,i]} = (\cdots, x_{i-1}, x_i)$ and $x_{[j,+\infty)} = (x_j, x_{j+1}, \cdots)$ denote the left and right half infinite subsequence respectively. For a finite sequence $a = (a_0, \cdots, a_{n-1})$, if there exists an $n \in Z$ such that $x_{n+k} = a_k$ $(k = 0, 1, \cdots, n-1)$, then a is a subword of x, denoted by $a \prec x$; otherwise $a \not\prec x$.

The left-shift σ_L and right-shift σ_R are defined by

$$\sigma_L(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) = (\cdots, x_0, \overset{*}{x}_1, x_2 \cdots)$$

and

$$\sigma_R(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) = (\cdots, x_{-2}, \overset{*}{x}_{-1}, x_0, \cdots)$$

respectively.

By a theorem of Hedlund [3], a map $f : \Sigma_2 \to \Sigma_2$ is a cellular automaton iff it is continuous and commutes

Table 1: Truth table of Boolean function of Rule 73

(x_{i-1}, x_i, x_{i+1})	$\hat{f}(x_{i-1}, x_i, x_{i+1})$
(0, 0, 0)	1
(0, 0, 1)	0
(0, 1, 0)	0
(0, 1, 1)	1
(1, 0, 0)	0
(1, 0, 1)	0
(1, 1, 0)	1
(1, 1, 1)	0

with σ, i.e. $\sigma \circ f = f \circ \sigma$, where σ is left-shift or right-shift. Moreover, for any CA f, (Σ_2, f) defines a dynamical system. A set $X \subseteq \Sigma_2$ is f-invariant if $f(X) \subseteq X$, and strongly f-invariant if $f(X) = X$. If X is a closed and f-invariant, then (X, f) or simply X is called a subsystem of f.

Each ECA rule can be expressed by a Boolean function. For example, the one of rule 73 is a local map $\hat{f}$:

$$\hat{f}(x_{i-1}, x_i, x_{i+1}) = \bar{x}_{i-1} \cdot x_i \cdot x_{i+1} \oplus x_{i-1} \cdot x_i \cdot \bar{x}_{i+1} \oplus \bar{x}_{i-1} \cdot \bar{x}_i \cdot \bar{x}_{i+1}, i \in Z,$$

where "$\cdot$", "$\oplus$" and "$-$" stand for "AND", "XOR" and "NOT" logical operations, respectively [7, 13]. The truth table of its Boolean function is shown in Table 1.

It is clear that its binary output sequence is 10010010. Thus, a global map $f_{73} : \Sigma_2 \to \Sigma_2$ with

$$f_{73}(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) = (\cdots, y_{-1}, \overset{*}{y}_0, y_1, \cdots)$$

can be induced by $\hat{f}$, where $y_i = \hat{f}(x_{i-1}, x_i, x_{i+1})$.

The n $(n \geq 2)$ times iteration of $\hat{f}$ is a map $\hat{f}^n$ from $\{0,1\}^{2n+1}$ to $\{0,1\}$ with

$$\hat{f}^n(a_{-n}, \cdots, a_0, \cdots, a_n) = \hat{f}(\hat{f}^{n-1}(a_{[-n,n-2]}), \hat{f}^{n-1}(a_{[-n+1,n-1]}), \hat{f}^{n-1}(a_{[-n+2,n]})).$$

2.2 Invariant subsets and subsystems of rule 73

In this subsection, some f_{73}-invariant subsets and subsystems of rule 73 are revealed.

Proposition 1: For rule 73, there is a f_{73}-invariant subset $\Lambda_0 \subset \Sigma_2$ such that $\forall\ x \in \Lambda_0$, $f_{73}(x) = x$, where $\Lambda_0 = \Lambda_{\mathscr{A}_0} = \{x \in \Sigma_2 | x_{[i-1,i+1]} \in \mathscr{A}_0, \forall\ i \in Z\}$, and $\mathscr{A}_0 = \{(0,0,1), (0,1,1), (1,0,0), (1,0,1), (1,1,0)\}$.

Remark 1: Obviously, Λ_0 is the set of fixed points of f_{73}, and $\mathscr{A}_0$ is the determinative block system of Λ_0, which is a 3-sequence set. It is clear that there are infinitely many fixed points of f_{73}, and furthermore, there is a subset $\tilde{\Lambda}_0 \subset \Lambda_0$ such that there is an one to one corresponding between $\tilde{\Lambda}_0$ and Σ_2, i.e., $\tilde{\Lambda}_0$ is equivalent to Σ_2. Thus, this proposition gives a rigorous mathematical analysis for the evolution of the stationary symbol sequences of rule 73 described in Wolfram's *A New kind of Science* [7]. For convenience, $\mathscr{A}_0$ can be denoted by its decimal code set $D(\mathscr{A}_0) = \{1, 3, 4, 5, 6\}_3$. This representation will be also applied to the determinative block systems in the following propositions.

Proposition 2: For rule 73, there is a f_{73}-invariant subset $\Lambda^* \subset \Sigma_2$ such that $f_{73}^2(x) = x,\ x \in \Lambda^*$, where $\Lambda^* = \{(1,0,0,0)^*\}$, and $(1,0,0,0)^*$ stands for the cycle configuration $(\cdots, \underline{1,0,0,0}, \underline{1,0,0,0}, \underline{1,0,0,0}, \cdots)$.

Remark 2: In Proposition 2 and the following discussion, if any symbol of a configuration $x \in \Sigma_2$ is not designated as the 0-th position, then it means any symbol of x can be designated as the position. For example, the cycle configuration $(1,0,0,0)^* = (\cdots, \underline{1,0,0,0}, \cdots)$ can stand for $(\cdots, \underline{\overset{*}{1},0,0,0}, \cdots)$, $(\cdots, \underline{1,\overset{*}{0},0,0}, \cdots)$, $(\cdots, \underline{1,0,\overset{*}{0},0}, \cdots)$ or $(\cdots, \underline{1,0,0,\overset{*}{0}}, \cdots)$.

In fact, Λ^* is an Isle of Eden for rule 73 [11-12].

Proposition 3: For rule 73, there are two f_{73}-invariant subsets $\Lambda_L^*, \Lambda_R^* \subset \Sigma_2$ such that $f_{73}^3(x) = x,\ x \in \Lambda_L^*$ or $x \in \Lambda_R^*$, where $\Lambda_L^* = \{(1,0,1,0,0,0,0,0,1,1,1,0)^*\}$ and $\Lambda_R^* = \{(0,1,1,1,0,0,0,0,0,1,0,1)^*\}$.

It is easily known that Λ^*, Λ_L^* and Λ_R^* are finite subsets of Σ_2, and $\Lambda_R^* = \{x \in \Sigma_2 | x_i = y_{-i}, y \in \Lambda_L^*, i \in Z\}$, where $x = (\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots)$ and $y = (\cdots, y_{-1}, \overset{*}{y}_0, y_1, \cdots)$.

Proposition 4: For rule 73, there exists a f_{73}-invariant subset $\Lambda_L' \subset \Sigma_2$ such that $f_{73}^5(x) = \sigma_L^2(x),\ x \in \Lambda_L'$, where $\Lambda_L' = \Lambda_{\mathscr{A}_1} = \{x \in \Sigma_2 | x_{[i-5,i+5]} \in \mathscr{A}_1, \forall\ i \in Z\}$, and the determinative block system $\mathscr{A}_1$ is a 11-sequence set, whose decimal code set is
$D(\mathscr{A}_1) = \{\ 1, 2, 4, 8, 16, 21, 32, 42, 64, 65, 84, 85, 129, 130, 169,\ 170,\ 224,\ 240, 258, 261, 338, 341, 383, 448, 480, 517, 522,\ 607,\ 663, 677, 683, 687, 703, 766, 767, 896, 960, 1016, 1020, 1024, 1025, 1034, 1045, 1136, 1144, 1215, 1327, 1355, 1367, 1375, 1407, 1532, 1534, 1536, 1592, 1596, 1792, 1820, 1822, 1920, 1934, 1935, 1991, 2019, 2033, 2040\}_{11}$.

Proof: In fact, for any $x \in \Lambda_L'$, one has $x_{[i-5,i+5]} \in \mathscr{A}_1$. For the map $\hat{f}_{73}^5 : \{0,1\}^{11} \to \{0,1\}$, it can be verified that $\hat{f}_{73}^5(x_{[i-5,i+5]}) = \hat{f}_{73}^5(x_{i-5}, \cdots, x_i, \cdots, x_{i+5}) = x_{i+2}$ for any $x_{[i-5,i+5]} \in \mathscr{A}_1$, i.e., $[f_{73}^5(x)]_i = x_{i+2}$ for $x \in \Lambda_L'$, where $[f_{73}^5(x)]_i$ denotes the i-th symbol of $f_{73}^5(x)$. This leads to $f_{73}^5(x) = \sigma_L^2(x)$ for $x \in \Lambda_L'$. It is easily validated that $f_{73}(x) \in \Lambda_L'$ for $x \in \Lambda_L'$. Thus, $f_{73}(\Lambda_L') \subset \Lambda_L'$ and $f_{73}^5|_{\Lambda_L'} = \sigma_L^2|_{\Lambda_L'}$. ■

Proposition 5: For rule 73, there exists a f_{73}-invariant subset $\Lambda_R' \subset \Sigma_2$ such that $f_{73}^5(x) = \sigma_R^2(x),\ x \in \Lambda_R'$, where $\Lambda_R' = \Lambda_{\mathscr{A}_2} = \{x \in \Sigma_2 | x_{[i-5,i+5]} \in \mathscr{A}_2, \forall\ i \in Z\}$, and the determinative block system $\mathscr{A}_2$ is a 11-sequence set, whose decimal code set is
$D(\mathscr{A}_2) = \{\ 1, 3, 7, 14, 15, 16, 28, 30, 32, 56, 60, 64, 113, 120,$

$128, 227, 241, 254, 255, 256, 336, 455, 483, 509, 510, 512, 516, 520, 596, 641, 642, 672, 680, 911, 967, 1018, 1021, 1024, 1025, 1032, 1040, 1151, 1192, 1282, 1284, 1322, 1344, 1345, 1360, 1364, 1599, 1685, 1706, 1823, 1866, 1877, 1935, 1957, 1962, 2002, 2005, 2025, 2026, 2036, 2037, 2042\}_{11}$.

The proof of proposition 5 is similar to proposition 4, the details are omitted here.

Proposition 6: (1) There exists a f_{73}^{10}-invariant subset $\Lambda_L \subset \Sigma_2$ such that $f_{73}^{10}(x) = \sigma_L^4(x),\ x \in \Lambda_L$, where $\Lambda_L = \Lambda_{\bar{\mathscr{A}}} = \{x \in \Sigma_2 | x_{[i-10,i+10]} \in \bar{\mathscr{A}}, \forall\ i \in Z\}$, and the determinative block system $\bar{\mathscr{A}}$ is a 21-sequence set, whose decimal code set is
$D(\bar{\mathscr{A}})$= $\{59918, 119837, 239674, 479349, 958698, 1917396, 1737640, 1378128, 659104, 1318209, 539267, 1078535, 119836, 239672, 479344, 958688, 1917376, 1737600, 1378048, 658944, 1317888, 538624, 1077248, 57344, 114688, 229377, 458754, 917509, 1835018, 1572884, 1048616, 80, 160, 321, 643, 1287, 2574, 5149, 10298, 20597, 41194, 82388, 164776, 329552\}_{21}$.

(2) $\Lambda_L'' = \Lambda_L \cup f_{73}(\Lambda_L) \cup f_{73}^2(\Lambda_L) \cup \cdots \cup f_{73}^9(\Lambda_L)$ is a f_{73}-invariant subset, and $f_{73}^{10}(x) = \sigma_L^4(x),\ x \in \Lambda_L''$. Furthermore, $\Lambda_L'' = \{x \in \Sigma_2 | x_{[i-10,i+10]} \in \mathscr{A}''\}$, and the determinative block system $\mathscr{A}''$ is a 21-sequence set, whose decimal code set is
$D(\mathscr{A}'') =$
$\{59918, 119837, 239674, 479349, 958698, 1917396, 1737640, 1378128, 659104, 1318209, 539267, 1078535, 119836, 239672, 479344, 958688, 1917376, 1737600, 1378048, 658944, 1317888, 538624, 1077248, 57344, 114688, 229377, 458754, 917509, 1835018, 1572884, 1048616, 80, 160, 321, 643, 1287, 2574, 5149, 10298, 20597, 41194, 82388, 164776, 329552, 1917397, 1737643, 1378135, 659119, 1318239, 539327, 1078655, 60159, 120319, 240639, 481279, 962559, 1925118, 1753084, 1409016, 720880, 1441760, 786369, 1572739, 1048327, 2096654, 2096157, 2095162, 2093173, 2089194, 2081236, 2065320, 2033488, 1969824, 1842497, 1587843, 539266, 1078532, 59912, 119824, 239648, 479296, 958592, 1917184, 1737216, 1377280, 657409, 1314818, 532485, 1064971, 32791, 65582, 131165, 262330, 524661, 1049322, 1492, 2984, 5968, 11936, 23873, 47747, 95495, 190990, 381981, 763962, 1527925, 479345, 958691, 1917383, 1737615, 1378079, 659007, 1318015, 538879, 1077758, 58364, 116728, 233457, 466914, 933829, 1867658, 1638164, 1179176, 261200, 522400, 1044801, 2089603, 2082055, 2066958, 2036765, 1976378, 1855605, 1614058, 1130964, 1737642, 1378133, 659114, 1318229, 539306, 1078612, 60072, 120144, 240288, 480576, 961153, 1922306, 1747461, 1397770, 698388, 1396776, 696400, 1392800, 688449, 1376899, 656647, 1313294, 529437, 1058874, 1318208, 539264, 1078528, 59904, 119809, 239619, 479239, 958479, 1916958, 1736764, 1376376, 655600, 1311200, 525249, 1050499, 3847, 7694, 15389, 30778, 61557, 123114, 246228, 492456, 984912, 59919, 119839, 239679, 479359, 958718, 1917437, 1737722, 1378292, 659433, 1318866, 540581, 1081163, 65175, 130350, 260701, 521402, 1042805, 2085610, 2074068, 2050984, 2004816, 1912480, 1727809, 1358467, 619783, 1239566, 479348, 958696, 1395624, 694096, 1388192, 1917392, 1737632, 1378113, 659074, 1318148, 539144, 1078288, 59424, 118848, 237697, 475394, 950789, 1901578, 1706004, 1314856, 532560, 1065120, 33089, 66179, 132359, 264718, 1378129, 659107, 1318215, 539278, 1078556, 59960, 119921, 239843, 479687, 959375, 1918750, 1740348, 1383544, 669936, 1339872, 582593, 1165187, 233223, 466446, 932893, 1865786, 1634421, 1171690, 1078533, 59914, 698741, 1397482, 697812, 119829, 239658, 479317, 958634, 1917269, 1737386, 1377620, 658089, 1316178, 535205, 1070411, 43671, 87342, 174685, 349370, 679233\}_{21}$.

Proposition 7: (1) There exists a f_{73}^{10}-invariant subset $\Lambda_R \subset \Sigma_2$ such that $f_{73}^{10}(x) = \sigma_R^4(x),\ x \in \Lambda_R$, where $\Lambda_R = \Lambda_{\bar{\bar{\mathscr{A}}}} = \{x \in \Sigma_2 | x_{[i-10,i+10]} \in \bar{\bar{\mathscr{A}}}, \forall\ i \in Z\}$, and the determinative block system $\bar{\bar{\mathscr{A}}}$ is a 21-sequence set.
(2) $\Lambda_R'' = \Lambda_R \cup f_{73}(\Lambda_R) \cup f_{73}^2(\Lambda_R) \cup \cdots \cup f_{73}^9(\Lambda_R)$ is a f_{73}-invariant subset, and $f_{73}^{10}(x) = \sigma_R^4(x),\ x \in \Lambda_R''$. Furthermore, $\Lambda_R'' = \{x \in \Sigma_2 | x_{[i-10,i+10]} \in \bar{\mathscr{A}}''\}$, and the determinative block system $\bar{\mathscr{A}}''$ is a 21-sequence set.

Remark 3: Due to space limitations, the proofs of Propositions 6 and 7 and the expressions of the decimal code sets of the determinative block systems $\bar{\bar{\mathscr{A}}}$ and $\bar{\mathscr{A}}''$ in Propositions 7 are omitted here.

Proposition 8: $\Lambda_L^* \subset \Lambda_L'$, $\Lambda_R^* \subset \Lambda_R''$.

From Proposition 1 to 7, eight subsystems of f_{73}: (Λ_0, f_{73}), (Λ^*, f_{73}), (Λ_L^*, f_{73}), (Λ_R^*, f_{73}), (Λ_L', f_{73}), (Λ_R', f_{73}), (Λ_L'', f_{73}) and (Λ_R'', f_{73}) are obtained. Additionally, there are symmetrical relations between Λ_R' and Λ_L', and between Λ_R'' and Λ_L''.

Proposition 9: $\Lambda_R' = \{x \in \Sigma_2 | x_i = y_{-i}, y \in \Lambda_L', i \in Z\}$, $\Lambda_R'' = \{x \in \Sigma_2 | x_i = y_{-i}, y \in \Lambda_L'', i \in Z\}$, where $x = (\cdots, x_{-1}, \overset{*}{x_0}, x_1, \cdots)$ and $y = (\cdots, y_{-1}, \overset{*}{y_0}, y_1, \cdots)$.

3. Topological dynamics of rule 73

3.1 Bernoulli-shift subsystem and Subshift of finite type

Definition *1:* (1) $\Lambda \subset \Sigma_2$ is a f-invariant subset, (Λ, f) is called a Bernoulli-shift subsystem if there exists an integer pair (q, p) with $p \geq q > 1$, such that $f^p(x) = \sigma^q(x), x \in \Lambda$, where σ is the left-shift σ_L or right-shift σ_R.

(2) If $\Lambda = \Lambda_{\mathscr{A}} = \{x \in S^Z |\ x_{[i-p,i+p]} \in \mathscr{A}, \forall\ i \in Z\}$, and the determinative block system $\mathscr{A}$ of Λ is a $(2p+1)$-sequence set, then the subsystem (Λ, f) is called a subshift of finite type of f.

If (Λ, f) is a subshift of finite type, let $\Lambda = \Lambda_{\mathscr{A}}$, then $\Lambda_{\mathscr{A}}$ can be described by a finite directed graph $G_{\mathscr{A}} = \{\mathscr{A}, \mathscr{E}\}$, where each vertex is labeled by a sequence in $\mathscr{A}$, and $\mathscr{E}$ is the edge set. Two vertices $a = (a_0, \cdots, a_{n-1})$ and $b = (b_0, \cdots, b_{n-1})$ are connected by an edge of $\mathscr{E}$ if and only if $a_k = b_{k-1}, k = 1, 2, \cdots, n-1$. Every edge $(a_0, \cdots, a_{n-1}) \to (b_0, \cdots, b_{n-1})$ of $\mathscr{E}$ is labeled by b_{n-1}. One can think of each element of $\Lambda_{\mathscr{A}}$ as a bi-infinite path on the graph $G_{\mathscr{A}}$. Whereas a directed graph corresponds to a square transition matrix $A = (A_{ij})_{m \times m}$ with $A_{ij} = 1$ if and only if there is an edge from vertex $b^{(i)}$ to vertex $b^{(j)}$, where $m = |\mathscr{A}|$ is the number of elements in $\mathscr{A}$, and i (or j) is the code of the vertex in $\mathscr{A}$, $i, j = 0, 1, \cdots, m-1$. Thus, $\Lambda_{\mathscr{A}}$ is precisely defined by the transition matrix A.

Remarkably, a square matrix A is irreducible if, for any i, j, there exists an n such that $A^n_{ij} > 0$; aperiodic if there exists an n, such that $A^n_{ij} > 0$, for all i, j, where A^n_{ij} is the (i, j) entry of A^n. If $\Lambda_{\mathscr{A}}$ is a subshift of finite type of the shift map σ, then the map is topologically transitive if and only if A is irreducible; the map is topologically mixing if and only if A is aperiodic. Equivalently, A is irreducible if and only if for every ordered pair of vertices $b^{(i)}$ and $b^{(j)}$ in $\mathscr{A}$ there is a path in the graph $G_{\mathscr{A}}$ starting at $b^{(i)}$ and ending at $b^{(j)}$; A is aperiodic if and only if it is irreducible and the numbers of the length of any two different closed paths in the graph $G_{\mathscr{A}}$ are coprime [18-20].

Based on the above definition, (Λ'_L, f_{73}), (Λ'_R, f_{73}), (Λ''_L, f_{73}) and (Λ''_R, f_{73}) derived from Propositions 5, 6 and 7 are subshifts of finite type of f_{73}.

3.2 Complicated dynamics of subsystems of rule 73

Since Λ_0, Λ^*, Λ^*_L and Λ^*_R are the set of fixed points or the set of periodic points of f_{73}, so their dynamical properties on these invariant sets are simple. In this section, the dynamics of f_{73} on Λ'_L, Λ'_R, Λ''_L and Λ''_R will be thoroughly investigated.

Lemma 1: Let (Λ, f) be a subshift of finite type of a CA f with $f^p(x) = \sigma^q(x),\ x \in \Lambda\ (p \geq q > 1)$, $\mathscr{A}$ be the determinative block system of Λ, and A be transition matrix corresponding to the finite directed graph $G_{\mathscr{A}} = \{\mathscr{A}, \mathscr{E}\}$, if A is aperiodic, then σ and f are both topologically mixing on Λ [18-19].

Lemma 2: Let (Λ, f) be a subshift of finite type of a CA f with $f^p(x) = \sigma^q(x),\ x \in \Lambda\ (p \geq q > 1)$, then
(1) the topological entropy of f on Λ is

$$\mathbf{ent}(f|_\Lambda) = \frac{q}{p} log(\rho(A)),$$

where $\rho(A)$ is the spectral radius of the transition matrix A corresponding to the finite directed graph $G_{\mathscr{A}} = \{\mathscr{A}, \mathscr{E}\}$;
(2) $\mathbf{ent}(f) \geq \mathbf{ent}(f|_\Lambda)$, where $\mathbf{ent}(f)$ is the topological entropy of f on total symbolic space Σ_2. [16-20]

Lemma 3: For a subshift of finite type (Λ, f), if f is topologically mixing on Λ, then
(1) f is chaotic in the sense of Devaney on Λ;
(2) f is chaotic in the sense of Li-Yorke. [14-15, 18-20]

Theorem 1: (1) f_{73} is chaotic in the sense of Devaney on Λ'_L;
(2) f_{73} is chaotic in the sense of Devaney on Λ'_R;
(3) f_{73} is chaotic in the sense of Devaney on Λ''_L;
(4) f_{73} is chaotic in the sense of Devaney on Λ''_R.

Proof: (1) Recall f_{73}-invariant set Λ'_L and its determinative block system $\mathscr{A}_1$ defined in Proposition 4, and it is easy to obtain the finite directed graph $G_{\mathscr{A}_1} = \{\mathscr{A}_1, \mathscr{E}\}$ which is shown in Figure 1. It is easily found that the numbers of the length of any two different closed paths in the graph are coprime, thus, the transition matrix A_1 corresponding to the graph is aperiodic, so the shift σ_L is mixing on Λ'_L. Since $f^5_{73}(x) = \sigma^2_L(x), x \in \Lambda'_L$, the topologically mixing property of σ_L implies the topologically mixing property of f^5_{73} on Λ'_L, and this educes the topologically mixing property of f_{73} on Λ'_L [24-25]. Thus, f_{73} is chaotic in the sense of Devaney on Λ'_L based on Lemma 3. The proofs of (2) to (4)

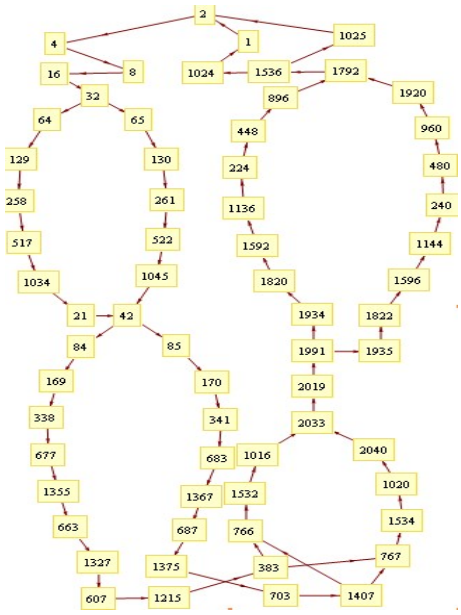


Fig. 1: Finite directed graph $G_{\mathscr{A}_1} = \{\mathscr{A}_1, E\}$.

are similar to that of (1). ■

Theorem 2: The topological entropies of f_{73} on these invariant sets Λ'_L, Λ'_R, Λ''_L and Λ''_R are respectively
(1) $\mathbf{ent}(f_{73}|_{\Lambda'_L}) = \mathbf{ent}(f_{73}|_{\Lambda'_R}) = \frac{2}{5} log(\rho(A_1)) = \frac{2}{5} log(\rho(A_2)) \approx \frac{2}{5} log(1.09433952) \approx 0.4 \times log(1.09433952) = 0.0360604$,
where $\rho(A_1)$ and $\rho(A_2)$ are the spectral radiuses of the transition matrices A_1 and A_2 corresponding to the finite directed graphs $G_{\mathscr{A}_1} = \{\mathscr{A}_1, \mathscr{E}\}$ and $G_{\mathscr{A}_2} = \{\mathscr{A}_2, \mathscr{E}\}$.
(2) $\mathbf{ent}(f_{73}|_{\Lambda''_L}) = \mathbf{ent}(f_{73}|_{\Lambda''_R}) = \frac{2}{5} log(\rho(\bar{A}_1)) = \frac{2}{5} log(\rho(\bar{A}_2)) \approx \frac{2}{5} log(1.081959) \approx 0.0315094$,
where $\rho(\bar{A}_1)$ and $\rho(\bar{A}_2)$ are the spectral radiuses of the transition matrices $\bar{A}_1$ and $\bar{A}_2$ corresponding to the finite directed graphs $G_{\mathscr{A}''} = \{\mathscr{A}'', \mathscr{E}\}$ and $G_{\bar{\mathscr{A}}''} = \{\bar{\mathscr{A}}'', \mathscr{E}\}$.

Remark 4: (1) The transition matrix A_1 corresponding to the finite directed graphs $G_{\mathscr{A}_1} = \{\mathscr{A}_1, \mathscr{E}\}$ is shown in Appendix, which is a 66×66 matrix. The topological entropies of f_{73} on Λ'_L and Λ'_R can be computed based on Lemma 2;
(2) the transition matrices $\bar{A}_1$ and $\bar{A}_2$ corresponding to $G_{\mathscr{A}''} = \{\mathscr{A}'', \mathscr{E}\}$ and $G_{\bar{\mathscr{A}}''} = \{\bar{\mathscr{A}}'', \mathscr{E}\}$ are too large to be listed here, both are 278×278 matrices;
(3) the calculation of these transition matrices and the characterization of the finite directed graphs in above theorems

can be completed by computer-aided methods.

It is well known that chaos in the sense of Devaney or positive topological entropy implies chaos in the sense of Li-Yorke [18-19, 26], thus, one has the following corollary.

Corollary 1: f_{73} is chaotic in the sense of Li-Yorke.

4. Conclusion

One of the main challenges is to explore the quantitative dynamics in cellular automata evolution [30]. Over the past few years, the material advances in this research field have been obtained [15-17, 21-25, 27-29]. By taking some advantages of complex Bernoulli-shift rules unclosed by Chua et al., this work has developed an elementary and rigorous proof to predict the rich and complex dynamics of rule 73 in view of symbolic dynamical systems. For example, the rule is topologically mixing and possesses positive topological entropies on some subsystems, and henceforth is chaotic in the sense of both Devaney and Li-Yorke. Indeed, the dynamics of rule 73 have not been completely revealed, therefore new analytical methods should be exploited to investigate it as well as other CA rules in future studies.

Acknowledgments

This research was jointly supported by the NSFC (Grant No. 60872093 and No. 10832006).

References

[1] von Neumann, J. *Theory of Self-reproducing Automata* (edited and completed by A. W. Burks), University of Illinois Press, Urbana and London, 1966.

[2] Gardner, M. The fantastic combinations of John Conway's new solitaire game 'life', *Scientific American*, 223:120-123, 1970.

[3] Hedlund, G. A. Endomorphisms and automorphism of the shift dynamical system, *Theory of Computing Systems*, 3:320-375, 1969.

[4] Wolfram, S. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 3:601-644, 1983.

[5] Wolfram, S. *Theory and Applications of Cellular Automata,* World Scientifc, Singapore, 1986.

[6] Wolfram, S. Universality and complexity in cellular automata, *Phys. D*, 10:1-35, 1984.

[7] Wolfram, S. *A New Kind of Science*, Champaign Illinois: Wolfram Media, 2002.

[8] Chua, L. O., Yoon, S., Dogaru, R. A nonlinear dynamics perspective of Wolfram's new kind of science. Part I: Threshold of complexity, *International Journal of Bifurcation and Chaos*, 12 (12):2655-2766, 2002.

[9] Chua, L. O., Sbitnev, V. I., Yoon, S. A nonlinear dynamics perspective of Wolfram's new kind of science. Part IV: From Bernoulli-shift to 1/f spectrum, *International Journal of Bifurcation and Chaos*, 15 (4):1045-1223, 2005.

[10] Chua, L. O., Sbitnev, V. I., Yoon, S. A nonlinear dynamics perspective of Wolfram's new kind of science. Part VI: From time-reversible attractors to the arrows of time, *International Journal of Bifurcation and Chaos*, 16 (5):1097-1373, 2006.

[11] Chua, L. O., Guan, J. B., Valery, I. S., Shin, J. A nonlinear dynamics perspective of Wolfram's new kind of science. Part VII: Isle of Eden, *International Journal of Bifurcation and Chaos*, 17 (9):2839-3012, 2007.

[12] Chua, L. O., Karacs, K., Sbitnev, V. I., Guan, J. B., Shin, J. A nonlinear dynamics perspective of Wolfram's new kind of science. Part VIII: More isles of Eden. *International Journal of Bifurcation and Chaos*, 17 (11):3741-3894, 2007.

[13] Guan, J. B., Shen, S. W., Tang, C. B., Chen, F. Y. Extending Chua's global equivalence theorem on Wolfram's new kind of science. *International Journal of Bifurcation and Chaos*, 17 (12):4245-4259, 2007.

[14] Devaney, R. L. *An Introduction to Chaotic Dynamical Systems*, Addison-Wesley, 1989.

[15] Favati, P., Lotti, G., Margara, L. Additive one-dimensional cellular automata are chaotic according to Devaney's definition of chaos. *Theoretical Computer Science*, 174:157-170, 1997.

[16] D'amico, M., Manzini, G., Margara, L. On computing the entropy of cellular automata. *Theoretical Computer Science*, 290:1629-1646, 2003.

[17] Culik, K., Hurd, L. P., Yu, S. Computation theoretic aspects of cellular automata. *Phys. D*, 45:357-378, 1990.

[18] Kitchens, B. *Symbolic Dynamics: One-sided, Two-sided and Countable State Markov Shifts*, Springer-Verlag, Berlin, NY, 1998.

[19] Zhou, Z. L. *Symbolic Dynamics*, Shanghai Scientific and Technological Education Publishing House, Shanghai, 1997.

[20] Xiong, J. C., Young, Z. Chaos caused by a topologically mixing map in *Dynamical Systems and Related Topics*, World Scientific, Singapore, 1992.

[21] Chen, F. Y., Jin, W. F., Chen, G. R., Chen F. F., Chen, L. Chaos of elementary cellular automata rule 42 of Wolfram's class II, *Chaos*, 19 (1):013140, 2009.

[22] Jin, W. F., Chen, F. Y., Chen, G. R., Chen L., Chen, F. F. Extending the symbolic dynamics of Chua's Bernoulli-shift rule 56, *Journal of Cellular Automata*, 5 (1-2):121-138, 2010.

[23] Jin, W. F., Chen, F. Y., Chen, G. R., Chen L., Chen, F. F. Complex symbolic dynamics of Chua's period-2 rule 37, *Journal of Cellular Automata*, 5 (4-5):315-331, 2010.

[24] Chen, F. F., Chen, F. Y. Complex dynamics of cellular automata rule 119, *Phys. A*, 388:984-990, 2009.

[25] Chen, F. F., Chen, F. Y., Chen, G. R., Jin W. F., Chen, L. Symbolics dynamics of elementary cellular automata rule 88, *Nonlinear Dynamics*, 58:431-442, 2009.

[26] Huang, W., Ye, X. D. Devaneys chaos or 2-scattering implies Li-Yorkes chaos, *Topol. Appl.*, 117 (3):259-272, 2002.

[27] Jin, W. F., Chen, F. Y. Temporal complexity of totalistic cellular automaton rule 52, *The 2010 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'10)*, Las Vegas, Nevada, USA, July 12-15, 2010.

[28] Chen, G. R., Chen F. Y., Guan J. B., Jin, W. F. Symbolic dynamics of some Bernoulli-shift cellular automata rules, *The 2010 International Symposium on Nonlinear Theory and its Applications (NOLTA 2010)*, Krakow, Poland, September 5-8, 2010.

[29] Jin, W. F., Chen F. Y. Global attractors and chaos of complex Bernoulli-shift rules, *The 2010 International Workshop on Chaos-Fractal Theory and its Applications (IWCFTA 2010)*, Kunming, Yunnan, China, October 28-31, 2010.

[30] Kari, J. Theory of cellular automata: A survey, *Theoretical Computer Science*, 334:3-33. 2005.

Appendix

The transition matrix A_1 corresponding to the finite directed graphs $G_{\mathscr{A}_1} = \{\mathscr{A}_1, \mathscr{E}\}$ in Figure 1 is

$$A_1 = \begin{pmatrix}
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0
\end{pmatrix}$$

Infinite Number of Chaotic Generalized Sub-shifts of Cellular Automaton Rule 180

Wei Chen, Fangyue Chen, Yunfeng Bian, and Jing Chen
School of Science, Hangzhou Dianzi University, Hangzhou, Zhejiang, P. R. China

Abstract— *This paper is devoted to an in-depth study of cellular automaton rule* 180 *under the framework of symbolic dynamics. Rule* 180*, a member of Wolfram's class IV and Chua's hyper Bernoulli shift rules, defines infinite number of generalized sub-shifts. An effective method of constructing the shift invariant sets of the rule's global map is proposed. It is noted that this method is also applicable to studying the dynamics of other rules. Furthermore, the rich and complex dynamical behaviors on these sub-shifts, such as positive topological entropies, topologically mixing, and chaos in the sense of Li-Yorke and Devaney, are revealed.*

Keywords: cellular automata; chaos; generalized sub-shift; symbolic dynamics; topologically mixing.

1. Introduction

Cellular automata (CA) are among the oldest models of natural computing, dating back over half a century [1]. The first CA studied by von Neumann in the late 1940s were biologically motivated: the goal was to design self-replicating artificial systems that are also computationally universal. Following suggestions by S. Ulam, he envisioned a discrete universe consisting of a two-dimensional mesh of finite state machines, called cells, interconnected locally with each other. The cells change their states synchronously depending on the states of some nearby cells, the neighbors, as determined by a local update rule. All cells use the same update rule so that the system is homogeneous like many physical and biological systems. These cellular universes are now known as CA. CA have been widely used to model a variety of dynamical systems in physics, biology, chemistry, and computer science in the recent decades.

The topological dynamics of CA began in 1969 with Hedlund who viewed one-dimensional CA in the context of symbolic dynamics as endomorphisms of the shift dynamical system. His main results are the characterizations of surjective and open CA [2]. In the 1980s, Wolfram proposed cellular automata as models for physical systems exhibiting complex or even chaotic behaviors. In his work, he divided the $2^{2^3} = 256$ elementary cellular automata (ECA) rules informally into four classes using dynamical concepts like periodicity, stability, and chaos [3-5]. In 2002, he introduced his monumental work *A New kind of Science* [6]. Based on this work, Chua *et.al* provided a nonlinear dynamics perspective to Wolfram's empirical observations from the viewpoint of mathematical analysis via the concepts like characteristic function, forward time-τ map, basin tree diagram, Isle-of-Eden digraph and so on [7-11].

Although there are 256 ECA rules, only 88 rules are globally independent from each other [12]. These 88 global independent ECA rules are also organized into 4 groups with distinct qualitative dynamics; 40 period-k ($k = 1, 2, 3, 6$), 30 topologically distinct Bernoulli shift rules, 10 complex Bernoulli shift rules and 8 hyper Bernoulli shift rules [8-11].

Due to the fact that many properties of the temporal evolution of CA, such as topological entropy, topologically sensitivity and topologically mixing are undecidable [13], one should, in principle, separately analyze time-asymptotic dynamics for each class of rules. It will be seen in the next sections that rule 180, a member of Wolfram's class IV and Chua's hyper Bernoulli shift rules, has infinite number of generalized sub-shifts with rich and complex dynamics. Precisely, it has positive entropies, and is topologically mixing on these generalized sub-shifts. This implies that rule 180 is chaotic in the sense of Li-Yorke and Devaney.

The rest of this paper is organized as follows. Section 2 presents the preliminaries of symbolic dynamical systems and CA. Section 3 explores infinite number of f_{180}-positively invariant subsets, which are hyper generalized sub-shifts of f_{180}. Section 4 demonstrates some complex dynamics of the rule. Finally, Section 5 concludes this paper.

2. Preliminaries

A word over $S = \{0, 1\}$ is a finite sequence $a = (a_0, \cdots, a_{n-1})$. The length of a is denoted by $l(a) = n$. If a is a finite or infinite word and $I = [i, j]$ is an interval of integers on which a is defined, then denote $a_{[i,j]} = (a_i, \cdots, a_j)$ and $a_{[i,j)} = (a_i, \cdots, a_{j-1})$. b is a subword of a, denoted by $b \prec a$, if $b = a_I$ for some interval $I \subseteq \mathbb{Z}$; otherwise, denoted by $b \not\prec a$.

A bi-infinite word is called a configuration, the collection of all configurations is

$$\Sigma_2 = S^{\mathbb{Z}} = \{(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) |\ x_i \in S, i \in \mathbb{Z}\}$$

and the product "d" induced by Hamming distance is defined as

$$d(x, y) = \sum_{i=-\infty}^{\infty} \frac{1}{2^{|i|}} \cdot |x_i - y_i|,$$

Table 1: Truth table of Boolean function of rule 180

(x_{i-1}, x_i, x_{i+1})	$\hat{f}_{180}(x_{i-1}, x_i, x_{i+1})$
(0, 0, 0)	0
(0, 0, 1)	0
(0, 1, 0)	1
(0, 1, 1)	0
(1, 0, 0)	1
(1, 0, 1)	1
(1, 1, 0)	0
(1, 1, 1)	1

for any $x, y \in \Sigma_2$. It is easy to know that Σ_2 is a Cantor complete metric space. The left-shift map $\sigma_{_L}$ and right-shift map $\sigma_{_R}$ are defined by $[\sigma_{_L}(x)]_i = x_{i+1}$, $[\sigma_{_R}(x)]_i = x_{i-1}$, for any $x \in \Sigma_2, i \in \mathbb{Z}$ respectively, where $[\sigma(x)]_i$ stands for the i-th symbol of $\sigma(x)$.

By a theorem of Hedlund [2], a map $f : \Sigma_2 \to \Sigma_2$ is a cellular automaton iff it is continuous and commutes with σ, i.e., $\sigma \circ f = f \circ \sigma$, where σ is left-shift or right-shift. Moreover, any CA f defines a dynamical system (Σ_2, f). A subset $X \subseteq \Sigma_2$ is f-invariant if $f(X) \subseteq X$, and strongly f-positively invariant if $f(X) = X$. If X is a closed and f-invariant, then (X, f) or simply X is called a subsystem of (Σ_2, f).

Each ECA rule can be expressed by a logical truth table, the one of rule 180 is shown in Table 1. It is clear that its binary output sequence is 00101101.

Let $\hat{f}_{180}$ be the local map defined by the truth table, and $f_{180} : \Sigma_2 \to \Sigma_2$ with $f_{180}(\cdots, x_{-1}, \overset{*}{x_0}, x_1, \cdots) = (\cdots, y_{-1}, \overset{*}{y_0}, y_1, \cdots)$ be the global map induced by $\hat{f}_{180}$, where $y_i = \hat{f}_{180}(x_{i-1}, x_i, x_{i+1})$.

The local map $\hat{f}_{180}$ also defines a map from S^{n+1} to S^{n-1} $(n \geq 2)$ with $\hat{f}_{180}(a_0, a_1, \cdots, a_n) = (b_0, b_1, \cdots, b_{n-2})$, where $b_i = \hat{f}_{180}(a_i, a_{i+1}, a_{i+2})$ $(i = 0, 1, 2, \cdots, n-2)$. The n times iteration of $\hat{f}_{180}$ is a map $\hat{f}^n_{180}$ from S^{2n+1} to S with

$$\hat{f}^n_{180}(a_{-n}, \cdots, a_0, \cdots, a_n) =$$

$$\hat{f}_{180}(\hat{f}^{n-1}_{180}(a_{[-n,n-2]}), \hat{f}^{n-1}_{180}(a_{[-n+1,n-1]}), \hat{f}^{n-1}_{180}(a_{[-n+2,n]})).$$

3. Generalized sub-shifts

Let $\bar{x} \in S^n$ be a finite word, $\bar{x} = (\bar{x}_0, \bar{x}_1, \cdots, \bar{x}_{n-1})$ with boundary condition $\bar{x}_0 = \bar{x}_{n-1} = 1$. Denote $x : \mathbb{Z} \to S$ with $i \mapsto x_i$ of the following form:

$$\exists\ m \in \mathbb{Z},\ x_{m+i} = \begin{cases} \bar{x}_i, & if\ 0 \leq i \leq (n-1); \\ 0, & otherwise. \end{cases}$$

These configurations are called bi-infinite extensions of the block $\bar{x}$ in the background of $0'$s [14, 15].

Define

$$\mathscr{F}^n = \{x \in \Sigma_2 | \bar{x} \in S^n, \bar{x}_0 = \bar{x}_{n-1} = 1\} \tag{1}$$

and

$$\mathscr{F} = \bigcup_{n \in \mathbb{N}} \mathscr{F}^n. \tag{2}$$

Any configuration of this set is said to be 0-finite [14].

Therefore a configuration x is 0-finite iff it is of the form $x = (\cdots, 0, 0, x_m, \cdots, x_p, 0, 0, \cdots)$, with m and p the minimum and the maximum site respectively for which $x_m = x_p = 1$. To each 0-finite configuration x we assign its length $l(x) = p - m + 1$. particularly, for the quiescent configuration $0 = (\cdots, 0, 0, 0, \cdots)$, $l(0) = 0$.

Definition *1:* (1) [14] A dynamical system (Σ_2, f) (or simply f, non-necessarily induced from a CA rule) is said to be a generalized shift iff there exists a map $M : \Sigma_2 \to \mathbb{N}$ such that for any $\alpha \in \Sigma_2$, and any $t \in \mathbb{N}$,

$$f^{M(\alpha)}(f^t(\alpha)) = \sigma^{M(\alpha)}(f^t(\alpha));$$

(2) [14] f is said to be a generalized sub-shift on the subset X of Σ_2 iff X is f-positively invariant $(f(X) \subset X)$, and there exists a map $M : X \to \mathbb{N}$ such that for any $x \in X$, and any $t \in \mathbb{N}$,

$$f^{M(x)}(f^t(x)) = \sigma^{M(x)}(f^t(x));$$

(3) f is said to be a hyper generalized sub-shift on the subset X of Σ_2 iff X is f-positively invariant $(f(X) \subset X)$, and there exists a number $M \in \mathbb{N}$ such that for any $x \in X$

$$f^M(x) = \sigma^M(x).$$

Remark 1: If f is a (hyper) generalized sub-shift on the subset X, then simply say X is a (hyper) generalized sub-shift of f.

Proposition 1: If X is a hyper generalized sub-shift of CA f, then there exists a set $\mathscr{A}$ of words of length $n = (2M + 1)$ such that $X = X_{\mathscr{A}} = \{x \in \Sigma_2 |\ x_{[i-n,i+n]} \in \mathscr{A}, \forall\ i \in \mathbb{Z}\}$, where $\mathscr{A}$ is said to be the determinative block system of X.

Proof: Without loss of generality, σ is chosen as the right shift with $[\sigma(x)]_i = x_{i-1}$ for $x \in X$. Since the local map $\hat{f} : S^3 \to S$, it can lead out of its M times iteration $\hat{f}^M : S^{2M+1} \to S$. Thus, $f^M(x) = \sigma^M(x), x \in X$, if and only if $\hat{f}^M(x_{[i-M,i+M]}) = [f^M(x)]_i = [\sigma^M(x)]_i = x_{i-M}$, for all $i \in \mathbb{Z}$. Let $\mathscr{A} = \{(a_0, \cdots, a_{2M}) \in S^{2M+1} | (a_0, \cdots, a_{2M}) = x_{[i-M,i+M]}, x \in X, i \in \mathbb{Z}\}$, $\mathscr{A}$ is a finite set since $|\mathscr{A}| < 2^{2M+1}$. Then it follows that $X = X_{\mathscr{A}}$. ■

It was known that f_{180} is not a generalized shift, but is a generalized sub-shift and hyper generalized sub-shift on the subset $\mathscr{F}$ [14, 15].

Lemma 1: [14, 15] For rule 180, there exists a map $M : \{0,1\}^* \to \mathbb{N}$ such that for any $\bar{x} \in \{0,1\}^*$,

$$f_{180}^{M(\bar{x})}(x) = \sigma_R^{M(\bar{x})}(x),$$

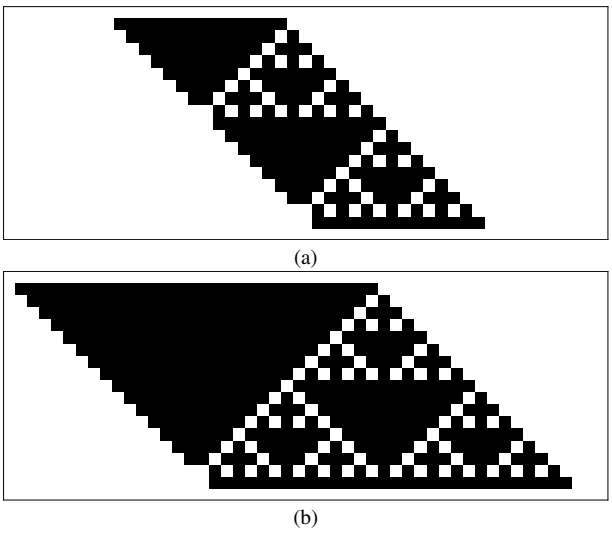
where $\{0,1\}^*$ is the set of all blocks of finite length defined over $S = \{0,1\}$ and the map M is a power of two, i.e., for any $\bar{x} \in \{0,1\}^n$, $M(\bar{x}) = 2^{E(\bar{x})}$ for a suitable map $E : \{0,1\}^* \to \mathbb{N}$. Where as usual $x(\in \mathscr{F} \subset \Sigma_2)$ denotes any bi-infinite extension of block $\bar{x}$ in a background of $0's$.

Lemma 2: [14, 15] The map which satisfies Lemma 1 is such that for every $n \in N$,

$$M(1^n) = 2^{[log(n+1)]},$$

where 1^n is the 1-constant block of length n.

The spatio-temporal evolution of the bi-infinite extensions of blocks 1^{14} and 1^{30} in a background of $0's$ is shown in Figure 1.



(a) (b)

Fig. 1: Spatio-temporal evolution of the bi-infinite extension of block 1^n in a background of $0's$, (a) $n = 14$; (b) $n = 30$.

Unfortunately, the f_{180}-positively invariant subset $\mathscr{F}$ defined in (2) is only a countable infinite set, the dynamics of f_{180} on the generalized sub-shift $\mathscr{F}$ is not enough to reveal the dynamics on the total phase space Σ_2. In the following discussion in this section, infinite number of generalized sub-shifts of f_{180} will be expressed.

For given $p \in \mathbb{N}$, let n satisfies $2^p - 1 \le n \le 2^{p+1} - 2$ (i.e., $p \le log(n+1) < p+1$), and construct a finite word

$$a_p^{(n)} = (\underbrace{0,\cdots,0}_{2^{p+1}+1}, 1^n, \underbrace{0,\cdots,0}_{2^{p+1}}).$$

$\mathscr{A}_p^{(n)}$ denotes the $(2^{p+1}+1)$-sequence set appeared in $a_p^{(n)}$, i.e.,

$$\mathscr{A}_p^{(n)} = \{(a_0, a_1, \cdots, a_{2^{p+1}}) | (a_0, a_1, \cdots, a_{2^{p+1}}) \prec a_p^{(n)}\}.$$

It is easily known that the number of symbols of $a_p^{(n)}$ is $(2^{p+1}+n+1)$, thus, the number of sequences of $\mathscr{A}_p^{(n)}$ is $(2^{p+1}+n+1)$, and $\hat{f}_{180}^{2^p}(a_0, a_1, \cdots, a_{2^{p+1}}) = a_0$, for any $(a_0, a_1, \cdots, a_{2^{p+1}}) \in \mathscr{A}_p^{(n)}$.

Now let

$$\Lambda_p^{(n)} = \{x \in \Sigma_2 |\ x_{[i-2^p, i+2^p]} \in \mathscr{A}_p^{(n)},\ i \in \mathbb{Z}\} \quad (3)$$

and

$$\tilde{\Lambda}_p = \bigcup_{n=2^p-1}^{2^{p+1}-2} \Lambda_p^{(n)}. \quad (4)$$

In (3), the $(2^{p+1}+1)$-sequence set $\mathscr{A}_p^{(n)}$ is the determinative block system of $\Lambda_p^{(n)}$.

Proposition 2: For given $p \in \mathbb{N}$, and n satisfies $2^p - 1 \le n \le 2^{p+1} - 2$, $\tilde{\Lambda}_p$ is a $f_{180}^{2^p}$-positively invariant set, and is a hyper generalized sub-shift of $f_{180}^{2^p}$.

Proof: In fact, for any $x \in \tilde{\Lambda}_p$, if $x \in \Lambda_p^{(n)}$, where n satisfies $2^p - 1 \le n \le 2^{p+1} - 2$, thus, $x_{[i-2^p, i+2^p]} \in \mathscr{A}_p^{(n)}$, so $\hat{f}_{180}^{2^p}(x_{[i-2^p, i+2^p]}) = x_{i-2^p}$, this implies $f_{180}^{2^p}(x) = \sigma_R^{2^p}(x) \in \Lambda_p^{(n)} \subset \tilde{\Lambda}_p$. ■

Moreover, let

$$\Lambda_p = \tilde{\Lambda}_p \cup f_{180}(\tilde{\Lambda}_p) \cup f_{180}^2(\tilde{\Lambda}_p) \cup \cdots \cup f_{180}^{2^p-1}(\tilde{\Lambda}_p). \quad (5)$$

Proposition 3: Λ_p is a f_{180}-positively invariant set, and is a hyper generalized sub-shift of f_{180}.

Proof: The result is obvious, the details are omitted here. ■

Based on above propositions, the following interesting result is obtained.

Theorem 1: For rule 180, there exists infinite number of f_{180}-positively invariant subsets Λ_p $(p = 1, 2, 3, \cdots)$ such that $f_{180}^{2^p}|_{\Lambda_p} = \sigma_R^{2^p}|_{\Lambda_p}$, i.e., Λ_p $(p = 1, 2, 3, \cdots)$ are the generalized sub-shifts of f_{180}.

Example 1: The structure of the generalized sub-shift Λ_1: when $p = 1$, then $n = 1, 2$. Thus, $a_1^{(1)} = (0000010000)$, $a_1^{(2)} = (00000110000)$, and

$$\tilde{\Lambda}_1 = \Lambda_1^{(1)} \cup \Lambda_1^{(2)},$$

where

$$\Lambda_1^{(1)} = \{x \in \Sigma_2 |\ x_{[i-2,i+2]} \in \mathscr{A}_1^{(1)},\ i \in \mathbb{Z}\},$$

$$\Lambda_1^{(2)} = \{x \in \Sigma_2 |\ x_{[i-2,i+2]} \in \mathscr{A}_1^{(2)},\ i \in \mathbb{Z}\},$$

$\mathscr{A}_1^{(1)} = \{(a_0, a_1, \cdots, a_4) | (a_0, a_1, \cdots, a_4) \prec a_1^{(1)}\}$
$= \{00000, 00001, 00010, 00100, 01000, 10000\}$ and
$\mathscr{A}_1^{(2)} = \{(a_0, a_1, \cdots, a_4) | (a_0, a_1, \cdots, a_4) \prec a_1^{(2)}\}$
$= \{00000, 00001, 00011, 00110, 01100, 11000, 10000\}$.
It follows that

$$\tilde{\Lambda}_1 = \{x \in \Sigma_2 |\ x_{[i-2,i+2]} \in \mathscr{A}_1,\ i \in \mathbb{Z}\},$$

where
$\mathscr{A}_1 = \mathscr{A}_1^{(1)} \cup \mathscr{A}_1^{(2)} = \{00000, 00001, 00010, 00100, 01000,$

$10000, 00011, 00110, 01100, 11000\}$.

Let $\Lambda_1 = \tilde{\Lambda}_1 \cup f(\tilde{\Lambda}_1)$, where

$$f(\tilde{\Lambda}_1) = \{x \in \Sigma_2 |\ x_{[i-2,i+2]} \in \mathscr{A}'_1,\ i \in \mathbb{Z}\},$$

and

$$\mathscr{A}'_1 = \{(b_0, b_1, \cdots, b_4) |\ (b_0, b_1, \cdots, b_4) = \hat{f}_{180}(a_0, a_1, \cdots, a_6),\ (a_i, a_{i+1}, \cdots, a_{i+4}) \in \mathscr{A}_1,\ i = 0, 1, 2\} = \mathscr{A}_1.$$

Remark 2: Generally, the f_{180}-positively invariant set Λ_p in (5) may be one of subset of the set which also is a generalized sub-shifts Δ_p with $f_{180}^{2^p}|_{\Delta_p} = \sigma_R^{2^p}|_{\Delta_p}$. For example, $p = 1$, it is easily investigated that there exists a f_{180}-positively invariant set $\Delta_1 = \{x \in \Sigma_2 |\ x_{[i-2,i+2]} \in \bar{\mathscr{A}},\ i \in \mathbb{Z}\}$ such that $f_{180}^2|_{\Delta_1} = \sigma_R^2|_{\Delta_1}$, where $\bar{\mathscr{A}} = \{00000, 00001, 00010, 00011, 00100, 00101, 00110, 01000, 01001, 01011, 01100, 01101, 10000, 10001, 10010, 10011, 10110, 11000, 11001, 11011\}$. Obviously, $\Lambda_1 \subset \Delta_1$.

4. Complex dynamics

If $\Lambda \subset \Sigma_2$ is a hyper generalized sub-shift of a CA f, and $\Lambda = \Lambda_{\mathscr{A}}$, $\mathscr{A}$ is the determinative block system of Λ, then $\Lambda_{\mathscr{A}}$ can be described by a finite directed graph $G_{\mathscr{A}} = \{\mathscr{A}, \mathscr{E}\}$, where each vertex is labeled by a sequence in $\mathscr{A}$, and $\mathscr{E}$ is the edge set. Two vertices $a = (a_0, \cdots, a_{n-1})$ and $b = (b_0, \cdots, b_{n-1})$ are connected by an edge of $\mathscr{E}$ if and only if $a_k = b_{k-1}, k = 1, 2, \cdots, n-1$. Every edge $(a_0, \cdots, a_{n-1}) \to (b_0, \cdots, b_{n-1})$ of $\mathscr{E}$ is labeled by b_{n-1}. One can think of each element of $\Lambda_{\mathscr{A}}$ as a bi-infinite path on the graph $G_{\mathscr{A}}$. Whereas a directed graph corresponds to a square transition matrix $A = (A_{ij})_{m\times m}$ with $A_{ij} = 1$ if and only if there is an edge from vertex $b^{(i)}$ to vertex $b^{(j)}$, where $m = |\mathscr{A}|$ is the number of elements in $\mathscr{A}$, and i (or j) is the code of the vertex in $\mathscr{A}$, $i, j = 0, 1, \cdots, m-1$. Thus, $\Lambda_{\mathscr{A}}$ is precisely defined by the transition matrix A.

Remarkably, a square matrix A is irreducible, if for any i, j, there exists an n such that $A_{ij}^n > 0$; aperiodic if there exists an n, such that $A_{ij}^n > 0$, for all i, j, where A_{ij}^n is the (i, j) entry of A^n. If $\Lambda_{\mathscr{A}}$ is a sub shift of finite type of the shift map σ, then the map is topological transitive if and only if A is irreducible; the map is topologically mixing if and only if A is aperiodic. Equivalently, A is irreducible if and only if for every ordered pair of vertices $b^{(i)}$ and $b^{(j)}$ in $\mathscr{A}$ there is a path in the graph $G_{\mathscr{A}}$ starting at $b^{(i)}$ and ending at $b^{(j)}$; A is aperiodic if and only if it is irreducible and the numbers of the length of any two different closed paths in the graph $G_{\mathscr{A}}$ are coprime [19-21].

Lemma 3: Let Λ be a hyper generalized sub-shift of a CA f with $f^M(x) = \sigma^M(x),\ x \in \Lambda$, $\mathscr{A}$ be the determinative block system of Λ, and A be transition matrix corresponding to the finite directed graph $G_{\mathscr{A}} = \{\mathscr{A}, \mathscr{E}\}$, if A is aperiodic, then σ and f are both topological mixing on Λ [19, 20].

Lemma 4: Let Λ be a hyper generalized sub-shift of a CA f with $f^M(x) = \sigma^M(x),\ x \in \Lambda$, then
(1) the topological entropy of f on Λ is

$$\mathbf{ent}(f|_\Lambda) = log(\rho(A)),$$

where $\rho(A)$ is the spectral radius of the transition matrix A corresponding to the finite directed graph $G_{\mathscr{A}} = \{\mathscr{A}, \mathscr{E}\}$; [19, 20]
(2) $\mathbf{ent}(f) \geq \mathbf{ent}(f|_\Lambda)$, where $\mathbf{ent}(f)$ is the topological entropy of f on total symbolic space Σ_2. [16-20]

Lemma 5: For a hyper generalized sub-shift (Λ, f), if f is topological mixing on Λ, then
(1) f is chaotic in the sense of Devaney on Λ [16, 19, 20];
(2) f is chaotic in the sense of Li-Yorke. [19, 27]

Theorem 2: (1) f_{180} is chaotic in the sense of Devaney on the hyper generalized sub-shift Λ_1 in Example 1;
(2) the topological entropy of f_{180} on Λ_1 is positive.

Proof: (1) At the time, $f_{180}^2(x) = \sigma_R^2(x), x \in \Lambda_1$. It is already known that the determinative block system of Λ_1 is $\mathscr{A}_1 = \{00000, 00001, 00010, 00100, 01000, 10000, 00011, 00110, 01100, 11000\}$, and the matrix corresponding to the finite directed graph $G_{\mathscr{A}_1} = \{\mathscr{A}_1, \mathscr{E}\}$ is

$$A_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

It follows that $A_1^n > 0$ $(n \geq 10)$, so A is aperiodic, σ_R and σ_R^2 are mixing on Λ_1, it implies f_{180} is mixing on Λ_1. Thus, f_{180} is chaotic in the sense of Devaney on Λ_1 based on Lemma 5.
(2) in fact, $\mathbf{ent}(f_{180}|_{\Lambda_1}) = log(\rho(A_1)) \approx log(1.4196) > 0$. ■

Remark 3: The result of the theorem can also be obtained from the coprime property of the numbers of the length of any two different closed paths in the finite directed graph $G_{\mathscr{A}_1} = \{\mathscr{A}_1, \mathscr{E}\}$.

Theorem 3: f_{180} is chaotic in the sense of Devaney on each hyper generalized sub-shift Λ_p $(p = 1, 2, 3, \cdots)$ in (3), (4) and (5).

Proof: Recall the structure of Λ_p $(p = 1, 2, 3, \cdots)$:

$$\Lambda_p = \tilde{\Lambda}_p \cup f_{180}(\tilde{\Lambda}_p) \cup f_{180}^2(\tilde{\Lambda}_p) \cup \cdots \cup f_{180}^{2^p-1}(\tilde{\Lambda}_p).$$

It is obvious that Λ_p is f_{180}-positively invariant set. Since $\tilde{\Lambda}_p$ is a hyper generalized sub-shift of f_{180}, by Proposition 1, there exists its determinative block system $\tilde{\mathscr{A}}_p$. It is clear that the $(2^{p+1}+1)-$word $0^{2^{p+1}+1}$ belongs to $\tilde{\mathscr{A}}_p$, this implies

there exists a closed path from the word to self whose length is 1 in the finite directed graph $G_{\tilde{\mathscr{A}_p}} = \{\tilde{\mathscr{A}_p}, \mathscr{E}\}$, so it is easy to know that σ_R and $f_{180}^{2^p}$ is topologically mixing on $\tilde{\Lambda}_p$. Moreover, f_{180} is a homeomorphic map from $\tilde{\Lambda}_p$ to $f_{180}(\tilde{\Lambda}_p)$, so it is easily proved that $0^{2^{p+1}+1}$ belongs to each $\tilde{\mathscr{A}_p}$ $(p = 2, 3, \cdots, 2^p - 1)$, thus, σ_R and $f_{180}^{2^p}$ is also topologically mixing on $f_{180}^i(\tilde{\Lambda}_p)$ $(i = 2, 3, \cdots, 2^p - 1)$. Finally, let $\mathscr{A}_p$ denote the determinative block system of Λ_p, then there still holds that $0^{2^{p+1}+1}$ belongs to $\mathscr{A}_p$, it implies that σ_R and f_{180} is topologically mixing on Λ_p, so f_{180} is chaotic in the sense of Devaney on Λ_p. Of course, it is chaotic in the sense of Li-Yorke. ■

5. Conclusion

As a particular class of dynamical systems, CA have been widely used for modeling and approximating many physical phenomena. Despite their apparent simplicity, CA can display rich and complex evolutions [22-26, 28, 29], and many properties of their spatio-temporal evolutions are undecidable [13]. This paper is devoted to an in-depth study of cellular automaton rule 180 in the framework of symbolic dynamics. It has been rigorously proved that its global map defines infinite number of generalized sub-shifts with rich and complex dynamical behaviors, such as topologically mixing, positive topological entropies and chaos in the sense of Li-Yorke and Devaney. Indeed, the dynamics of rule 180 have not been completely revealed, therefore new effective analytical methods should be exploited to investigate it as well as other CA rules in future studies.

Acknowledgments

This research was jointly supported by the NSFC (Grant No. 60872093 and No. 10832006).

References

[1] von Neumann, J. *Theory of self-reproducing automata* (edited and completed by A. W. Burks), University of Illinois Press, Urbana and London, 1966.

[2] Hedlund, G. A. *Endomorphisms and automorphism of the shift dynamical system*, Theory of Computing Systems, 3:320-375, 1969.

[3] Wolfram, S. *Statistical mechanics of cellular automata*. Rev. Mod. Phys., 3:601-644, 1983.

[4] Wolfram, S. *Theory and applications of cellular automata,* World Scientifc, Singapore, 1986.

[5] Wolfram, S. *Universality and complexity in cellular automata*, Phys. D, 10:1-35, 1984.

[6] Wolfram, S. *A new kind of science*, Champaign Illinois: Wolfram Media, 2002.

[7] Chua, L. O., Yoon, S., Dogaru, R. *A nonlinear dynamics perspective of Wolfram's new kind of science. Part I: Threshold of complexity*, International Journal of Bifurcation and Chaos, 12 (12):2655ÍC2766, 2002.

[8] Chua, L. O., Sbitnev, V. I., Yoon, S. *A nonlinear dynamics perspective of Wolfram's new kind of science. Part IV: From Bernoulli-shift to 1/f spectrum*, International Journal of Bifurcation and Chaos, 15 (4):1045-1223, 2005.

[9] Chua, L. O., Sbitnev, V. I., Yoon, S. *A nonlinear dynamics perspective of Wolfram's new kind of science. Part VI: From time-reversible attractors to the arrows of time*, International Journal of Bifurcation and Chao, 16 (5):1097-1373, 2006.

[10] Chua, L. O., Guan, J. B., Valery, I. S., Shin, J. *A nonlinear dynamics perspective of Wolfram's new kind of science. Part VII: Isle of Eden*, International Journal of Bifurcation and Chaos, 17 (9):2839-3012, 2007.

[11] Chua, L. O., Karacs, K., Sbitnev, V. I., Guan, J. B., Shin, J. *A nonlinear dynamics perspective of Wolfram's new kind of science. Part VIII: More isles of Eden*. International Journal of Bifurcation and Chaos, 17 (11):3741-3894, 2007.

[12] Guan, J. B., Shen, S. W., Tang, C. B., Chen, F. Y. *Extending Chua's global equivalence theorem on Wolfram's new kind of science*. International Journal of Bifurcation and Chaos, 17 (12):4245-4259, 2007.

[13] Culik, K., Hurd, L. P., Yu, S. *Computation theoretic aspects of cellular automata*, Phys. D, 45:357-378, 1990.

[14] Gianpiero, C., Luciano, M. *GeneraliZed sub-shift in elementary cellular automata: the "strang case" of chaotic rule 180*. Theoretical Computer Science, 201:171-187, 1998.

[15] Braga, G., Cattaneo, P., Quaranta Vogliotti, C., *Pattern growth in elementary cellular automata* , Addison-Wesley, 1989.

[16] Devaney, R. L. *An introduction to chaotic dynamical systems*, Addison-Wesley, 1989.

[17] Favati, P., Lotti, G., Margara, L. *Additive one-dimensional cellular automata are chaotic according to Devaney's definition of chaos*. Theoretical Computer Science, 174:157-170, 1997.

[18] D'amico, M., Manzini, G., Margara, L. *On computing the entropy of cellular automata*. Theoretical Computer Science, 290:1629-1646, 2003.

[19] Kitchens, B. *Symbolic dynamics: one-sided, two-sided and countable state markov shifts*, Springer-Verlag, Berlin, NY, 1998.

[20] Zhou, Z. L. *Symbolic dynamics*, Shanghai Scientific and Technological Education Publishing House, Shanghai, 1997.

[21] Xiong, J. C., Young, Z. Chaos caused by a topologically mixing map in *Dynamical Systems and Related Topics*, World Scientific, Singapore, 1992.

[22] Chen, F. Y., Jin, W. F., Chen, G. R., Chen F. F., Chen, L. *Chaos of elementary cellular automata rule 42 of Wolfram's class II*, Chaos, 19 (1):013140, 2009.

[23] Jin, W. F., Chen, F. Y., Chen, G. R., Chen L., Chen, F. F. *Extending the symbolic dynamics of Chua's Bernoulli-shift rule 56*, Journal of Cellular Automata, 5 (1-2):121-138, 2010.

[24] Jin, W. F., Chen, F. Y., Chen, G. R., Chen L., Chen, F. F. *Complex symbolic dynamics of Chua's period-2 rule 37*, Journal of Cellular Automata, 5 (4-5):315-331, 2010.

[25] Chen, F. F., Chen, F. Y. *Complex dynamics of cellular automata rule 119*, Phys. A, 388:984-990, 2009.

[26] Chen, F. F., Chen, F. Y., Chen, G. R., Jin W. F., Chen, L. *Symbolics dynamics of elementary cellular automata rule 88*, Nonlinear Dynamics, 58:431-442, 2009.

[27] Huang, W., Ye, X. D. *Devaneys chaos or 2-scattering implies Li-Yorkes chaos*, Topol. Appl., 117 (3):259-272, 2002.

[28] Jin, W. F., Chen, F. Y. *Temporal complexity of totalistic cellular automaton rule 52*, The 2010 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'10), Las Vegas, Nevada, USA, July 12-15, 2010.

[29] Kari, J. *Theory of cellular automata: A survey*, Theoretical Computer Science, 334:3-33. 2005.

Attractors and Subshifts of Finite Type of ECA 41

Yunfeng Bian, Fangyue Chen, Yi Wang, Jing Chen, and Wei Chen
School of Science, Hangzhou Dianzi University, Hangzhou, Zhejiang, P. R. China

Abstract— *In this paper, the dynamics of elementary cellular automaton rule 41 is investigated in the bi-infinite symbolic sequence space. In spite of rule 41 is not surjective, but it possess of rich and complex dynamical behaviors. The existence of attractors and subshifts of finite type of the rule's global map is strictly proved, some interesting dynamical properties on these subshifts, such as positive topological entropies, topological transitivity and topological mixing, chaos in the sense of Li-Yorke and Devaney, are revealed.*

Keywords: attractor; cellular automata; chaos; subshift of finite type; symbolic dynamics.

1. Introduction

Cellular automata (CA) are a class of spatially and temporally discrete, deterministic mathematical systems characterized by local interactions and an inherently parallel form of evolution. CA, formally introduced by von Neumann and Ulam in the 1940's to the 1950's, are able to produce complex dynamical phenomena by means of designing simple local rules [1]. Due to their simple mathematical constructions and distinguishing features, CA have been widely used to model a variety of dynamical systems. The study of topological dynamics of CA began with Hedlund in 1969, who viewed one-dimensional CA (1D CA) in the context of symbolic dynamics as endomorphisms of the shift dynamical system [2], where the main results are the characterizations of surjective and open CA. Based on the theoretical concept of universality, researchers have tried to develop even simpler and more practical architectures of CA which can be used for widely diverse applications. In the early 1980's, Wolfram introduced space-time representations of 1D CA and informally classified them into four classes by using dynamical concepts like periodicity, stability and chaos [3, 4]. In 2002, he introduced his monumental work *A New Kind of Science* [5]. To provide a rigorous foundation for Wolfram's empirical observations Chua *et al* derived a nonlinear dynamics perspective to elementary cellular automata (ECA) via the concepts like characteristic function, forward time-τ map, basin tree diagram and Isle-of-Eden digraph [6-8]. It was known that there are 256 ECA rules, only 88 rules are globally independent from each other [9]. These 88 global independent ECA rules are also organized into 4 groups with distinct qualitative dynamics: 40 period-k ($k = 1, 2, 3, 6$) rule classes, 30 topologically distinct Bernoulli shift rule classes, 10 complex Bernoulli shift rule classes and 8 hyper Bernoulli shift ones [6-9].

CA are dynamical systems with a very rich spectrum of dynamical properties. Although topological properties of CA can be explored, many of them such as topological entropy, sensitivity, topological mixing, topological transitivity and so on are undecidable. The relationship between positively expansive and mixing was investigated by Blanchard and Maass [10]. The transitive CA implies surjective and sensitive to initial conditions have been obtained by Margara and Kurka [11, 12]. The dynamics of a specific ECA on their Bernoulli-shift invariant subset was analysed [13-15]. When a cellular automaton is not surjective, the concept of an attractor is essential for its understanding. Some attractors of CA are subshifts and some are not. These two kinds of attractor have quite different properties [16].

ECA rule 41, which is not a surjective CA, possess of rich and complex dynamical behaviors. In this paper, the dynamics of the rule's global map is investigated in the bi-infinite symbolic sequence space. The existence of attractors and subshifts of finite type of the rule's global map is strictly proved, some dynamical properties on these subshifts are revealed. As an illustration, we give a simulation of the evolution of rule 41 with a random initial configuration in Figure 1, where the black pixel stands for 1 and white for 0.

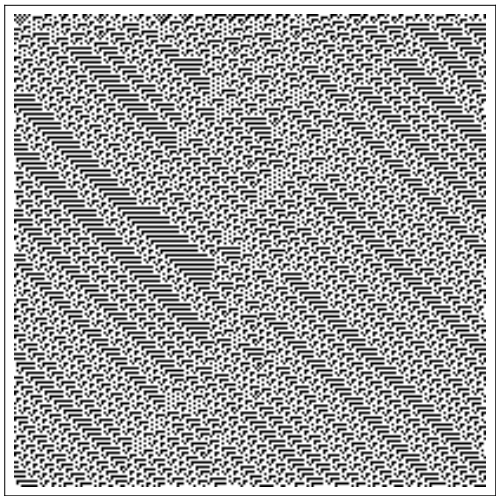


Fig. 1: The evolution of rule 41, from a random initial configuration.

The rest of this paper is organized as follows: Section 2 presents the preliminaries of symbolic dynamical systems and CA, and two lemmas. Section 3 explores two attractors

and some subshifts of finite type of rule's global map f_{41}. Section 4 demonstrates some complex dynamics of f_{41}. Finally, Section 5 concludes this paper.

2. Preliminaries

The bi-infinite binary symbols sequence space is a configuration set on $S = \{0, 1\}$:

$$\Sigma_2 = \{x = (\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) \mid x_i \in S, i \in Z\}$$

and the metric "d" on Σ_2 defined as

$$d(x, y) = \max_{i \in Z}\{\frac{\rho(x_i, y_i)}{2^{|i|}}\}$$

for any x, $y \in \Sigma_2$, where $\rho(\cdot, \cdot)$ is the metric on S defined as

$$\rho(x_i, y_i) = \begin{cases} 0, & if \quad x_i = y_i \\ 1, & if \quad x_i \neq y_i. \end{cases}$$

It is known that Σ_2 is a compact, perfect and totally disconnected metric space.

A finite sequence $a = (a_0, a_1, \cdots, a_{n-1})$ $(a_i \in S,\ i = 0, 1, 2, \cdots, n-1)$ is called a word over S. If $x \in \Sigma_2$ and $I = [i, j]$ is an interval of integers, put $x_{[i,j]} = (x_i, x_{i+1}, \cdots, x_j)$ $(i < j)$, $x_{[i,j)} = (x_i, \cdots, x_{j-1})$. For a word $a = (a_0, \cdots, a_{n-1})$, if there exists an $n \in Z$ such that $x_{n+k} = a_k$ $(k = 0, 1, \cdots, n-1)$, then say a be a subword of x, denoted by $a \prec x$; otherwise $a \not\prec x$.

The left-shift σ_L and right-shift σ_R are defined by

$$\sigma_L(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) = (\cdots, x_0, \overset{*}{x}_1, x_2 \cdots)$$

and

$$\sigma_R(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) = (\cdots, x_{-2}, \overset{*}{x}_{-1}, x_0, \cdots)$$

respectively.

By a theorem of Hedlund [3], a map $f : \Sigma_2 \to \Sigma_2$ is a cellular automaton iff it is continuous and commutes with σ, i.e. $\sigma \circ f = f \circ \sigma$, where σ is left-shift or right-shift.

A set $X \subseteq \Sigma_2$ is f-invariant if $f(X) \subseteq X$, and strongly f-invariant if $f(X) = X$. If X is a closed and f-invariant, then (X, f) or simply X is called a subsystem of f.

The omega-limit of a closed invariant set X is $\Omega_f(X) = \bigcap_{n \geq 0} f^n(X)$. A set $X \subset \Sigma_2$ is an attractor of f if there exists a clopen f-invariant set Y such that $\Omega_f(Y) = X$. The maximal attractor $\Omega_f = \Omega_f(\Sigma_2)$ is also called the limit set of f. A subshift is a non-empty subset $\Lambda \subset \Sigma_2$ which is strongly σ-invariant and closed. A subshift attractor of a cellular automaton is an attractor which is a subshift. For example, the maximal attractor is a subshift attractor [19].

Definition *1:* A closed and open (clopen) f-invariant set $U \subset \Sigma_2$ is spreading if $f^k(U) \subset \sigma_L(U) \cap U \cap \sigma_R(U)$ for some $k > 0$.

Lemma 1: [16] Let $f : \Sigma_2 \to \Sigma_2$ be a cellular automaton and U a clopen f-invariant set, then $\Omega_f(U)$ is a subshift attractor iff U is spreading.

Table 1: Truth table of Boolean function of Rule 41

(x_{i-1}, x_i, x_{i+1})	$\hat{f}(x_{i-1}, x_i, x_{i+1})$
(0, 0, 0)	1
(0, 0, 1)	0
(0, 1, 0)	0
(0, 1, 1)	1
(1, 0, 0)	0
(1, 0, 1)	1
(1, 1, 0)	0
(1, 1, 1)	0

For any subshift Λ, there exists a set $\mathscr{A}$ consisting of some words over $S = \{0, 1\}$ such that $\Lambda = \Lambda_{\mathscr{A}} = \{x \in \Sigma_2 |\ a \not\prec x, \forall\ a \in \mathscr{A}\}$, the set $\mathscr{A}$ named excluded block system. A subshift is of finite type if the set $\mathscr{A}$ is finite. The order of the finite type subshift denoted by N, which is the length of the longest word in $\mathscr{A}$.

Lemma 2: [17] For any subshift of finite type Λ, the followings statements are equivalent:
(1) there exists a set $\mathscr{A}$ consisting of some words over S such that $\Lambda = \Lambda_{\mathscr{A}} = \{x \in \Sigma_2 |\ a \not\prec x, \forall\ a \in \mathscr{A}\}$;
(2) there exists a set $\mathscr{B}$ consisting of some words over S such that $\Lambda = \{x \in \Sigma_2 |\ x_{[n,n+N-1]} \in \mathscr{B}, \forall n \in Z\}$, where N is the order of Λ. The set $\mathscr{B}$ named determinative block system of Λ.

It is well known that each ECA rule can be expressed by a Boolean function. For example, the one of rule 41 is a local map $\hat{f}$:

$$\hat{f}(x_{i-1}, x_i, x_{i+1}) = x_{i-1} \cdot \overline{x}_i \cdot x_{i+1} \oplus \overline{x}_{i-1} \cdot x_i \cdot x_{i+1} \oplus \overline{x}_{i-1} \cdot \overline{x}_i \cdot \overline{x}_{i+1}$$

where "$\cdot$", "$\oplus$", and "$-$" stand for "AND", "XOR" and "NOT" logical operations respectively [7, 13]. The truth table of its Boolean function is shown in Table 1.

It is clear that its binary output sequence is 10010100. Thus, a global map $f_{41} : \Sigma_2 \to \Sigma_2$ with

$$f_{41}(\cdots, x_{-1}, \overset{*}{x}_0, x_1, \cdots) = (\cdots, y_{-1}, \overset{*}{y}_0, y_1, \cdots)$$

can be induced by $\hat{f}$, where $y_i = \hat{f}(x_{i-1}, x_i, x_{i+1})$.

The n $(n \geq 2)$ times iteration of $\hat{f}$ is a map $\hat{f}^n$ from $\{0, 1\}^{2n+1}$ to $\{0, 1\}$ with

$$\hat{f}^n(a_{-n}, \cdots, a_0, \cdots, a_n) = \hat{f}(\hat{f}^{n-1}(a_{[-n,n-2]}), \hat{f}^{n-1}(a_{[-n+1,n-1]}), \hat{f}^{n-1}(a_{[-n+2,n]})).$$

3. Attractors and Subshifts of Finite Type

In this section, two attractors and some subshifts of finite type of the dynamical system (Σ_2, f_{41}) induced by rule 41

are revealed.

3.1 Attractors

Proposition 1: For rule 41, there exists an invariant subset $\Lambda \subset \Sigma_2$, such that $f_{41}(x) \in \Lambda, \forall\ x \in \Sigma_2$, where $\Lambda = \{x \in \Sigma_2 |\ a \not\prec x, \forall\ a \in \mathscr{A}_1\}$, and $\mathscr{A}_1 = \{(1,0,1,1,1), (1,1,1,0^{3k+1},1,1), (1,1,1,0^{3k+2},1,1,1),$ $k \in Z^+\}$, where 0^n is 0-constant block of length n, $n = 3k+1,\ 3k+2$.

Proof: Let $y = f_{41}(x)$, if $y_{[i,i+n-1]}$ is a word of y, then there exists a word $x_{[i-1,i+n]}$ of x, which is a pre-image of $y_{[i,i+n-1]}$, such that $\hat{f}_{41}(x_{[i-1,i+n]}) = y_{[i,i+n-1]}$.

Assume that $y_{[i,i+4]} = (1,0,1,1,1) \in \mathscr{A}_1$, and its a pre-image is $x_{[i-1,i+5]}$. It is easy to know that the pre-image set of $y_{[i,i+3]} = (1,0,1,1)$ is $\{(0,1,1,0,1,1), (1,0,1,0,1,1)\}$, and the pre-image set of $y_{[i+1,i+4]} = (0,1,1,1)$ is $\{(1,0,0,0,0,0)\}$. Since $y = f_{41}(x)$, so the pre-image of y must satisfy $x_{[i-1,i+4]} = \{(0,1,1,0,1,1)\}$ or $x_{[i-1,i+4]} = (1,0,1,0,1,1)$, and $x_{[i,i+5]} = (1,0,0,0,0,0)$. This lead to a contradiction, so the pre-image of $y_{[i,i+4]} = (1,0,1,1,1)$ is empty. Similarly, these words $(1,1,1,0^{3k+1},1,1)$ and $(1,1,1,0^{3k+2},1,1,1)$ must have no pre-image ($n = 3k+1, 3k+1, k \in Z^+$). This implies that $f_{41}(\Sigma_2) \subset \Lambda$ and $f_{41}(\Lambda) \subset \Lambda$. ■

Theorem 1: $\Omega_f(\Lambda)$ is a subshift attractor of f_{41}, where Λ is the invariant set obtained in Proposition 1.

Proof: In fact, Σ_2 is σ_L-invariant and σ_R-invariant, and Σ_2 is a clopen set, so Σ_2 is spreading, thus, $\Omega_f(\Sigma_2) = \bigcap_{n\geq 0} f_{41}^n(\Sigma_2)$ is a subshift attractor of f_{41}. Λ obtained in Proposition 1 must satisfy $f_{41}(\Sigma_2) \subset \Lambda$ and $f_{41}(\Lambda) \subset \Lambda$. This implies that $\Omega_f(\Sigma_2) = \Omega_f(\Lambda)$, thus $\Omega_f(\Lambda)$ is a subshift attractor of f_{41}. ■

Proposition 2: For rule 41, there exists an invariant subset $\Lambda_0 \subset \Lambda$, such that $\Omega_f(\Lambda_0) = \{0^*, 1^*\}$, where $\Lambda_0 = \{x \in \Sigma_2 |\ x_{[i,i+5]} \in \mathscr{A}_0\}$, and $\mathscr{A}_0$={(0, 0, 1, 0, 0, 1), (0, 1, 0, 0, 1, 0), (1, 0, 0, 1, 0, 0), (1, 1, 0, 0, 1, 0), (1, 1, 1, 0, 0, 1), (1, 1, 1, 1, 0, 0), (1, 1, 1, 1, 1, 0), (1, 1, 1, 1, 1, 1)}, 0^* and 1^* are the cycle configurations $0^* = (0^\infty)$ and $1^* = (1^\infty)$.

Proof: The result can be directly validated. ■

Theorem 2: $\{0^*, 1^*\}$ is a local attractor of f_{41}.

Proof: It is easily verified that $f_{41}(x) \in \{0^*, 1^*\}$, for any $x \in \Lambda_0$. ■

3.2 Subshifs of finite type

In this subsection, some subshifts of finite type are given out. Based on a computer-aided method, the following propositions can be easily verified:

Proposition 3: For rule 41, there exists a subset $\Lambda_1 \subset \Sigma_2$, such that $f_{41}|_{\Lambda_1} = \sigma_L|_{\Lambda_1}$. where $\Lambda_1 = \Lambda_{\mathscr{B}_1} = \{x \in \Sigma_2 |\ x_{[i-1,i+1]} \in \mathscr{B}_1, \forall\ i \in Z\}$ and the determinative block system $\mathscr{B}_1$ is a 3-sequence set, whose binary code set is $\mathscr{B}_1 = \{(0,1,0), (0,1,1), (1,0,1), (1,1,0)\}$.

For convenience, $\mathscr{B}_1$ can also be denoted by its decimal code set $D(\mathscr{B}_1) = \{2,3,5,6\}$.

Proposition 4: For rule 41, there exists a subset $\Lambda_2 \subset \Sigma_2$, such that $f_{41}^4|_{\Lambda_2} = \sigma_L^3|_{\Lambda_2}$. where $\Lambda_2 = \Lambda_{\mathscr{B}_2} = \{x \in \Sigma_2 |\ x_{[i-4,i+4]} \in \mathscr{B}_2, \forall\ i \in Z\}$ and the determinative block system $\mathscr{B}_2$ is a 9-sequence set, whose decimal code set is $D(\mathscr{B}_2)$={4, 8, 14, 15, 16, 20, 29, 30, 32, 40, 41, 58, 60, 64, 65, 76, 80, 82, 100, 116, 120, 129, 131, 133, 144, 147, 153, 161, 164, 200, 201, 232, 233, 241, 258, 263, 266, 285, 286, 288, 294, 306, 322, 328, 329, 398, 399, 400, 403, 417, 420, 455, 464, 466, 483}.

Proposition 5: For rule 41, there exists a subset $\Lambda_3 \subset \Sigma_2$,such that $f_{41}^4|_{\Lambda_3} = \sigma_R^4|_{\Lambda_3}$, where $\Lambda_3 = \Lambda_{\mathscr{B}_3} = \{x \in \Sigma_2 |\ x_{[i-4,i+4]} \in \mathscr{B}_3, \forall\ i \in Z\}$ and the determinative block system $\mathscr{B}_3$ is a 9-sequence set, whose decimal code set is $D(\mathscr{B}_3)$={0, 1, 2, 4, 5, 8, 10, 16, 20, 21, 32, 33, 34, 40, 42, 50, 57, 60, 62, 63, 64, 65, 66, 68, 76, 78, 79, 80, 85, 100, 101, 114, 120, 121, 124, 126, 127, 128, 129, 130, 133, 136, 153, 156, 158, 159, 160, 161, 170, 182, 201, 202, 214, 218, 228, 229, 241, 242, 248, 249, 252, 254, 255, 256, 257, 258, 261, 266, 273, 281, 284, 286, 287, 294, 295, 298, 306, 313, 316, 318, 319, 320, 321, 322, 341, 347, 363, 365, 396, 398, 399, 403, 405, 429, 437, 454, 455, 457, 458, 483, 484, 485, 497, 498, 504, 505, 508, 510, 511}.

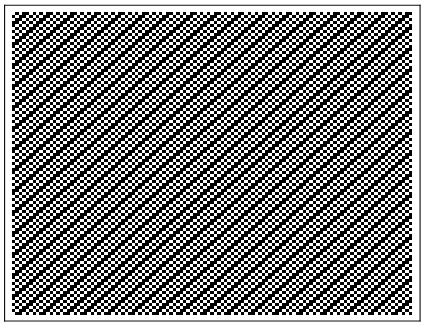


Fig. 2: The evolution of rule 41 from an initial configuration of Λ_1.

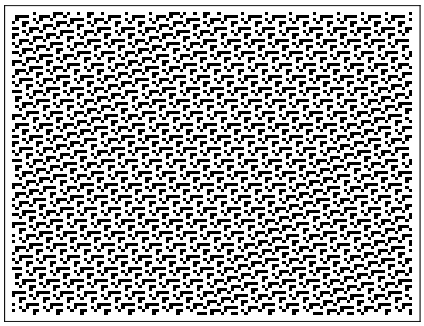


Fig. 3: The evolution of rule 41 from an initial configuration of Λ_2.

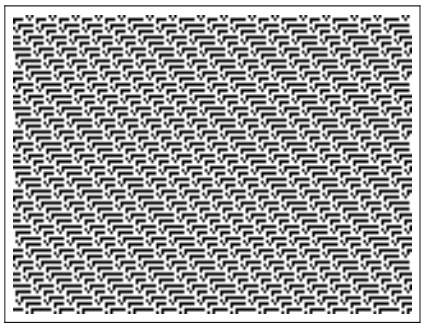


Fig. 4: The evolution of rule 41 from an initial configuration of Λ_3.

Thus, Λ_1, Λ_2 and Λ_3 are subshifts of finite type of f_{41}. For illustration, simulations on Λ_1, Λ_2 and Λ_3 are shown in Figure 2, 3 and 4.

4. Complex Dynamics

In this section, the complexity and chaotic dynamics of f_{41} are explored. Since the topological dynamics of a subshift of finite type is largely determined by the properties of its transition matrix, it is helpful to briefly review some definitions about irreducible and aperiodic [18]. A matrix A is positive if all of its entries are non-negative, irreducible if $\forall\ i, j$, there exists n such that $A_{ij}^n > 0$, aperiodic if there exists N, such that $A_{ij}^n > 0$, $n > N$, $\forall i, j$. If $\Lambda_{\mathscr{A}}$ is a two-order subshift, then it is topologically mixing if and only if A is aperiodic, topological transitive if and only if A is irreducible. A is the associated transition matrix of subshift with $A_{ij} = 1$, if $(i, j) \prec \mathscr{A}$; otherwise $A_{ij} = 0$.

The topologically conjugate relationship between $(\Lambda_{\mathscr{A}}, \sigma)$ and a two-order subshift of finite type can be established, and the dynamical behavior of f_{41} on $\Lambda_{\mathscr{A}}$ can be discussed based on existing results.

Let $\hat{S} = \{s_0, s_1, s_2, s_3\}$ be a new symbolic set, where s_i $(i = 0, 1, 2, 3)$ stand for (0, 1, 0), (0, 1, 1), (1, 0, 1) and (1, 1, 0) appeared in Proposition 3, then one can construct a new symbolic space $\hat{S}^Z$ on $\hat{S}$. Let
$\hat{\mathscr{A}} = \{(s, s')|\ s = (b_1, b_2, b_3),\ s' = (b_1', b_2', b_3') \in \mathscr{B}_1, b_j = b_{j-1}', 2 \le j \le 3\}$, where $\mathscr{B}_1$ is the determinative block system of Λ_1 in Proposition 3. Further, the 2-order subshift $\Lambda_{\hat{\mathscr{A}}}$ of σ is defined by $\Lambda_{\hat{\mathscr{A}}} = \{(\dots, r_{-1}, r_0, r_1, \dots) \in \hat{S}^Z|\ r_i \in \hat{S}, (r_i, r_{i+1}) \prec \hat{\mathscr{A}}, \forall\ i \in Z)\}$. Thus, the transition matrix A_1 of the subshift $\Lambda_{\hat{\mathscr{A}}}$ is

$$A_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Proposition 6: $(\Lambda_{\mathscr{B}_1}, \sigma)$ and $(\Lambda_{\hat{\mathscr{A}}}, \sigma)$ are topologically conjugate; namely, $(\Lambda_{\mathscr{B}_1}, f_{41})$ and $(\Lambda_{\hat{\mathscr{A}}}, \sigma)$ are topologically conjugate.

Proof: Define a map from $\Lambda_{\mathscr{B}_1}$ to $\Lambda_{\hat{\mathscr{A}}}$ as follows :

$$g : \Lambda_{\mathscr{B}_1} \to \Lambda_{\hat{\mathscr{A}}}$$

$$(\dots, x_{-1}, \overset{*}{x}_0, x_1, \dots) \to (\dots, r_{-1}, \overset{*}{r}_0, r_1, \dots)$$

where $r_i = (x_i, x_{i+1}, x_{i+2}) \in \mathscr{B}_1$, $\forall\ i \in Z$. One can easily check that g is a homeomorphism and $g \circ \sigma = \sigma \circ g$. Therefore, $(\Lambda_{\mathscr{B}_1}, \sigma)$ and $(\Lambda_{\hat{\mathscr{A}}}, \sigma)$ are topologically conjugate. ■

It is known that σ is topological mixing on $\hat{\mathscr{A}}$ if and only if the transition matrix A_1 is aperiodic. Thus, the following theorem is obtained.

Theorem 3:
(1) $f_{41}|_{\Lambda_1} = \sigma_L|_{\Lambda_1}$ is topological mixing;
(2) The topological entropy of $f_{41}|_{\Lambda_1}$ satisfies $ent(f_{41}|_{\Lambda_1}) = log(\rho(A_1)) \approx 0.1221$, where $\rho(A_1)$ is the spectral radius of the transition matrix A_1.

A positive topological entropy is often taken as a signature of chaos. A system with topological mixing property has many chaotic properties in different senses such as Devaney and Li-Yorke.

Corollary 1:
(1) $f_{41}|_{\Lambda_1}$ is chaotic in the sense of Li-Yorke;
(2) $f_{41}|_{\Lambda_1}$ is chaotic on Λ_1 in the sense of Devaney.

Theorem 4:
(1) $f_{41}|_{\Lambda_2}$ is topological transitive on Λ_2;
(2) $f_{41}|_{\Lambda_2}$ is chaotic in the sense of Li-Yorke on Λ_2.

Proof: (1) It can be proved that the transition matrix A_2 corresponding to the subshift Λ_2 is irreducible, so $\sigma_L|_{\Lambda_2}$ is topological transitive. Thus, $\sigma_L^3|_{\Lambda_2}$ is topological transitive. It follows from proposition 4 that $f_{41}^4|_{\Lambda_2} = \sigma^3|_{\Lambda_2}$, so $f_{41}^4|_{\Lambda_2}$ is topological transitive. To prove $f_{41}|_{\Lambda_2}$ is topological transitive, one only need to check that for any two open sets $U,\ V \subset \Lambda_2, \exists N > 0$, such that $f_{41}^N(U) \bigcap V \neq \phi$, since $f_{41}^4|_{\Lambda_2}$ is topological transitive, thus, $\exists\ k > 0$, such that $(f_{41}^4)^k(U) \bigcap V \neq \phi$. Hence $\exists N = 4k$, such that $f_{41}^N(U) \bigcap V \neq \phi$, this means $f_{41}|_{\Lambda_2}$ is topological transitive.

(2) It follows that $ent(\sigma_L^3|_{\Lambda_2}) = 3 \cdot ent(\sigma_L|_{\Lambda_2})$. It is easy to compute that $ent(f_{41}|_{\Lambda_2}) = \frac{ent(f_{41}^4|_{\Lambda_2})}{4} = \frac{3 \times log(\rho(A_2))}{4} \approx 0.10657$. ■

Similarly, the dynamical behavior of f_{41} on Λ_3 can be analyzed via the property of the corresponding transition matrix. But the transition matrix A_3 is neither irreducible nor aperiodic. In order to investigate the dynamic behavior on the subshift of finite type Λ_3. Now, take two invariant sets $\Lambda_3^1,\ \Lambda_3^2 \subset \Lambda_3$, their determinative block systems are 9-sequence sets $\mathscr{B}_3^1$ and $\mathscr{B}_3^2$ respectively. Let their decimal codes are $D(\mathscr{B}_3^1)$={21, 42, 85, 101, 170, 202, 229, 298, 341, 405, 458, 485} and $D(\mathscr{B}_3^2)$={0, 1, 2, 4, 5, 8, 10, 16, 20, 32, 33, 34, 40, 50, 57, 60, 62, 63, 64, 65, 66, 68, 76, 78, 79, 80, 100, 114, 120, 121, 124, 126, 127, 128, 129, 130, 133, 136, 153, 156, 158, 159, 160, 161, 182, 201,

214, 218, 228, 241, 242, 248, 249, 252, 254, 255, 256, 257, 258, 261, 266, 273, 281, 284, 286, 287, 294, 295, 306, 313, 316, 318, 319, 320, 321, 322, 347, 363, 365, 396, 398, 399, 403, 429, 437, 454, 455, 457, 483, 484, 497, 498, 504, 505, 508, 510, 511}. Moreover, the topological entropy of $f_{41}|_{\Lambda_3^2}$ is $ent(f_{41}|_{\Lambda_3^2}) = log(\rho(A_3^2)) \approx 0.294$, where A_3^2 is the transition matrix corresponding to Λ_3^2.

Theorem 5: $f_{41}|_{\Lambda_3^2}$ is chaotic in the sense of Li-Yorke.

It is obviously that the subshift of finite type $\Lambda_1, \Lambda_2, \Lambda_3 \subset \Lambda$. The dynamical behavior of rule 41 are very complex on its subshift attractor. However, the dynamics should be explored on the invariant set Λ. It has been proved that the only subshift attractor of a surjective cellular automata is the full space [19]. Additive one-dimensional cellular automata defined on a finite alphabet of prime cardinality are chaotic in the sense of Devaney [22]. Rule 41 is not a surjective CA, since it is useful to study the dynamical behavior and to find its maximum attractor. In spite of the dynamics of f_{41} on the finite type subshifts has been made clear, but the ones on Λ should be further studied.

5. Conclusion

One of the main challenges is to explore the quantitative dynamics in cellular automata evolution [24]. this work has developed an elementary and rigorous proof to predict the rich and complex dynamics of rule 41 in view of symbolic dynamical systems. For example, the rule is topological mixing, topological transitive, and possesses positive topological entropies on some subshifts of finite type, thus, it is chaos in the sense of Li-Yorke or Devaney. At the same time, an invariant set is found, which includes the maximum attractor of rule 41. Indeed, the dynamics of rule 41 have not been completely revealed. Some new methods should be exploited to investigate the subshift in future study.

Acknowledgments

This research was jointly supported by the NSFC (Grant No. 10832006 and No. 60872093).

References

[1] J. von Neumann, *The Theory of Self-reproducing Automata,* A. W. Burks (ed), Univ. of Illinois Press, Urbana and London, 1996.

[2] G. A. Hedlund, *Endomorphisms and automorphism of the shift dynamical system*, Theory of Computing Systems, 3: 320-375, 1969.

[3] S. Wolfram, *Statistical mechanics of cellular automata*. Rev. Mod. Phys., 3:601-644, 1983.

[4] S. Wolfram, *Theory and applications of cellular automata,* World Scientifc, Singapore, 1986.

[5] S. Wolfram, *A new kind of science*. Wolfram Media, Inc, 2002.

[6] L. O. Chua, V. I. Sbitnev, and S. Yoon, *A nonlinear dynamics perspective of Wolfram's new kind of science. Part IV: From bernoulli shift to 1/f spectrum*. International Journal of Bifurcation and Chaos, 15(4): 1045-1183, 2005.

[7] L. O. Chua, V. I. Sbitnev and S. Yoon, *A nonlinear dynamics perspective of Wolfram's new kind of science. Part VI: From time-reversible attractors to the arrows of time*. International Journal of Bifurcation and Chaos, 16(5): 1097-1373, 2006.

[8] L. O. Chua, J. B. Guan, I. S. Valery and J. Shin, *A nonlinear dynamics perspective of Wolfram's new kind of science. Part VII: Isle of Eden*, Int. J. Bif. Chaos, vol. 17 (9): 2839-3012, 2007.

[9] J. B. Guan, S. W. Shen, C. B. Tang and F. Y. Chen, *Extending Chua's global equivalence theorem on Wolfram's new kind of science*. International Journal of Bifurcation and Chaos, 17(12): 4245-4259, 2007.

[10] F. Blanchard and A. Maass, *Dynamical properties of expansive one-sided cellular automata*, Israel J. Math. 99 149-174, 1997

[11] B. Codenotti and L. Margara, *Transitive cellular automata are sensitive*, The American Mathematical Monthly, 103(1): 55-62, 1996.

[12] P. Kurka, *Languages equicontinuity and attractors in cellular automata*, Ergodic Theory Dynamical Systems 17 417-433,1997

[13] F. Y. Chen, W. F. Jin, G. R. Chen, F. F. Chen and L. Chen, *Chaos of elementary cellular automata rule 42 of Wolframs class II.* Chaos, 19(1), 013140 1-6, 2009.

[14] W. F. Jin, F. Y. Chen, G. R. Chen, L. Chen and F. F. Chen, *Extending the symbolic dynamics of Chua's Bernoulli-shift rule 56*, Journal of Cellular Automata, 5 (1-2): 121-138, 2010.

[15] G. R. Chen, F. Y. Chen, J. B. Guan and W. F. Jin, Symbolic dynamics of some Bernoulli-shift cellular automata rules,*The 2010 International Symposium on Nonlinear Theory and its Applications (NOLTA 2010)*, Krakow, Poland, September 5-8, 2010.

[16] E. Formenti and P. Kurka, *Subshift attractors of cellular automata* Nonlinearity, 20: 105-117, 2007.

[17] Z. L. Zhou, *Symbolic Dynamics*, Shanghai Scientific and Technological Education Publishing House, Shanghai, 1997.

[18] B. Kitchens, *Symbolic Dynamics: One-sided, Two-sided and Countable State Markov Shifts*, Springer-verlag, Berlin, 1998.

[19] A. Maass, *On the sofic limit set of cellular automata*. Theory Dyn. Syst. 15 663-84, 1995

[20] R. L. Devaney, *An introduction to chatic dynamical systems,* Addison-Wesley, Reading, MA, 1989.

[21] P. Kurka, *On the measure attractor of a cellular automaton*, Discrete Contin. Dyn. Syst., 524-35, 2005

[22] P. Favati, G. Lotti and L. Margara, *Additive one-dimensional cellular automata are chaotic according to Devaney's definition of chaos*. Theoretical Computer Science, 174: 157-170, 1997.

[23] W. F. Jin and F. Y. Chen, *Global attractors and chaos of complex Bernoulli-shift rules*, The 2010 International Workshop on Chaos-Fractal Theory and its Applications (IWCFTA 2010), Kunming, Yunnan, China, October 28-31, 2010.

[24] J. Kari, *Theory of cellular automata: A survey*, Theor. Comput. Sci., 334: 3-33, 2005.

[25] Chen, G. R., Chen F. Y., Guan J. B., Jin, W. F. Symbolic dynamics of some Bernoulli-shift cellular automata rules, *The 2010 International Symposium on Nonlinear Theory and its Applications (NOLTA 2010)*, Krakow, Poland, September 5-8, 2010.

Cellular automata modeling of nanopore formation in passive layers

W. Chmielewski[1], **D. di Caprio**[2,3], **and J Stafiej**[1]
[1]Department of Complex Systems and Chemical Processing of Information,
Institute of Physical Chemistry, Polish Academy of Sciences, Warsaw, Poland.
[2]Chimie ParisTech, Laboratory of Electrochemistry, Chemistry of Interfaces and
Modelling for Energy (LECIME), 75005 Paris, France
[3]CNRS, UMR 7575, 75005 Paris, France

Abstract— *Nanopore formation phenomena in passive layers on anodized metal surfaces are modeled based on cellular automata approach. The preliminary simulation results obtained in 2D and 3D systems show self ordering of the pore structure.*

Keywords: Cellular automata modelling, self organization of nanopores in passive layers

1. Introduction

Cellular automata (CA) approach can successfully describe many real systems such as: ecosystems [1], forest systems in fire spreading problems [2], traffic in public transport systems [3], cell cultures [4], bacterial colonies [5], plant morphogenesis [6]. Here we consider cellular automata approach for a physicochemical systems - the growth of a porous oxide layer on a passivating metal surface subject to intense anodization. Cellular automata based computer simulations have been successfully used to study the related field of surface growth phenomena [7]. They provide basic tests of the validity of scaling concepts for these phenomena. Various universality hypothesis for the crossover transitions could be successfully demonstrated using the simulations [8]. In their applications to physical chemistry and related material science problems cellular automata models are the method of choice where the atomic level microscopic modeling is hardly applicable. This is the case of corrosion and passivation phenomena where the crucial events of passivity breakdown occur in time scales making microscopic approach impractical if not impossible [9]-[14]. The cellular automata description is used to construct a general model for a wider range of systems to describe the features that these systems share independent of their chemical specificity and molecular background having a common physicochemical origin. The corrosion and passivation phenomena on metal surfaces meet this requirement. Many metal materials cover with the passivation layer in contact with the atmosphere in an ambient environment. Very recently we have published a cellular automata model for this phenomenon illustrating a well known fact that the higher reactivity of the surface gives a faster passive layer formation and thus the surface is better protected in a more aggressive environment characterized by higher bare corrosion rates rather than in milder environment where passive layer cannot readily form [15], [16]. The pitting corrosion has been subject to many studies also by CA models as it is the primary cause of material destruction [17]. In our previous work we show how the phenomenon of spontaneous symmetry breaking leading to a formation of cathodic and anodic zones on the corroding surface arises in the CA-type simulations [18], [19]. In this paper we treat a certain positive aspect of passive layer dissolution mechanisms. As shown by a numerous literature the anodic dissolution of passivating metals leads to a formation and regular spacial organization of nanopores in the passive layer [20]. The widest known example is the anodized alumina oxide (AAO) layer obtained on aluminum in contact with various electrolytes and under various anodization protocols. It finds numerous applications in nanotechnology. There have been several attempts to describe the phenomenon based on macroscale physicochemical laws for electrostatics and chemical kinetics [20]. Here we give an alternative approach by a CA-type model based simulations. The arguments for doing this are the following. The pores have a mesoscopic size and arrangement. They appear on a variety of systems. They appear on polycrystalline materials and the underlying structure of the granular material has little if any effect on them. It follows that their formation and arrangement are independent of atomic scale structure. However, the reminiscent of the microscopic atomic level nature of phenomena is the stochastic character of processes as it is in the Brownian motion of mesoscopic sized particles suspended in a solvent.

2. Cellular automata model for nanopore formation on anodized surfaces

2.1 Lattice representation of the passivating system

As sketched in Figure 1 we use square lattice with von Neumann [21] four and six nearest neighbor connectivity in 2D and 3D respectively. The lattice sites are found in four main states or, as we say in physical chemistry, can be occupied by four species. These four species are metal

Fig. 1: Schematic representation of the states in the CA model for passive layer. The grey scale codes are as follows. Metal (M) bulk sites (on the left hand side) and solution (S) bulk sites (on the right) are blank as no storage is reserved for them in the program. The bar indicates the grey code for inactive oxide, active metal, active oxide, active solvent and active walker from left to right respectively.

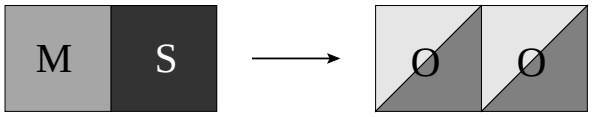


Fig. 2: Oxide creation event. Metal site M in contact with solvent site S are transformed to two oxide sites with probability P_{OX}. These can be active if in contact with solution sites (dark gray) and inactive otherwise (light gray).

(M), solution (S), oxide (O) and walker (W). Because of the transformation rules adopted these species can come in two flavors – active or inactive. The inactive species can change their state only because at least one of the neighboring site has changed.

2.2 Transformation rules for the system evolution

There are six transformation rules governing the behavior of our system presented in Figures 2-8. Some transformations modify the number of sites in a given state, that is the number of given species. This is the case depicted in Figure 2, where the creation mechanism of the oxide layer

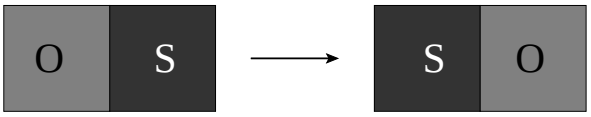


Fig. 3: Oxide random walk by swapping with the solution site with probability P_{swap}.

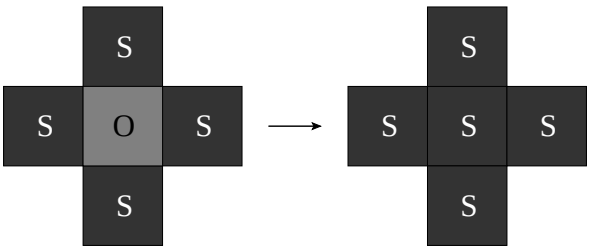


Fig. 4: Annihilation of a single oxide site in the solution.

is presented. Other transformations like swaps, only modify the spatial distribution of the species. As in [15], [16], the swaps of the oxide sites are associated to the probability of swap, P_{swap}. This is calculated on the basis of the number of broken bonds N_{broken} that is the number of nearest neighbor O-O pairs if the swap be performed less the number of nearest neighbor O-O pairs in the actual configuration. This algorithm provides a certain cohesion to the formed layer, as the probability of an oxide detaching from the layer is less than that for attaching. The oxide sites appear to *stick* to one another. The probability is expressed as

$$\begin{aligned} P_{swap} &= P_{bond}^{N_{broken}} \text{ if } N_{broken} > 0 \\ P_{swap} &= 1 \text{ otherwise} \end{aligned} \quad (1)$$

where P_{bond} is the probabilty of breaking a single bond. As seen in Figure 1, in some cases, the oxide sites may detach from the layer forming single site islands wandering in the solution. We make such islands disappear with a probability P_{die}. This mechanism mimics the dissolution of the formed oxide layer. The balance between processes of Figures 2 and 4, creation and dissolution, determines the layer thickness. A similar set of rules apply to walker creation, diffusion and annihilation. Walker sites can execute random walk by swapping with the neighboring O sites. In this case, there is no restriction to the swaps as such a move is performed at each step. If in the random walk the walker encounters a solution site it may annihilate as shown in Figure 7. It is required that the site S on which the walker attempts to step

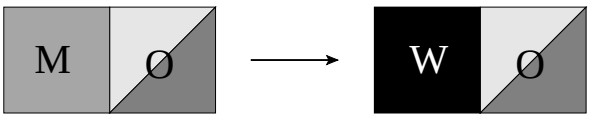


Fig. 5: Walker creation with probability P_{CW}.

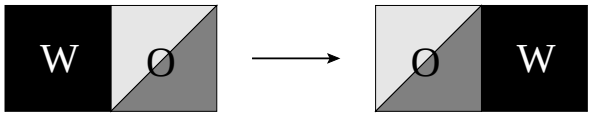


Fig. 6: Walker random walk by swapping with oxide sites with probability P_{OW}.

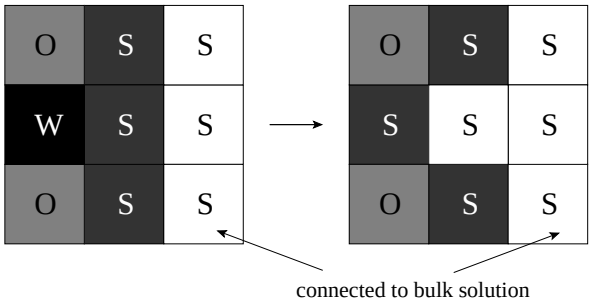


Fig. 7: Walker annihilation with probability P_{WA}.

is connected to the bulk solution or in other words belongs to the main solution front. During the random walks the site to swap with is a randomly selected nearest neighbor. For the oxide no action is taken if the selected site is not solution. Similarly walker swaps only with an oxide site. When the selected neighbor solution site is from, so to say, a sea or a bay or an ocean connected to the ocean of bulk solution the walker disappears and its site becomes the part of the oceanic connected set. If the solution site belongs to an inland lake like in Figure 8 nothing happens. We use asynchronous updating both for creation and random walk. The sites to update are selected randomly from a given type until the number of selections equals number of sites. It may happen that a given site is selected more than one time while another site is never selected.

3. Results

Preliminary simulation results obtained both in 2D and 3D are shown in Figures 9-12. The values of the parameters are $P_{OX} = P_{OW} = P_{WA} = 1$, $P_{die} = 0.001$, $P_{bond} = 0.01$ and $P_{CW} = 3.33 \cdot 10^{-4}$. They have been selected by a trial and

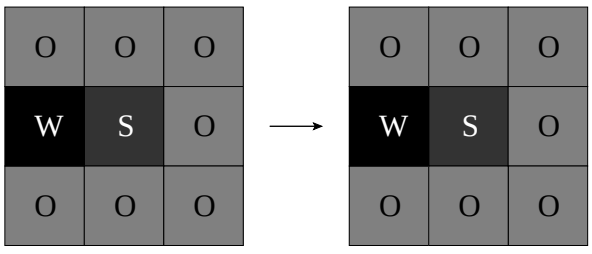


Fig. 8: If the neighboring site is a solution site trapped in the oxide the walker does not annihilate and no action is taken when it attempts to step on the solution site.

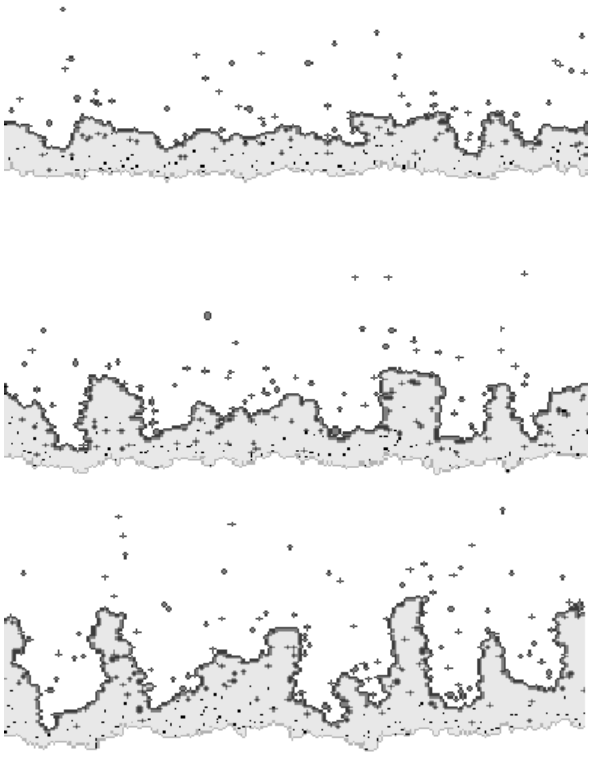
error method as our first goal is to document the existence of the rough effect of pore formation and pore self organization. The box width is 256 and 128x128 in 2 and 3D respectively. The height is adjusted according to the layer evolution.
In Figure 9, we illustrate the formation and evolution of a 2D layer. The snapshots show that starting from a featureless roughness, the premices of a self organization are visible. This behaviour is confirmed in Figure 10, by the appearance of a characteristic wave length in the Fourier transform of the layer-solution interface. A similar scenario is also observed in 3D. In this case, the Fourier transform pattern also seems to show a relative decay of the four folded symmetric pattern characteristic of the lattice and the appearance of a pattern with a distinct symmetry.
The evolution of the layer in the course of the simulation, first involves and instability created by the coupling between the corrosion mechanism at the metal-layer interface with the dissolution at the layer-solution interface due to the diffusion of the W species. Then a more stable regime which appears to be self-organized is reached dynamically as the result of the balance between the evolution of the two fronts and the diffusion characteristic length of the walkers. We do not know for now the exact competition mechanism between the different length scales in the system.
Finally, the coupling between metal-layer and layer solutionn interfaces through the walker species appears to be essential. In the framework of the electrochemistry, the walker species can represent chemical species such as excess of metal or oxygen acidic species between the two fronts. The role of those species is to accelerate the dissolution of the layer-solution interface. Another possible interpretation of the walkers, is to mimic the effect of the electric potential distribution and the electric field, which is known to be highly inhomogeneous for anodic dissolution experiments where high voltages are applied. This comes from the fact that stationary walker distribution obeys the Laplace equation of the same form as the electric potential.

4. Conclusion

The presented CA based approach reproduces qualitatively the nanopore formation mechanism. We show that a simple model involving the dynamics of two fronts: a corrosion front and a dissolution front coupled by the diffusion of a species can lead from a disordered situation to the gradual appearance of a seemingly self-organized system. From the first encouraging results of this minimal model, we need further refinements for a clearer hexagonal arrangement of the pores in 3D.

Acknowledgments

Work supported by Polish Ministry of Science and Higher Education, grant N N204 139038.



a

b

c

Fig. 9: Snapshots of fragments of the simulated layer in 2D at 20, 40 and 80 thousand time steps from up to down respectively.

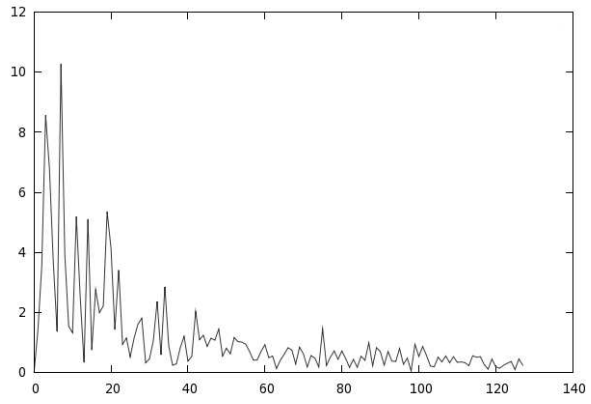


Fig. 10: The absolute value of the FFT transformed layer front showing a characteristic wavelength of ca 50 nodes.

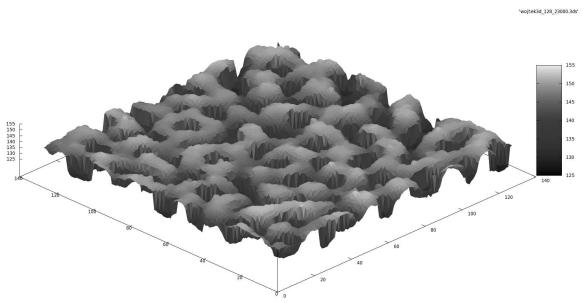


Fig. 11: View of the surface layer surface in 3D after 25 000 time steps.

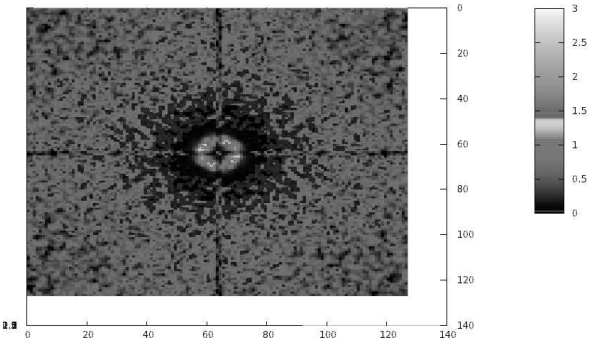


Fig. 12: FFT transform of the layer surface after 25000 time steps.

References

[1] A.G. Dunn and J. D. Majer, *Simulating Weed Propagation Via Hierarchical, Patch-Based Cellular Automata*, in Computational Science, ser. Lecture Notes in Computer Science - ICCS 2007. Berlin: Springer, 2007, vol. 4487, pp. 762–769.

[2] A. Hernández Encinas, L. Hernández Encinas, S. Hoya White, A. Martín del Rey and, G. Rodríguez Sánchez, "Simulation of forest fire fronts sing cellular automata", *Advances in Engineering Software*, vol. 38,pp. 72–378, 2007.

[3] D. Chowdhury, L. Santen, and A. Schadschneider, "Statistical physics of vehicular traffic and some related systems", *Physics Reports*, vol. 329, pp. 199–329, 2000.

[4] Y. Lee, S. Kouvroukoglou, L. McIntire and, K. Zygourakis, *Biophysical Journal*, vol. 69, pp. 1284–1298, 1995.

[5] J.W. T. Wimpenny and, R. Colasanti, *FEMS Microbiology Ecology*, vol. 22, pp. 1–16, 2006.

[6] G.G. Lazareva, V.V. Mironova, N. A. Omelyanchuk, I. V. Shvab, V.A. Vshivkov, D.N. Gorpinchenko, S.V. Nikolaev and, N.A. Kolchanov, *Numerical Analysis and Applications*, vol. 1, pp. 123–134, 2008.

[7] A.L. Barabási and, H.E. Stanley, *Fractal Concepts in Surface Growth*, Cambridge, Cambridge University Press, 1995.

[8] F.D.A.A. Reis, "Scaling in the crossover from random to correlated growth", *Phys. Rev. E*, vol. 73, pp. 021605-1–021605-7, 2006.

[9] D. di Caprio and, J.Stafiej, "Cellular automata approach to corrosion and passivity phenomena," in *Proceedings of the International Conference on Scientific Computing (CSC'10), ISBN 1-60132-137-6*, pp.62–66.

[10] Li Lei, Li Xiaoganga, Dong Chaofang and, Huang Yizhong, "Computational simulation of metastable pitting of stainless steel", *Electrochimica Acta*, vol. 54, pp. 6389–6395, 2009.

[11] L. Li, X. Li, C. Dong, K. Xiao and, L. Lu, "Cellular automata modeling on pitting current transients", *Electrochemistry Communications*, vol. 11, pp. 1826–1829, 2009.

[12] Xiaofei Yu, Shenhao Chen, Ying Liu and, Fengfeng Ren, "A study of intergranular corrosion of austenitic stainless steel by electrochemical potentiodynamic reactivation, electron back-scattering diffraction and cellular automaton", *Corrosion Science*, vol. 52, pp. 1939–1947, 2010.

[13] S.V. Lishchuka, R. Akida, K. Wordenb and, J. Michalskic "A cellular automaton model for predicting intergranular corrosion", *Corrosion Science*, in press, 2011.

[14] A. Taleb and, J. Stafiej, "Numerical simulation of the effect of grain size on corrosion processes: Surface roughness oscillation and cluster detachment ", *Corrosion Science*, in press, 2011.

[15] D. di Caprio and, J. Stafiej, "Simulations of passivation phenomena based on discrete lattice gas automata", *Electrochimica Acta*, vol. 55, pp. 3884–3890,2010.

[16] D. di Caprio and, J. Stafiej, "The role of adsorption in passivation phenomena modelled by discrete lattice gas automata", *Electrochimica Acta*, vol. 56, pp. 3963–3968, 2011.

[17] Z. Szklarska-Smialowska, *Pitting and Crevice Corrosion*, Houston, NACE International, 2005.

[18] C. Vautrin-Ul, A. Chausse , J. Stafiej and, J. P. Badiali, "Simulations of corrosion processes with spontaneous separation of cathodic and anodic reaction zones", *Pol. J. Chem.*, vol. 78, pp. 1795–1810, 2004.

[19] F.D.A.A. Reis, J. Stafiej and, J.P. Badiali, "Scaling theory in a model of corrosion and passivation", *J. Phys. Chem. B*, vol. 110, pp 17554–17562, 2006.

[20] L. Stanton and, A. Golovin, "Effect of ion migration on the self-assembly of porous nanostructures in anodic oxides", *Phys.Rev.B* vol.79 pp 035414:1-7,2009.

[21] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems*, Cambridge, Cambridge University Press, 1998.

Cycles, Transients, and Complexity in the Game of Death Spatial Automaton

K.A. Hawick and C.J.Scogings
Computer Science, Institute for Information and Mathematical Sciences,
Massey University, North Shore 102-904, Auckland, New Zealand
email: k.a.hawick@massey.ac.nz, c.scogings@massey.ac.nz
Tel: +64 9 414 0800 Fax: +64 9 441 8181

ABSTRACT
Cellular automaton models such as Conway's Game of Life have long been shown to exhibit a high degree of spatial complexity. Spatial patterns can be analysed and categorised in this and other models and used as a basis for cataloging other related models and their behaviour classes. An interesting variation arises when a third state is introduced and we explore the consequences of this in models like Silverman's "Brian's Brain" – sometimes known as the "Game of Death" where "zombies" are introduced into the spatial model. The third state and the microscopic rules associated with the three constituent species gives rise to a rich new set of phases and behaviours with can be simulated and catalogued statistically. We focus on the early transient behaviour following a random system initialisation and the initial thinning out following by a subsequent explosion in species diversity.

KEY WORDS
multi-agent model; cellular automata; biodiversity; cyclic states.

1 Introduction

Conway's Game of Life [1] (GoL) has served for some time as a key cellular automaton (CA) [2] model for comparing other model systems. CAs are useful tools for investigating complexity [3, 4] and spontaneous emergent properties because they are generally simpler than other models that yield insights into fundamental properties such as universality [5], growth [6], and other statistical mechanical properties such as scaling [7], the onset of chaos [8], phase transitions [9], fluid-flow miscibility and entropy [10, 11] and game-theoretic predictions [12].

A range of CAs [13] have proved able to capture some of the fundamental properties [14] on more sophisticated artificial life models [15] developed for the study of population dynamics and species diversity. CA models typically show a rich set of spatial patterns which can often be related to fundamental driving forces to catalogue and explain complex emergent phenomena [16] such as spiral formation [17, 18] and fluctuations in population dynamics [19].

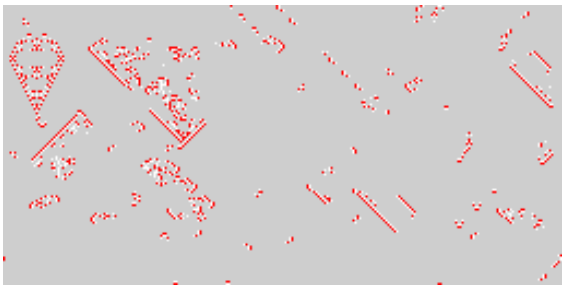


Figure 1: A typical configuration of the 3-state Game of Death on a 256×128 cell mesh, after 640 time-steps.

A great deal of research effort has been reported on extensions and variations of the GoL [20]. Recent research work has highlighted the significance of cyclic rules in automaton and related models with symmetry and other properties having been identified as drivers behind some of the complex behaviour in rock-paper-scissors and related models. Silverman introduced a cycle into the GoL to produce what was colloquially known as the "Game of Death" [21] or "Brian's Brain" [22]. In this model (as shown in Figure 1) a third zombie state is introduced and the automaton rules extended accordingly.

CAs are usually studied as computer simulations [23, 24] with important technical considerations such as finite-sizing [25], the introduction of randomness and stochasticity [26], synchronicity of spatial updates [27] and performance analysis [28] all giving rise to discussion in the literature. Important work has covered self-reproduction [29] and ratcheting effects [30] within CA rule spaces as well as attempts to characterise the rules spaces themselves [31].

Much reported research has focused on the specific automaton patterns that reproduce, or self-sustain. In this

present work we look at statistical properties rather than those from specific patterns of automaton cells. We report on experiments with the Game of Death 3-state spatial automata played with a Moore cell neighbourhood in both two and three dimensions. Figure 1 shows a typical configuration of the Game of Death played with a Moore cell neighbourhood of eight cells on a square periodic mesh. The configuration shows spaceships, rakes and other common patterns reported in the literature [21]. We particularly focus on the early transient stage following a random initialisation of spatial automata system. This is interesting to investigate as a platform to investigate why does the system seek out a particular sort of configuration, independent of the microscopic noise in the starting state.

Our article is structured as follows: In Section 2 we describe the automaton rules for the 3-state cyclic Game of Death automaton and describe our simulation method. We present some typical configurations and measurement analysis results in Section 3. In Section 4 we discuss some of the implications for cyclic CA models as well as offering some concluding remarks and areas for further work in Section 5.

2 Spatial Automaton Rules

Automaton games such as the Game of Life are typically played using totalistic rules applied to spatial cells arranged on a regular mesh. Many physics-oriented automata and models use strictly nearest neighbour sets where there are four neighbours in two dimensions and six in three dimensions. The totalistic rules used in GoL like models however are expressed in terms of the Moore neighbourhoods which include diagonally-touching neighbours that are at distances $\sqrt{2}$, $\sqrt{3}$ as well as unit-distance nearest-neighbours. In two dimensions there are 8 Moore neighbours and in three dimensions there are 26.

Conway's GoL rules apply to dead (0) or live(1) cells:

- any live cell with fewer than two ($r_4 = 2$) live neighbours dies
- any live cell with more than three ($r_3 = 3$) live neighbours dies
- any live cell with exactly two or three live neighbours lives on
- any dead cell with exactly three ($n >= r_1, n <= r_2 : r_1 = r_2 = 3$)live neighbours becomes live

We have given the R_i values in this form to emphasise that there is a family of GoL-like games that arise from adjusting these parameters.

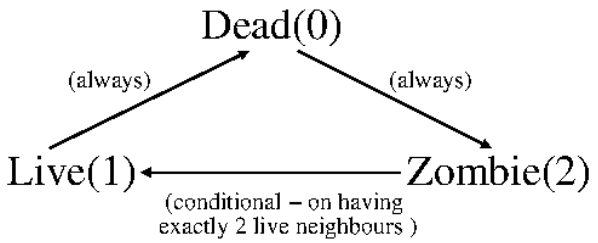


Figure 2: Cyclic transition diagram between states in the Game of Death.

The Game of Death model however is different because of the cyclic relationship between the three species. The third "zombie" state is introduced with the following rule set used with cells that are dead(0),live(1) or zombie(2):

- any dead cell becomes a zombie $0 \rightarrow 2$
- any zombie becomes live iff it has exactly two live neighbours $2 \rightarrow 1$
- any live cell dies $1 \rightarrow 0$

Figure 2 shows the cyclic relationship between the three states. For technical reasons it is easier to insert the zombie state as "2" so that the simulation software can continue to refer to dead cells as "0" and live ones as "1." We coded these simulations using a custom generated C++ program and fast library routines for hyper-dimensional rectilinear system manipulation. This meant we could readily experiment with different neighbourhood logics as well as being able to easily switch from two to three dimensions using the same simulation program.

Figure 3 shows the time progression of the Game of Death along with snapshots emphasising the location of the dead cells and also typical configurations of the Game of Death at similar times following random initialisation.

The Game of Death shows a marked dip in the density of live and dead cells following a uniform random initialisation. The spaceships and rakes and other reported patterns [21] propagate as alternating patterns of live and dead cells in a background of zombie cells. The zombie cell density rises dramatically after initialisation. This is in contrast to the Game of Life where live cells are the minority against a majority background of dead or vacant cells.

3 Simulation Results

In the work reported here we simulate Life and Death on a (periodic) square and cubic mesh, using the Moore neighbourhood of 8 cells in two dimensions and 26 neighbours in three dimensions. The periodicity limits the size of very

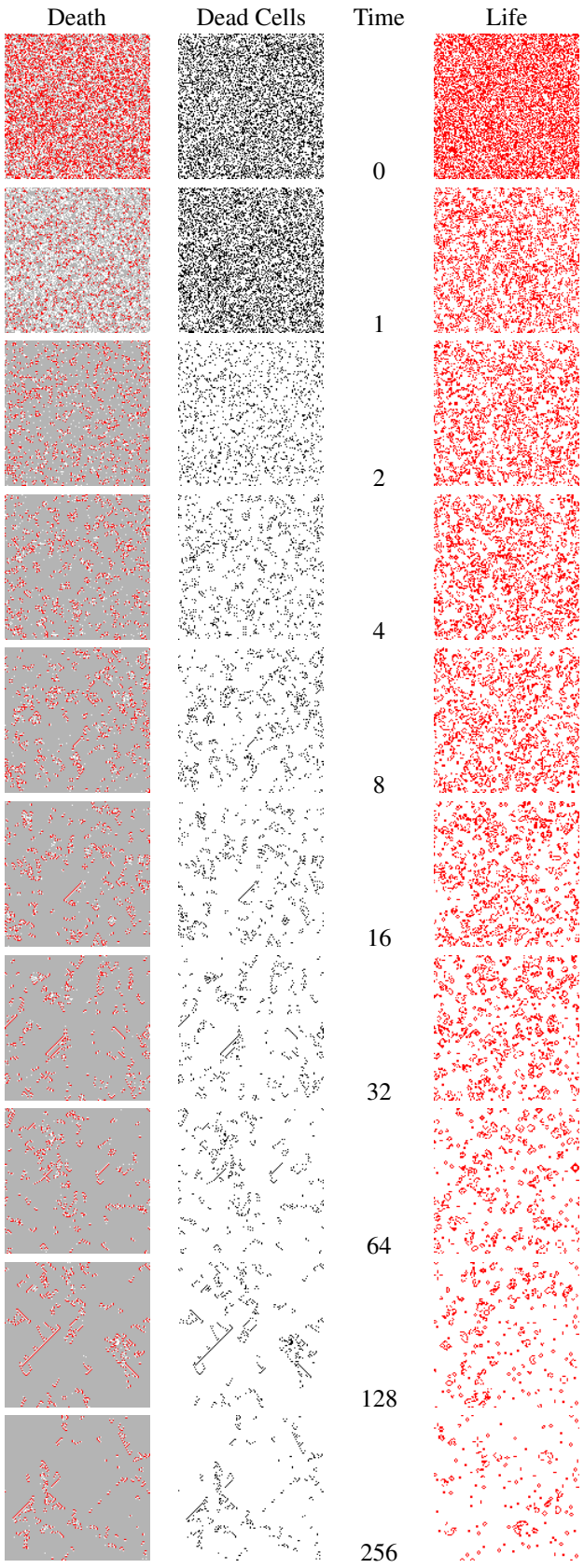


Figure 3: The Game of Death (left); the dead cells in it; and the Game of Life (right), all played on a 128×128 cell mesh. Red = Live, White = Dead, Grey=Zombie.

large individual structures, but is useful for the statistical analysis we report in this paper as we can ignore special rules required for boundary conditions – all cells experience the same neighbourhood structure. The systems are initialised randomly with an equal and random mix of all Q species for the relevant model. $Q = 2$ for the Game of Life, $Q = 3$ for the Game of Death.

In addition to being able to count and track the average fraction of live or dead cells in the model configuration, we can also investigate a simple count of the like-like species bonds or nearest neighbour relationships, defined as:

$$N_{\text{same}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i = s_j \qquad (1)$$

where N is the number of automaton cells or sites in the d-dimensional lattice and each $s_i = 0, 1, 2$ for the $Q = 3$-state Game of Death.

Similarly $N_{\text{diff}} = 1 - N_{\text{same}}$ is the fraction of possible bonds in the system that are different. We also define a correlation function:

$$N_{\text{corr}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i^{0,1}(t) = s_i^{0,1}(t-1) \qquad (2)$$

It is convenient to define this so that both live(1) and dead(0) cells correlate together since the configuration snapshots suggest live and dead behave in a similar manner to one another within a background "sea of zombies."

Since our Game of Death rules involve a cycle in the species number, it is useful to define a selection metric that correlates when the rule applied was in a particular direction around the cycle.

$$N_{\text{sel}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i = x; s_j = x - 1, \forall x \qquad (3)$$

This is in accordance with the cycle direction shown in Figure 2 although for a 3-cycle it can be applied in reverse with the same result. We can also define a rule neutral fraction $N_{\text{neut}} = 1 - N_{\text{sel}}$.

3.1 Two Dimensional Models

We can examine these metrics for the Game of Death system in more detail for two dimensional model systems.

Figure 4 shows the metrics for the Game of Death plotted on a log-log scale. The early straight line regions thus suggest power-law behaviours. The correlation and fraction of live, vacant(dead), like-like(same) and different cell metrics are shown along with two others for selection and neutral interactions. These latter two measure the fraction of activity or change that takes place in a model [32] and

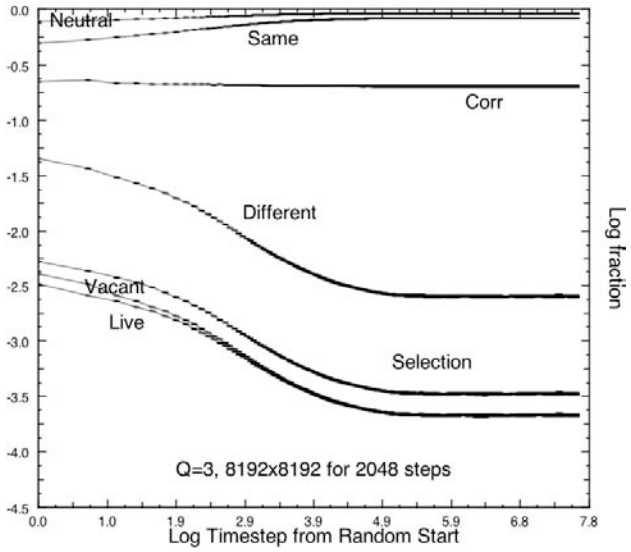


Figure 4: Fractional variation of the composition of the 3-State Game of Death over time following a random uniform start.

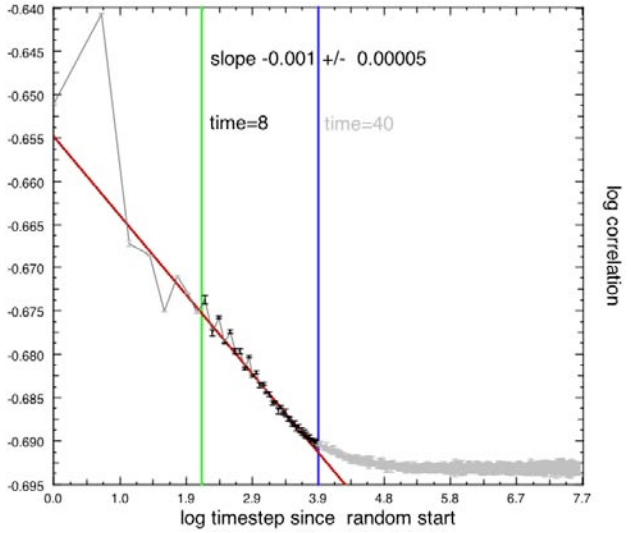


Figure 5: Single step time correlation function for Game of Death model on log-log scale.

are seen to be closely related to the vacant (dead) cell population.

Figure 5 shows the correlation function defined in equation 2 on an enhanced plot scale. The straight line regime shown on the log-log scale indicates a power-law behaviour with a small exponent - that we speculate may be related to (reciprocal) system edge length $L = 1024^2$. The straight line region shows the measured data oscillating at alternate time steps around a trend. This pattern persists despite averaging over 100 independently initialised runs and is thus intimately related to the synchronous CA update mechanism where all cells are updated at once, in synchrony.

The configuration snapshots of the Game of Death model suggest that activity is driven by the combined presence of live and dead cells against a background of zombie states. Although the snapshots shown were initialised with equal proportions of live, dead and zombie states, it is clear the zombie states rapidly dominate. Furthermore this symmetry between "live" and "dead" suggests that one of these two could be used as a variable parameter. The fraction of dead or "vacant" cells in the initial configuration is thus used as a parameter in the data shown in the parametric surface plots shown below. The experiments are averaged over 100 independent runs using the stated $pVacant$ fraction of dead cells in the initial configuration, with the zombie and live states receiving $\frac{1-p}{2}N$ initial cells each. The metric fractions are shown over power-of-two times.

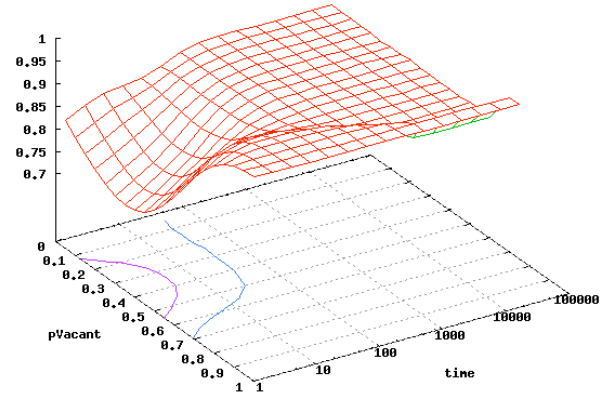


Figure 6: Average fraction of like-like cells in 2D Death model, shown at power-of-two times $0, 1, 2, 4, 8, 16, ..., 16384$, as the initial number of dead or "vacant" sites is varied

Figure 6 shows the fraction of like-like cell neighbour bonds encountered in the model system with time, with the vacant/dead cell initialisation parameter scan shown. If the system is initialised entirely dead, it stays that way but if there are no dead cells, just live and zombie states, then the fraction of dead cells does reach a steady dynamic equilibrium value. There is an interesting saddle-point in the surface near a dead initialisation fraction of $p \approx 0.3$. It is not yet clear if this is really an approximation of $\frac{1}{3}$ or a less trivial phase transitional value. The long-term behaviour is almost independent of the dead initialisation fraction. Providing there are some appreciable fraction of dead cells to interact with live cells, the system eventually reaches an unvarying fraction of like-like neighbouring cells.

Figure 7 shows a simple population fraction of the live cells. This parametric surface also shows long term insensitivity to the dead cell initialisation fraction, as long as it is less than around ≈ 0.7, above which the system stays uniformly dead. The same saddle-point is shown at around

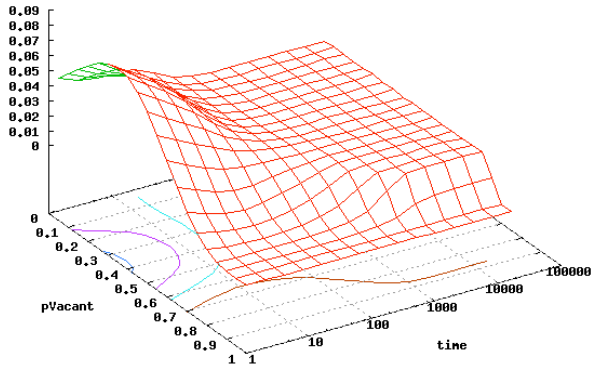


Figure 7: Average fraction of live cells in 2D Death model, shown at power-of-two times $0, 1, 2, 4, 8, 16, ..., 16384$, as the initial number of dead or "vacant" sites is varied

0.3 and is again seen to affect only the early time regime of the model. After a number of time steps that is comparable with the system length traces of the initialisation fraction are seen to be largely dissipated.

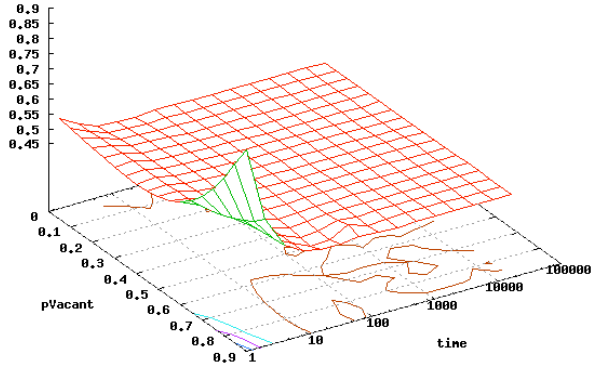


Figure 8: Average 1 time step correlation function in 2D Death model, where neighbouring cells are the same, shown at power-of-two times $0, 1, 2, 4, 8, 16, ..., 16384$, as the initial number of dead or "vacant" sites is varied

Figure 8 shows the correlation function as defined in equation 2. The surface correlates live and dead cells together without distinction, so it neglects zombie-zombie cell effects which would otherwise completely dominate. The correlation rapidly washes out any fluctuations after a time comparable to that necessary for information to have propagated across the entire model. The early regime correlation surface is relatively smooth but does show a turn-up for configurations where too many of the initial cells are dead.

3.2 Three Dimensional Models

It is interesting to examine the Game of Death model in three dimensions to try to determine if the early time regime phenomena carry over or are unique to the two dimensional case.

The Game of Life with the stated cutoffs does not lead to interesting behaviour in 3D on Moore neighbourhood. All cells die out after a single time step and it is necessary to adjust the totality thresholds r_i to make a sustainable three dimensional Game of Life. However the unaltered Game of Death *does* show interesting transients followed by a complex dynamical equilibrium as shown in figure 9.

Figure 9 shows that a uniform random initial configuration does undergo a dramatic change at the first few time steps with the system nearly dying out, leaving just a few viable combinations - by statistical chance. These viable combinations rapidly grow to fill the space and then "jostle" with one another creating ongoing interference patterns a nd surface waves that maintain a dynamical equilibrium in overall population density.

We observe that the system must be big enough for these viable combinations to arise statistically. We speculate that it should be possible to compute the statistical likelihood of viable combinations arising by chance by a systematic study of different sized model systems. As preliminary bounds on this size and probability, we found that a 40^3 system would nearly always recover viable patterns, whereas a 16^3 system typically would not.

4 Discussion

The Game of Death system shows that "the right" combinations of live and dead cells located nearby in a background of zombie cells can grow and self-perpetuate, and even when these independent fluctuations encounter one another they still manage to co-exist in the long term, creating three dimensional waves and interacting patterns. Overall this gives rise to a system of dynamic equilibrium. A great deal of cell activity occurs, but if large enough, the overall populations reach steady sustainable values.

This behaviour is similar to the overall sustainable population averages found in Lotka-Volterra and predator-prey animat models [16]. In those models however there are typically much slower periodic boom-bust envelops superposed on the flat averages.

The cyclic relation between the 3-states of the Game of Death goes some way to explaining why zombie states dominate. The rules as shown in Figure 2 show that while live cells *always* become zombies and dead cells *always* die, zombies tend to become stuck as zombies since there

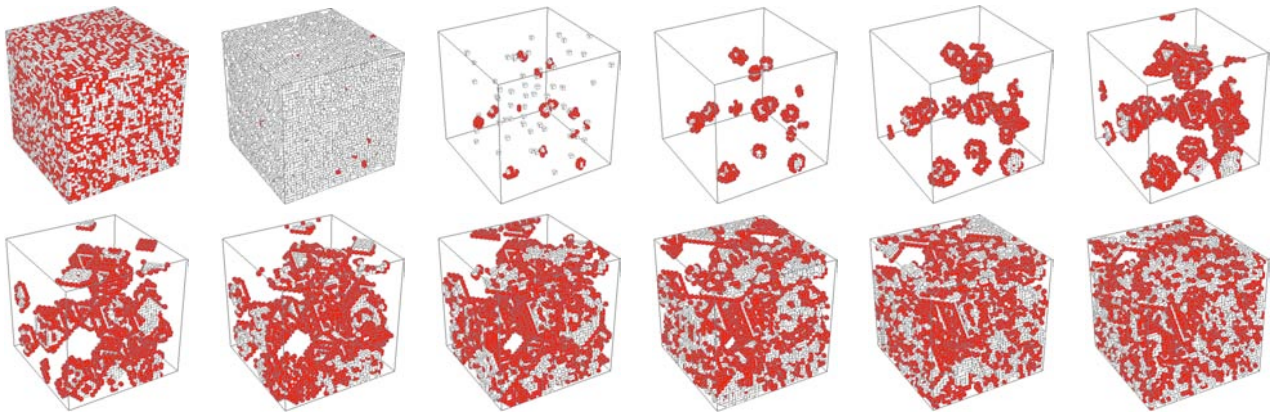


Figure 9: The Game of Death on a $40 \times 40 \times 40$ mesh, zombie states left transparent, at times $0, 1, 2, 3, ..., 11$.

is a special condition (having exactly 2 live neighbours) for a cell to break out of the zombie state. This condition is unlikely to happen just by chance with probability $2/8$ or $2/26$ in two or three dimensions respectively. Patterns in the CA subsist or reproduce however precisely because the neighbour hoods are not random - after the transient temporal stage is over. Once the dynamical equilibrium phase has begun the probability of "the right combination" of live neighbour cells is high enough to maintain the overall population densities.

5 Conclusions

We have analysed Silverman's Game of Death on two-dimensional and three-dimensional spatial periodic meshes and identified both qualitatively different behaviours from the Game of Life, as well as employing a number of signature metrics that quantify these distinctions. We have compared the typical configurations that arise following a random initialisation and have found that the Game of Death model exhibits two distinct temporal regimes. The initial transient stage involves a massive near-extinction of statistically random patterns that are not viable. This stage is followed by the re-population of sustainable patterns from the statistical remnants of the initial near-extinction. This manifests itself a large dip in the density of live and dead cells which then can stabilise to steady state values characteristic of a dynamic equilibrium.

Providing the system is large enough there are typically enough self-sustaining or reproducing patterns present to repopulate - although for systems that are too small for these to occur statistically, then the whole system dies out. This behaviour is in contrast to the Game of Life behaviour where a minority of live cells requires just the right neighbourhood combinations to survive or reproduce against a majority background of dead cells.

We experimented with a deliberate adjustment to the number of dead cells in the model initialisation and parametric scans show that providing there are some, then the exact details of the initialisation are washed out in time and thus the model does not retain much memory of its state beyond the time for information to propagate across the spatial edge length.

We note that for the Game of Life the Moore neighbourhood works in two dimensions but not in three unless we adjust the r_i neighbour count cutoffs. Without such an adjustment all Game of Life cells die out immediately. In contrast however the Game of Death model works unaltered in 3D with Moore neighbourhood and live and dead cells persist albeit at low densities compared to the zombie cells.

Unlike much of the research reported on Game of Life like models, we have not focused on specific structures but there is scope to do so for the Game of Death and in particular to look at three dimensional features and viable patterns. A detailed statistical analysis of the motifs and other common patterns in the Game of death would also be worthwhile. There is also scope for further work adjusting the cutoffs $r_1, r_2, ...$ in Game of Life in 3D and also to adjust the two-live neighbour zombie promotion rule in the Game of Death.

The transient statistical behaviour in the early stages of these models has some implications for viable biodiversities of many specied complex systems. Only quite particular combinations of the many possible patterns that are formed by chance initially can actually survive and reproduce. Computational experiments with a microscopically simple sort of CA model such as the Game of Death could feasibly map out this statistical model space and link it to reductionist theory, whereas this approach continues to prove difficult for real world biological systems [33–35].

References

[1] Gardner, M.: Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life". Scientific American **223** (1970) 120–123

[2] Ganguly, N., Sikdar, B.K., Deutsch, A., Canright, G., Chaudhuri, P.P.: A survey on cellular automata. Technical report, Dresden University of Technology (2003)

[3] Wolfram, S.: Cellular Automata as models of complexity. Nature **311** (1984) 419–424

[4] Freire, J.G., Owen J, B., Gallas, J.A.C.: Exact quantification of the complexity of spacewise pattern growth in cellular automata. J. Phys. A. Math. & Theor. **42** (2009) 395003

[5] Wolfram, S.: Universality and complexity in cellular automata. Physica D **10** (1985) 1–35

[6] Eppstein, D.: Growth and Decay in Life-Like Cellular Automata. In: Game of Life Cellular Automata. Springer (2010) 71–98

[7] Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev.Mod.Phys **55** (1983) 601–644

[8] Lynch, J.F.: On the threshold of chaos in random boolean cellular automata. Random Structures & Algorithms **6** (1995) 239–260

[9] Kaneko, K., Akutsu, Y.: Phase Transitions in two-dimensional stochastic cellular automata. J.Phys.A. Letters **19** (1986) 69–75

[10] Boghosian, B.M., IV, W.T., Rothman, D.H.: A cellular automata simulation of two-phase flow on the CM-2 Connection Machine computer. Thinking Machines (1988)

[11] Rothman, D.H., Keller, J.M.: Immiscible cellular-automaton fields. J.Stat.Phys. **52** (1988) 1119–1127

[12] Fort, H., Viola, S.: Spatial patterns and scale freedom in prisoner's dilemma cellular automata with pavlovian strategies. Journal of Statistical Mechanics: Theory and Experiment (2005) P01010

[13] Wolfram, S.: Theory and Applications of Cellular Automata. World Scientific (1986)

[14] Hawick, K.A., Scogings, C.J.: A minimal spatial cellular automata for hierarchical predator-prey simulation of food chains. In: International Conference on Scientific Computing (CSC'10), Las Vegas, USA, WorldComp (2010) 75–80

[15] Adamatzky, A., Komosinski, M., eds.: Artificial Life Models in Software. Springer (2005) ISBN: 1852339454.

[16] Hawick, K.A., James, H.A., Scogings, C.J.: Manual and semi-automated classification in a microscopic artificial life model. In: Proc. Int. Conf. on Computational Intelligence (CI'05), Calgary, Canada. (2005) 135–140

[17] Hawick, K.A., James, H.A., Scogings, C.J.: A zoology of emergent patterns in a predator-prey simulation model. In Nyongesa, H., ed.: Proceedings of the Sixth IASTED International Conference on Modelling, Simulation, and Optimization, Gabarone, Botswana (2006) 84–89

[18] Hawick, K.A., Scogings, C.J.: Spatial pattern growth and emergent animat segregation. Web Intelligence and Agent Systems **8** (2010) 165–179

[19] Ginn, T.R., Loge, F.J., Scheibe, T.D.: Explaining noise as environmental variations in population dynamics. Computing in Science & Engineering **March** (2007) 40–49

[20] Adamatzky, A., ed.: Game of Life Cellular Automata. Number ISBN 978-1-84996-216-2. Springer (2010)

[21] Resnick, M., Silverman, B.: Exploring emergence: The brain rules. `http://llk.media.mit.edu/projects/emergence/mutants.html` (1996) MIT Media, Laboratory, Lifelong Kindergarten Group.

[22] Evans, M.S.: Cellular Automata - Brian's Brain. `(http://www.msevans.com/automata/briansbrain.html` (2002) Department of Neurology, Southern Illinois University School of Medicine.

[23] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc. (2002) ISBN 1-57955-008-8.

[24] Stauffer, D.: Computer simulations of cellular automata. J.Phys.A:Math. Gen **24** (1991) 909–927

[25] Frobose, K.: Finite-size effects in a cellular automaton for diffusion. J.Stat.Phys. **55** (1989) 1285–1292

[26] Mattos, T.G., Moreira, J.G., Atman, A.P.F.: A discrete method to study stochastic growth equations: a cellular automata perspective. J. Phys. A: Math. Theor. **40** (2007) 13245–13256

[27] Capcarrere, M.S.: Evolution of asynchronous cellular automata. In: Parallel Problem Solving from Nature Proc. PPSN VII. Volume 2439. (2002)

[28] Yey, W.C., Lin, Y.C., Chung, Y.Y.: Performance analysis of cellular automata monte carlo simulation for estimating network reliability. Expert Systems with Applications **36** (2009) 3537–3544

[29] Nehaniv, C.L.: Self-reproduction in asynchronous cellular automata. In: Proc. Evolvable Hardware 2002, NASA/-DoD. (2002)

[30] Hastings, M.B., Reichhardt, C.J.O., Reichhardt, C.: Ratchet cellular automata. Phys. Rev. Lett. **90** (2003) 247004

[31] de Oliveira, G.M.B., Siqueira, S.R.C.: Parameter characterization of two-dimensional cellular automata rule space. Physica D: Nonlinear Phenomena **217** (2006) 1–6

[32] Hawick, K.: Cycles, diversity and competition in rock-paper-scissors-lizard-spock spatial game simulations. Technical Report CSTN-129, Computer Science, Massey University, Auckland, New Zealand (2011) Submitted to WorldComp ICAI, Las Vegas 2011.

[33] Kauffman, S.: Homeostasis and differentiation in random genetic control networks. Nature **224** (1969) 177–178

[34] May, R.M.: Will a large complex system be stable? Nature **238** (1972) 413–414

[35] Fenyo, D., ed.: Computational Biology. Number ISBN 978-160-7618-416. Springer, NY, USA. (2010)

The development model of Munster town in multi-agent system

M. Hadi Kaboli[a,b], Jean-Luc Mercier[b], Benjamin Soulet[c]
[a]Islamic Azad University, Damavand branch, Jilard, Damavand, Iran,
[b]Université de Strasbourg, 3 Rue de l'Argonne, 67000 Strasbourg, France,
[c]Communauté Urbaine de Strasbourg, 1, parc de l'Etoile - 67076 Strasbourg Cedex, France.

Abstract- *In this study, Munster town, located in the east of France is modeled in a multi-agent system and the urban growth has been studied in the referred year 2002 from the initial state 1979. For studying the urban growth, aerial photos and geographical data have been exploited as the primary materials to make the model in two different time sections.*[1]
The verisimilitude of simulation and real growth has been tested by observation and statistical comparison.
With controlling some agents, the variety of states is modeled for the year 2025.
By measuring the forces that feed the growth and the natural and artificial obstacles that limit it, this simulation provides the urban planners and deciders with some tool for controlling the process. The different modes of simulation can help to define strategies for future.
Keywords: *Multi agent system, Cellular Automata, urban growth, complex system.*

1. Introduction

The urban growth has been one of the major items for urban planners and geographers. In recent years, urban modeling showed the attempt to shift from "classic" models to "evolutionary" models (as evolutionism appeared to be the symbol of complex systems). [2]

- Cellular automata, as a tool for urban modelling, are gaining a big popularity, because they imply inputs and outputs in a qualitative and cartographic form, which is, for urban planners, more usual than numerical representations. In addition, this different form of representation allows an easy interface with GIS, which show an increasing diffusion;
- The working mechanism of cellular automata is relatively simple, because it consists mainly in the transition rules between one state and another of a cell, which depend on the state of the cell itself and of its neighbouring (appropriately defined). This simplicity makes operatively easier the connection with the "learning system".[3]

These models are criticized for that instead of showing the complexity which is resulted by activating several factors simultaneously, the introduction of new ingredients into the simulation soup may have unpredictable and unforeseen consequences for the flavor of the resulting model. [4]
Nonetheless the configuration of the factors, giving weight to them, and combining several factors with their personal weights, make the models good toolboxes that can take into consideration the results of increases and decreases in the number of factors and their weights.
It is possible to change some of the factors that have impact on the speed, orientation and magnitude of growth, and by this, the changes in horizontal growth in borders of city or vertical growth in varied parts of city is resulted.

This work is realized on Munster town, located in the middle of regional natural park Ballons des Vosges. A MAS (Multi Agent System) model is applied in this little mountain town where the heterogeneous space constrains it. Munster town is surrounded by mountains that impede the easy construction and the regulations do not let the construction on the slopes. However there has been a destroy of 3% of urban and 10% of Natural park environment as the result of construction.
In this study, the dynamic aspects of urban growth are analyzed, understood and anticipated and therefore a simple and powerful tool for controlling the urbanization policies in a space constrained by nature (slope, forest, biotope, humid zones, infrastructure, ...) and human activities (agriculture, tourism, ...) is Prepared.

2. The model

2.1 Making model

From different sources like maps and administrative documents, aerial photographies (IGN, Institut Géographique National), Orthophotographic database, and topography database (IGN) the data are issued. The geometry is extracted from town data and geometric data (punctual and surficial).
Totally 15 different files of data are combined to construct the model.
With Arcgis9.1 software which works on the digital model of terrain and create and analyze the grids, the data are treated to adapt to matrices (177, 97). The codes are written in the open source software "Netlogo" which is a 3D world. This tool lets to create a regular mesh of (25m) square and transfer the values from the entities of one layer. The pixels are all the same size and can't simulate the multipixel buildings (factories, supermarkets…)

2.2 Multi agent system

A multi-agent system (MAS) is the set of agents situated in a certain environment and interact according to a certain organization.
What is simulated is the Munster town, but the space goes more than the town limits. The environment is discretized in 25 meters (x,y) and 10 meters (z) units.

The first inputs of the town in 1979 were manually reconstituted by degrading 2002, because there was lack of precise information. The non-constructible zones (cemetery, stadium, humid zones, and biotope) were delimited in 2002.

The system is based on different factors which permits or not the construction in the town.

The adjustable parameters are:
The duration of simulation (0-50 years)
Number of buildings to be constructed in simulation period (0-1000)
Maximum permitted slope (0-30°)
Number of construction permitted by pixels (1-6)
Empty pixels around the constructions (1-16)
Construction in the center of town (yes/no)
Construction in the forest (yes/no)

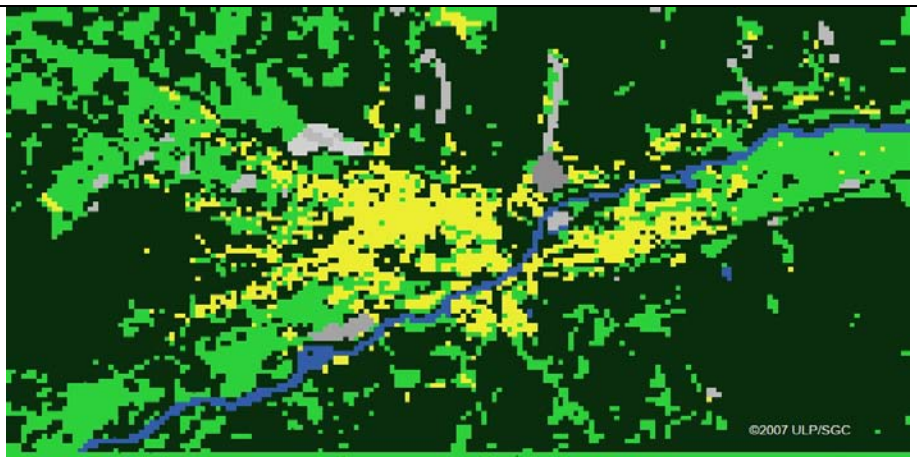


Fig.1 Munster town 1979

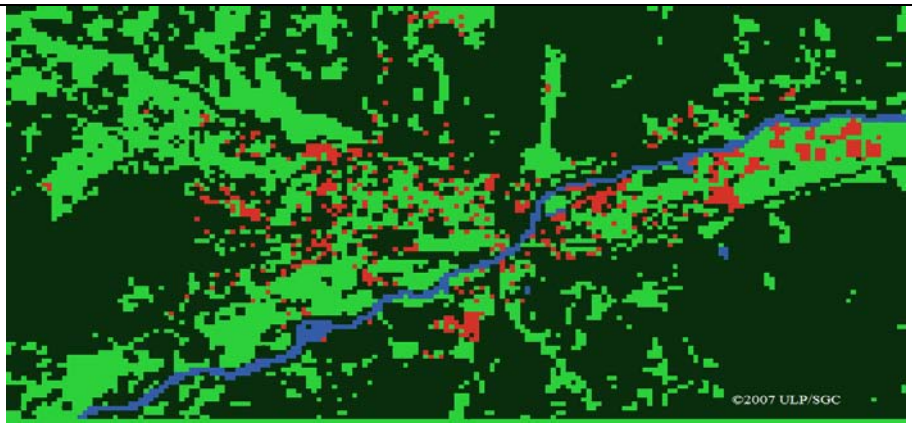


Fig.2 Munster town Evolution

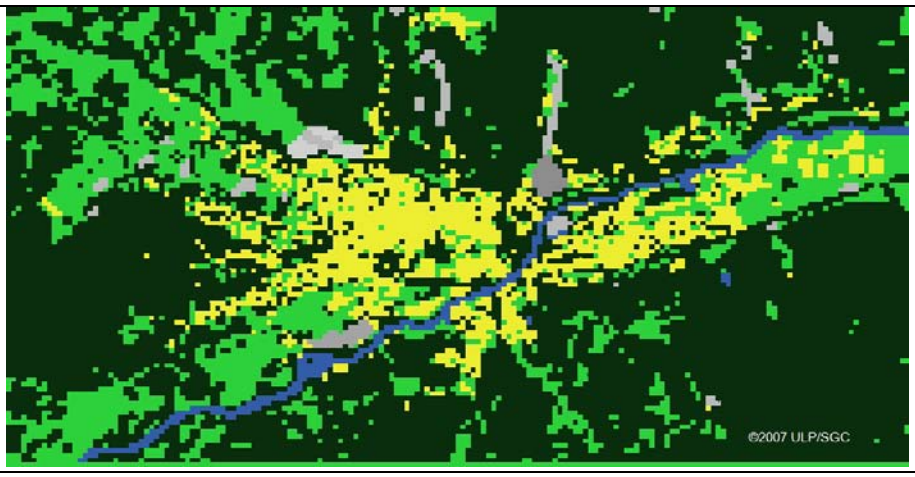
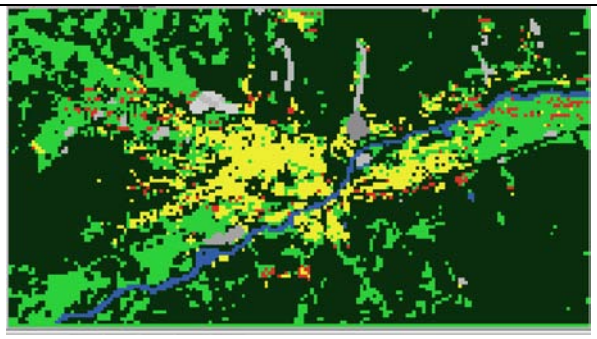


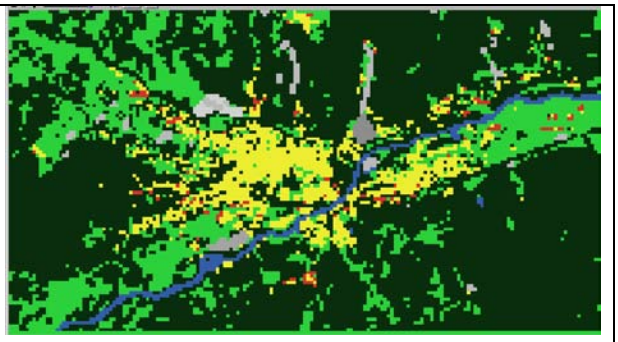
Fig.3 Munster town 2002

1979-2002 simulation



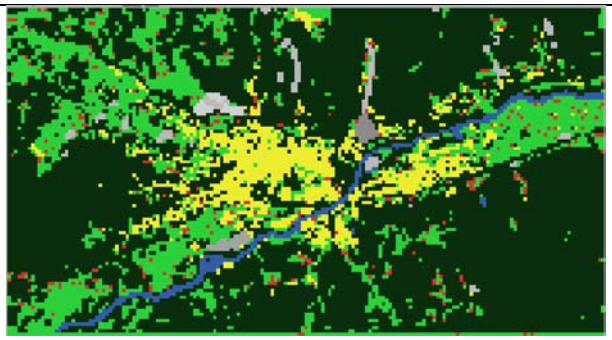
Simulation period	23	neighbors	3
Building wanted	394	Construction in center	no
slope	5°	Construction in forest	no
densification	5	**[F1=0.443 F2=0.678]**	

Fig.4 Building densification 1 periphery



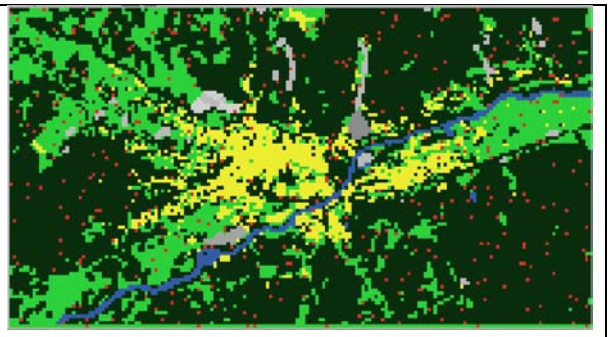
Simulation period	23	neighbors	1
Building wanted	394	Construction in center	yes
slope	5°	Construction in forest	no
densification	6	**[F1=-0.025 F2=0.639]**	

Fig.5 Building densification 1 Maximal



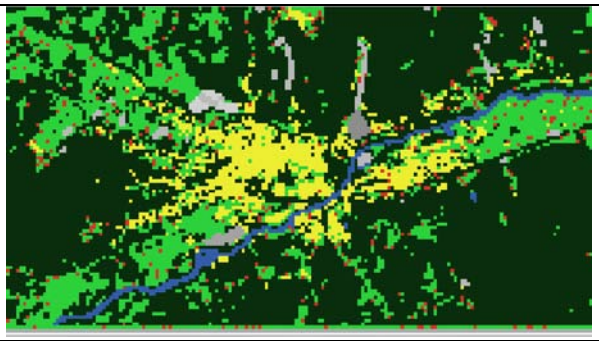
Simulation period	23	neighbors	8
Building wanted	394	Construction in center	no
slope	12°	Construction in forest	no
densification	2	**[F1=0.690 F2=0.714]**	

Fig.6 Building dispersion 1



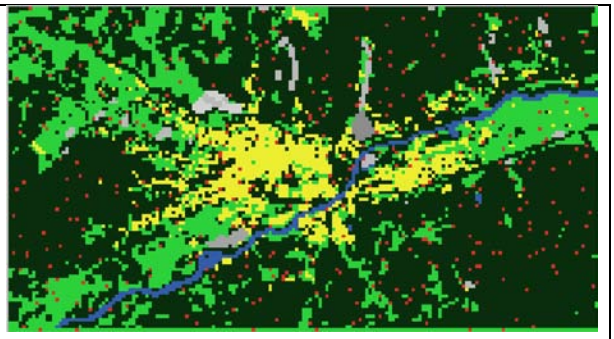
Simulation period	23	neighbors	16
Building wanted	394	Construction in center	no
slope	30°	Construction in forest	yes
densification	1	**[F1=0.780 F2=0.731]**	

Fig.7 Building dispersion 2



Simulation period	23	neighbors	8
Building wanted	394	Construction in center	no
slope	30°	Construction in forest	no
densification	2	**[F1=0.726 F2=0.720]**	

Fig.8 Construction on the slopes



Simulation period	23	neighbors	8
Building wanted	394	Construction in center	no
slope	30°	Construction in forest	yes
densification	2	**[F1=0.777 F2=0.732]**	

Fig.9 Construction on the slopes and in the forest

2.3 The validation and the test

For measuring the quality and the verisimilitude of simulation, some tool should be applied for comparing the results of simulation with observation. For this aim, two approaches were applied to compare the two real and simulated states. "Classic Statistical" and "spatial analyze" approach.

In statistic approach, the quality of simulation is measured by comparing the exact data.

Reference Year: 2002→Matrix $[A_{2002}]$.

Simulation results→Matrix $[B_{sim}]$.

The data are all in matrix form (177, 97).

The comparison is made by considering two criteria:

The quadratic mean error criteria:

$$\left[\frac{\sum_{i=1}^{i=97}\sum_{j=1}^{j=177}(\boldsymbol{A_{2002}}-\boldsymbol{B_{sim}})^2}{\sum_{i=1}^{i=97}\sum_{j=1}^{j=177}(\boldsymbol{A_{2002}}-<A_{sim2002}>)^2}\right] \quad \textbf{(1)}$$

And absolute criteria:

$$\left[\frac{\sum_{i=1}^{i=97}\sum_{j=1}^{j=177}|\boldsymbol{A_{2002}}-\boldsymbol{B_{sim}}|}{\sum_{i=1}^{i=97}\sum_{j=1}^{j=177}|\boldsymbol{A_{2002}}-\langle\boldsymbol{A_{2002}}\rangle|}\right] \quad \textbf{(2)}$$

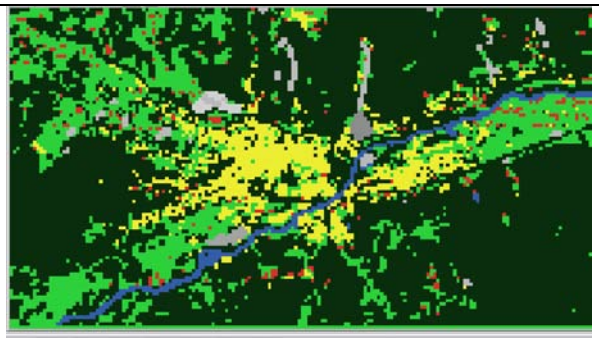
These two criteria vary from $[-\infty, 1]$; The more they are close to 1, the more the two distribution are similar. One negative value indicates that the model best explain spatialised constant value.

Three evolution scenarios were used (densification, dispersion, urbanization of slopes), everyone with two intensities. The application of these schemas in the period 1979-2002 showed that development of Munster is close to "densification 2"

Prevision: The model was used in the period 2002-2025 with the same schema of evolution.

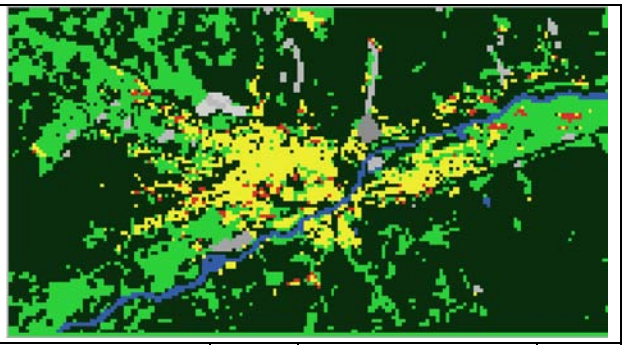
The second approach can be applied a little more empiric, it consists the construction of indices and comparing the two. It is done in 3 spatial scales: the landscape, the building type, the pixels.

2002-2025 simulation



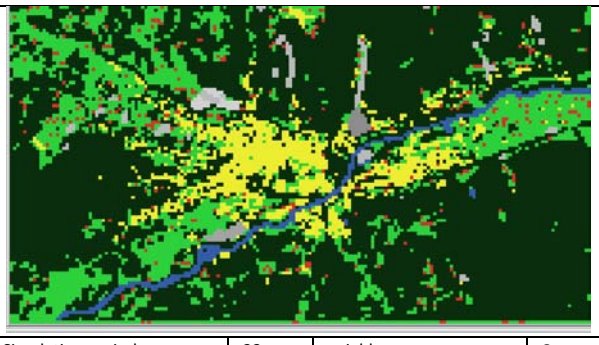
Simulation period	23	neighbors	3
Building wanted	394	Construction in center	no
slope	5°	Construction in forest	no
densification	5	**[F1=0.479 F2=0.724]**	

Fig.10 Building densification 1 periphery



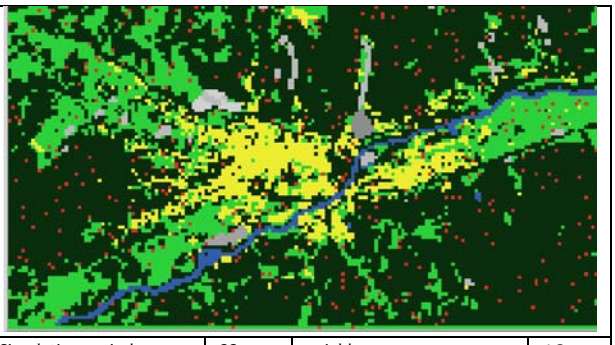
Simulation period	23	neighbors	1
Building wanted	394	Construction in center	yes
slope	5°	Construction in forest	no
densification	6	**[F1=0.145 F2=0.701]**	

Fig.11 Building densification 2 periphery



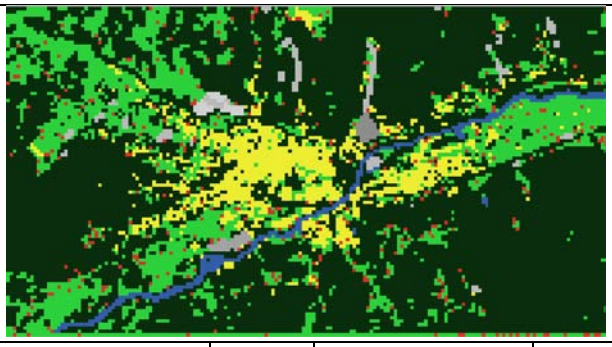
Simulation period	23	neighbors	8
Building wanted	394	Construction in center	no
slope	12°	Construction in forest	no
densification	2	**[F1=0.705 F2=0.750]**	

Fig.12 Building dispersion 1



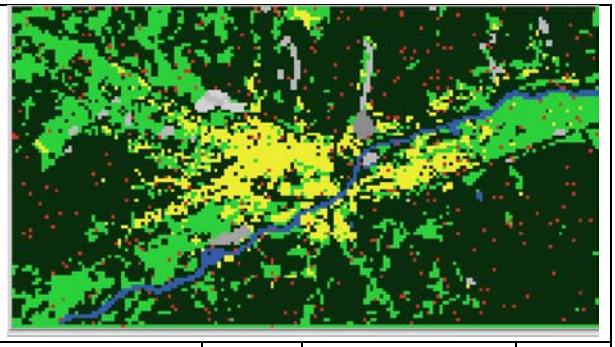
Simulation period	23	neighbors	16
Building wanted	394	Construction in center	no
slope	30°	Construction in forest	yes
densification	1	**[F1=0.793 F2=0.764]**	

Fig.13 Building dispersion 2



Simulation period	23	neighbors	8
Building wanted	394	Construction in center	no
slope	30°	Construction in forest	no
densification	2	**[F1=0.752 F2=0.759]**	

Fig.14 Construction on the slopes



Simulation period	23	neighbors	8
Building wanted	394	Construction in center	no
slope	30°	Construction in forest	yes
densification	2	**[F1=0.786 F2=0.764]**	

Fig.15 Construction on the slopes and in the forest

3. Conclusion

This study has realized a temporal and spatial modelization. The support of the study is the evolution of urbanization in Munster at the center of PNRBV (Parc Naturel Regional des Ballons des Vosges).

The model is developed by discretized simulation in space and in time. The model in 2D and 3D version can describe the urban sprawl and densification, based on application of simple anthropic and natural factors.

The original data are: the topography (x,y,z, slope, orientation), the plant (forest, prairie), the building (5 types), and the non constructible zones (4 types). The reference year is 2002. The year 1979 was manually reconstructed from maps and orthophotos.

Six factors permit the hypothesis construction on urban evolution: Duration of simulation, number of houses, density of buildings, contiguity of neighborhood, construction in the forest and/or in the center of town.

In future, some other tests will compare the simulation and actual state, some other factors like the distance from the city center and the impact of neighbor cells on the referred cell will be taken to account. As the center of city has played an important role in urban growth[5], and on the other hand by changing the transportation facility and the ease of access to the central parts[6], and because of the emergence of new centers in cities[7], the distance from the center and the impact of neighbor cells will be the aim of future study to make the models more dynamic.

References

[1] Mercier J.L., Breting J., Soulet B., *Simulation de l'étalement et de la densification du bâti dans la commune de Munster : Utilisation d'un SMA, 2D et 3D*, Université de Strasbourg, 2007.

[2] Antonio Colonna, Vittorio Di Stefano, Silvana Lombardo, Lorenzo Papini and Giovanni A. Rabino, *Learning Cellular Automata: Modelling Urban Modelling*, Geocomputation conference, 1998, University of Bristol.

[3] Antonio Colonna, Vittorio Di Stefano, Silvana Lombardo, Lorenzo Papini and Giovanni A. Rabino, *Learning Cellular Automata: Modelling Urban Modelling*, Geocomputation conference, 1998, University of Bristol.

[4] Torrens, P. M. and D. O'Sullivan, *Cities, cells and complexity: developing a research agenda for urban geocomputation*. Presented at *Geocomputation 2000*, University of Greenwich, London, 25 - 28 August.

[5] Burgess E.W. (1924)"*The growth of the city: an introduction to a research project*" Publications of the American Sociological Society, 18:85-97

[6] Hoyt H (1939): "*The structure and growth of residential neighborhoods in American cities*" Washington DC; Federal Housing Administration

[7] Harris C D and Ullman E L (1945), "*The nature of cities*" Annals of the American Academy of Political and Social Science 242: 7-17

Post modern comfort as a factor in gentrification of city, modeled in Cellular Automata

Leila Zare[a], M. Hadi Kaboli[b],
[a]Department of Art and Architecture, Science and Research Branch, Islamic Azad University, Tehran, Iran.
[b]Université de Strasbourg, 3 Rue de l'Argonne, 67000 Strasbourg, France,

Abstract- *The aim of this study is to investigate the discrete factors of gentrification modeled by cellular Automata. The gentrification is the arriving of middle and high social classes in the inner-city areas. It corresponds to the residential mobility of solvent households, attracted by residential comfort and the central retailing equipment. The proximity, diversity and quality of this equipment become criteria of the residential choice, as important as the criterion of the residential comfort. The cellular automata in this study, models the constituents of post modern comfort in residence, its environment and the centrality which altogether and simultaneously make the city inhabitants aspiration. During the spatiotemporal change, the aspiration in central parts of city is studied.*

Keyword: Gentrification; Cellular Automata; Complex System, postmodern comfort.

1. Introduction

The social world is getting more complex, more people on various levels participate in decisions through democratization and decentralization. Globalization and new technologies increase their interrelations and people become more diverse and through an increase in wealth become more individual. As is the notion of decentralized decision making, the physics of far-from-equilibrium structures is important. Processes that lead to surprising events, to emergent structures not directly obvious from the elements of their process but hidden within their mechanism, new forms of geometry associated with fractal patterns, and chaotic dynamics, all are combining to provide theories that are applicable to highly complex systems such as cities.[1]

The complex system of cities are comprised of some more simple systems which are working at the same time and also there is always uncertainty about the outcome of the process of changes that originate from the bottom-up order.[2]

The imbalance in interurban of Strasbourg exemplifies some state in macro scale which is the result of some changing in micro scale. The mobility of residents and the changing in socio-spatial disparity in the city leads to the socio-spatial phenomenon "gentrification".

For explanation of an observed socio-spatial phenomenon, there has been a tendency to grow it rather than explain it. Artificial society modeling allows us to "grow" social structures *in silico* demonstrating that certain sets of micro specification are sufficient to generate the macro phenomena of interest. For matching between the true, observed structures and the generative models, statistics can be used for testing the results.[3]

Cellular automata are applied for modeling an urban system that enacts several factors making the mobility of residents in Strasbourg. Some factors that make them settle or stay in their residences and some that make them move into other parts of the city.

In this study, two main categories of comfort are taken to study the satisfaction of people. The comfort factors are important for deciding to stay or move.

The two comfort categories are modern comfort, that are measurable standard factors that enhance the residence and neighborhood, and the second, discrete comfort, that gives aspiration to the residents to continue to live in their residence

and consequently in their neighborhood. The latter are some facilities like centrality or in another word having easy access to urban equipments. [4]

The two main categories are modeled as the result of statistical outputs of a study in Strasbourg in three time sections: 1975, 1982, and 1992.
All of the factors enhance the quality of life in a neighborhood, however at the same time they elevate the cost of living in these areas.
In the last study, the two theories of gentrification were combined to see the demand and supply side of gentrification at the same time. The proposition was to give more aspiration to the residents of a depreciating neighborhood to make them rest in the neighborhood and at the time of aging the buildings, keep them ready for the shock of sudden elevation in prices after some renovation.[5]
Now in this study knowing that all the promotions in the level of life make it more costly, all the factors are brought in modeling to see their role in mobility of residents from some neighborhood in/to the/other neighborhood.

2.1. Gentrification

Gentrification is the displacement or replacement of a low ranking socio-economic group by a higher status socio-economic group in the inner city, and it involves the renovation of previously downgraded buildings for residential use.

Not all definitions of gentrification include the displacement of the lower-income residents. Some observers argue that displacement is not a necessary outcome of gentrification if original residents cannot afford to move elsewhere or are attached to the neighborhood, or if higher-income households are able to occupy vacant properties or move into newly constructed developments.[6]

2.2 Post modern comfort

Gentrification is mostly described as the result of enhancing the quality of living area in some concrete aspect of life such as the buildings and their quality. The other aspect of quality enhancing which generally concerns the life quality take into account the discrete aspects. That can be mentioned as postmodern approach.
The post-modern comfort is a new concept which understands and articulates the social mutation better. It is not just a kind of enrichment of the tenants which end to reembourgeoisement of central parts, but is an effort to understand other options for residence choosing in non equal social classes. The evolution of socio-residential segregation reflect this malfunction of inter and intra-urban.
There is a measurable postmodern comfort in intra-urban scale according to eco-socio-spatial logic in gentrification.

The definitions like residential comfort, equipment comfort, life level, life style, etc that are mostly immeasurable, are inferred from a qualitative explanatory method. By using factor analysis, the latent dynamism of the social system is defined.

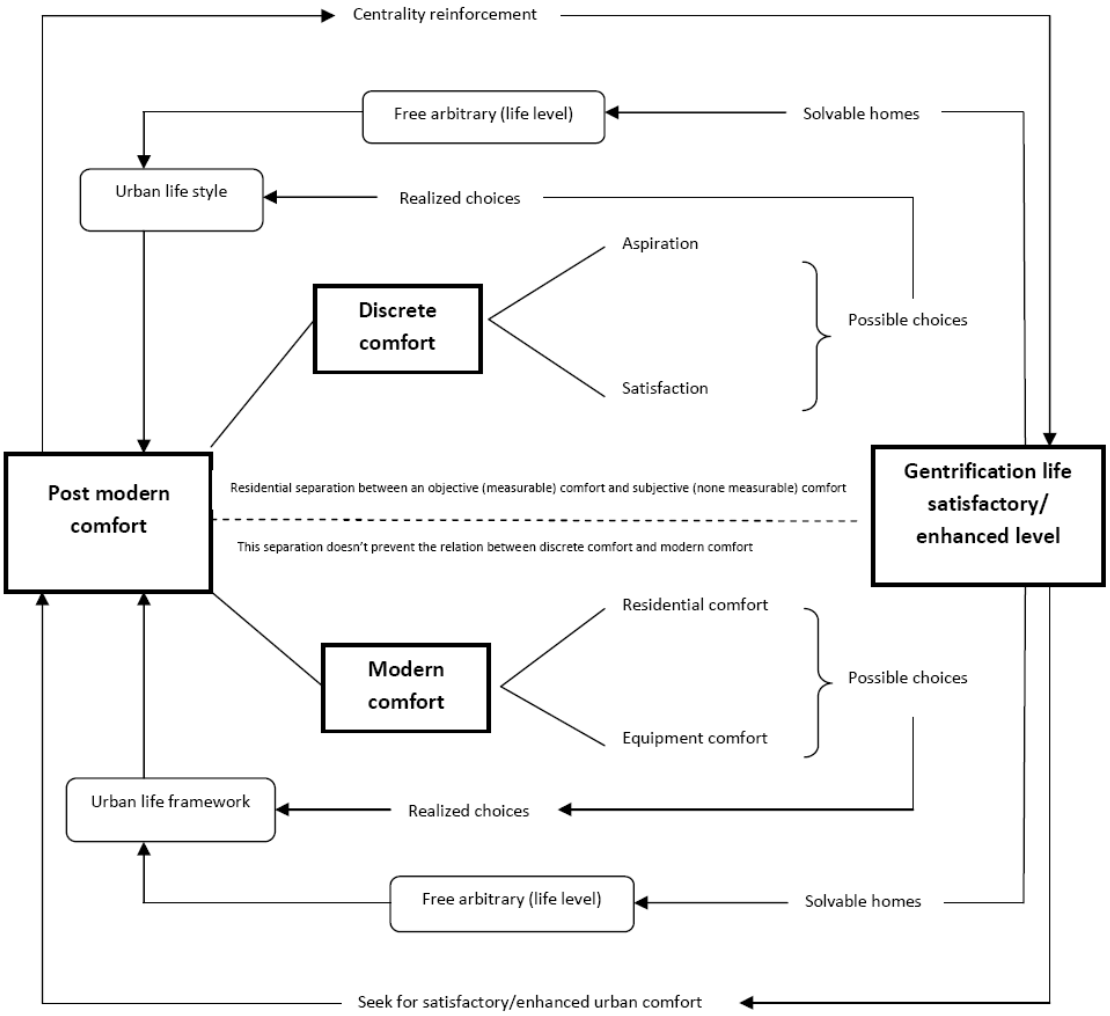


Fig. 1 Construction of relation between postmodern comfort and gentrification

2.3 Centrality

Due to the new means of communication and transportation, the socio-spatial mobility is growing. The replacement of deindustrialization with service sectors accelerates this process. The "heap of sand" logic recalls that the work evolution reinforces the concentration and spatialization. With redevelopment of central urban functions and the diversification of the activities of service sectors, two spatial forms spread in fewer than 20 years:

The first is a central form –centralization- with the new commercial centers, hotel complexes, offices, big markets. The "Halles" [markets] of Strasbourg or that one of Paris which are located in the very center of city are the examples that form the functional and material centers and it strengthens the concentration role in the center of city.

The second one is characterized with a peripheral spatial form –sprawl- installed in the proximity of the main axes of transport: supermarkets and other big specialized surfaces in different fields (gardening, entertainment, tinkering, automobiles, etc). The centrality is not limited to the center of cities and the activities extend to peri-urban zones, responding to the demands of habitants and the competition between the commerce and services.

According to the repartitioning of facilities and equipments in the city, the centrality of parts of city is calculated as below:

$$d_{ij} = \frac{f_j}{\sum f_j} = \frac{f_j}{s_j} \quad (1)$$

Which d_{ij} is the Scarcity Index, f is the number of specific institutions in different neighborhoods, and s is the sum of those institutions in all neighborhoods.

$$\text{Centrality index} = c_i = \sum_i d_j \quad (2)$$

And the Benisson Index introduces the rate of centrality of a given location.

$$\text{Bennison Index} = \frac{\frac{c_i}{\sum c}}{\frac{pop_i}{\sum pop}} \quad (3)$$

Which c_i is the centrality of i,
$\sum c$ the sum of centralities,
pop_i the population of i, and
$\sum pop$ is the sum of population in the zone ofstudy
The technical comfort which is the standard facilities that make a residence more comfortable and they are the results of statistics issued by INSEE.[i]

$$RC = \frac{\sum residence\ with\ WC + \sum residence\ with\ shower + \sum residence\ with\ central\ heating}{3 \sum residence} \quad (4)$$

RC (Residential Comfort) ranges between 0, 1. And the comfort found in a spacious residence which is based on the number of people living in each piece.

$$\text{Person per piece} = [0, 4.6] \quad (5)$$

In the fig. 2 the repartitioning of urban equipment is modeled in two time sections and the evolutions of the centralities are modeled.

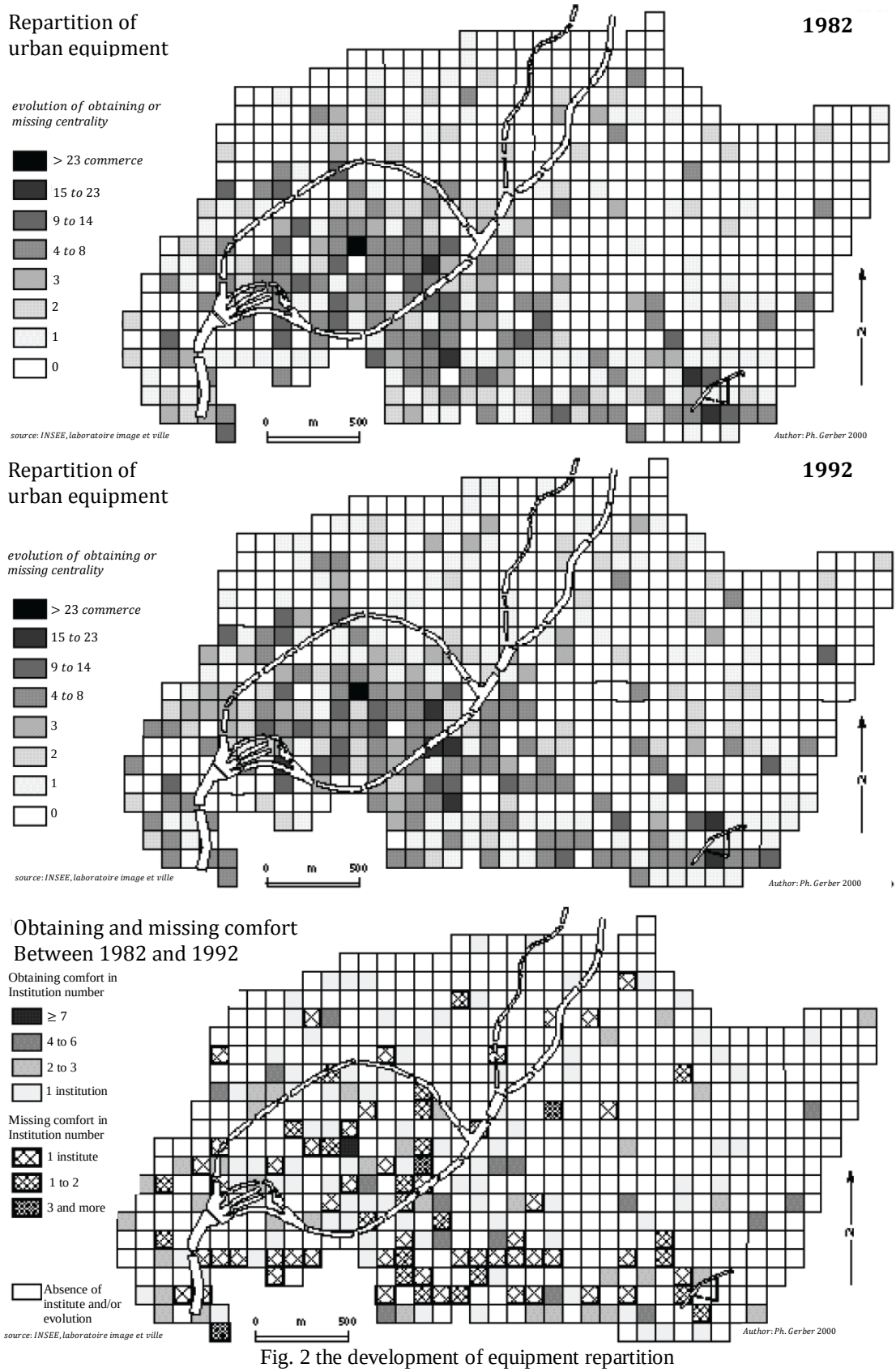


Fig. 2 the development of equipment repartition

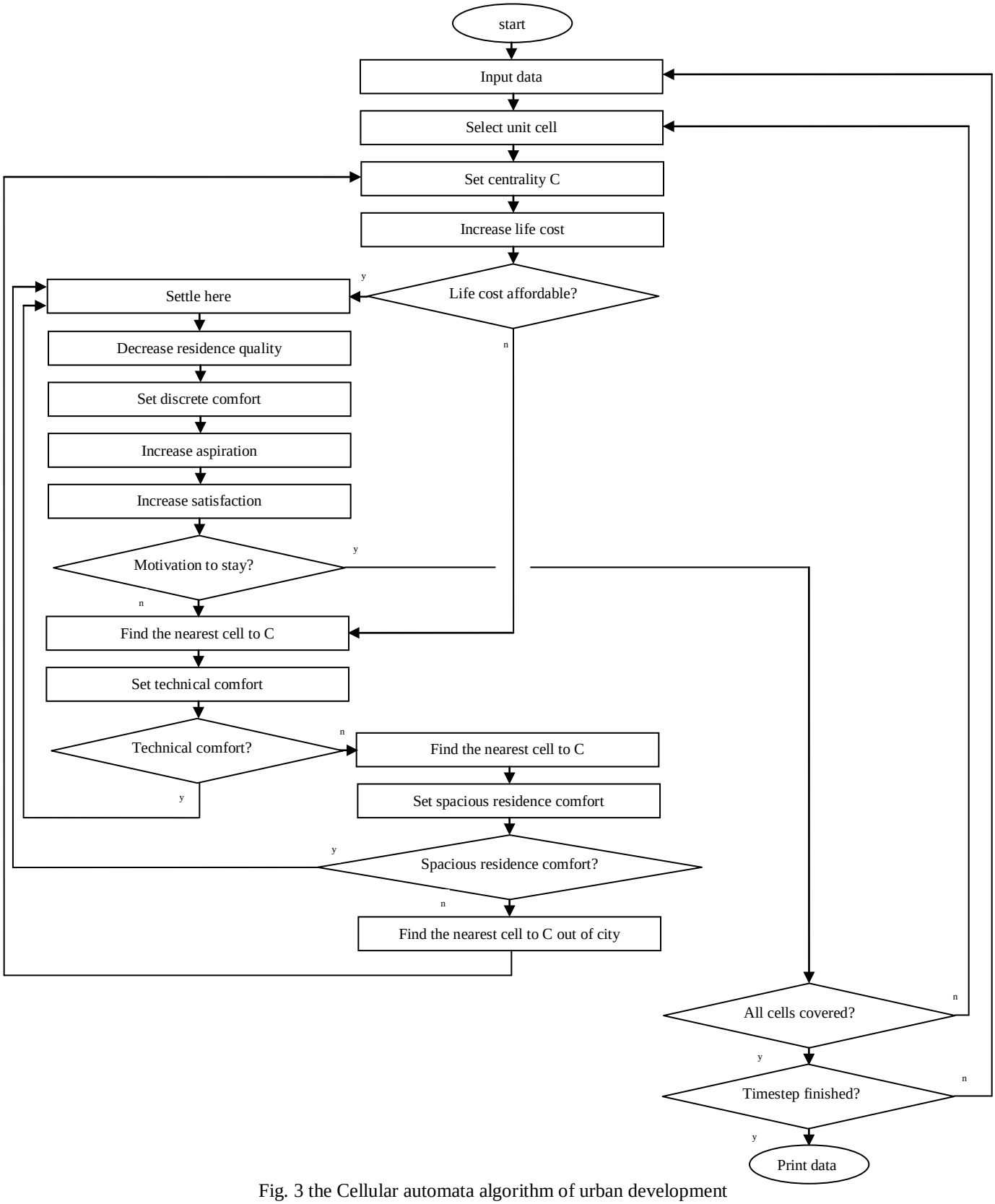


Fig. 3 the Cellular automata algorithm of urban development

3. Result and discussion

The simulation in Cellular automata

In the algorithm fig. 3 the impact of technical comfort and spacious residence comfort on settling and leaving the residents is modeled. The decision is based on closeness to the centralities. The modern and discrete comforts are the incentives for settling in a cell, and the depreciation of the residences and elevating the life costs are the constraints that make the resident leave.

Gentrification is based on a bottom-up, self organizing system where space and neighboring relationships are crucial. The cells are chosen as (100m×100m) almost equal to a block.

The two main constituents of gentrification as a multi-agent system are:

- The actors and their behaviors by social, spatial and economic *agents*
- The blocks by *cells* that change state on the base of agent actions that influence the cell's nearest neighbors.

4. Conclusion

By studying gentrification in complex system, some other agents can be added to make the models more real.

Based on models that study the formation of cities and their development as the result of mobility of the population, the impact of gentrification on the city development will be studied in three urban growth models:

Concentric distribution, with an urban extension in rururban,

The multi-polar situation in locating social residence,

And sectorial repartition.

References

[1] Michael Batty 2005, "*Cities and Complexity: Understanding Cities with Cellular Automata, Agent Based Modeling, and Fractals*" Press, Cambridge, Mass. pp.589.

[2] Ferner C., 2008 *complexity and agent-based modeling in urban research,* Studies for PhD at center of Forest, Landscape and Planning, University of Copenhagen, pp.12.

[3] Joshua M. Epstein, 2006 *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University. pp. 352.

[4] Gerber, P. 2000, *Gentrification et confort postmoderne elements emergents de nouvelle centralite, these de doctorat de Géographie*, Université de Strasbourg., pp.581.

[5] Kaboli, M. Hadi, 2010, *Cellular Automata in gentrification of the city, CSC10, Las Vegas*.

[6] Vigdor, Jacob L., 2002, *Does Gentrification Harm the Poor?,* Brookings-Wharton Papers on Urban Affairs pp. 133-182.

[i] Institut national de la statistique et des études économiques

From Complexity to Random Behaviors; Generate Random Numbers by Confusion in Cellular Automata State's

Seyed Morteza Hosseini[1], Hossein Karimi[2], Majid Vafaei Jahan[3]
[1,2,3] Department of Software Engineering, Mashhad Branch - Islamic Azad University, Mashhad, Iran

Abstract- *Cellular Automata(CA) with evolutionary and complex behaviors are used in several applications such as generating random numbers and cryptography. Because of the intrinsic self-organizing property, pure CA cannot produce a long sequence of random numbers. For increasing the sequence of produced numbers, non-uniform CA, controllable/programmable CA, stimulating factors or combination of several automata may be used. In this paper combined non-uniform cellular automata as a random number generator with contribution of Langton's ant is presented. Langton's ant has a complicated behavior that a combination of some Langton's ants gives them a chaotic behavior combination of chaotic behavior with complex behavior of cellular automata causes great efficiency in generating random sequence. Experimental results show that, in spite of our expectation, combination of Langton's ant and cellular automata does not have chaotic behavior (does not depend on initial value) but illustrates the random behavior. Which, it results in cycles of very long period lengths with limited number of cells such that a period length 2^{3n} is obtained by n cells.*

Keywords: Random Number Generator, Langton's Ant, Cellular Automata, Diehard Test, Long Period Length.

1 Introduction

Random number generators (RNGs) play an important role in several computational fields, including Monte Carlo techniques [1], Brownian dynamics [2], stochastic optimization methods [2, 3] and key-based cryptography [4]. It is usual to use mathematical or even evolutionary methods to construct RNGs that yields high quality generators. The quality of generators that determined by statistical tests have a great important role; for example, in cryptography, low quality of RNGs causes easily breaking the encrypted context [4]. In solving optimization problems, as shown in [5], performance and speed of algorithms directly depend on quality of used RNGs. To measure the quality of RNGs, some various statistical tests such as Entropy, Chi-Square, Diehard and NIST tests are used. In this paper, a new RNG algorithm based on combination of complexity methods is represented. Experiments show, the generated numbers do not depend on initial values and are independent of each other with uniform distribution. This generator is a two dimensional $n \times m$ cellular automata (CA) with eight different rules and some Langton's ants. The Langton's ants not only determine the CA rules but also give a random state to cellular automata by disturbing cellular automata state. The results show, this RNG passed all the mentioned quality tests such as entropy, chi-square, avalanche, Diehard and NIST test. Uniformity of generated numbers, high speed parallel processing and sensitivity to bit changes in special applications such as cryptography are beneficial features of presented RNG.

Following the introduction, in Section 2, related works are discussed. In Section 3, an overview on basic concepts of CA and Langton's ant are described. In Section 4, the proposed RNG algorithm and its behavior are discussed. In Section 5, the experimental data and test results are illustrated and at last in Section 6, the conclusion and future works are discussed.

2 Related works

The first work to apply CA as RNG was done by Wolfram in 1986. His work shows the ability of CA to generate random bits [6, 7]. Basic researches on CA are on producing RNG by one dimensional CA with 3 neighbors [7]. Other researches are focused on increasing CA's complexity with combinations of controllable cells [4, 8] or increasing CA`s complexity with increasing dimensionality. RNG are produced by using one dimensional CA studied in [9, 10, 11, 12, 13] and two dimensional CA in [14, 15, 16] and three dimensional CA in [17]. Hortensius proposed the first non-uniform CA or programmable CA (PCA) by using of the combination of two rules, 90 and 150 in 1989[9].

PCA is a non-uniform CA that allows different rules to be used at different time steps on the same cell. He also represented another generator using the combination of rules 30 and 45 in [10] that its output bits have more dependencies to each other rather than rules 90 and 150. Recently, extensive studies have been done on PCA for generating random numbers [11, 15, 16, 18, 19]. First works on two dimensional CA represented by Chaudhuri et al. in 1994 [14]. Their results show that produced generator using this CA works better rather than one dimensional CA with the same size. In [20, 21] all 256 (simple) elementary cellular automata were investigated (including those with rules given 90 and 150). It was found that CA with nonlinear rules 45 (or its equivalent rules 75, 89 or 101) exhibit chaotic (or pseudo-random) behaviors similar to those obtained in LFSRs.

3 Cellular Automata and Langton's Ants

3.1 Cellular Automata

A cellular automaton (CA), introduced by Von Neumann in 1940s, is a dynamic system in which its time, space and states are all discrete. The CA evolves deterministically in discrete time steps and each cell takes its value from a finite set S, called the State Set. A CA is named Boolean if s = $\{0,1\}$. The $i-th$ cell is denoted by ‹i› and the state of cell ‹i› at time t is denoted by a_i^t . For each cell ‹i›, called central cell, a symmetric neighborhood of radius r is defined by (1):

$$v_i = \{‹i-r›, \dots., ‹i›, \dots., ‹i+r›\} \quad (1)$$

the value of each cell ‹i› is updated by a local transition function f_i-called rule- which for a symmetric neighborhood with radius r is defined as follows (2):

$$a_i^{t+1} = f(a_{i-r}^t, \dots, a_i^t, \dots, a_{i+r}^t) \quad (2)$$

or equivalently by (3):

$$a_i^{t+1} = f(v_i^t) \quad (3)$$

Such that v_i^t is as follows (4):

$$v_i^t = f(a_{i-r}^t, \dots, a_i^t, \dots, a_{i+r}^t) \quad (4)$$

To represent a symmetric rule of radius r for a Boolean CA, a binary string of length L is used, where $L = 2^{2r+1}$, Table 1 Shows the rule 90 of radius one (r=1).

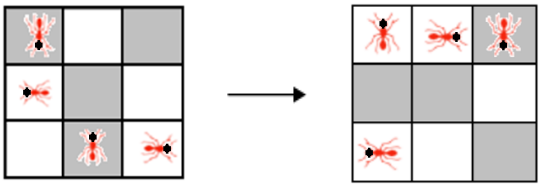
Table 1. The Rule Representation Of Boolean Symmetric Rule 90 Of Radius One

Neighborhood Number	7	6	5	4	3	2	1	0
v_i^t	111	110	101	100	011	010	001	000
$f(v_i^t)$	0	1	0	1	1	0	1	0

If all CA cells obey the same rule, then the CA is said to be a uniform CA; otherwise, it is a non-uniform CA[22]; in addition, a CA is said to be a CA with periodic boundary condition if the extreme cells are adjacent to each other else it called null-boundary CA. If a CA rule involves only XOR logic, it is called a linear rule; rules involving XNOR logic are referred to complemented rules. A CA with all cells having linear rules is called linear CA, whereas a CA having a combination of linear and complemented rules is called an additive CA [23]. In this paper, we used a non-uniform two dimensional CA with periodic boundary and {0, 1} as its states and state of each cell depends on the state of itself and its neighbors.

3.2 Langton's Ant

Langton's ant is a two-dimensional Turing machine with a very simple set of rules but complicated emergent behavior. It was invented by Chris Langton in 1986 and runs on a square lattice of black and white cells (zero or one). Each ant has been put arbitrary in one of black or white cells. They can move to each of their four neighbors according to these rules: (1) At a white square, turn 90° right, flip the color of the square, move forward one unit. (2) At a black square, turn 90° left, flip the color of the square, move forward one unit (Fig. 1).These simple rules cause very complicated and chaotic behavior. In this paper, we consider 1 as the value of white cell and 0 for black.



Before move After move

Fig. 1. Langton's ants behavior in 3 × 3 square lattice

4 The proposed generator based cellular automata and Langton's ants

4.1 The proposed generator

In this scheme a two dimensional, non-uniform $n \times m$ CA with periodic boundary condition are used to generate random numbers by using 8 rules: 153, 30, 90, 165, 86, 105, 110, 150. The Boolean expression of each CA rule is shown in Table 2. According to [24], generated numbers by these rules have the best results in different tests such as entropy, chi-square and diehard.

Table 2. The Detail and Boolean Expression of Each CA Rule

Rule Name	Possible Input Configuration								Boolean Representation
	111	110	101	100	011	010	001	000	
101	0	1	1	0	0	1	0	1	$[x_{i-1}\ nor\ x_{i+1}]\ or\ [(x_i\ xor\ x_{i+1})\ and\ x_{i-1}]$
105	0	1	1	0	1	0	0	1	$Not[x_{i-1}\ xor\ x_i\ xor\ x_{i+1}]$
86	0	1	0	1	0	1	1	0	$[x_{i-1}\ nor\ x_i]\ xor\ [not(x_{i+1})]$
165	1	0	1	0	0	1	0	1	$[x_{i-1}]\ xnor\ [x_{i+1}]$
90	0	1	0	1	1	0	1	0	$[x_{i-1}]\ xor\ [x_{i+1}]$
30	0	0	0	1	1	1	1	0	$[x_{i-1}]\ xor\ [x_i\ or\ x_{i+1}]$
153	1	0	0	1	1	0	0	1	$[x_i]\ xnor\ [x_{i+1}]$
150	1	0	0	1	0	1	1	0	$[x_{i-1}]\ xor\ [x_i]\ xor\ [x_{i+1}]$

Each ant is placed on corresponding cell of CA. The position of ant shows the rule number of cell automata. Furthermore each ant has exactly two fixed neighbors and doesn't depend on the position of ants in CA. Here as Fig. 2 Shows, the numbers of i^{th} ant's neighbors that are i-1 and i+1.

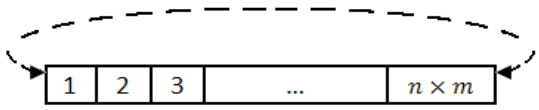


Fig. 2. Selection of ant's neighborhood.

For generating random numbers, the initial values of CA, the position and direction of ants are determined randomly. In each run, a rule is determined for each cell (based on the lookup Table 3); then CA is updated and ants move one step.

Table 3: CA rules lookup table

0	1	2	3	4	5	6	7
165	105	90	153	150	101	30	86

A rule is chosen for each cell as shown in the lookup Table 3. For selection of rule for each cell, the value of cells which the ants (with the same cell's number) and their neighbor ants are there, generate a number between 0..7. Then by Table 3, the rule of cell is selected according to rows. For example the number of cells is shown in Fig. 3 for a 3×3 CA. In the second ant the neighbors are always first and third ants.

1	2	3
4	5	6
7	8	9

Fig. 3. CA arrangement for rule selection.

If the value of cells for first, second and third were 1, 1, 0 respectively, the chosen rule number for second cell will be $(011)_2 = 3$. i.e. The rule of cell in position [1, 2] (row 1 and column 2), according to Table 3 is 153. It should be mentioned that the neighbors of first ant are second ant and $(m \times n)^{\text{th}}$ ant. Ants only move according to the rules. So, this maybe yields that one cell has more than one ant or even has no ants. After rule selection for each cell, the rules have been applied on CA by row. If the ants were more than $m \times n$, the extra ants have been used for stimulate the CA and have no effect on rule selection.

4.2 How does Langton's Ant Disturb Cellular Automata and Generate Random Numbers?

4.2.1 Langton's Ant Role

Role of Langton's ant in this generator, in addition to determine executive rule for a cell, is to actuate the maximum disturbance in cellular automata to prevent cycle formation and to reach the maximum entropy in cellular automata. What important is that disturbance in cellular automata cells state should be the maximum at each of ants movement (%50 of change). This disturbance causes that the number of very long iteration for reaching the current state to be needed for few cells and a sequence of pseudo-random generated bits to be of high quality. Table 4 represents change degree of cellular automata cells states at every movement of ants for different number of ants. For this reason, the algorithm was performed 100 times and each time, 5000 times with different number of ants. To calculate changes, percentage of imposed changes by cellular automata rules is neglected.

Table 4. Changing percentage of cellular automata state for ants movement

Ant Number	Percent Change		
	Min	Max	Average
25	0.4414	0.4445	0.4430
50	0.4867	0.4897	0.4893
75	0.4984	0.5001	0.5001
100	0.4999	0.4985	0.5007
250	0.4994	0.5009	0.5002
500	0.4991	0.5004	0.4998

As it is shown in Table 4, changes percentage for the increased number of ants to 4 times of cells is almost 0.5 and by more increasing of ants, changes percentage remains the same as 0.5 with a little difference.

4.2.2 Combining Cellular Automata and Langton's Ant

As it was stated in 4.1, the proposed generator is a combination of cellular automata and Langton's ants. According to Table 4, for the same number of ants and cells, changes percentage is a little less than 0.5 that is because of being located of some ants in one cell and another change of modified cell by next ants. Table 5 shows changes percentage for movement of ants and implementation cellular automata rules.

Table 5. Change Percentage of Cellular Automata State For Movement of Ants and Implementation of Cellular Automata Rules

Ant Number	Percent Change		
	Min	Max	Average
25	0.4982	0.5016	0.5000
50	0.4990	0.5017	0.4999
75	0.4992	0.5009	0.5006
100	0.4984	0.5010	0.5002

As it is observed in Table 5, changes in this table to Table 4 are better with the same number of ants and for less number of ants reaches to 0.5.

5 Experimental Results

A 3×3 CA is used for tests. In all cases which the numbers of ants are chosen less than 9, the purpose is the number of stimulus ants (those who change the value of a cell), while the ants that select the rules are those 9. In other words, there are at least 9 ants. For k ants which $k < 9$ ants, $9 - k$ extra ants are considered, that move according to the rules but have no effect on their cell's value. In the cases that the number of ants is greater than 9, only first 9 ants will determine the rules and remained ants only is used as the stimulator of CA.

5.1 Several Basic Statistical Tests For PRNG

Let s $= s_0, s_1, s_2, \ldots, s_{n-1}$ be a binary sequence of length n. This subsection presents several basic statistical tests that are commonly used for determining whether the binary sequence s possesses some specific characteristics that a truly random sequence would be

likely to exhibit. It is emphasized again that the outcome of each test is not definite, but rather probabilistic.

5.1.1 Frequency Test (Monobit Test)

The purpose of this test is to determine whether the number of $0's$ and $1's$ in s are approximately the same, as would be expected for a random sequence. Let n_0, n_1 denote the number of $0's$ and $1's$ in s, respectively. The statistic used is:

$$x1 = \frac{(n_0 - n_1)^2}{n} \quad (5)$$

which approximately follows a x^2distribution with 1 degree of freedom if $n \geq 10$. For a significance level of $\alpha = 0.05$, the threshold values for this test is 3.8415 [25].

5.1.2 Serial Test (Two-Bit Test)

The purpose of this test is to determine whether the number of occurrences of $00, 01, 10,$ and 11 as subsequences of s are approximately the same, as would be expected for a random sequence. Let n_0, n_1 denote the number of $0's$ and $1's$ in s, respectively, and let $n00, n01, n10, n11$ denote the number of occurrences of $00, 01, 10, 11$ in s, respectively. Note that $n00 + n01 + n10 + n11 = (n - 1)$ since the subsequences are allowed to overlap. The statistic used is:

$$\frac{4}{n-1}(n00^2 + n01^2 + n10^2 + n11^2) - \frac{2}{n}(n_0^2 + n_1^2) + 1 \quad (6)$$

which approximately follows a x^2distribution with 2 degrees of freedom if $n \geq 21$. For a significance level of $\alpha = 0.05$, the threshold values for this test is 5.9915 [25].

5.1.3 Poker Test

Let m be a positive integer such that $\left\lfloor \frac{n}{m} \right\rfloor \geq 5.(2^m)$ and let $k = \left\lfloor \frac{n}{m} \right\rfloor$. Divide the sequence s into k non-overlapping parts each of length m, and let n_i be the number of occurrences of the i^{th} type of sequence of length $m, 1 \leq i \leq 2m$. The poker test determines whether the sequences of length m each appear approximately the same number of times in s, as would be expected for a random sequence. The statistic used is:

$$x3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k \quad (7)$$

Which approximately follows a x^2 distribution with $2^m - 1$degrees of freedom. Note that the poker test is a generalization of the frequency test: setting m = 1in the poker test yields the frequency test. . For a significance level of $\alpha = 0.05$, the threshold values for this test is 14.0671 [25].

5.1.4 Autocorrelation Test

The purpose of this test is to check for correlations between the sequence s and (non-cyclic) shifted versions of it. Let d be a fixed integer,$1 \leq d \leq \lfloor n/2 \rfloor$. The number of bits in s not equal to their d-shifts is $A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$ where $\oplus$ denotes the XOR operator. The statistic used is:

$$x_5 = 2\left(A(d) - \frac{n-d}{2}\right)/\sqrt{n-d} \quad (8)$$

Which approximately follows an $N(0; 1)$ distribution if $n - d \geq 10$. Since small values of $A(d)$ are as unexpected as large values of $A(d)$, a two-sided test should be used. . For a significance level of $\alpha = 0.05$, the threshold values for this test is 1.96 [25]. In Table 6, values of discussed tests are presented for the proposed generator. For this, a sequence of random numbers is generated with 1 million bits and discussed tests are implemented on it. This procedure is repeated 100 times and its average is given, too.

Table 6. Values of 4 basic statistical test

MCD	TESTS				Pass
	Frequency	Serial	Poker	Autocorrelation	
2	122.146	312.038	1075.00	1.5276	1/4
4	40.424	109.886	356.893	0.1668	1/4
6	8.9880	29.1744	84.1101	1.1336	1/4
8	4.6742	13.5709	32.5310	1.3829	1/4
9	0.0876	4.1185	8.3874	1.989	4/4
≥ 10	0.732	3.326	6.761	0.245	4/4

As it is shown in Table 6, generator with more than 8 ants is able to pass all tests.

5.2 ENT Test

The ENT test is useful for evaluating pseudorandom number generators for encryption and statistical sampling applications, compression algorithms, and other applications where the information density of a file is of interest [26]. The ENT test is a collective term for three tests, known as the Entropy test, Chi-square test, and Serial correlation coefficient (SCC) test. Table 7 shows values of this test for the proposed generator with 1 ant, 3, 6 and 9 ants. In entropy test, its maximum value is 4 and Chi- Square test with freedom degree 4 and precision of 0.1 is used. For doing these tests, a sequence of length 2^{17} is used.

Table 7. ENT Test

Number of Ant's	Entropy	Chi-Square	SCC
1	3.9944	21.8743	0.00865
3	3.9992	9.5983	0.00022
6	3.9996	7.4759	0.00017
9	3.9999	6.8734	0.00002

As it is represented in above table, generated sequence by 6, 3 and 9 ants is able to pass all tests successfully and with good result.

5.3 PRNG Quality Evaluation

To compare how our PRNG performs against several different PRNGs, we use Diehard[27] And NIST test suite [28] . For this reason, the proposed generator based on obtained score from DIEHRD and NIST test is compared with other generators.

5.3.1 Based on Diehard Test Suite

we used Johnson's scoring scheme [29]: we initialized (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7) with 32 different random values obtained from http://randomnumber.org, got 32 different 10MB files, and then assigned scores based on the results of the Diehard tests. The PRNGs we have compared to ours are of several different kinds: Linear Congruential Generators (rand [30], rand1k [31], pm [32]), Multiply with Carry Generators (mother [33]), Additive and Subtractive Generators (add [30], sub [32]), Compound Generators (shsub [30], shpm [32], shlec [32]), Feedback Shift Register Generators (tgfsr [34], fsr [35]), and Tausworthe Generators (tauss [36]). Each of the Diehard tests produces one or more p-values. We categorize them as good, suspect or rejected. We classify a p-value as rejected if $p \geq 0.998$, and as suspect if $0.95 \leq p < 0.998$; all other p-values are considered to be good. We assign two points for every rejection, one point for every suspect classification, and no points for the rest. Finally, we add up these points to produce a global Diehard score for each PRNG, and compute the average over the 32 evaluations:

low scores indicate good PRNG quality. The information relating to the different PRNGs was taken from [31, 37]. The results are presented in Table 8. We note that our PRNG is outstandingly better than the rest of the analyzed PRNGs: the lowest scores correspond to shsub (17.125) and fsr (17.90625), significantly greater than our PRNG (12.90625). On the other hand, the average scores increase up to 50.59375 (pm), 66.53125 (rand), and even 291.78125 (rand1k).

Table 8. PRNG Diehard Scores

PRNG	Total Score	Mean
rand	2129	66.531250
rand1k	9337	291.78125
pm	1619	50.593750
mother	602	18.812500
add	577	18.031250
sub	655	20.468750
shsub	548	17.125000
shpm	799	24.968750
shlec	751	23.468750
fsr	573	17.906250
tgfsr	584	18.250000
tauss	935	29.218750
Proposed PRNG	413	12.906250

5.3.2 Based On NIST Test Suite

The NIST Statistical Test Suite is widely proverbial, which provides advice on developing a testing strategy for pseudorandom bit sequence generator. The NIST Statistical Test Suite supplies the user with nine pseudo-random number generators. In this section, the proposed generator is compared with these 9 generators with respect to obtained score from NIST test. Once the generator has been selected, a series of binary sequences are to be generated and to be saved to analyze. A set of tests are used to the saved sequences. The P-value is then to be examined to determined pass of failure of sequences. Then, the proportion of sequences passing the tests should be considered relative to a normal distribution confidence interval. If the proportion is within the confidence interval, the generator will be acceptable. NIST test includes 16 tests. For comparing by the proposed generator, 100 sequences of random numbers of 12 mega bites are generated with the random initial values. Then, NIST tests are executed on it. Obtained values of these tests contain 188 parts and because the maximum value for each of these tests is one, hence, the maximum summation of these tests is 188. For obtaining the average of obtained values for every generator, values obtained from NIST test for each sequence are summed together and finally are divided by the number of sequences.

The best case occurs when the average to be 188. A test is not passed when not be located in defined confidence interval. To obtain the average number of failures in NIST test for each generator, the number of failures for all sequences are summed and divided by the number of sequences The average of obtained values for every generator and also the average number of failures of generators are given in Table 9. To reach a general index for comparing generators with each other, the average of values is divided by average number of failures. The bigger number value is, better quality of generator is.

Table 9. PRNG NIST Scores

PRNG	mean Score	Mean Fail rate	Total score
Linear Congruential	185.5212	5.17	35.8841
Quadratic Congruential Ⅰ	185.7121	7.41	25.0623
Quadratic Congruential II	186.0181	3.51	52.9966
Cubic Congruential	185.0396	15.76	11.7410
XOR	95.7669	175.50	0.5456
Modular Exponentiation	186.0254	4.59	40.5284
Blum-Blum-Shub	186.2247	1.94	95.9921
Micali-Schnorr	185.9812	3.12	59.6093
G Using SHA-1	185.8135	7.47	24.8746
Our PRNG	186.1374	1.26	147.7280

As it is clear in Table 9, the proposed generator can reach the best score among other generators that shows high quality of this generator.

5.4 Sequence Period Length

The length of a CA's state cycle is very important in determining the suitability of the CA as a generator of random numbers. In general, the longer the cycle-the better the CA acts as an RNG. For instance, typical Monte Carlo applications may require on the order of 10^9 pseudorandom numbers. According what mentioned in [4]: Ideally, an arbitrary n-cell CA RNG should have a maximum cycle length, i.e., it should start repeating itself only after 2^n time steps since this will result in the longest possible pseudorandom sequence. Chaotic behavior of the movement of ants among cellular automata cells leads to complicated behaviors contributed with great disturbance in cellular automata, such that it is possible to reach a sequence of random

numbers of very long period with a few cells. Table 10 shows generated cycle by 3 up to 9 ants and cells and also the maximum generated cycle by the proposed generator in [4]. The maximum computation power for measuring period length is 2^{27} bits. For obtaining the average period length, the sequence of random numbers is generated for each ant and cell and then the average period length is calculated. As it is observed, length of generated cycle for n cells and ants is almost $(2^n)^3 = 2^{3n}$.

Table 10. The Generated Cycle by The Proposed Generator of Different Cells and Ants

Ants/Cells Number	3	4	5	6	7	8	9	≥10
2^n	8	16	32	64	128	256	512	1024
Cycle Length	$2^{9.5}$	$2^{10.3}$	$2^{16.1}$	$2^{16.2}$	$2^{21.3}$	$2^{22.01}$	$2^{27.1}$	?

5.5 Avalanche Effect

Desirable property of any cryptographic algorithm is that a small change in either plaintext or the key should result in a significant change in the ciphertext. Changing value of one randomly chosen bit in the plaintext or in the key should produce change of nearly half of the values of the ciphertext. This is so called avalanche property. It was introduced by H.Feistel in 1973 [38]. Represented generator could be used to generate key. For generate a unique key in encryption and decryption sides, initial values of CA, ant positions and directions must be available in both sides. Thus for high security, the generated stream bits should have a high dependency to this parameters. Effects of small changes of parameters on generated bits represented in continue.

Investigated changes are: 1) Reversing in one of the cells of CA. 2) changing in one ant's direction. 3) Changing in position of one ant to one of its four neighbors. The parameters of evaluation of changes are: the percent of changed bits in generated bit stream. Table 11 shows the change results in a 3×3 CA with 9 ants, on the generated sequence of one cell. Three samples are investigated for each change that will be discussed.

Table 11: Changes Effects on Generated Bits Sequence

Change Type	Percent Change
Change a cell of cellular automata	49.25
	48.7
	50.7
Change direction of an ant	50.45
	50.95
	50.25
Move an ant to the neighbor cell	49.10
	48.10
	51.65
Consecutive bits in a sequence	51.1
	50.05
	50.25

As shown, a small change yields about 50 percent changes in generated bits sequence.

6 Conclusion

In this paper a new random number generation method based on CA and Langton's ants was presented. By combining chaotic behavior of various ants and complex and self organized behavior of cellular automata, a qualified generator is presented for generating sequence of random numbers. Langton's ant moves on a square grid with white and black cells that yet the simplicity has a chaotic behavior. CA has been updated with respect to the rows using the combination of 8 rules: 165, 105, 90, 150, 153, 101, 30, 86. In each run of program, at first as rule has been determined for each cell according to the ant's position. Then automata have been updated by new selection rules and finally ant's moves one step forward. The tests results on generated random numbers show that the proposed generator has maximum entropy and passing the several static test, it is able to generate high quality Sequences of random bits with uniform distribution. This generator has an acceptable speed and holds the ability of parallelism of CA. Farther more; passing all parts of diehard and NIST test shows the high quality of generated random bits by this generator. This property is justifier for cryptography applications.

Reference

[1] J. Gentle, "Random number generation and Monte Carlo methods," Springer New York 2003, 2th edition, 2004, ISBN-10: 0387001786

[2] A. Reese, "Random number generators in genetic algorithms for unconstrained and constrained optimization," Nonlinear Analysis: Theory, Methods & Applications, Vol. 71, pp: 679 - 692, 2009.

[3] P.L.Ecuyer, "Random numbers for simulation," Communications of the ACM, Vol. 33, pp:85-97, 1990.

[4] M. Tomassini, M. Sipper, and M. Perrenoud, "On the generation of high quality random numbers by two-dimensional Cellular Automata," IEEE Transactions on Computers, Vol. 49, pp: 1146 –1151, 2000.

[5] J.Carmelo, A. Bastos-Filho, D. Jaulio, D. Andrade, R. Marcelo, S. Pita, D. Ramos, "Impact of the Quality of Random Numbers Generators on the Performance of Particle Swarm Optimization," IEEE International Conference on Systems, Man and Cybernetics , pp: 4988–4993, 2009.

[6] S. Wolfram, "Cryptography with cellular automata," in Proc. CRTPTO 85 Advances in Cryptography, Vol. 218, pp: 429–432, 1985.

[7] S. Wolfram, "Theory and Applications of Cellular Automata," River Edge, NJ: World Scientific, pp:1983–1986, 1986.

[8] S.-U. Guan and S. Zhang, "A family of controllable cellular automata for pseudorandom number generation," International Journal of Modern Physics C, Vol. 13, Issue 8, pp:1047-1073 2002.

[9] P. D. Hortensius, R. D. Mcleod, and H. C. Card, "Parallel random number generation for VLSI system using cellular automata," IEEE Transactions on Computers, Vol. 38, pp: 1466–1473, 1989.

[10] P. D. Hortensius, R. D. Mcleod,W. Pries, D. M. Miller, and H. C. Card, "Cellular automata-based pseudorandom number generators for built-in self-test," IEEE Transactions on Computers, Computer-Aided Design, Vol. 8, pp: 842–859, 1989.

[11] P. Anghelescu "Encryption Algorithm using Programmable Cellular Automata", World Congress on Internet Security (WorldCIS), pp: 233 – 239, 2011

[12] S.H. Shin, K.Y. Yoo, "Analysis of 2-State, 3-Neighborhood Cellular Automata Rules for Cryptographic Pseudorandom Number Generation ",International Conference on Computational Science and Engineering, CSE '09, pp: 399 – 404,2009.

[13] X.Xuewen, L.Yuanxiang, X.Zhuliang, W.Rong, "Data Encryption Based on Multi-Granularity Reversible Cellular Automata", International Conference on Computational Intelligence and Security, 2009. CIS '09, pp: 192 – 196, 2009.

[14] D. R. Chowdhury, I. S. Gupta, and P. P. Chaudhuri, "A class of two-dimensional cellular automata and applications in random pattern testing," Journal of Electronic Testing: Theory and Applications, Vol. 5, pp: 65–80, 1994.

[15] B.H.Kang, D.H.Lee, C.P.Hong, "High-Performance Pseudorandom Number Generator Using Two-Dimensional Cellular Automata", 4th IEEE International Symposium on Electronic Design, Test and Applications, pp:597 - 602, 2008

[16] B.H.Kang, D.H.Lee, C.P.Hong, "Pseudorandom Number Generation Using Cellular Automata", Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics, pp:401-404, 2008.

[17] S.H.Shin, G.D.Park, K.Y.Yoo, "A Virtual Three-Dimension Cellular Automata Pseudorandom Number Generator Based on the Moore Neighborhood Method", 4th International Conference on Intelligent Computing, ICIC 2008, pp: 174-181, 2008.

[18] A. Ray, D. Das, "Encryption Algorithm for Block Ciphers Based on Programmable Cellular Automata", Information Processing and Management, Vol.70, pp:269-275, 2010.

[19] N.S.Maiti, S.Ghosh, B.K.Shikdar, P.P.Chaudhuri , "Programmable Cellular Automata (PCA) Based Advanced Encryption Standard (AES) Hardware", 9th International Conference on Cellular Automata for Research and Industry, ACRI, pp:271-274, 2010.

[20] R. Dogaru, I. Dogaru, and H. Kim, "Binary chaos synchronization inelementary cellular automata", Int. J. Bifurcation and Chaos, 19, 2009.

[21] R. Dogaru, I. Dogaru, H.Kim,"Synchronization in elementary cellular automata", Proceedings of the 1Oth International Workshop on Multimedia Signal Processing and Transmission (MSPT'08), Jeonju, Korea, July 21-22, pp. 35-40, 2008.

[22] I.Kokolakis, I.Andreadis, and P. Tsalids, "Comparison between cellular automata and linear feedback shift registers based pseudo-random number generators," Microprocessors and Microsystems, Vol. 20, pp: 643–658, 1997.

[23] S. Nandi, B. K. Kar, and P. P. Chowdhuri, "Theory and applications of cellular automata in cryptography," IEEE Transactions on Computers, Vol. 43, pp:1346–1357, 1994.

[24] F. Seredynski, P. Bouvry, and A.Y. Zomaya, "Cellular automata computations and secret key cryptography," Parallel Computing, Vol.30, pp: 753-766, 2004.

[25] Alfred J. Menezes, Paul C. van Oorschot, Scott A.Vanstone "Handbook of Applied Cryptograph", CRC Press; 1 edition, pp:181-183, 1996, ISBN-10: 0849385237.

[26] ENT Test Suite, http:// www.fourmilab.ch/random

[27] G.Marsaglia, Diehard test, http://stat.fsu.edu/ ~geo/diehard.html, 1998.

[28] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E.Barker, S. Leigh, M. Levenson, M. Vangel, D.Banks, A. Heckert, J. Dray, San Vo. "A statistical test suite for random and pseudorandom number generators for cryptographic applications". NIST special publication 800-22 (with revision dated May 15, 2001.

[29] B.C. Johnson. Radix-b extensions to some common empirical tests for PRNGs. ACM Trans. on Modeling and Comp. Sim., 6(4):261–273, 1996.

[30] D.E. Knuth. The Art of Computer Programming, Volume 2: Seminumerical Algorithms. Addison-Wesley, 3rd edition, 1998.

[31] M.M. Meysenburg and J.A. Foster. Randomness and GA performance, revisited.In Proc. of GECCO'99, volume 1, pages 425–432. Morgan Kaufmann, 1999.

[32] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical Recipes in C. Cambridge University Press, 2nd edition, 1992.

[33] G. Marsaglia. Yet another RNG. Posted to sci.stat.math, 1994.

[34] M. Matsumoto and Y. Kurita. Twisted GFSR generators. ACM Trans. on Modeling and Comp. Sim., 2(3):179–194,1992.

[35] B. Schneier. Applied Cryptography. John Wiley and Sons, 1994.

[36] S. Tezuka and P. L'Ecuyer. Efficient and portable combined Tausworthe Random Number Generators. ACM Trans. on Modeling and Comp. Sim., 1(2):99–112, 1991.

[37] M.M. Meysenburg and J.A. Foster. The quality of PRNGs and simple genetic algorithm performance. In Proc. of the 7th Int. Conference on Genetic Algorithms, pp: 276–281, 1997.

[38] W. Stallings, "Cryptography and Network Security," Prentice Hall, 3th Edition, 2002, ISBN-10: 0130914290.

SESSION

NOVEL APPLICATIONS AND ALGORITHMS + LINEAR AND STOCHASTIC PROGRAMMING + DATA AND SIGNAL PROCESSING + SIMULATION AND HPC

Chair(s)

Prof. Hamid R. Arabnia

Non-linear Analysis of Psychophysiological Effects of Auditory Stimuli using Fractal Mining

M. Sink[1], M. Hossain[1], and T. Kato[2]
[1]Department of Computer Science, Fairmont State University, Fairmont, WV, USA
[2]Department of Behavioral Science, Fairmont State University, Fairmont, WV, USA

Abstract— *While spectral analysis (e.g., Fast Fourier Transformation) of electroencephalogram (EEG) has been one of the most well established parameters in psychophysiology, physiological implication of fractal analysis has not been established. Further, systematic examination of the association between the waveforms of auditory stimuli and EEG is an untouched area of research. In the present study, we used fractal analysis and data mining techniques, and created a method for finding the association between fractal dimensions of auditory stimuli and fractal dimensions of EEG. Applying our method, we found strong associations between signal complexity in auditory input and the resulting EEG data, with confidence values exceeding 90% in several of the associations. Our success in this initial application could potentially be generalized to further brain activity analysis.*

Keywords: Fractal Dimensions, Association Mining, Electroencephalogram

1. Introduction

Music is believed to have differential psychophysiological effects on humans. The notion on the effects of music dates back to the ancient era when Pythagoras created several diatonic scales and discussed their psychophysiological effects. Studies have shown that music can affect and stimulate different parts of the brain and can help with stress reduction, depression alleviation, and information recall. Such effects of music can be quantitatively studied using Electroencephalogram (EEG) [1], [2], [3], [4]. EEG refers to the electrical activity of the brain neurons captured from scalp surface. Each EEG electrode reflects electrical activity of about 100 neurons underneath the electrode and it produces tracing of pulses at various frequencies.

Even though studying the brain activity using linear analysis of electroencephalogram (EEG) is one of the most widely accepted research techniques in psychology and neuroscience, nonlinear analysis of EEG has not been extensively explored. Also, to the best of our knowledge, a systematic examination of the mathematical relationship between auditory stimuli and EEG has not been reported. The purpose of this work was to examine the psychophysiological effects of various auditory input in the form of synthetic music using fractal analysis and data mining techniques. In particular, we discovered the association between the auditory stimuli and the resulting EEG using fractal dimensions. Psychologists believe that there is a particular fractal dimensionality in nature and when the incoming stimuli imitates this fractal dimension, the nervous system would resonate with this fractal dimension and show a particular pattern.

The auditory stimuli and EEG are both high-dimensional time-series datasets that contain a very large number of features, some of which are highly correlated. This high-dimensionality of the data can make the data analysis task extremely difficult and time-consuming. A "fractal" [5] is defined to be a self-similar set of data points that consists of pieces similar to the original set, e.g., Sierpinski's Triangle. The "fractal dimension" is an estimate of the degrees of freedom of a data set [6]. The fractal dimension estimates the intrinsic dimension of the data and is a good indicator of the spread of the data. It is a useful tool to characterize the non-linearity and complexity of a given dataset. The fractal dimension of a dataset can make the data mining task more efficient and effective. The fundamental principle of fractal analysis is to identify the number of data points that self-correlate across scales, each of which is considered as a "dimension". Fractal dimension has been utilized as an effective tool for modeling various real world time series data with high complexity and irregularity [7], [8].

Our objective was to develop a working method for providing meaningful analysis of psychophysiological experiments. We analyzed EEG data collected from subjects who were exposed to auditory input, as well as the auditory data itself. We used fractal dimension analysis [9] and association mining [10] to provide the psychology researchers with information about their tests that they couldn't have otherwise discovered. To the best of our knowledge, this approach hasn't been implemented prior to this work.

The remainder of the paper is organized as follows. In section 2, we present the background pertaining to this work. In section 3, we present our analysis methods, describe our datasets, and present the experimental results. Finally, in section 4, we will provide concluding remarks and scope of future research.

2. Background

2.1 Fractal Dimension

The fractal dimension of a dataset is the degree of self-similarity that exists within the data. A fractal dimension is a non-negative real value that quantifies the complexity and irregularity of a dataset. Several methods have been developed for computing the fractal dimension of data.

Box-counting dimension [6] is by far the most commonly used fractal dimension. If $N(\epsilon)$ is the minimum number of n-dimensional boxes with sides of ϵ needed to cover the fractal, then the box-counting dimension is expressed as:

$$d_b = -\lim_{\epsilon \to 0} \frac{\ln N(\epsilon)}{\ln(\epsilon)}$$

Correlation dimension [11] is widely used when the data is available as a set of isolated points and is particularly suitable for time series data. It is easy to calculate, but its effectiveness is reduced with the presence of noise in data. If $C(\epsilon)$ is the fraction of pairs of points within a distance of ϵ, then the correlation dimension is defined as:

$$d_c = \lim_{\epsilon \to 0} \frac{C(\epsilon)}{\ln(\epsilon)}$$

Regularization dimension [9] is computed by smoothing (or regularizing) data by convoluting it with a Gaussian kernel. The regularization dimension quantifies how the length of a smoothed signal converges to infinity as Gaussian kernel width approaches 0. It is very effective in dealing with noisy data. If δ is the Gaussian kernel width and l_δ is the length of the smoothed signal, then the regularization dimension is formally expressed as:

$$d_r = 1 - \lim_{\delta \to 0} \frac{\ln l_\delta}{\ln \delta}$$

2.2 Association Mining

The goal of association mining is to derive correlations among multiple features of a dataset [10]. An association rule is an implication of the form $X \Rightarrow Y_{[Supp,Conf]}$, where X and Y are disjoint itemsets, $Supp$ is the *support* of $X \cup Y$ indicating the percentage of total records that contain both X and Y, and $Conf$ is the *confidence* of the rule that is defined as $Supp(X \cup Y)/Supp(X)$. The intuitive meaning of such a rule is that records of the dataset that contain X tend to contain Y.

A typical example of an association rule obtained from music experiment is $2.55 \geq FD(Music) > 2.45 \Rightarrow FD(EEG_{T6}) > 1.86_{[0.12,0.94]}$. This implies 94% of the time when the fractal dimension of music is between 2.45 and 2.55, the fractal dimension of the EEG response at the right temporal lobe T6 will be more than 1.86, this constitutes 12% of the data records. Here the confidence of the rule is 94% and the support of the rule is 12% .

The goal in a particular application is to find all association rules that satisfy user-specified minimum support and minimum confidence constraints. Association rules are generated in two steps. The itemsets having minimum support (called *large itemsets*) are discovered first and then these large itemsets are used to generate the association rules with minimum confidence. The *Apriori* association mining algorithm [10] has widely been accepted as the algorithm of choice in many applications. The process of generating large itemsets in Apriori consists of several passes and the large itemsets found in one pass are used to generate large itemsets for the next pass. In the k^{th} pass, the candidate itemsets of length k (C_k) are generated by joining large itemsets of length $k-1$ (L_{k-1}) and leaving out itemsets containing any non-large subset. Formally, $L_{k-1} * L_{k-1} = \{X \cup Y | X, Y \in L_{k-1}, |X \cap Y| = k-2\}$. All candidate k-itemsets having support values greater than the minimum support threshold constitute the large k-itemsets L_k. Formally, $L_k = \{X | X \in C_k, Supp(X) \geq Supp_{min}\}$. After all the large itemsets are generated, for every large itemset L, the following set of rules are generated: $\{A \Rightarrow (L - A) \mid A \subset L, A \neq \emptyset, Supp(L)/Supp(A) \geq Conf_{min}\}$.

2.3 Previous Work

Fractal dimensions have been used widely to analyze music. Gunasekaran and Revathy [12] used fractal dimensions of music segments to identify musical instruments using neural network classifiers. Das and Das [13] showed how the fractal dimensions calculated from the same song varies when it is performed by different singers. Bigerelle and Iost [7] used fractal dimensions to classify different types of music and demonstrated that fractal dimensions can distinguish different aspects of music effectively.

Fractal analysis of EEG signals have been found to be effective in neuroscience. Preissl et al. [14] showed how fractal dimension can be used to characterize the complexity of short-duration EEG signals. Jacquin et al. [15] combined wavelet and fractal analysis of EEG signals to detect seizures. Chouvarda et al. [16] used the fractal dimensions of EEG signals to study the different sleep stages in individuals. Easwaramoorthy and Uthayakumar [8] proposed a method for discriminating healthy and the epileptic individuals using a multi fractal analysis of EEG signals.

The influence of music on EEG has also been investigated for studying brain activities. Yuan et al. [1] studied the effect of music on EEG power spectrum. They showed that the presence of music makes significant changes in certain EEG power spectrum that are closely related to the emotional state of the nervous system. Bhattacharya et al. [17] analyzed the interdependency between different brain regions based on asymmetric similarity of EEG signals in response to music. Jausovec et al. [2] investigated the influence of music on brain activity during learning and showed that classical music can result in better task performance and less complex EEG patterns. Srinivasan et al. [3] investigated the effect of

music on mental fatigue by performing statistical analysis (ANOVA) on EEG signals and showed that the presence of music reduces mental fatigue during physical activities like jogging. Lin et al. [4] used machine-learning algorithms identify different emotional states based on EEG responses to music. Ito et al. [18] studied the association between an individual's egogram score based personality and the EEG pattern in response to music.

3. Methods and Experiments

3.1 Overview

As described in the previous section, fractal analysis has been used for analyzing music and EEG signals. The influence of music on brain activities has also been studied via EEG analysis. But, to the best of our knowledge, no computational framework has been presented to investigate the association between the fractal dimensions of music stimuli and the fractal dimensions of EEG responses. Fig. 1 shows the workflow of our method for finding the associations between auditory stimuli and the resulting multichannel EEG signals using fractal dimensions. The first step was to pre-process the auditory stimuli and EEG data. In our experiments, the auditory stimuli consisted of different pieces of synthetic music, varying in scale and note. The next step was to compute fractal dimensions from the auditory stimuli and the resulting EEG signals. The final step was to discover the associations between the fractal dimensions of the auditory stimuli and the fractal dimensions of the multi-channel EEG signals.

We believe that the fractal dimension computed from a music segment is a good measure of its pitch variation and thus will be helpful in a meaningful analysis of the psychophysiological effects of music. We used the "regularization dimension" approach [9] for fractal dimension calculation since it is more effective in dealing with noisy data. Because of the adaptive nature of signal smoothing, the regularization dimension is robust and allows for small step variation in the Gaussian kernel [19].

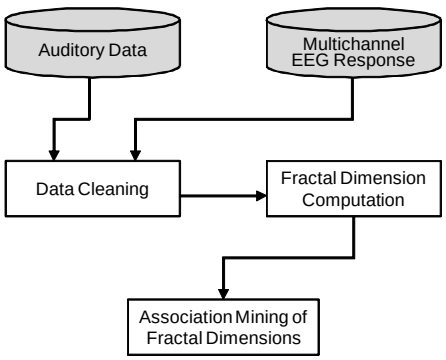


Fig. 1: The workflow for analyzing the EEG responses to auditory stimuli.

3.2 Data Collection

Our data collection involved ten healthy adult female subjects who were exposed to eight 2-minute long synthetic music pieces with varying degrees of fractal dimensions in random order. In order to keep all the parameters constant except for the fractal dimensions, a software package called FractMus[1] was used to generate the music pieces with different degrees of randomness, although all were composed in natural minor mode (one of the common modes local population is used to hearing in Irish folk songs) with three different pitches of electronic piano sound in 8 beats, 16 beats, and 32 beats, respectively. Throughout the music-listening task, eight-channels of EEGs based on the International 10-20 Method were measured using Biopac system at frontal lobe (F3, F4, F7, & F8) and at temporal lobe (T3, T4, T5, & T6). The channels F3, F4, F7, and F8 were selected in order to identify the emotional balance and function and the channels T3, T4, T5, and T6 were selected in order to identify the brain activity associated with sound processing function, respectively.

3.3 Data Analysis

The music files were generated in Waveform Audio File Format (WAV) and EEG data were generated in ASCII format. The WAV format was used because of its high quality, but the WAV files contain many features that are not relevant to the fractal analysis task. Both the music and EEG datasets were converted into a set of vectors using MATLAB. Each music piece was divided into four segments (30 seconds each). This created a total of thirty two 30-second long audio segments. For each audio segment, eight EEG channels were recorded; resulting in 256 EEG signals for each subject.

Fig. 2(a) shows the waveform of a 30-second music segment and Fig. 2(b) shows the waveform of this segment zoomed into the one second interval between 15 and 16 seconds. Figs. 2(c) and 2(d) show the EEG signals measured for a randomly selected subject at channels F3 (left frontal lobe) and T6 (right temporal lobe) in response to this music segment. The graphs clearly show the fractal aspects of these complex time series data. It is also evident that different EEG patterns are produced at different channels.

Our goal was to generalize any association that may exist between the music stimuli and the resulting EEG. Each audio and EEG signal was analyzed using the FracLab toolbox of MATLAB. The *regularization dimension* was computed for each 30-second music segment and for the corresponding EEG signals from eight channels. Fig. 3 shows the fractal dimensions computed from all eight music pieces with each music piece split into 30-second segments. It can be seen that different fractal patterns exist in the music pieces.

[1] http://www.gustavodiazjerez.com/fractmus_overview.html

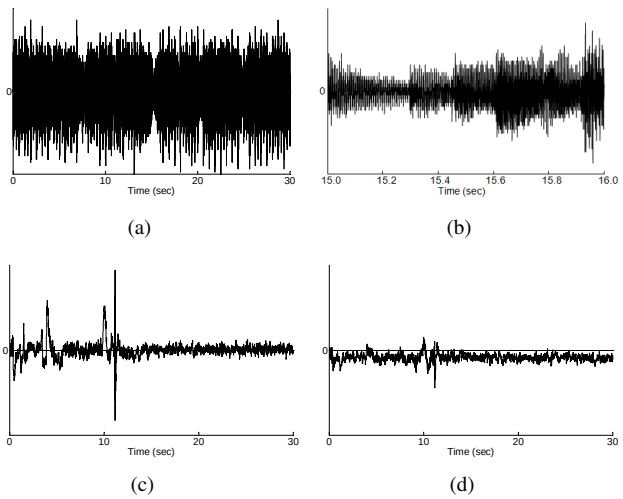


Fig. 2: The waveform of a 30-second music segment and the resulting EEG signals measured for a randomly selected subject - (a) The 30-second music segment, (b) Zoomed into one second, (c) EEG at F3, and (d) EEG at T6.

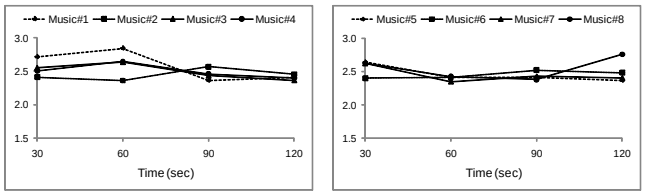
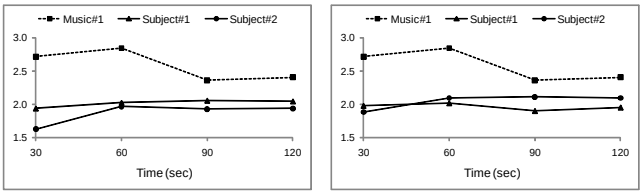


Fig. 3: The fractal dimensions of eight music pieces with each music piece split into 30-second segments.

Fig. 4 shows the fractal dimensions of EEG signals measured for two randomly selected subjects in response to music#1. Fig. 4(a) shows the EEG fractal dimensions for channel F3 and Fig. 4(b) shows the EEG fractal dimensions for channel T6. It can be seen that different subjects respond to the same music stimuli in different ways and the same individual exhibit different patterns at different channels. Therefore, it is not possible to discover any relationship between music stimuli and EEG responses using linear regression methods. That is why we chose to apply data mining to discover any such relationship.

After the fractal analysis was completed, data mining was performed on the fractal dimensions. The data mining package Weka[2] was used for the data mining task. First, the fractal dimensions were used to create Weka formatted files. Since the fractal dimensions are continuous real values, these values were converted into discrete categories. The unsupervised attribute discretizer was used for this purpose. It is an entropy-based method that performs discretization using density estimation and computes the leave-one-out

[2]http://www.cs.waikato.ac.nz/ ml/weka



(a) Channel F3 (b) Channel T6

Fig. 4: The fractal dimensions of EEG signals (F3 and T6) measured for two randomly selected subjects in response to a complete music piece split into 30-second segments.

cross-validation log-likelihood of the fit. Five bins were created for the music fractal dimensions and two bins were created for the EEG fractal dimensions.

After the fractal dimensions were discretized, association mining was performed using the *Apriori* algorithm [10] to examine the associations between the fractal dimensions of music segments and the fractal dimensions of resulting multichannel EEG signals. The Weka implementation of *Apriori* was modified to generate rules that only have one antecedent and one consequent, i.e., rules of the form $X \Rightarrow Y$, where $|X| = |Y| = 1$. Moreover, we restricted the antecedent of each rule to consist of a music fractal dimension and the consequent to consist of an EEG fractal dimension. A minimum confidence value of 0.70 was used for association mining.

3.4 Results

The results of the association analysis is presented in Table 1. The first column represents the fractal dimensions of music that associate with the fractal dimensions of multi-channel EEG. The second column represents the fractal dimensions of the EEG with the EEG channel specified in third column. The last column represents the confidence values of the mined association rules. The subjects, who happened to be all female, demonstrated strong association between the fractal dimension of music and fractal dimension of EEG at various EEG channels. Among those, the strongest association was observed between music fractal dimensions in the range of $(2.447 - 2.546]$ and the EEG fractal dimensions at right temporal lobe (T4 and T6) in the range of > 1.854 and > 1.864 respectively. This was followed by reasonably strong associations between the same music fractal dimension range and EEG fractal dimensions of > 1.947 at left temporal lobe (T5) and EEG fractal dimensions of > 1.830 at right frontal lobe (F4) (confidence levels $\geq 88\%$). Strong association discovered in females is an implication that the further research into the association between the fractal dimension of the sound stimuli and the fractal dimension of the EEG may give us a new insight into the selection of effective music in the context of music therapy as an alternative medicine.

Table 1: Results of association mining between the fractal dimensions of auditory stimuli and EEG responses

FD of Music	$\Rightarrow$	FD of EEG	EEG Channel	Confidence
(2.447-2.546]		$>$ 1.854	T4	0.98
(2.447-2.546]		$>$ 1.864	T6	0.94
$\leq$ 2.447		$>$ 1.864	T6	0.93
(2.447-2.546]		$>$ 1.947	T5	0.92
$\leq$ 2.447		$>$ 1.854	T4	0.9
$\leq$ 2.447		$>$ 1.830	F4	0.89
(2.447-2.546]		$>$ 1.881	T3	0.88
(2.447-2.546]		$>$ 1.830	F4	0.88
$\leq$ 2.447		$>$ 1.947	T5	0.87
$\leq$ 2.447		$>$ 1.881	T3	0.86
(2.546-2.644]		$>$ 1.854	T4	0.85
(2.546-2.644]		$>$ 1.864	T6	0.85
$\leq$ 2.447		$>$ 1.927	F3	0.8
(2.546-2.644]		$>$ 1.881	T3	0.8
(2.546-2.644]		$>$ 1.947	T5	0.8
(2.447-2.546]		$>$ 1.927	F3	0.8
(2.546-2.644]		$>$ 1.830	F4	0.76
$\leq$ 2.447		$>$ 1.793	F8	0.7

4. Conclusions

In the present study, we analyzed the fractal dimensions of auditory stimuli and the resulting multi-channel EEG responses. A robust level of correlation between the fractal dimension of the auditory stimuli and the fractal dimension of the EEG was established in female test subjects via association mining. These results had significance at two different levels. First, it implied a promising future of the application of nonlinear analysis of the time-series waveforms in the field of human electrophysiology. Further, it also suggested a significant mathematical association between auditory stimuli in the environment and physiological process in the human body. These implications of our study suggested the importance of further investigations in these two areas. Aside from the needs for further investigation on the significance of the fractal dimension of EEG itself, it is also important to examine the generalizability of present study results to male subjects as well as the generalizability to the use of other forms of auditory stimuli in order to examine the general principles of fractal dimension range of auditory stimuli that produces strong association with a certain range of electrophysiological process in human body. Although the present study was meant to be one case study of the application of data mining methods, the results suggested noteworthy implications in the direction of future research areas in human physiology.

References

[1] Q. Yuan, X. Liu, D. Li, H. Wang, and Y. Liu, "Effects of noise and music on EEG power spectrum," *Space Medicine & Medical Engineering*, vol. 13, no. 6, pp. 401–406, 2000.

[2] N. Jausovec, K. Jausovec, and I. Gerlic, "The influence of mozart's music on brain activity in the process of learning," *Clinical Neurophysiology*, vol. 117, no. 12, pp. 2703–2714, 2006.

[3] J. Srinivasan, A. Kumar, and V. Balasubramanian, "Cognitive effect of music for joggers using EEG," in *Proceedings of 13th International Conference on Biomedical Engineering (ICBME 2008), Singapore, Dec 3-6*, 2008, pp. 1120–1123.

[4] Y.-P. Lin, C.-H. Wang, T.-P. Jung, T.-L. Wu, S.-K. Jeng, J.-R. Duann, and J.-H. Chen, "EEG-based emotion recognition in music listening," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 7, pp. 1798–1806, 2010.

[5] B. Mandelbrot, *The Fractal Geometry of Nature*. New York, NY: W.H. Freeman, 1983.

[6] M. Schroeder, *Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise*. W.H. Freeman and Company, NY, 1991.

[7] M. Bigerelle and A. Iost, "Fractal dimension and classification of music," *Chaos, Solitons & Fractals*, vol. 11, no. 14, pp. 2179–2192, 2000.

[8] D. Easwaramoorthy and R. Uthayakumar, "Improved generalized fractal dimensions in the discrimination between healthy and epileptic eeg signals," *Journal of Computational Science*, vol. 2, no. 1, pp. 31–38, 2011.

[9] F. Roueff and J. L. Vehel, "A regularization approach to fractional dimension estimation," in *Proceedings of Fractals 98, Malta, October, 1998*.

[10] R. Agrawal and A. Swami, "Fast algorithms for mining association rules," in *Proceedings of 20th International Conference on Very Large Databases (VLDB 1994), Santiago, Chile, September 12-15*, 1994, pp. 487–499.

[11] P. Grassberger, "Generalized dimensions of strange attractors," *Physics Letters A*, vol. 97, no. 5, pp. 227–230, 1983.

[12] S. Gunasekaran and K. Revathy, "Fractal dimension analysis of audio signals for indian musical instrument recognition," in *Proceedings of International Conference on Audio, Language, and Image Processing (ICALIP 2008), Shanghai, China, Jul 7-9*, 2008, pp. 257–261.

[13] A. Das and P. Das, "Fractal analysis of songs: Performer's preference," *Nonlinear Analysis: Real World Applications*, vol. 11, no. 3, pp. 1790–1794, 2010.

[14] H. Preissl, W. Lutzenberger, F. Pulvermüller, and N. Birbaumer, "Fractal dimensions of short eeg time series in humans," *Neuroscience Letters*, vol. 225, no. 2, pp. 77–80, 1997.

[15] A. Jacquin, E. Causevic, and E. R. John, "Automatic identification of spike-wave events and non-convulsive seizures with a reduced set of electrodes," in *Proceedings of 29th International Conference of IEEE Engineering in Medicine and Biology Society (EMBS 2007), Lyon, France, Aug 22-26*, 2007, pp. 1928–1931.

[16] I. Chouvarda, V. Rosso, M. Mendez, A. Bianchi, L. Parrino, A. Grassi, M. Terzano, and S. Cerutti, "Assessment of the EEG complexity during activations from sleep," *Computer Methods and Programs in Biomedicine*, 2010.

[17] J. Bhattacharya, H. Petsche, and E. Pereda, "Interdependencies in the spontaneous eeg while listening to music," *International Journal of Psychophysiology*, vol. 42, no. 3, pp. 287–301, 2001.

[18] S. I. Ito, Y. Mitsukura, K. Sato, S. Fujisawa, and M. Fukumi, "Study on association between user's personality and individual characteristic of left prefrontal pole EEG activity," in *Proceedings of 6th International Conference on Natural Computation (ICNC 2010), Shandong, China, Aug 10-12*, 2010, pp. 2163–2166.

[19] Z. Feng, M. J. Zuo, and F. Chu, "Application of regularization dimension to gear damage assessment," *Mechanical Systems and Signal Processing*, vol. 24, no. 4, pp. 1081–1098, 2010.

Monte Carlo Stochastic programming applied to Asset Allocation

Gavriel Yarmish
Department of CIS
Brooklyn College
Room 2109 N
Brooklyn, NY 11210 USA
718 951 5000 ext 2071
yarmish@sci.brooklyn.cuny.edu

Harry Nagel
Department of CIS/DS
Tobin College of Business
St. John's University
Jamaica, NY 11439 USA
718 990 7368
nagelh@stjohns.edu

Robert Fireworker
Department of CIS/DS
Tobin College of Business
St. John's University
Jamaica, NY 11439 USA
718 390 4517
fireworr@stjohns.edu

Abstract *In this paper we describe applications of Monte Carlo methods to stochastic asset allocation models. The application of robust Monte Carlo methods to asset allocation can be extremely useful both to individuals and more importantly to fund managers. Considering the loss of equity in many people's pensions over the last few years, this is an important topic for fund managers.*

Keywords Linear programming, asset allocation, stochastic, Monte Carlo

1. Overview

The Asset Allocation Problem considers the question of how a portfolio should be weighted with different security assets in order to satisfy an investor's objective.

The idea is to invest and divide up the portfolio in a way to maximize profit while remaining within given constraints. Unfortunately, very often the assumptions made when setting up a mathematical program are exactly that – assumptions. When those assumptions turn out to be mistaken there is often a hefty price to be paid. One method that is used is Monte Carlo analysis. We simulate many possible scenarios and finally, solve the mathematical program in the context of expected value.

We organize this paper as follows. Section 2 is a review of a basic mathematical program, in this formulation, linear. Section 3 expands the formulation to stochastic programming, describes the method of applying Monte Carlo simulation and shows how to apply the scenarios of the Monte Carlo simulation to mathematical programming. Section 4 mentions a number of other applications and section 5 is a summary.

2. Mathematical Program Formulation

A linear program is of the matrix form:

$$\begin{aligned} &\textit{Minimize} \quad c^T x \\ &\textit{Subject to} \quad Ax \geq b,\ x \geq 0 \\ &\textit{where } c, x, A \textit{ and } b \textit{ are vectors} \end{aligned}$$

Fig 1

Expanded for two variables x_1 and x_2:

$$\begin{aligned} &\textit{Minimize} && c_1x_1 + c_2x_2 \\ &\textit{Subject to} && a_{11}x_1 + a_{12}x_2 \geq b_1 \\ & && a_{21}x_2 + a_{22}x_2 \geq b_2 \quad x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Fig 2

Where x_i would be the number of the i^{th} security to purchase, c_i is the cost of purchasing the i^{th} security, b_j is the total dollar amount of the liability obligation for year j and a_{ij} is the cash flow that security i will give in year j.

The idea of this linear program is to seek the minimum total cost c_i and still satisfy our obligations b_j. This objective will be minimized by picking securities with the highest cash flows a_{ij}.

3. Stochastic Linear Programs

Unfortunately, it is often the case that either the objective function coefficients c, the right hand side values b or the coefficients a are not known with certainty but are only known within a probability distribution. As an example, say an insurance company uses one of the values to represent expected cash available for investments net insurance claims. In that case it may very well turn out that net insurance claims were not as expected which would render the assumption of the program incorrect.

This leaves the practitioner with a dilemma if the numbers turn out to be wrong. A good method of dealing with this is to utilize a probability distribution based on past performance and to simulate numerous values for the uncertain group of numbers using that probability distribution.

Figure 3 is the formulation of the stochastic program that corresponds with the linear program in figure 1.

$$\begin{aligned} &\textit{Minimize} && E[c_S^T x_S] \quad \forall s \\ &\textit{Subject to} && A_S x_S \geq b,\ x_S \geq 0 \quad \forall s \\ &\textit{where } c, x, A \textit{ and } b \textit{ are vectors and} \\ &S \textit{ means Monte Carlo scenarios} \end{aligned}$$

Fig 3

As an example let us assume that both b_1 and b_2 of figure 2 are known but that the four values a_{ij} and the two values c are not known with certainty but, instead, follow a probability distribution. We then generate numerous scenarios. For this example we generate four scenarios but keep in mind that in real situations we may generate thousands of scenarios.

Scenario 1: $\{c_{1,1}\ c_{2,1}\ a_{11,1}\ \ a_{12,1}\ a_{21,1}\ a_{22,1}\}$
Scenario 2: $\{c_{1,2}\ c_{2,2}\ a_{11,2}\ \ a_{12,2}\ a_{21,2}\ a_{22,2}\}$
Scenario 3: $\{c_{1,3}\ c_{2,3}\ a_{11,3}\ \ a_{12,3}\ a_{21,3}\ a_{22,3}\}$
Scenario 4: $\{c_{1,4}\ c_{2,4}\ a_{11,4}\ \ a_{12,4}\ a_{21,4}\ a_{22,4}\}$

Figure 4 shows what the mathematical program from figure 3 would look like when utilizing the four generated scenarios. Note that there are now four groups of constraints corresponding to the four scenarios. The objective function now minimizes the expected value over the probability distribution. In this simple example we only generated 4 scenarios – the more scenarios generated the more realistic the problem but the larger the problem grows. Within these constraints we minimize the expected cost over these scenarios.

$$
\begin{array}{lllll}
\textit{Minimize} & E\left[c_{1,1}x_{1,1}+c_{2,1}x_{2,1} \quad c_{1,2}x_{1,2}+c_{2,2}x_{2,2} \quad c_{1,3}x_{1,3}+c_{2,3}x_{2,3} \quad c_{1,4}x_{1,4}+c_{2,4}x_{2,4}\right] & & & \\
 & a_{11,1}x_{1,1}+a_{12,1}x_{2,1} & & & \geq b_1 \\
 & a_{21,1}x_{2,1}+a_{22,1}x_{2,1} & & & \geq b_2 \\
 & a_{11,2}x_{1,2}+a_{12,2}x_{2,2} & & & \geq b_1 \\
\textit{Subject to} & a_{21,2}x_{2,2}+a_{22,2}x_{2,2} & & & \geq b_2 \\
 & & a_{11,3}x_{1,3}+a_{12,3}x_{2,3} & & \geq b_1 \\
 & & a_{21,3}x_{2,3}+a_{22,3}x_{2,3} & & \geq b_2 \\
 & & & a_{11,4}x_{1,4}+a_{12,4}x_{2,4} & \geq b_1 \\
 & & & a_{21,4}x_{2,4}+a_{22,4}x_{2,4} & \geq b_2
\end{array}
$$

$$x_{i,s} \geq 0$$

Fig 4

4. Other Stochastic Programs

Stochastic programs have been used for many applications. Zenios [1991, 1994] modeled Mortgage-backed security portfolios. The Frank Russel Company modeled a large stochastic program for a large insurance Company [Cariño, Kent et al., 1994, 1998].

Mathematical programs have been used to model uncertainty at least as far back as the 60's and 70's. Reservoir systems [Houck et al, 1978], Planing models [Tintner and Raghavan, 1970], models for multi-national firms [Salmi, 1974; Fourcans and Hindelang, 1974], labor input decisions of a typical rural household with risky agricultural technologies, off-farm employment opportunities [Becker, 1990] and wind generation [Bloom, 2010] are other examples. More recently Moriggia et al [1998] discusses attempts to use parallel computers to help solve large stochastic programs for bond portfolio applications

5. Summary

In this article we described the use of Monte Carlo scenario generation as applied to mathematical programming. We first identify coefficients, right hand sides and objective function values that are not known with certainty. We then utilize their probability distributions to generate many scenarios of values. Once we have those scenarios the mathematical program is transformed into a stochastic program that takes all scenarios into account and solves by minimizing (or maximizing) the expected values over all scenarios.

References

April, Jay; Fred Glover, James Kelly "Risk analysis: OptQuest software tutorial: portfolio optimization for capital investment projects" ACM *SIGSIM Proceedings of the 34th conference on Winter simulation: exploring new frontiers* December 2002

Becker, H. "Labour Input Decisions of Subsistence Farm Households in Southern Malawi" *Journal of Agricultural Economics*, v. 41, iss. 2, pp. 162-71 May 1990

Bloom, Jeremy "Integrating Wind Generation into the Power Grid" *International Conference on Stochastic Programming.* 2010.

Cariño, D. R., T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, and W. T. Ziemba. "The Russell-Yasuda Kasai Model: An Asset/Liability Model for a Japanese Insurance Company Using Multistage Stochastic Programming" Interfaces 24 (1): 29-49. 1994

Cariño, D.R. and W.T. Ziemba. "Formulation of the Russell Yasuda Kasai Financial Planning Model" Operations Research 46, 433-449 1998.

Fourcans, Andre, Thomas J. Hindelang "Working capital management for the multinational firm: A simulation model" *ACM SIGSIM Winter Simulation Conference Proceedings of the 7th conference on Winter simulation* Volume 1 pp. 141-149 1974

Houck, Mark Hedrich; Cohon, Jared L. "Sequential Explicitly Stochastic Linear Programming Models: A Proposed Method for Design and Management of Multipurpose Reservoir Systems" *Water Resources Research*, v. 14, iss. 2, pp. 161-69 April 1978

Markowitz, Harry M., "Portfolio Selection: Efficient Diversification of Investments," New York: John Wiley & Sons, 1959.

Moriggia, Vittorio; Marida Bertocchi, Jitka Dupaková "Highly parallel computing in simulation on dynamic bond portfolio management" *ACM SIGAPL Proceedings of the APL98 conference on Array processing language*, Volume 29 iss. 3 July 1998

Salmi, Timo "Joint Determination of Trade, Production, and Financial Flows in the Multinational Firm; A Stochastic Linear Programming Model Building Approach" *Liiketaloudellinen Aikakauskirja*, v. 23, iss. 3, pp. 222-37 1974

Tintner, Gerhard; Raghavan, N. S. "Stochastic Linear Programming Applied to a Dynamic Planning Model for India" *Economia Internazionale*, v. 23, iss. 1, pp. 105-17 Feb. 1970

Zenios, Stavros A., "Parallel Monte Carlo Simulation of Mortgage Backed Securities," *in Financial Optimization*, Cambridge University Press, 1991.

Viscosity of Suspensions: A Theoretical Study

K. Alammar
Department of Mechanical Engineering, King Saud University, Riyadh, Saudi Arabia

Abstract - *In this work, Einstein's formula for the viscosity of suspensions is revised to include effect of particle interactions. The revision is based on assumption of linear behavior of particle interactions with distance between particles. The resulting formula is tested over a range of particle size and concentrations. Good agreement is attained between theory and previously published measurements*.

Keywords: Einstein formula; reheology; viscosity; suspension; nanofluid

1 Introduction

Particle suspensions can change properties of base fluids, leading to great potential for enhancements. Previously, using micro-scale particles led to pipe erosion and pump damage, as well as sedimentation and fouling. However, recent advances in materials technology has made it feasible to produce nanoparticles that can overcome such limitations. Nanoparticles can have better properties than conventional materials, including thermal, mechanical, and electrical [6]. Fluids suspended by nanoparticles have been termed nanofluids [3]. Comprehensive reviews on the research on nanofluids are found in [3, 4, 5].

Due to diverse and important applications in industry, reheology of suspensions has been a subject of research for many years. Early in the twentieth century, Einstein [7] presented his famous formula, eq. (1), for calculating the effective viscosity of dilute suspensions.

$$\mu_r = 1 + c\nu \tag{1}$$

His formula suggests that the increase in viscosity is independent of the size or material of the particles, and was derived with the underlying assumption of spherical particles and no particle interaction. As such, the equation gives good results for low concentrations when particle interaction effects are negligible.

It is expected that increased particle interactions give rise to higher viscosity. The interactions are greater when particles are closer to each other. Increasing the concentration brings particles closer to each other, and hence lead to higher viscosity than predicted by Einstein formula. Moreover, and for a given concentration, smaller particles lead to closer particles, and hence higher viscosities. This is observed in viscosity measurements of suspensions over the years, e.g., Williams [1] and Williams et al. [2]. Particles used in Williams [1] were of the microscale, whereas those in Williams et al. [2] were of the nanoscale. Viscosity measurements in both experiments differ by orders of magnitudes.

In this work, Einstein [7] formula for viscosity of spherical suspensions is revised to include the effect of particle interactions. The revision is based on assumption of linear behavior of particle interactions with the distance between particles. The resulting equation is tested over a range of particle size and concentrations.

2 Theory

Particle interaction is a function of the distance between the particles. At sufficiently large distances, it is reasonable to assume a linear behavior. As such, one can determine the particle separation changes with volume concentration and particle size, and in turn introduce the effects in Einstein formula.

In three-dimensional space, it can be shown that the distance between particles is proportional to $\sqrt{\nu}$, where ν is the particle volume concentration. Accordingly, the relative viscosity could be revised as follows:

$$\mu_r = 1 + c\nu^{3/2} \tag{2}$$

Introducing the interaction factor as such is in line, for example, with theoretical development of Happel [8]. On the other hand, if the particle diameter is doubled for a given volume concentration, the distance between the particles would increase proportional to $d^{2/3}$, where d is the particle diameter. Accordingly, Eq. (2) would be revised further as follows:

$$\mu_r = 1 + c\nu^{3/2}(\delta / d)^{2/3} \tag{3}$$

Here, δ is a dimensional constant possibly attributed to the base fluid. Combining the two constants, we have

$$\mu_r = 1 + \nu^{3/2}(a / d)^{2/3} \tag{4}$$

Where a is a dimensional constant.

3 Results and Discussion

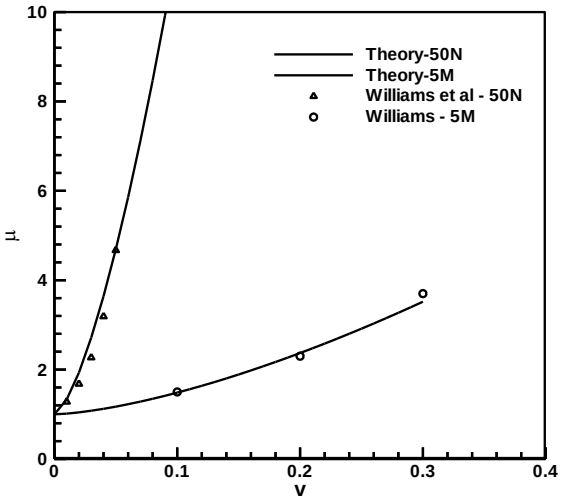


Figure 1: Relative viscosity as a function of particle volume concentration.

The relative viscosity as a function of particle volume concentration is depicted in Fig. 1. Measurements of Williams et al [2] were conducted for 50-nm particles. The base fluid was water. In the case of Williams [1], the particles were in the order of 5 microns and the base fluid was a mixture of water and glycerin glycol. Eq. (4) and measurements are in good agreement. The empirical constant a was determined to be 3.0e-4.

4 Conclusions

In this work, Einstein's formula for the viscosity of spherical particle suspensions was revised to include the effect particle interactions. The revision was based on assumption of linear behavior of particle interactions with the distance between particles. Good agreement between the revised theory and measurements was attained within the range tested.

5 References

[1] P. S. Williams, "Flow of Concentrated Suspensions "; J. App. Chem., 3, 120 (1953)

[2] W. C. Williams, J. Buongiorno, L. W. Hu, "Experimental Investigation of Turbulent Convective Heat Transfer and Pressure Loss of Alumina/Water and Zirconia/Water Nanoparticle Colloids (Nanofluids) in Horizontal Tubes"; J. Heat Transfer, Vol. 130, 042412, Apr 2008.

[3] W. Yu, D. France, S. Choi, J. Routbort, "Review and Assessment of Nanofluid Technology for Transportation and Other Applications"; Argonne National Laboratory Technical Report, ANL/ESD/07-9, (2007).

[4] V. Trisaksri and S. Wongwises, "Critical review of heat transfer characteristics of nanofluids. Renewable and Sustainable Energy Reviews. 11, 3: 512-523, (2007).

[5] S. Das, S. Choi, W. Yu, T. Pradeep, "Nanofluids: Science and Technology"; John Wiley & Sons, (2008).

[6] G. Schmid, "Nanoparticles from Theory to Application"; J. Wiley & Sons, Hoboken, (2004).

[7] A. Einstein, "Eine neue bestimmung der molekuldimensionen"; Ann. Physik. 19: 289, (1906).

[8] J. Happel, "Viscosity of Suspensions of Uniform Spheres"; Journal of Applied Physics, V. 28, No. 11: 1288-92, (1957,2006).

A Cloud Oriented Framework for Scientific Data Processing

R. A. Wasniowski

Computer Science Department
California State University, CA, 93012

Abstract - *Our work concerns the use of cloud computing for data processing. To study the problem, we designed a framework for experimenting with various cloud-computing arrangements. The main goal is a framework that has the ability to automatically track the configuration, providing event management, performance measurement and testing for small-scale cloud computing. We show how such a framework can be integrated and accessed in a practical manner using multiple data processing tools.*

Keywords: *Cloud Computing, Data Processing*

1 Introduction

In recent years, there has been an exponential growth in the amount of data that needs to be processed. This is especially true in the research teams where large data sets are routinely produced by experiments. As a result of this challenge, computation is rapidly moving toward cloud computing and the software industry's focus is shifting from developing applications for PCs to developing data centers and the cloud technology that enables millions of users to make use of software simultaneously. This is creating a huge demand for workers with skills in this area. Educational and research organizations require a platform that can support multiple models of application programming, multiple types of cloud deployments (private, public, or hybrid), and an extensible framework enabling educators/researchers to develop their own programming models and application.

2 Cloud Computing

In today's research and education increasingly vital role is played by technologies that enable efficient processing of large data sets. Current challenges for computing resources associated with processing large data sets far exceed the capacity of personal computers. To achieve satisfactory performance applications often need clusters with hundreds of nodes. In recent years, a new technology cloud computing emerged to manage a distributed storage and processing of large sets of data [2, 3, 7, 19-22]. The basic principle of cloud computing involves the idea that users do not need to run the required applications on their individual personal computers; the applications can instead be run by a server running on a cloud.

In this article, we provide a brief introduction to cloud computing technology and Platform as a Service, we examine the offerings in this category, and we provide the basis to help readers understand basic application platform opportunities in the cloud, including Microsoft Azure, Sales Force, and Google Apps. Cloud environments can be classified as public, private, or hybrid, depending on the model of deployment [7]. A public cloud environment is made available in a pay-as-you-go manner to the general public. A private cloud environment is a data center of an organization that is not made available to the general public. A hybrid cloud environment involves the seamless use of a public cloud environment combined with a private cloud, when needed. In a typical public cloud environment scenario, a third-party vendor delivers services, such as computation, storage, networks, virtualization and applications, to various customers. In a private cloud environment, internal information technology resources are used to serve their internal users and customers. Businesses are adopting public cloud services to reduce capital expenditures and operational costs by leveraging the cloud's elastic scalability and market-oriented cost features. What makes cloud computing different from traditional approaches is the focus on service delivery and the consumer utilization model.

Companies that have been involved in the development of cloud computing include Amazon, Google, Microsoft, IBM, Oracle, Yahoo and other companies. Amazon's cloud

computing services are collectively called Amazon Web Services. Google's cloud computing platform makes use of Google's scale to provide a cloud-computing infrastructure for distributed computing. It provides developers with integration of the host server and can automatically update the online application service. For user-written applications, Google provides operation and maintenance support of the required application for all of the platform resources. Microsoft's cloud computing strategy is "software plus services". Microsoft launched the Windows Azure operating system [31]. The main objective of Windows Azure is to provide a platform for developers to develop the server that is running in the cloud, data center, Web, and PC on the application. IBM announced the "Blue Cloud" program [32], a computing plan that involves building large-scale distributed computing using IBM's expertise, hardware, software, technical support and service support of open standards and open-source software.

Cloud computing architecture is generally grouped into the following three service categories [7]:

(1) Infrastructure as a Service (IaaS), which offers basic storage and computing capabilities, including rapid provisioning of resources, i.e., software, hardware, the ability to scale those resources and pay-per-use convenience;

(2) Software as a Service (SaaS), which provides the ability to access software over the Internet; and

(3) Servers, which involves the ability to secure one or more servers.

3 Framework

In the practice of teaching a course on Distributed Computing the author has gained experience of using the technology in question by developing a set of experiments for academic applications. The main goal of this course was to develop practical skills using modern technology, parallel and distributed computing, mostly in solving problems related to data processing and analysis. One of the problem was to develop practical skills in working with large data sets on a platform of Hadoop [23]. The next task was intended to reuse practical skills to work with Hadoop platform to implement a non-trivial applications for processing and analyzing large data sets. MapReduce [28] programs can be implemented in various programming languages. However, in practice students prefer Java or Python languages. MapReduce [28] - the model for processing of large volumes of data, developed and used at Google for a wide range of applications. MapReduce model is straightforward and relatively easy to use, hiding from the user details of the algorithms on a cluster system. Distributed computing platform Hadoop [23] is developed within the organization Apache Software Foundation as open source. The platform is focused on support for processing large data sets on cluster systems.

We have tested the following providers: Amazon, PiCloud, Eucalyptus, and Cloudo.

Table 1: Cloud computing providers tested

Provider	Link
Amazon	aws.amazon.com
PiCloud	www.picloud.com
Eucalyptus	www.eucalyptus.com
Claudo	www.cloudo.com

The Amazon Elastic Compute Cloud (EC2) [19] is a web service that provides resizable computing capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon EC2 gives the user full control of their environments. The user can manage and obtain the resources that they need to have. Amazon EC2 is also a well-established provider with a number of features and services. Some of those services have the following characteristics:

- Elastic: Amazon EC2 enables the user to increase or decrease capacity in a matter of minutes rather than hours or days.
- Flexible: The user can choose multiple instance types, with different configurations, operating systems, software packages, etc.
- Reliable: The service runs within Amazon's proven network infrastructure and data centers.
- Completely Controlled: Users have complete control over their instances. User can save the instance and reboot it with Amazon Web Services.

PiCloud [20] allows the user to easily run any Python code on an auto-scaling, high-performance cluster. The main objective of PiCloud is offloading the computation from the user's computer onto the cloud. It handles the algorithm or the highly scalable web application. The user does not need to worry about server management as PiCloud handles the server management part.

The PiCloud Platform also offers an excellent management platform for tracking computations, analyzing performance on a function-by-function basis, and accessing a fully integrated technical suite. They have published a library that anyone can use. With the library, any function can be passed in with the parameters. This approach frees up computing resources on the local computer. Using the library is very straightforward.

The Eucalyptus Community Cloud [21] is a sandbox environment in which members can run trials and experiment with Eucalyptus, which is the software foundation for the cloud computing. Eucalyptus implements what is commonly referred to as Infrastructure as a Service.

The Cloudo [22] design is elastic in nature, which means it will only use the number of machines that it needs. You do not have to specify how many of them to use. It automatic does the scaling. Some of the benefits of Cloudo are listed below:

- You do not need to download any software that needs to be configured to connect to a private/public cloud.
- Because Cloudo is a web-based cloud-computing environment, it is accessible anywhere in the world.
- Cloudo provides an application programming interface (API) that allows the user to develop, maintain and share applications with other Cloudo users. Cloudo can be accessed from anywhere in the world.

Because Cloudo is web-based cloud IaaS, the end user will need a fast Internet connection; otherwise, it will take a very long time to load and configure the virtual operating system. Currently, Cloudo is still under development, so some of the JavaScript applications might not work, i.e., GUI-based apps. Cloudo is a true IaaS service provider. Cloudo provides an operating system with software preinstalled in an image and scalable hardware resources. Cloudo also provides a JavaScript application framework for cloud application development.

For small research groups, particularly those in an academic environment, computer-intensive labs, such as those containing clusters and grids, are becoming expensive to maintain and support. Many universities have started infrastructure and administrative expense and time and improve hardware utilization.

We discuss how cloud-oriented technologies can be used to implement distributed, reconfigurable small data centers and services to support research in academic institutions. The physical heart of the system is the server, which integrates other servers, network-based storage and software applications into effective systems.
A typical user accesses the system using a web-based interface. After the appropriate authentication and validation steps, the user is presented with a set of menu options. The user accesses the system through a web interface to select a combination of applications and services he or she needs. If a specific combination is not available as an image, an authorized user can construct his or her own image from available components. The web interface provides privileged users the ability to grant varying levels of control to other users and provides a method to schedule the resources in the pool. The web Interface is developed using an open-source Apache web server.

Once the user enters his or her login and password, the user is presented with a Linux-based virtual machine image that is loaded into the browser.

4 Experiments

Researchers, students and many others are often running experiments and evaluating results of those experiments. Although the situations differ from one domain to another, the standard procedures are generally the same: run various experiments, repeat them, sometimes with slight variations of data, try a new collection of parameters. A lot of time is usually spent on implementing and refining the experiments. The cost to implement the appropriate experiment, especially when dealing with developing new algorithms or dealing with large sets of data is crucial. Therefore, it is necessary to develop and use a tool that takes care of the whole configuration of the experiment and process computationally intensive problems.

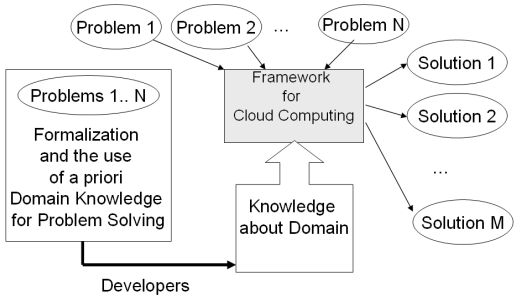


Fig 1: Framework used

In the subsequent section, we describe how to integrate the experiment with cloud computing, how to set up experiments and demonstrate ways to evaluate results. We select Python as a programming language, because it offers attractive packages especially suited for scientific data processing such as numpy and scipy [29, 30].

Table 2: Modules used

Module	Implementation
Neural nets	Python
Polynomial nets	C/Python
Semantic nets	Java
Image processing	C/C++, Python
Data Fusion	C/Python
SVM classifier	C, Python
Particle filter	C
Potts net	C/Python

4.1 Amazon EC2

Once an account is set up, the user can invoke a web application service called the AWS Management Console and create cloud instances. (We select the Amazon Linux.) Once the instance is created, the user can log into the instance using ssh from another linux/unix machine. We used Mac, and scp to copy files back and forth from the local Mac machine and the EC2 cloud.

The test program that we selected was a multistage version of the Ensemble Kalman filter [26] and Polynomial filter [35]. We created kalman() and poly() which are run from inside of a thread. We wanted to be able to run a multi-threaded program on an instance of Linux on EC2 and compare that to the performance of a Mac. The results of the EC2 proved to be faster than the local machine for this test.

4.2 PiCloud

Then in the Python, you import the cloud, set the keys to the cloud, define the function and assign it a job id and afterward you can call the function from the cloud and see the result. When running the same calculation on the PiCloud, we did 1000 iterations for kalman() almost instantly. The Picloud interface is neat in that you can see the jobs that are processing and refresh to see if the jobs finish. We then updated calculation by a factors of 10, 100, 1000 etc. At 100,000 iterations it took about 10-15 seconds to complete, and at 1,000,000 iterations about 4 minutes to complete. Running this function on the PiCloud is much faster then running it on a local Mac alone as one would expect it to be. These examples are relatively simple, but ideal ways to highlight usefulness of cloud computing particularly when teaching distributed computing. It is worth to note that one can use Numpy, Scipy on PiCloud account.

4.3 Eucaliptus, Cloudo

There are several other experiments under development such as binary neural Potts type of nets, radial basis network, simulated annealing, optimization and search, patterns classification using Cloudo and Eucaliptus [17].

5 Conclusions

The article briefly reviews the new technology of processing of large data sets using cloud computing, and describes the experience in applying these technologies. As cloud computing technology becomes an integral part of modern data processing there is a need to develop frameworks for academic environments aimed at storing and analyzing large amounts of data. We have presented a scalable framework for cloud-oriented data processing. The framework exploits cloud-oriented properties to achieve better scalability. Many data analysis tasks can be broken into multiple subtasks and executed in the cloud. The proposed framework is domain independent. We are planning to extend the framework to other tasks, such as data mining and data retrieval.

6 References

[1] Bolle D., Dupont P. & van J. Mourik. Stability properties of Potts neural networks with biased patterns and low loading. Journal of Physics A, 24, 1065-1081 1991.

[2] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, Elsevier, The Netherlands, June 2009.

[3] Cloudo: The Pretty WebOS Formerly Known As Xindesk byDucan Riley Feb 22, 2008.

[4] J. Callan, Distributed information retrieval, In W.B. Croft, editor, Advances in information retrieval, chapter 5, pages 127-150, Kluwer Academic Publishers, 2000.

[5] D. Chappell, Introducing the Windows Azure Platform, David Chappell and Associates, October 2010.

[6] Cook J. The mean-field theory of a Q-state neuralnetwork model. Journal of Physics A, 22, 2000-2012, 1989.

[7] Cloud Computing, Principles and Paradigms, Edited by Rajkumar Buyya, James Broberg, Andrzej Goscinski, John Wiley & Sons, 2011.

[8] Dean, J. and Ghemawat, S. MapReduce, Simplified data processing on large clusters. In Proceedings of Operating Systems Design and Implementation (OSDI). San Francisco, CA 2004.

[9] Dovey, M. J. (2002). Music GRID: A Collaborative Virtual Organization for Music Information Retrieval Collaboration and Evaluation. In the MIR/MDL Evaluation Project White Paper Collection (2nd ed., pp. 50--52), Champaign, IL: GSLIS.

[10] G. Fox, D. Gannon, and M. Thomas, "Editorial: A Summary of Grid Computing Environments." Concurrency and Computation: Practice and Experience, Vol. 14, No. 13- 15, pp. 1035-1044, 2002.

[11] Ghemawat, S., Gobioff, H., and Leung, S.-T. The Google file system. In 19th Symposium on Operating Systems Principles, Lake George, NY, 2003.

[12] S Ghemawat and J Dean, MapReduce: Simplified Data Processing on Large Clusters, Proceedings of the 6 Symposium on Operating System Design and Implementation (OSDI'04), San Francisco, CA, USA, 2004.

[13] Ido Kanter. Potts-glass models of neural networks. Physical Review A, 1988, v.37 (7), pp. 2739-2742.

[14] Jithesh Moothoor, Vasvi Bhatt, A Cloud Computing Solution for Universities: Virtual Computing Lab., IBM 2009.

[15] Yangwoo Kim, Grid Information Retrieval System for Dynamically Reconfigurable Virtual Organization, Memo for Grid Information Retrieval Working Group (GIR-WG).

[16] Vogt H., Zippelius A. Invariant recognition in Potts glass neural networks. Journal of Physics A, 25, 2209-2226, 1992.

[17] R. Wasniowski, Cloud Computing, TR, 2010.

[18] R. Yan, M.-O. Fleury, M. Merler, A. Natsev, and J. R.Smith. Large-scale multimedia semantic concept modeling using robust subspace bagging and MapReduce. In LS-MMRM'09, 2009.

[19] http://aws.amazon.com/ec2/

[20] http://www.picloud.com/faq/

[21] http://www.eucalyptus.com/

[22] http://www.cloudo.com/press.htm

[23] http://hadoop.apache.org/

[24] http://www.globus.org/toolkit/

[25] http://en.wikipedia.org/wiki/Support_vector_machine

[26] http://en.wikipedia.org/wiki/Kalman_filter

[27] http://en.wikipedia.org/wiki/Cloud_computing

[28] http://mapreduce.org/

[29] http://numpy.scipy.org/

[30] http://www.scipy.org/

[31] http://www.microsoft.com/windowsazure/

[32] http://www.ibm.com/cloud-computing/us/en/

[33] http://www.oracle.com/us/technologies/cloud/index.htm

[34] http://labs.yahoo.com/Cloud_Computing

[35] http://en.wikipedia.org/wiki/GMDH

Fourier-Legendre Spectral Method for Spherical Advection Equation with Solid-Body-Rotation Flow

Hyeong-Bin Cheong[1] and Ja-Rin Park[2]
([1]hbcheong@pknu.ac.kr, [2]jrpark@pknu.ac.kr)

Department of Environmental Atmospheric Sciences, Pukyong National University
(599-1 Daeyeon-3-dong, Namgu, Busan 608-737, Korea)

Abstract

A spectral method to solve the advection equation on the sphere with solid-body rotation flow is described. Dependent variables are expanded with spherical harmonics which are represented with Fourier series both in zonal and meridional directions. Once the initial fields are transformed into 2D Fourier space, the advection equation can be time-integrated without transforming spectral coefficients back to grid space. The operation count is $O(N^3)$ per one time-stepping for $O(N^2)$ spectral coefficients. The accuracy was found comparable to the spectral transform method using spherical harmonics.
Keywords: Spherical advection equation, Fourier series, spherical harmonics, Fourier representation of Legendre function, solid-body rotation

1 Introduction

Advection equation with a constant velocity is the simplest partial differential equation describing linear movement of passive scalar variable [e.g., 1]. In the spherical surface, the constant velocity should be replaced by solid-body rotating basic flow. By the nature of spherical coordinate system, the advection equation includes a singular term for zonal advection term, and thus requires a special numerical treatment. Spherical harmonics spectral-transform method is known to provide robust and accurate solutions for this problem via transform method in which spectral coefficients are transformed into grid space and vice versa at every timestep [1, 2]. In this study, a spectral method which does not require transforms between wave and grid space during time-stepping is proposed for the advection equation with rigid-body rotation.

2 Solid-body rotation advection equation and spherical harmonics analysis with double Fourier series

Advection equation on the unit-radius sphere (scaled by Earth's rotation rate and radius) is written as

$$\frac{\partial h}{\partial t} = -\frac{u}{\sin\phi}\frac{\partial h}{\partial \lambda} - v\frac{\partial h}{\partial \phi}\,, \tag{1}$$

where t means time, λ is the longitude, $\phi = \text{latitude} + \pi/2$, h is a scalar variable, and u and v are longitudinal and latitudinal velocities, respectively. The streamfunction and velocity components of a rigid-rotation are specified as

$$\begin{array}{l} \psi \;=\; \psi_1^0 \cos\phi + \psi_1^{-1} \sin\phi\cos\lambda + \psi_1^1 \sin\phi\sin\lambda \\ u \;=\; \psi_1^0 \sin\phi - \psi_1^{-1} \cos\phi\cos\lambda - \psi_1^1 \cos\phi\sin\lambda \\ v \;=\; -\psi_1^{-1} \sin\lambda + \psi_1^1 \cos\lambda \end{array}, \tag{2}$$

where ψ_1^0, ψ_1^{-1}, and ψ_1^1 are constant spectral components of streamfunction of zonal-mean and zonal wavenumber-one. The axis of solid-body rotation is tilted to the north by the angle $\alpha = \arctan\left[\sqrt{(\psi_1^{-1})^2 + (\psi_1^1)^2}/\sqrt{(\psi_1^0)^2}\right]$, and the mean flow rotates eastward for $\psi_1^0 > 0$. If the scalar variable h is expanded with zonal harmonics as

$$\begin{array}{rl} h(\lambda,\phi) \;=\; \displaystyle\sum_{m=0}^{N} h_m^C(\phi)\cos m\lambda + \sum_{m=1}^{N} h_m^S(\phi)\sin m\lambda \\ h_m^C \;=\; a\int_0^{2\pi} h\cos m\lambda d\lambda \\ h_m^S \;=\; a\int_0^{2\pi} h\sin m\lambda d\lambda \end{array} \tag{3}$$

with $a = (2\pi)^{-1}$ for $m = 0$ and π^{-1} for $m > 0$, (1) can be written in terms of zonal wave components as

$$\begin{array}{l} \dfrac{\partial h_m^C}{\partial t} \;=\; A_1 + A_2 + A_3 + A_4 + A_5 \\ \dfrac{\partial h_m^S}{\partial t} \;=\; B_1 + B_2 \end{array}, \tag{4}$$

where $h_m^S(\phi)$ and $h_m^C(\phi)$ are zonal sine and cosine coefficients of h, and A_i and B_i are shown in Appendix. The meridional function in (4) is further expanded with half-ranged sine or cosine series:

$$h_m(\phi) = \begin{cases} \displaystyle\sum_{k=0}^{N} h_{m,k}\cos k\phi, & m = \text{even} \\ \displaystyle\sum_{k=1}^{N} h_{m,k}\sin k\phi, & m = \text{odd} \end{cases}, \tag{5}$$

where superscripts 'C' and 'S' as in (4) are dropped for simplicity.

3 Spectral equation and advection of Gaussian bell

The spherical harmonics coefficients ($\hat{h}_{n,m}$) associated with (4) are obtained by multiplying Legendre functions and integrating between poles:

$$\begin{array}{l} \dfrac{d}{dt}\hat{h}_{n,m}^C \;=\; A_1^S + A_2^S + A_3^S + A_4^S \\ \dfrac{d}{dt}\hat{h}_{n,m}^S \;=\; B_1^S + B_2^S + B_3^S \end{array}, \tag{6}$$

where A_i^S and B_i^S are tendencies for spectral coefficients as shown in Appendix. Tendencies in the equations A_i^S and B_i^S include some Fourier-related integrals defined as:

$$\begin{aligned}
S_{n,m,k} &= \begin{cases} \int_0^\pi P_n^m \cos k\phi \sin\phi d\phi & (m = \text{even}) \\ \int_0^\pi P_n^m \sin k\phi \sin\phi d\phi & (m = \text{ odd}) \end{cases} \\
D_{n,m,k} &= \begin{cases} \int_0^\pi P_n^m \sin k\phi d\phi & (m = \text{even}) \\ \int_0^\pi P_n^m \cos k\phi d\phi & (m = \text{ odd}) \end{cases} . \\
\bar{D}_{n,m,k} &= (D_{n,m,k+1} + D_{n,m,k-1})/2 \quad (k \geq 1) \\
\bar{D}_{n,m,0} &= \begin{cases} 0 & (m = \text{even}) \\ D_{n,m,1} & (m = \text{ odd}) \end{cases}
\end{aligned} \tag{7}$$

Detailed method to compute the Fourier-related integrals can be found in [3] where stable recursion equations of the associated Legendre functions [4, 5] were incorporated with a modification for normalization.

4 Result and Disccusion

The spectral equation (6) is time-integrated with initial condition of Gaussian bell function located at $(90°\text{E}, 0°\text{N})$. Eq. (6) includes no singular term related with $(\sin\phi)^{-1}$ because the Fourier integrals of (7) were used along with weighting factor $\sin\phi$, therefore, pole problem is not expected to be encountered with this method. The spherical harmonics obtained through time-integration should be transformed to Fourier-coefficients at every timestep. When (6) is formulated in matrix equations, only forward operation of matrices is required in this method. The solid-body basic flow was given to make the Gaussian bell rotate along the great-circle which has an angle of $\alpha = \pi/2 - 0.05$ from the poles. Examples of 12-day time-integration for $N = 200$ and the basic flow of $(\pi/6)$ rad/day are illustrated in Fig. 1, where a Gaussian bell of $e^{-200\sin^2(\theta/2)}$ is used as initial condition. Super rotation flow is specified as $\psi_1^0 = (1/12)\cos\alpha$, $\psi_1^1 = (1/12)\cos(\pi/2)\sin\alpha$, and $\psi_1^{-1} = (1/12)\sin(\pi/2)\sin\alpha$. Timestep size and Asselin time filter coefficient were given $1/120$ day and 0.01, respectively. Grid point values were obtained from the spherical harmonics coefficient with either Fourier method or Gauss Legendre. The height error by 12 day is extremely small, giving only maximum value of 0.03. The result demonstrates that the Fourier method can be used to time-integrate the advection equation even with full-grid system, which includes the South and North poles, without polar singularity.

Appendix

Zonal cosine (A_i) and sine (B_i) spectral coefficients of (4) are written as

$$A_1 = \begin{cases} \dfrac{\cos\phi}{2\sin\phi} \begin{Bmatrix} -\psi_1^1[(m+1)h_{m+1}^C(\phi) - (m-1)h_{m-1}^C(\phi)] \\ +\psi_1^{-1}[(m+1)h_{m+1}^S(\phi) + (m-1)h_{m-1}^S(\phi)] \end{Bmatrix} \\ +\dfrac{1}{2}\left[-\psi_1^1\left(\dfrac{\partial h_{m+1}^C}{\partial\phi} + \dfrac{\partial h_{m-1}^C}{\partial\phi}\right) + \psi_1^{-1}\left(\dfrac{\partial h_{m+1}^S}{\partial\phi} - \dfrac{\partial h_{m-1}^S}{\partial\phi}\right)\right] \end{cases} \quad (m \geq 2) \tag{A1}$$

$$A_2 = \frac{\cos\phi}{2\sin\phi}\left[-\psi_1^1(m+1)h_{m+1}^C(\phi) + \psi_1^{-1}(m+1)h_{m+1}^S(\phi)\right] \qquad (m = 0 \text{ or } 1)$$

(A2)

$$A_3 = \frac{1}{2}\left[-\psi_1^1\left(\frac{\partial h_{m+1}^C}{\partial\phi} + 2\frac{\partial h_{m-1}^C}{\partial\phi}\right) + \psi_1^{-1}\frac{\partial h_{m+1}^S}{\partial\phi}\right] \qquad (m = 1)$$

(A3)

$$A_4 = \frac{1}{2}\left[-\psi_1^1\frac{\partial h_{m+1}^C}{\partial\phi} + \psi_1^{-1}\frac{\partial h_{m+1}^S}{\partial\phi}\right] \qquad (m = 0)$$

(A4)

$$A_5 = -\psi_1^0 m h_m^S(\phi) \qquad (m \geq 1)$$

(A5)

$$B_1 = \begin{cases} \dfrac{\cos\phi}{2\sin\phi}\left\{\begin{array}{l} -\psi_1^{-1}[(m-1)h_{m-1}^C + (m+1)h_{m+1}^C] \\ +\psi_1^1[(m-1)h_{m-1}^S - (m+1)h_{m+1}^S] \end{array}\right\} + \psi_1^0 m h_m^C(\phi) \\ +\dfrac{1}{2}\left[\psi_1^{-1}\left(\dfrac{\partial h_{m-1}^C}{\partial\phi} - \dfrac{\partial h_{m+1}^C}{\partial\phi}\right) - \psi_1^1\left(\dfrac{\partial h_{m-1}^S}{\partial\phi} + \dfrac{\partial h_{m+1}^S}{\partial\phi}\right)\right] \end{cases} \qquad (m \geq 2)$$

(A6)

$$B_2 = \begin{cases} \dfrac{\cos\phi}{2\sin\phi}\left[-\psi_1^{-1}(m+1)h_{m+1}^C - \psi_1^1(m+1)h_{m+1}^S\right] + \psi_1^0 m h_m^C(\phi) \\ +\dfrac{1}{2}\left[\psi_1^{-1}\left(2\dfrac{\partial h_{m-1}^C}{\partial\phi} - \dfrac{\partial h_{m+1}^C}{\partial\phi}\right) - \psi_1^1\dfrac{\partial h_{m+1}^S}{\partial\phi}\right] \end{cases} \qquad (m = 1).$$

(A7)

The components of spherical harmonics coefficients (6) are given in terms of Fourier-related integrals as (the summation is for $k = 0, 1, \cdots, N$):

$$A_1^S = \frac{1}{2}\begin{cases} \psi_1^1\left[(m-1)\sum h_{m-1,k}^C \bar{D}_{n,m,k} - (m+1)\sum h_{m+1,k}^C \bar{D}_{n,m,k}\right] \\ +\psi_1^{-1}\left[(m-1)\sum h_{m-1,k}^S \bar{D}_{n,m,k} + (m+1)\sum h_{m+1,k}^S \bar{D}_{n,m,k}\right] \\ -2\psi_1^0 m\sum h_{m,k}^S S_{n,m,k} \end{cases} \qquad (m \geq 1)$$

(A8)

$$A_2^S = \frac{1}{2}\begin{cases} -\psi_1^1\left[\sum h_{m-1,k}^C(-1)^m k S_{n,m,k} + \sum h_{m+1,k}^C(-1)^m k S_{n,m,k}\right] \\ +\psi_1^{-1}\left[-\sum h_{m-1,k}^S(-1)^m k S_{n,m,k} + \sum h_{m+1,k}^S(-1)^m k S_{n,m,k}\right] \end{cases} \qquad (m \geq 2)$$

(A9)

$$A_3^S = \frac{1}{2}\begin{cases} -\psi_1^1\left[2\sum h^C_{m-1,k}(-1)^m kS_{n,m,k} + \sum h^C_{m+1,k}(-1)^m kS_{n,m,k}\right] \\ +\psi_1^{-1}\sum h^S_{m+1,k}(-1)^m kS_{n,m,k} \end{cases} \quad (m=1) \tag{A10}$$

$$A_4^S = \frac{1}{2}\begin{cases} -\psi_1^1\sum h^C_{m+1,k}(-1)^m kS_{n,m,k} + \psi_1^{-1}\sum h^S_{m+1,k}(-1)^m kS_{n,m,k} \\ -\psi_1^1(m+1)\sum h^C_{m+1,k}\bar{D}_{n,m,k} + \psi_1^{-1}(m+1)\sum h^S_{m+1,k}\bar{D}_{n,m,k} \end{cases} \quad (m=0) \tag{A11}$$

$$B_1^S = \frac{1}{2}\begin{cases} \psi_1^{-1}\left[-(m-1)\sum h^C_{m-1,k}\bar{D}_{n,m,k} - (m+1)\sum h^C_{m+1,k}\bar{D}_{n,m,k}\right] \\ +\psi_1^1\left[(m-1)\sum h^S_{m-1,k}\bar{D}_{n,m,k} - (m+1)\sum h^S_{m+1,k}\bar{D}_{n,m,k}\right] \\ +2\psi_1^0 m\sum h^C_{m,k}S_{n,m,k} \end{cases} \quad (m\geq 1) \tag{A12}$$

$$B_2^S = \frac{1}{2}\begin{cases} \psi_1^{-1}\left[\sum h^C_{m-1,k}(-1)^m kS_{n,m,k} - \sum h^C_{m+1,k}(-1)^m kS_{n,m,k}\right] \\ -\psi_1^1\left[\sum h^S_{m-1,k}(-1)^m kS_{n,m,k} + \sum h^S_{m+1,k}(-1)^m kS_{n,m,k}\right] \end{cases} \quad (m\geq 2) \tag{A13}$$

$$B_3^S = \frac{1}{2}\begin{cases} \psi_1^{-1}\left[2\sum h^C_{m-1,k}(-1)^m kS_{n,m,k} - \sum h^C_{m+1,k}(-1)^m kS_{n,m,k}\right] \\ -\psi_1^1\sum h^S_{m+1,k}(-1)^m kS_{n,m,k} \end{cases} \quad (m=1). \tag{A14}$$

Acknowledgements

This work was supported from the Korea Meteorological Administration (KMA) Research and Development Program under Grant CATER 2007-2206.

References

[1] H.-B. Cheong, Double Fourier Series on a Sphere: Applications to Elliptic and Vorticity Equations, J. Comput. Phys. 157 (2000) 327-349.

[2] P. N. Swarztrauber, The vector harmonic transform method for solving partial differential equations in spherical geometry, Mon. Wea. Rev. 121 (1993) 3415-3437.

[3] H.-B. Cheong, and J.-R. Park, Fourier-Series Representation and Projection of Legendre Functions, submitted to SIAM J. on Scientific Computing (2011).

[4] S. I. Moriguchi, K. H. Udakawa, and H. M. Shin, Formulas for mathematical functions III, 5th ed. (1990), Iwanami Shoten, Tokyo, 310pp.

[5] T. Nehrkorn, On the computation of Legendre functions in spectral models, Mon. Wea. Rev. 118 (1990) 2248-2251.

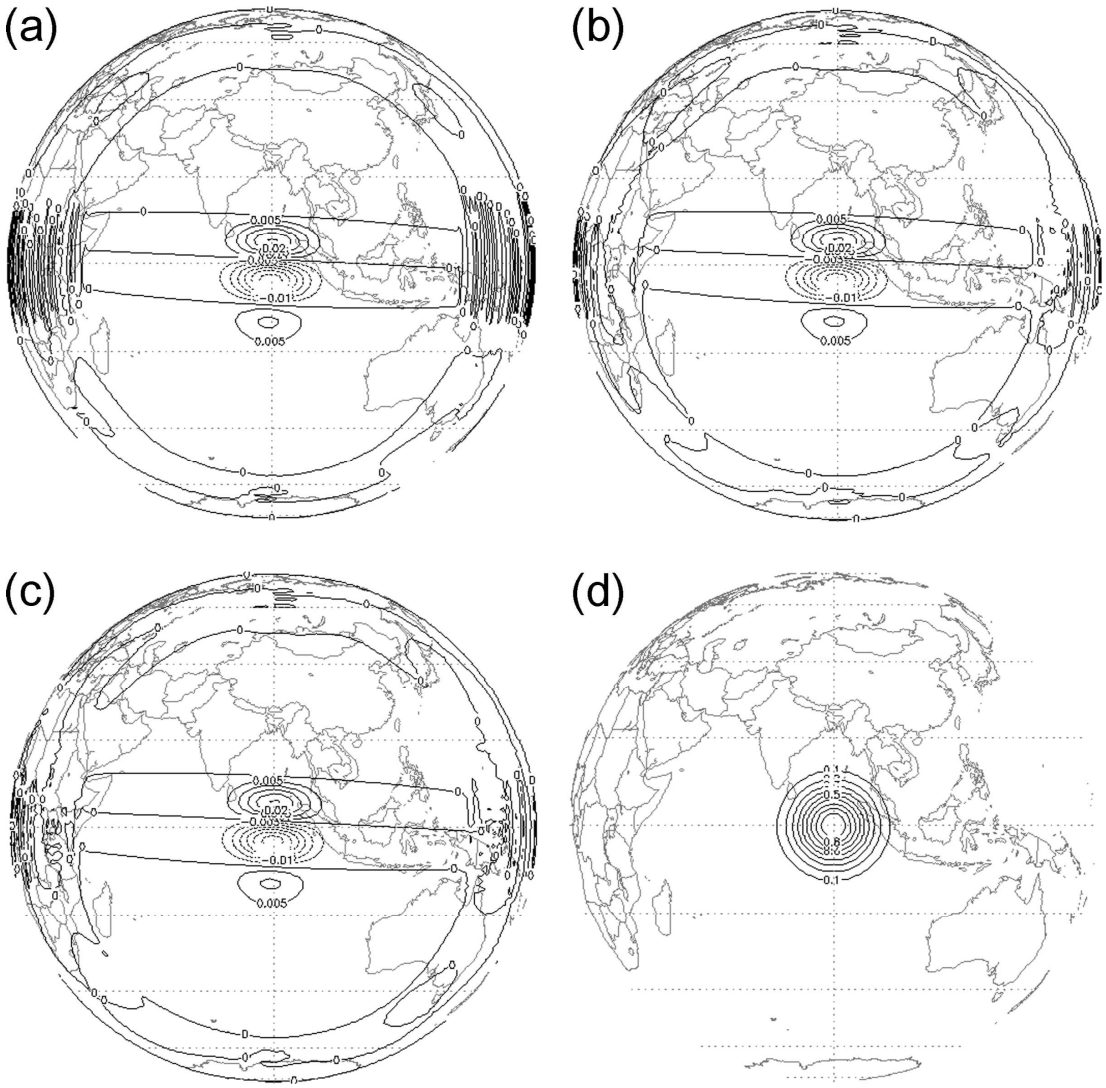


Figure 1. (a) Height error as a difference between day 12 and day 0, transformed to half-grids using Fourier-Legendre method. (b) Same as (a) but full-grids. (c) Result obtained by Gauss Legendre method. (d) Initial field.

Numerical simulation of MHD flow and heat transfer over a permeable stretching surface in a porous medium with variable parameters using FEM/EFGM

R. Bhargava[1,*], Rajesh Sharma[2]
Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee-247667, India
[1]Email: rbharfma@iitr.ernet.in
[2]Email: raj.juit@gmail.co.in

***Abstract*-The present paper deals with the study of thermal radiation effect on unsteady flow and heat transfer characteristics of viscous fluid with variable viscosity over a porous stretching sheet placed in a porous medium in the presence of viscous dissipation. A uniform magnetic field is applied transversely to the direction of the flow. Similarity transformations are used to convert the governing time dependent nonlinear boundary layer equations into a system of non-linear ordinary differential equations which are solved numerically by finite element method (FEM) and element free Galerkin method (EFGM). The influence of unsteadiness parameter (S), temperature-dependent fluid viscosity parameter (A) and radiation parameter (R), on the velocity and temperature profiles are shown graphically. The impact of radiation parameter on heat transfer rate is also shown. The present problem finds application in polymer production, metal casting etc.**

***Keywords*-** Porous medium, unsteady stretching sheet, MHD, temperature-dependent fluid viscosity, EFGM.

I. INTRODUCTION

The heat transfer in the laminar boundary layer flow on a stretching sheet has many practical applications in industrial manufacturing processes. This phenomenon is applied in wire and fiber coatings, food stuff processing reactor fluidization and transpiration cooling etc. The prime aim in almost every extrusion is to maintain the surface quality of the extrudate. The dynamics of the boundary layer flow over a stretching surface originated from the pioneering work of Crane [1]. Subsequently, it was extended by many authors to explore various aspects of the flow and heat transfer occurring in an infinite domain of the fluid surrounding the stretching sheet [2–6].

To achieve a better control on the rate of cooling, considerable models have been developed in recent years. Among other methods, it has been proposed that it may be advantageous to alter flow kinematics in such a way as to ensure a slower rate of solidification. A less intrusive methodology has been to employ transverse magnetic fields to exploit the electrically-conducting nature of many chemical engineering fluids. Important studies in this regards include Mansour et al [7], Cheng and Huang [8] and Chamkha [9]. In high-temperature chemical engineering operations in the industrial design and combustion, and fire science, it also becomes necessary to simulate thermal radiation heat transfer effects in combination with conduction, convection and also mass transfer e.g. radiative-convection heat transfer flows arising in industrial furnace systems [10], astrophysical flows [11], fire spread in buildings [12] etc. Wu et al [13] studied radiative-conductive heat transfer within porous polymer materials. Other applications where thermal radiation may be significant include solar receiver-reactors [14], steam-cracking furnaces [15] etc. Chen [16] studied the MHD mixed convection of a power-law fluid past a stretching surface in the presence of thermal radiation and internal heat generation/absorption effect. Recently, El-Aziz [21] has studied the thermal radiation effects on heat transfer over an unsteady stretching sheet. In stretching sheet processes, the radiative heat transfer properties of the cooling medium may also be manipulated judiciously influence the rate of cooling. Many other effects such as porosity of the medium may also be used effectively to control the rate of cooling. A combination of different thermophysical effects may be employed in order to obtain the best results.

Most of the studies in this regard confined their discussions by assuming uniformity of fluid viscosity. However, it is known that the physical properties of fluid may change significantly with temperature. The increase of temperature leads to a local increase in the transport phenomena by reducing the viscosity across the momentum boundary layer and so rate of heat transfer at the wall is also affected. Therefore, to predict the flow behavior accurately, it is necessary to take into account the viscosity variation for incompressible fluids. Gary et al. [17] and Mehta and Sood [18] showed that, when this effect is included the flow characteristics may change substantially as compared to the case of constant viscosity. Recently Mukhopadhyay [19] investigated the MHD unsteady boundary layer flow with variable fluid viscosity and thermal diffusivity past a porous stretching sheet.

In this paper we consider the unsteady case of a viscous fluid flow past a horizontal stretching sheet through porous medium in the presence of Magnetic field and viscous dissipation effect with heat transfer. The viscous dissipation effect is modeled in according to Al-Hadhrami et al. [20]. In this study, EFGM has been used as a tool for the numerical simulation. Comparisons of the results are done with those obtained by FEM also.

* Contact Author

II. MATHEMATICAL MODEL

Consider the flow of a viscous, incompressible, electrically conducting and radiation emitting fluid on a porous horizontal sheet, which comes through a slot at the origin and embedded in a porous medium. The Rosseland approximation is used to describe the radiative heat flux in the energy equation. The radiative heat flux in the x-direction is negligible in comparison with that in the y-direction. The fluid motion arises due to the stretching of the sheet. The continuous sheet coinciding with the plane $y=0$, moves in its own plane with a velocity $u_w(x,t)$, the temperature distribution $T_w(x,t)=T_\infty+T_{ref}\ cx^2/2\nu\ \ 1-\alpha t\ ^{-3/2}$ which vary along the sheet and with time. An external magnetic field is applied normal to sheet. Magnetic field is sufficiently weak to ignore induction effects i.e. magnetic Reynolds number is small. Hall and ionslip currents do not arise in the regime. The velocity and temperature fields in the boundary layer are governed by the following two-dimensional boundary layer equations for mass, momentum and thermal energy, given by:

Continuity equation:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{1}$$

Equation of momentum:

$$\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=\frac{1}{\rho}\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right)-\frac{\mu_e}{\rho k}u-\frac{\sigma B^2 u}{\rho} \tag{2}$$

Equation of energy:

$$\left(\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}\right)=\frac{k_f}{\rho C_p}\frac{\partial^2 T}{\partial y^2}+\frac{1}{C_p}\left(\frac{\mu_e}{\rho k}u^2+\nu\left(\frac{\partial u}{\partial y}\right)^2\right) -\frac{1}{\rho C_p}\frac{\partial q_r}{\partial y} \tag{3}$$

The corresponding boundary conditions for the regime are:

$$At\ y=0: u=u_w(x,t)=\frac{cx}{1-\alpha t},\ v=v_w(t), T=T_w(x,t) \tag{4}$$

$$At\ y\to\infty: u\to 0, T\to T_\infty \tag{5}$$

where u,v are the fluid velocity components in the x and y directions, T is the temperature in the boundary layer, μ,μ_e,ρ,k_f denote the effective fluid viscosity, the dynamic viscosity of fluid, fluid density and the thermal conductivity, respectively, c_p is the specific heat at constant pressure, q_r is the radiative heat flux, T_w and T_∞ are the plate temperature and the fluid free-stream temperature respectively, $v_w(t)=-v_0\ \ 1-\alpha t\ ^{-1/2}$ is the suction velocity $(v_0>0)$ of the fluid and c,α are positive constant with dimension of t^{-1}. Here, c is the initial stretching rate, whereas $c/\ 1-\alpha t$ is the effective stretching rate.

Thermal radiation is simulated using the Rosseland diffusion approximation (Kim et al. [22]) and in accordance with this, the radiative heat flux q_r is given by:

$$q_r=\frac{-4\sigma^*}{3k^*}\frac{\partial T^4}{\partial y} \tag{6}$$

where σ^* is the Stefan–Boltzman constant and k^* is the Rosseland mean absorption coefficient.

The temperature-dependent fluid viscosity is given by (Mukhopadhyay [19]),

$$\mu=\mu^*\left[a+b\ \ T_w-T\ \right] \tag{7}$$

where μ^* is the constant value of the coefficient of viscosity far away from the sheet and a,b are constants and $b>0$.

III. TRANSFORMATIONS

We now introduce dimensionless variables f and θ and similarity variable η as

$$\psi=\left(\frac{\nu c}{1-\alpha t}\right)^{1/2} xf(\eta), \theta(\eta)=\frac{T-T_\infty}{T_{ref}\left(\frac{cx^2}{2\nu\ \ 1-\alpha t\ ^{3/2}}\right)}, \tag{8}$$

$$\eta=\left(\frac{c}{\nu\ \ 1-\alpha t}\right)^{1/2} y$$

where ψ is the stream function which automatically satisfies the continuity equation. The velocity components are then derived from the stream function expression and obtained as

$$u=\frac{\partial\psi}{\partial y}=\left(\frac{cx}{1-\alpha t}\right)f'(\eta),\ v=-\frac{\partial\psi}{\partial x}=-\sqrt{\frac{\nu c}{1-\alpha t}}f(\eta) \tag{9}$$

Governing equations $(1)-(3)$ are then transformed into a set of differential equations and associated boundary conditions as given below:

$$S\left(\frac{\eta}{2}f''+f'\right)+f'^2-ff''=(a+A)f'''-A\theta f'''-A\theta' f'' -(a+A)\beta f'+A\beta\theta f'-Mf' \tag{10}$$

$$\left(1+\frac{4}{3R}\right)\theta''-2\Pr f'\theta+\Pr f\theta'-\frac{\Pr S}{2}\ 3\theta+\eta\theta'\ +\Pr Ec\beta\ \ f'\ ^2+Ec\ \ f''\ ^2=0 \tag{11}$$

$$f'(\eta)=1,\ f(\eta)=\lambda,\ \theta(\eta)=1 \text{ at } \eta=0 \tag{12}$$

$$f'(\eta)\to 0,\ \theta(\eta)\to 0 \quad \text{as} \quad \eta\to\infty \tag{13}$$

where $\beta=\frac{\mu_e}{\rho\nu}\frac{\mathrm{Re}_x}{\mathrm{Re}_k^2}$ is the local porous parameter, $\mathrm{Re}_x=\frac{u_w x}{\nu}$ is the local Reynolds number, $\mathrm{Re}_k=\frac{u_w\sqrt{k_f}}{\nu}$, $S=\frac{\alpha}{c}$ is the unsteadiness parameter, $M=\frac{\sigma B_0^2}{\rho c}$ is the magnetic parameter, $\Pr=\frac{\nu}{k_f}$ is the

Prandtl number, $R = {k^* k_f}/{4\sigma^* T_\infty^3}$ is the radiation parameter, $Ec = \dfrac{u_w^2}{c_p\ T_w - T_\infty}$ is local Eckert number, $A = b\ T_w - T_\infty$ is the temperature dependent viscosity parameter and the prime indicates differentiation with respect to η.

The physical quantity of interest in this problem is the local Nusselt number Nu_x which is defined as

$$Nu_x = \frac{-x}{(T_w - T_\infty)} \frac{\partial T}{\partial y}\bigg|_{y=0}, \text{ then } \frac{Nu_x}{\sqrt{\mathrm{Re}_x}} = -\theta'(0) \qquad (14)$$

IV. NUMERICAL SIMULATION

The two-point unsteady nonlinear boundary value problem defined by the system of ordinary differential equations (10)-(11), together with their corresponding boundary conditions (12)-(13), is solved using EFGM [23] and the results are provided graphically. Two point Gaussian quadrature formulae have been used to evaluate the integral values. Owing to the nonlinearity of the system of equations an iterative scheme is required to solve the nonlinear algebraic matrix system. The system is linearized by incorporating known function, which is solved efficiently employing the Gauss-elimination technique while sustaining an accuracy of 0.0001.

The accuracy of EFGM is also compared with the results of El. Azis [21] in Table 1 for local Nusselt number and the results show that as we increases the node points the Nusselt number value will converge to the result obtain by El. Aziz [21].

In order to see the effect of node points, we have run the code for our model with different uniform node points $N = 51, 101, 201, 401, 801$ and a very little change in the result has been observed after 401 node points; the results are omitted herein for brevity. Therefore, the whole domain is represented by 401 uniform node points. The results are also computed using FEM in Figs. 1-2. It shows that the EFGM method is in good agreement with the finite element solution value of h and θ.

V. DISCUSSION OF THE RESULTS

The velocity and temperature profiles for different physical parameters such as S, A, and R are shown in Figs. 3-8. A selected set of results has been obtained covering the ranges $0.0 \le S \le 0.4,\ 0 \le A \le 2,\ 0.1 \le R \le 1.5,\ \mathrm{Pr} = 0.71,\ \lambda = 1.0,$ $\beta = 1.0,\ M = 1.0,\ a = 1.0$ and $Ec = 0.3$.

Figs. 3-4 show the effect of unsteadiness parameter S $S = 0.0, 0.2, 0.4$ on velocity and temperature profiles, when all other parameters are kept constant. $S = 0$, gives steady state flow and $S > 0$, an unsteady flow. The results show that the velocity and temperature decreases with increase in the values of the unsteadiness parameter.

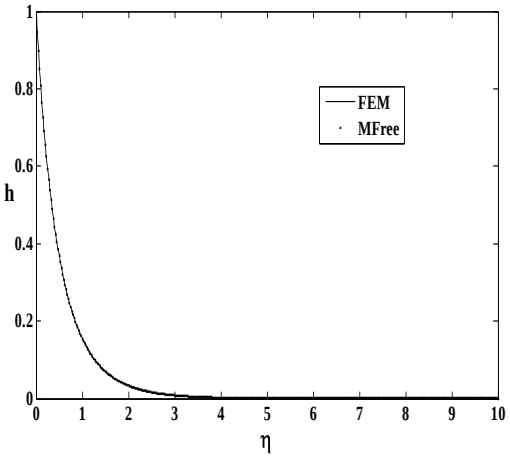


Fig. 1: Comparison of Velocity plot (S=0.1, A=1, R=1.0)

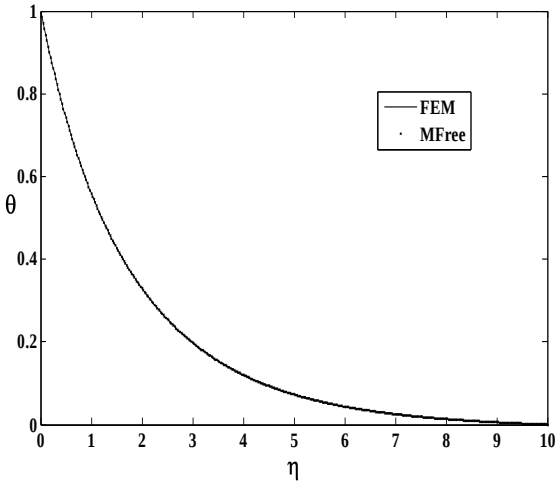


Fig. 2: Comparison of temperature plot (S=0.1, A=1, R=1.0)

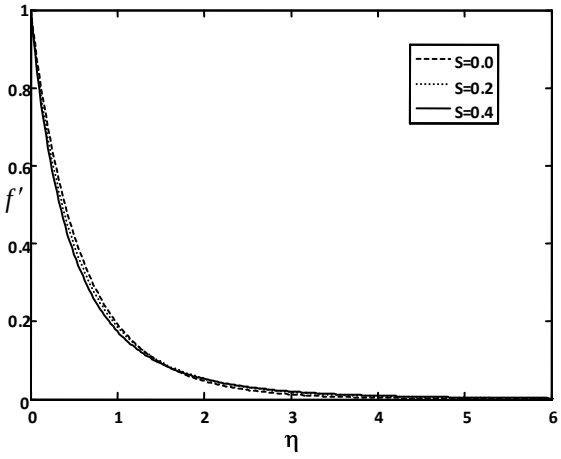


Fig. 3: Velocity plot with S (A=1.0, R=1.0)

In Figs. 5-6, we elucidate the effect of temperature-dependent fluid viscosity parameter A $A=0,1,2$ on the velocity and temperature profiles. Fluid velocity decreases with increasing A within a particular range of η. Fig. 6 exhibits that the temperature also decreases with the increasing value of A in the boundary layer. These results are in well agreement with the results obtained by Mukhopadhyay [19].

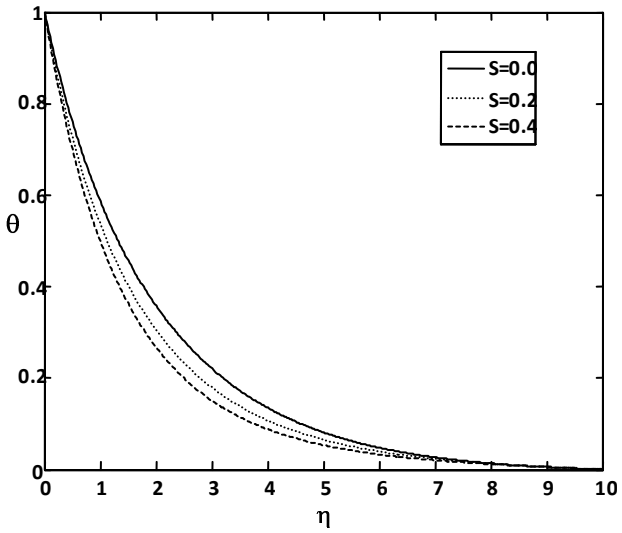
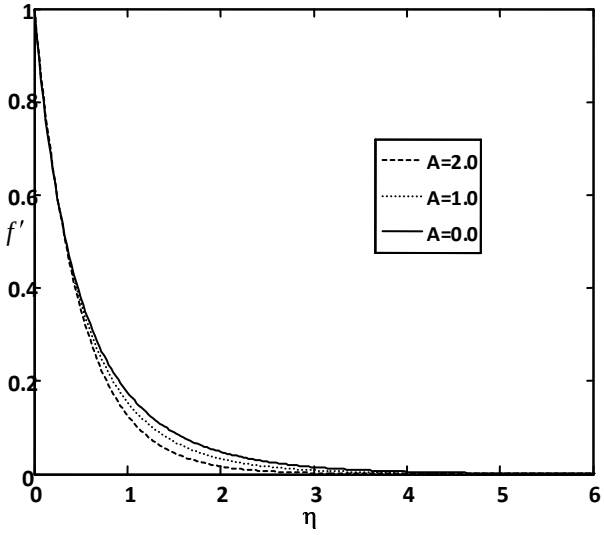


Fig. 4: Temperature plot with S (A=1.0, R=1.0)



Figs. 5: Velocity plot with A (S=0.1, R=1.0)

Fig. 7 displays the effect of radiation parameter R on the dimensionless temperature θ. Increasing the radiation parameter R implies to decreasing the temperature throughout the boundary layer region. This result can be explained by the fact that a decrease in the values of R for given k and T_∞ means a decrease in the Rosseland radiation absorptivity k^*. Since divergence of the radiative heat flux $\partial q_r / \partial y$ increases as k^* decreases which in turn increases the rate of radiative heat transferred to the fluid and hence the fluid temperature increases. In view of this explanation, when $R \to 0$ the effect of thermal radiation becomes more significant and the effects of radiation can be neglected when $R \to \infty$.

The local Nusselt number variation with radiation parameter R and viscosity parameter A are presented in Fig. 8. It is seen from this figure that the local Nusselt number is increased for all values of A and R. We can also observe that for fixed values of A, the local Nusselt number increases as the radiation parameter R increases and vice-versa.

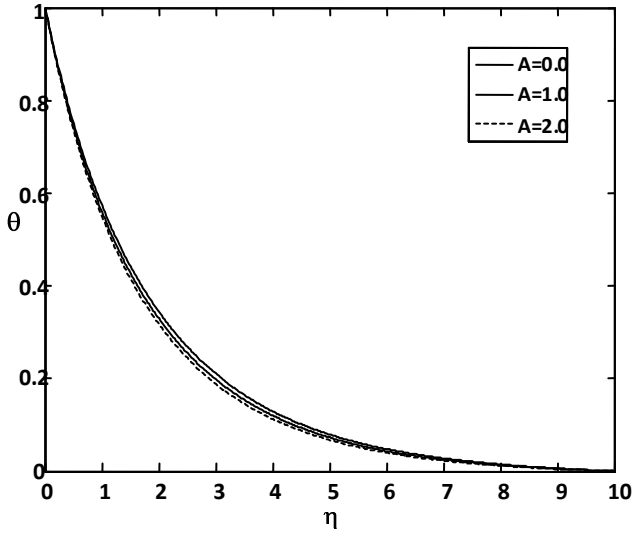


Fig. 6: Temperature plot with A (S=0.1, R=1.0)

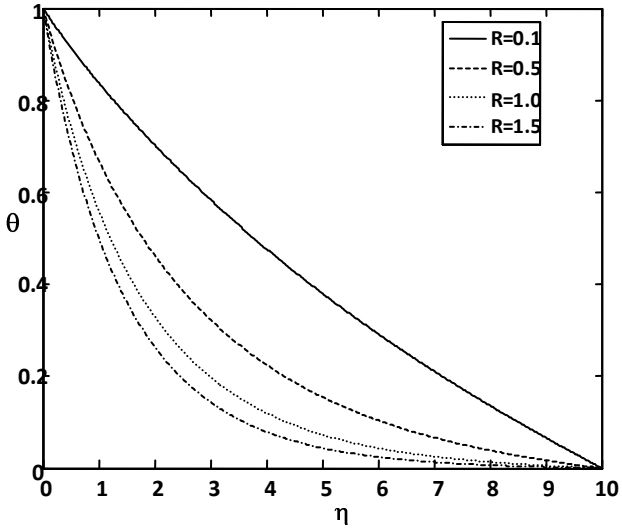


Fig. 7: Temperature plot with R (S=0.1, A=1.0)

VI. CONCLUSIONS

The following conclusions can be drawn as a result of the computations:

1. With increasing values of unsteadiness parameter, fluid velocity and temperature are found to decrease.
2. Velocity is decreased with increasing temperature-dependent fluid viscosity parameter.
3. Temperature of fluid in the boundary layer region decreases with the increase in radiation parameter.
4. The rate of heat transfer increases with an increase in temperature-dependent fluid viscosity parameter and radiation parameters. Thus, fast cooling of the plate can be achieved by incorporating these parameters.
5. The limiting case of our results are in excellent agreement with the earlier steady state results of El. Aziz [21].

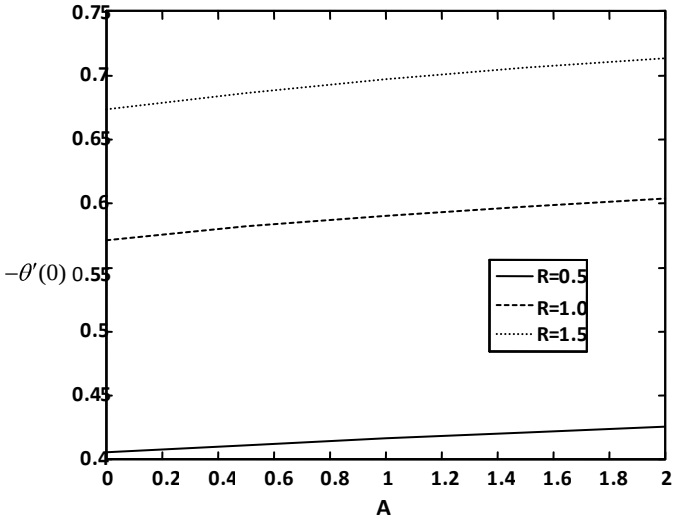


Fig. 8: Nusselt number plot with A & R
(S=0.1, A=1.0)

Pr	El. Aziz	Present Result with different node points				
	[21]	N=51	N=101	N=201	N=401	N=801
0.1	0.4517	0.4485	0.4509	0.4517	0.4520	0.4521
1.0	1.6728	1.6092	1.6498	1.6644	1.6695	1.6714
10	5.7050	4.8202	5.3446	5.5724	5.6569	5.6869

Table 1: Value of $Nu_x \mathrm{Re}_x^{-1/2} = -\theta'(0)$ for several values of Pr with $S = 0.8, A = 0.0, \lambda = 0.0, Gr = 0, M = 0, \beta = 0, Ec = 0.0, R \to \infty$

REFERENCES

[1] L. J. Crane, Flow past a stretching plate, Z. Angrew. Math. Phys. 21 (1970) 645-647.
[2] P. S. Gupta, A. S. Gupta, Heat and mass transfer on a stretching sheet with suction or blowing, Canadian J. Chemical Engineering. 55 (1977) 744-746.
[3] B. K. Datta, P. Roy, A. S. Gupta, Temperature field in the flow over a stretching sheet with uniform heat flux, Int. Comm. Heat Mass Transfer. 12 (1985) 89-94.
[4] C. K. Chen, M. I. Char, Heat transfer of a continuous stretching surface with suction or blowing, J. Math. Anal. Appl. 135 (1988) 568–580.
[5] E. Vajravelu, Viscous fluid over a non-linearly stretching sheet, Applied mathematics and computation 124 (2001) 281-288.
[6] R. Bhargava, L. Kumar, H. S. Takhar, Finite element solution of mixed convection microploar fluid driven by a porous stretching sheet, Int. J. Eng. Sci. 41 (2003) 2161–2178.
[7] M. A. Mansour, N. F. El-Anssary, A. M. Aly, Effects of chemical reaction and thermal stratification on MHD free convective heat and mass transfer over a vertical stretching surface embedded in a porous media considering Soret and Dufour numbers Chemical Engineering J. 145 (2008) 340-345.
[8] W. T. Cheng and C. N. Huang, Unsteady flow and heat transfer on an accelerating surface with blowing or suction in the absence and presence of a heat source or sink, Chemical Engineering Science 59 (2004) 771-780.
[9] A. J. Chamkha, Transient hydromagnetic three-dimensional natural convection from an inclined stretching permeable surface, Chemical Engineering J. 76 (2000) 159-168.
[10] Modest, M.F., Radiation Heat Transfer, MaGraw, New York, 1993.
[11] M. H. Mansour, Radiative and free convection effects on the oscillatory flow past a vertical plate, Astrophysics Space Science J. 166 (1990) 26-45.
[12] R. O. Weber, Analytical model for fire spread due to radiation, Combusition and Flame J. 78 (1989) 398-408.
[13] H. Wu, F. Jintu and D. Ning, Thermal energy transport within porous polymer materials: Effects of fiber characteristics, J. Applied Polymer Science 106 (2007) 576-583.
[14] L. O. Schunk, W. Lipinski and A. Steinfeld, Heat transfer model of a solar receiver-reactor for the thermal dissociation of ZnO-Experimental validation at 10 kW and scale-up to 1 MW, Chemical Engineering J. 150 (2009) 502-508.
[15] G. D. Stefanidis, K. M. Van Geem, G. J. Heynderickx and G. B. Marin, Evaluation of high-emissivity coatings in steam cracking furnaces using a non-grey gas radiation model Chemical Engineering J. 137 (2008) 411-421.
[16] C. H. Chen, Magneto-hydrodynamic mixed convection of a power-law fluid past a stretching surface in the presence of thermal radiation and internal heat generation/absorption, Int. J. Non-Linear Mechanics 44 (2009) 596-603.
[17] J. Gary, D. R. Kassoy, H. Tadjeran, A. Zebib, The effects of significant viscosity variation on convective heat transport in water saturated porous medium, J. Fluid Mech. 117 (1981) 233-249.
[18] K. N. Mehta, S. Sood, Transient free convection flow with temperature dependent viscosity in a fluid saturated porous medium, Int. J. Engg. Sci. 30 (1992) 1083-1087.
[19] S. Mukhopadhyay, G. C. Layek, S. A. Samad, Study of MHD boundary layer flow over a heated stretching sheet with variable viscosity, Int. J. Heat Mass Transfer 48 (2005) 4460-4466.
[20] A. K. Al-Hadhrami, L. Ellott, D. B. Ingham, A new model for viscous dissipation in porous media across a range of permeability values, Transport in Porous media 53 (2003) 117-122.
[21] M. A. El-Aziz, Radiation effect on the flow and heat transfer over an unsteady stretching sheet, Int. Comm. Heat Mass Transfer 36 (2009) 521−524
[22] Y.J. Kim and A. G. Fedorov, Transient mixed radiation convection flow of a micropolar fluid past a moving semi-infinite vertical porous plate, Int. J. Heat and Mass Transfer 46 (2003) 1751-1758.
[23] Liu G.R., Mesh free method-Moving beyond the Finite element method, CRC Press, London (2003).

The 5th Umpire: Cricket's Edge Detection System

R. Rock, A. Als, P. Gibbs, C. Hunte
Department of Computer Science, Mathematics and Physics
University of the West Indies (Cave Hill Campus)
Barbados

Abstract— *The game of Cricket, and the use of technology in the sport, have grown rapidly over the past decade. However, technology-based systems introduced to adjudicate decisions in run outs, stumpings, boundary infringements and close catches are still prone to human error, and thus their acceptance has not been fully embraced by cricketing administrators. In particular, technology is not currently employed for bat-pad decisions. Although the Snickometer may assist in adjudicating such decisions, it depends heavily on human interpretation. The aim of this study is to investigate the use of Wavelets in developing crickets' edge-detection adjudication system. Live audio samples of ball-on-bat, and ball-on-pad events from a cricket match, will be recorded. Wavelet analysis and feature extraction will then be employed on these samples. Results will show the ability to differentiate between these different audio events. This is crucial to developing a fully automated system.*

Keywords*:* Cricket, Wavelets, Edge-detection, feature extraction

I. Introduction

Sports play a major role in generating entertainment revenue. At the 2000 Sydney Olympic Games, the organizing committee generated an income of US $1.756 billion [1]. In 2009, the Indian Premier League (IPL) offered paychecks as high as US$1.55 million to top class cricketers for a five week contract [2]. Consequently, to deter event-fixing and ensure legitimate results, it is not surprising that the use of technology in sports has steadily increased over the years and now plays a major role in adjudicating the outcome of events. Moreover, it has become crucial in protecting the livelihoods of sportsmen who make a living from sports. Incorrect decisions in sporting events could affect a player's confidence, the outcome of a game and even end a player's sporting career. This view was supported by Mr. Nariman Jamshedji "Nari" Contractor while addressing the Fourteenth Frank Worrell Memorial Lecture at the University of the West Indies. He stated that the use of technology increases the likelihood of a correct decision by the referee and this greatly assists the players in earning a livelihood from their sports.

Unfortunately, Cricket is no exception and the prevalence of incorrect decisions, in spite of the exposure to various sporting technology, continues to be disconcerting to its wide fan base. Reasons range from the inefficiency of the equipment (e.g. 'blind spots' due to players or umpires blocking the view of cameras used in decision making processes) to human error by the on-field umpires. In international cricket, a Third Umpire consults with the on-field umpires using wireless technology. More specifically, the Third Umpire uses television replays in situations, such as disputed catches and boundary infringements, to appropriately advise the on-field umpires. The Third Umpire is also called upon to adjudicate on run out decisions.

There are also a number of devices being used to assist the umpires in making decisions and for the entertainment of television audiences. One such device 'Hawk-Eye', is a computer system which traces a ball's trajectory, with a claimed accuracy of 5mm, and sends the data to a virtual-reality machine [3]. Hawk-Eye uses six or more computer-linked television cameras situated around the cricket field of play. The computer acquires the video in real time, and tracks the path of the cricket ball on each camera. These six separate views are then combined to produce an accurate 3D representation of the path of the ball. The system was first used during a Test match between Pakistan and England at Lord's Cricket Ground, on 21 April 2001, in the TV coverage by Channel 4 [3]. Since then it has been an indispensable tool for cricket commentators around the world. It is used primarily by the majority of television networks to track the trajectory of balls in flight, mostly for analyzing Leg Before Wicket (LBW) decisions. In this case, Hawk-Eye predicts the most likely path of the ball, and determines if it would go on to hit the wicket. Although Hawkeye is very accurate in measuring the actual path of a ball, when it comes to predicting the future path of the ball, such as in LBW decisions, it is not as clear. If the ball is heading to the pitch (ground), Hawk-Eye cannot determine if it will skid a bit more than normal or hit a crack, bit of grass, or worn patch of the pitch. The predicted path of the ball is based on the average and expected pathway [3] and does not take into effect any deviation that can be caused by imperfections on the ground surface.

Another item of equipment being used in cricket is the Snickometer (also known as 'snicko'). This was invented by English Computer Scientist, Allan Plaskett, in the mid-1990s. The Snickometer is composed of a very sensitive microphone, located behind the stumps, and an oscilloscope (wirelessly connected) which displays traces of the detected sound waves. These traces are recorded and synchronized with the cameras located around the ground. For edge-decisions, the oscilloscope trace is shown alongside the slow motion video of the ball passing the bat. By the transient shape of the sound wave, the viewer(s) first determines whether the noise detected by the microphone coincides with

the ball passing the bat, and second, if the sound appears to come from the bat hitting the ball or from some other source. This technology is currently only used as a novelty tool to give the television audience more information regarding if the ball actually hit the bat. Umpires do not enjoy the benefit of using 'snicko' but must rely instead on their senses of sight and hearing, as well as personal judgment and experience. In many instances, there are coinciding events that may be confused with the sound of ball-on-bat. These include the bat hitting the pad during the batman's swing or the bat scuffing the ground at the same time the ball passes the bat. The shape of the recorded sound wave is the key differentiator as a short, sharp sound is associated with bat on ball. The bat hitting the pads, or the ground, produces a 'flatter' sound wave. The signal is purportedly different for bat-pad and bat-ball however, this is not always clear to the natural eye [4]. The aim of this paper is to employ wavelet analysis and feature extraction to differentiate between bat-on-ball and bat-on-pad audio sounds in cricket. Results could then be employed in an automated decision making process. It is expected that this will give teams a fairer chance on the outcome of a match (game) by minimising the number of decision errors currently observed in the game.

II. Background

It is well known that the continuous wavelet transform (CWT) may be used to analyze audio signals. The CWT provides another view of temporal signals as it transforms the regular time vs. amplitude signal to time vs. scale, where scale can be converted to a pseudo-frequency. This method allows one to examine the temporal nature of audio events and the corresponding frequencies involved. In essence, the correlation values, produced during the transformation process, provide critical information on the characteristics of the signal. By exploiting these characteristics a distinction can be made between different audio events. Significant applications of this new 'dimension' are widely reported in the literature. Lambrou et al used the wavelet transform to extract statistical features from audio data to successfully distinguish between three different musical styles of rock, piano and jazz [5] In another documented application, the wavelet-packet transform was used to extract spatiotemporal characteristic features for vehicular detection [6].

Ting et al introduced a novel time-frequency based pattern recognition technique for a proposed effective cricket decision making system using snicko-signals [7]. In their work, experiments were conducted to simulate and record snickometer signals using a PC microphone with the help of RAVEN LITE software. A simple time-frequency based pattern recognition technique was developed and tested.

It should be noted that this software employs the short-time Fourier transform (STFT) to generate the time-frequency data used in the pattern recognition analysis.

It is believed that this is the first time that wavelet analysis has been employed in classifying the audio signals recorded during the game of cricket. This innovation will be very useful in live broadcast of cricket match for the benefit of audience and adjudicators.

III. Methodology

The equipment setup shown in Fig. 1 was tested at various local hard-ball cricket matches throughout Barbados. This setup is identical to that used for international matches. More specifically, the microphone transmitter is covered in a small hole directly behind the stumps. The receiver and the laptop are assembled inside the players' pavilion and recordings are made using the laptop's sound recorder.

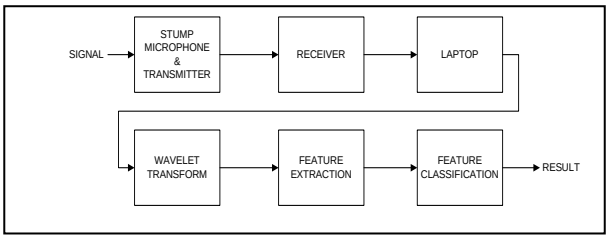


Figure 1: Schematic of experimental setup

The key specifications for equipment used in recording the audio data are listed in Table 1.

The received audio signals were first digitised using a 16-bit pulse coded modulation (PCM) scheme at a sample rate of 44,100 kHz and then stored as a stereo .WAV file on a laptop computer for offline wavelet analysis. The primary software tool for this analysis was MATLAB. It must be noted that there are many available wavelet transforms and scale ranges that can be used for testing. However, the wavelet used for analysis in this project is the Bi-orthogonal 3.3, and the scale chosen for this particular wavelet is from one to four hundred and fifty. Fifty-two samples were taken; twenty-six of ball hitting bat and twenty-six of ball hitting pad. Samples included deliveries from both fast- and slow-bowlers and the batsmen's' equipment (e.g. pads, bat) also vary in make and model.

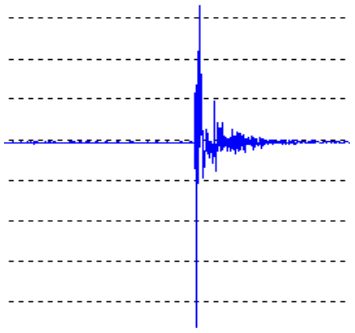
TABLE I. EQUIPMENT PARAMETERS

EQUIPMENT	SPECIFICATIONS
	WL184 Supercardiod Lavalier Condenser Mic:
	Supercardioid pickup pattern for high noise rejection and narrow pickup angle
Shure SLX14/84 Wireless Lavalier Microphone System	**SLX1 Body pack Transmitter:**
	518 - 782 MHz operating range
	SLX4 Wireless Receiver:
	960 Selectable frequencies across 24MHz bandwidth
Mobile Precision M6400 Notebook Computer	Precision M6400, Intel Core 2 Quad Extreme Edition QX9300 2.53GHz, 1067MHZ

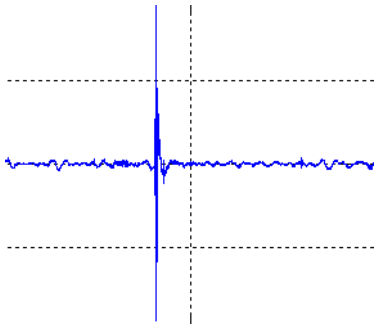
IV. Results

Recordings of the impact of ball hitting bat and ball hitting pad were successfully compiled. In many cases, attempts to categorise these signals by visual inspection of the data in the time domain, proved to be inconclusive as they are similar in appearance. This is highlighted in Figures 2(a) and 2(b). An adjudicating official could not reliably use this method to determine the source of the recorded noise as is done using the Snickometer.

However, results showed that when using the wavelet transform, there is a noticeable difference between ball hitting bat and ball hitting pad. Figures 3 and 4 show the results from the continuous wavelet transform of the two previous sound files. Note that the x-axis shows how the transform varies across time and the y-axis shows the coefficient value of the wavelet and the z-axis shows the scale. It is observed from these figures that for the ball hitting pad, the best correlation value is at a higher scale (i.e. lower pseudo-frequency) than for the ball hitting bat, where the best correlation value also observed that the correlation value for a ball hitting bat (12) was higher that the value for ball hitting pad (2.5).



(a)



(b)

Figure 2: Sound samples taken of (a) ball on pad, (b) ball on bat

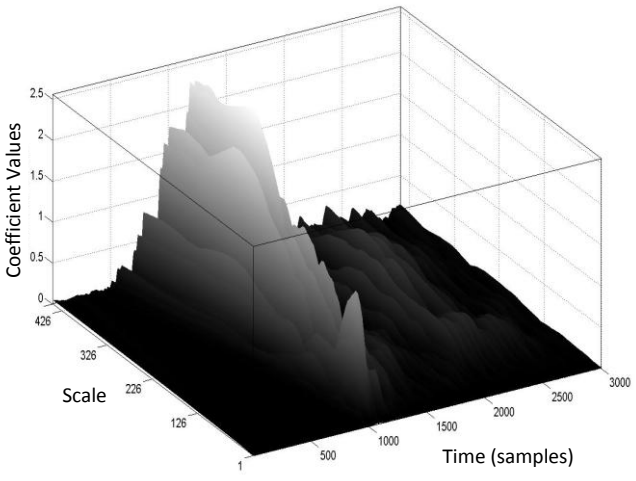


Figure 3: 3D wavelet transform of ball hitting pad

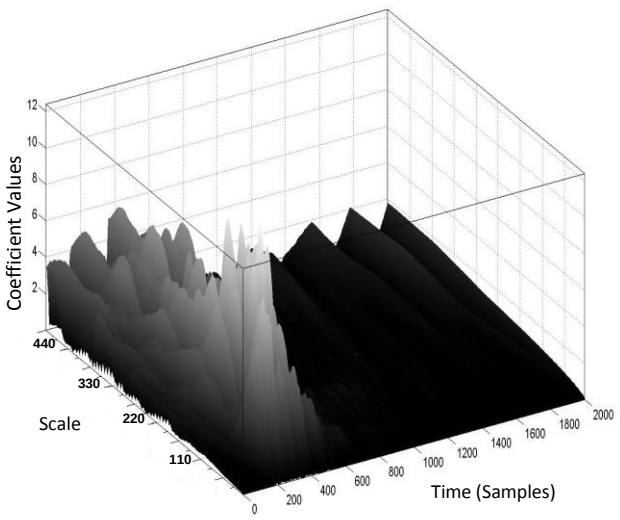
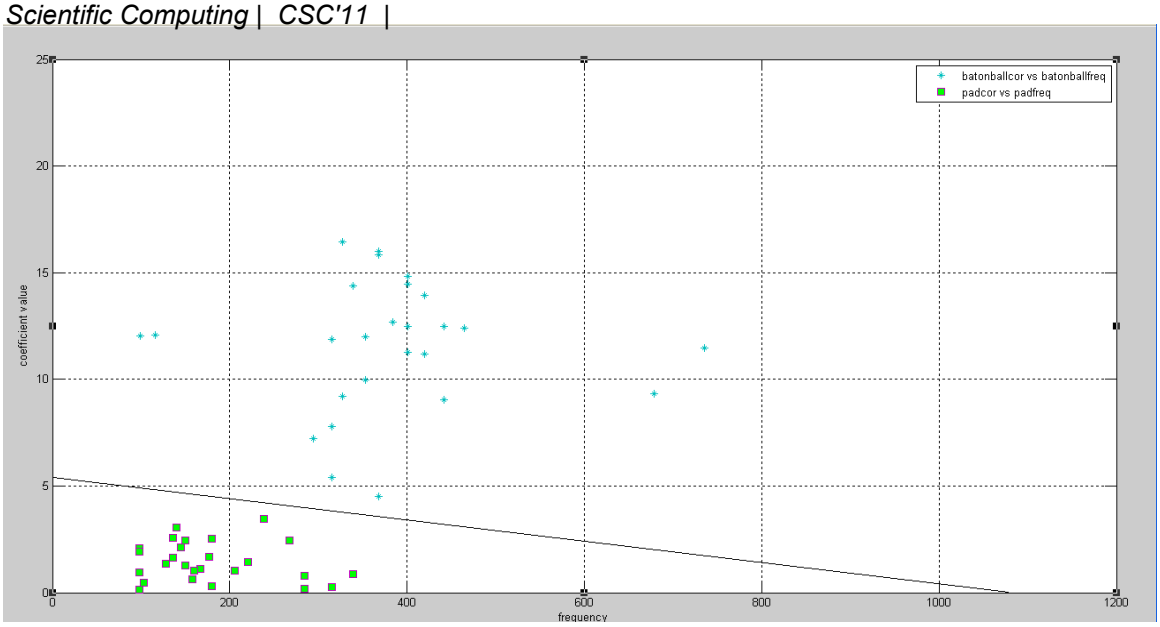


Figure 4: 3D wavelet transform of ball hitting bat



Figuure 5: Difference between bat on ball (stars), and ball on pad (squares).

In Figure 5, twenty-six wavelet coefficient values are plotted against their corresponding pseudo-frequencies for both ball-on-bat and ball-on-pad data categories. Observe that there are two general areas which represent contact between bat/ball, and ball/pad, respectively. Moreover, there is a linear partition separating the two categories of signals. This partition, which is represented as a line drawn between the two categories of signals serves as a possible decision boundary for classification.

V. Conclusion

Results showed that viewing a sound file in the time domain proved to be ambiguous when adjudicators need to determine the difference between key sounds in cricket such as ball hitting bat and ball hitting pad. When the sound files are analyzed using the wavelet transform, noticeable differences between the two signal classifications are revealed by the correlation values of the wavelet transform and the corresponding pseudo-frequency at the points interest. These differences were plotted on a graph for easy viewing. We are now closer to designing a fully automated system which, when given a sound file of a noise, can determine if it was bat-on-ball or ball-on-pad. This distinction provides the means for which an automation tool can be developed. Such a tool will be very useful in live broadcast of cricket match for the benefit of audience and adjudicators.

The wavelet chosen for testing is just one of many available wavelets. Future research will be carried out to determine which wavelet gives optimum results. In this regard it should be noted that it is possible to design a wavelet which best represents a particular sound. The scale range of the wavelets employed also needs to be examined, as changing the scale range of the wavelet may produce improved results. Therefore, an optimum scale range for each wavelet must also be established. Also, other characteristics given by the wavelet transform will be investigated to determine if there are other noticeable differences between the signals. This work will also be enhanced to a real-time application and extended to include bat-pad adjudication decisions.

Acknowledgment

The authors would like to acknowledge Mr. Simon Wheeler (Executive Producer/Director of TWI and IMG media company), Mr. Mike Mavroleon (Senior Engineer IMG media) and Mr. Collin Olive (Asst. Sound Engineer IMG media) for their direction in acquiring the equipment needed to record the data samples used in this paper. These are the persons responsible for technological aspects of the broadcast of international cricket (i.e. snickometer, hawk-eye, TV replays).

References

1. Olympaid, Beijing. 2011. *Profit or Loss?* China Internet Information Center, [Online], last cited 2011, January, 01], Available: http://www.china.org.cn/english/sports/111340.htm

2. Peter J. Schwartz, Chris Smith, 2011, *The World's Top-Earning Cricketers*, [Online], last cited 2011, January, 01], Available: http://www.forbes.com/2009/08/27/cricket-ganguly-flintoffl-business-sports-cricket-players.html.

3. Wood, Rob J., 2005, *Hawk-Eye System in Cricket*, [Online], last cited 2010, January, 21]. Available: http://www.topendsports.com/sport/cricket/equipment-hawkeye.htm.

4. Wood, Rob J., 2005, *Cricket Snicko-Meter*, [Online], last cited 2010, January, 21, Available: http://www.topendsports.com/sport/cricket/equipment-snicko-meter.htm.

5. Lambrou, T., P. Kudumakis, R. Speller, M. Sandler, and A. Linney. "Classification of audio signals using statistical features on time and wavelet transform domains," In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.*

6. Schclar, A., A. Averbuch, N. Rabin, V. Zheludev, and K. Hochman. "A diffusion framework for detection of moving vehicles," *Digital Signal Processing, vol* 20, no 1, pp. 111-122, Feb. 2009.

7. Ting, S., and M. V. Chilukuri. "Novel pattern recognition technique for an intelligent cricket decision making system." In *Proc. IEEE Instrumentation and Measurement Technology Conference, I2MTC '09*, 2009.

A Clustering-Based Matrix Multiplication Algorithm

Abdullah N. Arslan and Arvind Chidri
Department of Computer Science and Information Systems
Texas A & M University - Commerce
TX 75429, USA
{$Abdullah\_Arslan@tamu-commerce.edu$, $kashinathc@gmail.com$}
Contact Author: Abdullah N. Arslan

Abstract— *We present a simple matrix multiplication algorithm that multiplies two input matrices with rows (in one matrix) and columns (in the other matrix) within a small diameter d (distances are measured using the Hamming distance). This algorithm runs in time $O(dn^2)$ for matrices of size $n \times n$. We then propose a more general clustering-based matrix multiplication algorithm. For a given integer $k \leq n$, this algorithm first uses a clustering algorithm that places rows of one input matrix and separately columns of the other input matrix into k clusters with approximately the smallest possible radius (cluster size) s (within a factor of two of the minimum possible). Second, it uses our first algorithm as a subroutine to multiply the original input matrices. We have implemented this algorithm. The time complexity of this implementation is $O((k+s)n^2)$. We also describe how to achieve $O((\log k + s)n^2 + k^2 n)$ worst case time complexity.*

Keywords: algorithm, matrix multiplication, clustering, approximation algorithm

1. Introduction

Matrix multiplication is a fundamental operation. Efficient matrix multiplication would yield efficient algorithms for many other problems such as solving systems of linear equations, and for solving even some basic graph problems [12].

The naive matrix multiplication algorithm takes $\Theta(n^3)$ time to multiply two $n \times n$ matrices. Intuitively, the problem seems to require $\Omega(n^3)$ time. Strassen's $O(n^{2.81})$-time algorithm [13] for this problem was a big surprise for the scientific world. The fastest known algorithm to date runs in $O(n^{2.38})$ time [5]. An interesting recent approach makes evident a connection between fast matrix multiplication and group theory, which can potentially reduce the time complexity of the problem further [4]. Many researchers conjecture that an $O(n^2)$-time matrix multiplication algorithm exists. We believe that there exist near optimal algorithms that are not based on numerical techniques (at least for special cases of matrices). In the literature, sometimes similar optimal results (in terms of time complexity or approximation ratio) achieved by numerical methods have also been achieved by simple non-numerical algorithms (for example, consider vertex cover by linear programming and Gavril's graph based simple method, which give similar approximation performance). In the case of matrix multiplication, all existing asymptotically fastest methods use numerical techniques.

Several studies for matrix multiplication in the literature have proposed non-numerical approaches that exploit special structure in the matrices (after possibly reordering rows and columns, or representing matrices by graphs) [2], [3], [6]. The Cuthill-McKee algorithm is proposed to reduce the bandwidth of sparse symmetric matrix [6]. Arslan and Chidri [2] propose an algorithm specialized for matrices in which the elements are drawn from a fixed finite set, and the matrices are *thin*, or the matrices have many common prefixes in their rows and in their columns. A matrix is thin if one of its dimensions is very small compared to the other. Arslan and Chidri [2] represent each matrix by a trie, which is a compact representation for these special cases of matrices they study. Multiplication is done by simultaneous and synchronized tree traversals on two tries, depth-first traversal on one and breath-first traversal on the other. This way previously calculated partial results can be reused. Their algorithm is much faster than the $O(n^3)$-time naive algorithm when input matrices have many common prefixes in their rows and columns, or when input matrices are thin [2].

In the literature, the closest work to the current paper is [3] in which Björklund and Lingas present an algorithm for computing the product of two $n \times n$ matrices A and B. For an $n \times n$ Boolean matrix C, let G_C be the complete weighted graph on the rows of C where the weight of an edge between two rows is equal to their Hamming distance, i.e. the number of entries different in the same corresponding positions. Also let $MWT(C)$ be the weight of a minimum spanning tree of G_C. The Boolean matrix multiplication algorithm in [3] runs in expected time $\tilde{O}(n(n + \min\{MWT(A), MWT(B^t)\}))$, where B^t stands for the transposed matrix B, and $\tilde{O}(f(n))$ means $O(f(n)poly\text{-}\log n)$. This algorithm performs well on Boolean matrices whose minimum-weight spanning trees have small weight, however, it is noted that both $MWT(A)$ and $MWT(B^t)$ can be $\Omega(n^2)$ since the Hamming distance

between any two rows or columns can be $\Omega(n)$.

In the current paper, we continue studying non-numerical approaches for matrix multiplication. We present an algorithm specialized for matrices that form clusters with small radii. For a given integer $k \leq n$, the worst case time complexity of our algorithm is $O((k+s)n^2)$ achieved by placing the rows of one matrix and columns of the other in k clusters for the purpose of minimizing the maximum cluster radius s over all these clusters. The worst case time complexity of $O((\log k + s)n^2 + k^2 n)$ can be achieved by using more efficient clustering in our algorithm.

The outline of this paper is as follows: In Section 2 we present a multiplication algorithm for multiplying two input matrices with all rows in one matrix, and all columns in the other matrix within diameter d. We generalize this algorithm, and use a clustering algorithm as a subroutine to develop another algorithm for the case of matrix multiplication where rows and columns of matrices form multiple clusters (separately). We describe this algorithm in Section 3. We include pointers for additional computational problems that our approach yields, and discuss potential future work in Section 4. We summarize our results and conclude in Section 5.

We would like to mention one notational convenience that we follow throughout this paper. We often use product of vectors. We omit the product symbol '.' (dot). These vector products are always for a matrix row vector and a matrix column vector, therefore, for simplicity we do not use the usual vector notation (variable name with an arrow cap), and we do not use transpose symbol for the column vector. We believe that they are easily understood from the context.

2. Rows and columns within diameter d

Let M_1 and M_2 be two input matrices each of size $n \times n$. We assume that all rows are within Hamming distance d in M_1, and similarly all columns are within Hamming distance d in M_2. Hamming distance between two rows (or two columns) is the number of positions at each of which the elements in both vectors are different. For example, the Hamming distance between $(0, 1.23, 3.7, 2.125)$ and $(5, 1.23, 3.7, -7)$ is two because these vectors have different elements in the first and last positions. To compute the pairwise product of all row and column vectors, we can arbitrarily pick one representative row r from matrix M_1 and arbitrarily pick one representative column c from M_2. The observation we use in this case is the following: for all rows r_i in M_1 and all columns c_j in M_2, $r_i c_j = rc - \Delta_i^r c - \Delta_j^c r + \Delta_i^r \Delta_j^c$, where $\Delta_i^r = r - r_i$, and $\Delta_j^c = c - c_j$. Figure 1 summarizes this equality.

We propose the following matrix multiplication algorithm in this case: Arbitrarily pick some row r from the rows of M_1, and arbitrarily pick some column c from the columns of M_2. Calculate rc in time $O(n)$. For all i,j, $1 \leq i, j \leq n$, calculate $\Delta_i^r = r - r_i$, and $\Delta_j^c = c - c_j$. For all i,j, $1 \leq i, j \leq n$, compute $r_i c_j = rc - \Delta_i^r c - \Delta_j^c r + \Delta_i^r \Delta_j^c$.

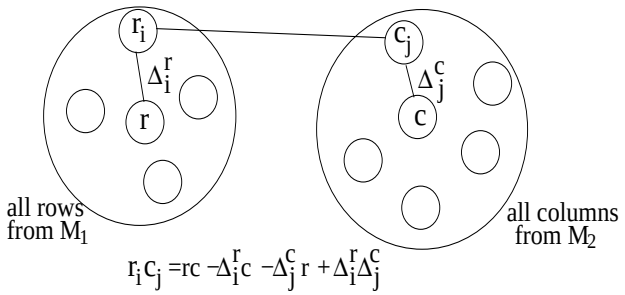


Fig. 1
ALL ROWS IN M_1 AND ALL COLUMNS IN M_2 ARE WITHIN HAMMING DISTANCE d

We store at each node only non-zero elements, and represent Δ_i^r and Δ_j^c in linear linked lists of size (size is in terms of number of nodes) at most d, where in each node we include a vector element and its position (index) in the original vector. Computing all Δ_i^r and Δ_j^c takes $O(n^2)$ time. We compute $\Delta_i^r c$ in $O(d)$ time. We traverse the linked list for Δ_i^r and for every element in the list we read the element of vector c in the same position by using a single read operation in constant time. We multiply these elements and add the product to the running sum for $\Delta_i^r c$. We compute $\Delta_j^c r$ in a similar way in time $O(d)$. We compute $\Delta_i^r \Delta_j^c$ also in time $O(d)$. To do this, we traverse two linked lists (each of size at most d) in parallel linearly (as in the merge step of mergesort), and multiply and add to the running sum when the indices agree. Since rc is already computed and available, computing $r_i c_j$ takes only $O(d)$ time. Therefore, the total time complexity of our algorithm is $O(dn^2)$.

We have implemented our algorithm and performed tests on a computer with 2 GHz processor and 3GB RAM (our programs can be obtained by contacting the authors). Given distance d, and size n, we generate one row randomly and another $n-1$ rows differing from this row in d random positions. This way we generate matrix M_1. We generate columns of matrix M_2 similarly based on an initial randomly generated column. We run our algorithm on matrices generated for different n ($n \in [1000, 2000, 4000, 8000, 16000, 32000]$) and d ($d \in [2, 4, 8, 16, 32]$). We plot the execution time versus n for all d values separately. We show the results in Figure 2. These experimental results verify the theoretical $O(dn^2)$ time complexity, and it shows that our algorithm is very practical for multiplying two input matrices one with rows and the other with columns within diameter d.

We note that for this case of the matrix multiplication (i.e. rows M_1 and columns in M_2 are within diameter d), some other existing methods may also offer time efficient solutions (e.g. matrix multiplication based on LSP decomposition

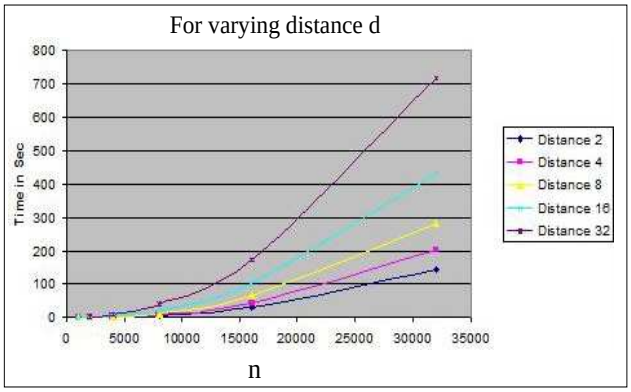


Fig. 2
EXECUTION TIME OF OUR FIRST ALGORITHM VERSUS n FOR VARIOUS DISTANCES d SHOWN IN DIFFERENT COLORS

[11]). However, the algorithm we present in this section is efficient enough for our purposes and it is the basis for a more general algorithm that we present in the next section.

3. Multiple clusters of rows and columns

In this case, we consider placing the rows of input matrix M_1 in k_1 clusters, and similarly, the columns of M_2 in k_2 clusters. We assume that $k_1 = k_2 = k$ for a given integer k ($\leq n$) for simplicity of explanations. We again use the observation that product $r_i c_j$ can be computed using the equation $rc - \Delta_i^r c - \Delta_j^c r + \Delta_i^r \Delta_j^c$, where $\Delta_i^r = r - r_i$, and $\Delta_j^c = c - c_j$, and where r and c (differently from the case in the previous section), respectively, the *representative* (or the head) of clusters that contain r_i and c_j. Again, at each node in linear linked lists for Δ_i^r and Δ_j^c, we store only the non-zero elements along with the indices in the original vectors. We summarize this case in Figure 3.

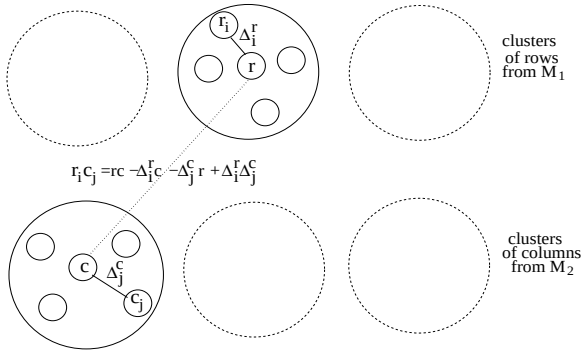


Fig. 3
ROWS OF M_1 AND COLUMNS OF M_2 ARE IN MULTIPLE CLUSTERS

We propose the following algorithm in this case: Given M_1 and M_2, and integer k ($\leq n$), we create k clusters for rows of M_1 and k clusters for columns of M_2. The objective of this step is to minimize the maximum *cluster size* (or equivalently the *cluster radius*). This is an NP-hard problem [9], but we use an approximation algorithm for this step. The algorithm that we use in this step also identifies a representative (head) for each cluster. Then for all i,j, $1 \leq i, j \leq k$, where r_i is the head of row cluster i, and c_j is the head of column cluster j, we compute $r_i c_j$. Computing all $r_i c_j$ takes $O(k^2 n)$ time. Consider a row cluster u whose head is row r, and a column cluster v whose head is column c. For all non-head rows r_i in row cluster u, and for all non-head columns c_j in column cluster v, we compute $r_i c_j = rc - \Delta_i^r c - \Delta_j^c r + \Delta_i^r \Delta_j^c$, where r and c are the cluster heads, and rc has already been computed. Let s be the minimum of the maximum *cluster radius* over all clusters. The cluster radius for a given cluster is defined as the maximum distance to a "center" from all members in the cluster. As we did in the previous section, we store Δ_i^r and Δ_j^c in linear linked lists. The sizes of these linked lists are at most s . Computing all Δ_i^r and Δ_j^c takes $O(n^2)$ time. For every r_i and c_j, computing $-\Delta_i^r c - \Delta_j^c r + \Delta_i^r \Delta_j^c$ takes $O(s)$ time by the same reasons that we discussed for the performance of our algorithm in the previous section. We can see that the total time requirement of our algorithm is $O(sn^2 + k^2 n)$ ignoring the time complexity of the clustering algorithm we use.

As part of our matrix multiplication algorithm, we solve a case of the so-called k-*center problem* [1]. This problem is defined as follows: Given a set S of n points in a metric space, and integer $k \leq n$, compute a k-*clustering* of S of the smallest possible size. A k-clustering of S is a partition of S into k subsets (clusters) $S_1, S_2, \ldots, S_k$. The cluster size is the maximum distance from a fixed point (called the center of the cluster) to members of the cluster. In this case, we use the Hamming distance and we use cluster radius to mean cluster size. The clustering problem we solve is to find a k-clustering with the minimum maximum cluster radius over all clusters. Finding the actual minimum is an NP-hard problem [9]. We use an approximation algorithm presented by Gonzalez [9] (we address another algorithm later). Gonzalez' algorithm [9] guarantees a k-clustering with which the maximum cluster radius over all clusters is at most twice the minimum possible. This algorithm assumes a complete input graph G. We run this algorithm separately for rows of M_1 and columns of M_2. For rows, each row is a vertex in G and we imagine that every two rows are connected by an edge whose weight is the Hamming distance between them. The algorithm is very fast: it does not examine all edges. The case of columns is the same except that the vertices are created for columns. Gonzalez' algorithm [9] is shown in Figure 4.

Gonzalez' algorithm [9] requires $O(kn)$ distance computations, and runs in time $O(kn^2)$ in our case since each Hamming distance computation takes $O(n)$ time. This algorithm

```
Algorithm Gonzalez((G,E,W),k)
Graph G has vertices V, edges E with weights
W
precondition: G is a complete graph, and
k < |V|
Set B_1 = V
Pick one vertex in B_1 and label it head_1
for j = 2 to k do
   Let v_i be a vertex in B_1, B_2, ..., B_{j-1} whose
   distance to the head of the cluster
   it belongs is maximum
   Move v_i to B_j and label it head_j
   For all v_l in {B_1, B_2, ..., B_{j-1}}, move v_l to B_j
   if its distance to v_i is not larger than
   the distance to the head of the cluster it
   belongs
```

Fig. 4

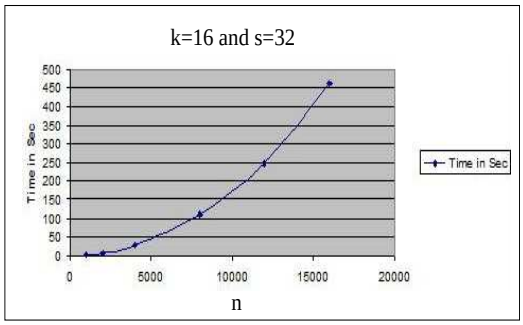
GONZALEZ' CLUSTERING ALGORITHM [9] WHICH APPROXIMATES k-CLUSTERING TO MINIMIZE MAXIMUM CLUSTER RADIUS

guarantees maximum cluster radius at most $2s$, where s is the minimum of the maximum cluster radius over all possible k clusterings. Including this algorithm's time complexity, our algorithm's time complexity is $O(sn^2 + k^2n + kn^2)$ or $O((k+s)n^2)$ since $k \leq n$.

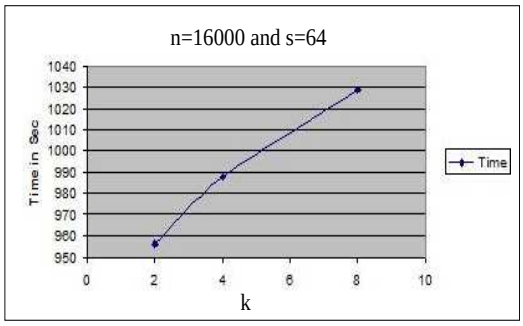
We have implemented Gonzalez' algorithm [9], and our matrix multiplication algorithm for multiple clusters, and performed tests on a computer with 2 GHz processor and 3GB RAM (our programs can be obtained by contacting the authors). Given size n, number of clusters k ($\leq n$), and radius s, we randomly generate k clusters of n rows in which each row differs in s random positions from the representative row, which is also randomly generated. These rows form input matrix M_1. We similarly generate n columns for the other input matrix M_2. We run our algorithm on matrices generated for different n, k, and s, where $n \in [1000, 2000, 4000, 8000, 12000, 16000]$, $1 \leq k \leq \lfloor \log_2 n \rfloor$, and $1 \leq s \leq \lfloor \sqrt{n} \rfloor$. We plot the execution time versus n, k, and s separately when any two of the parameters n, k, and s are fixed. All charts obtained by fixing two parameters to all possible values in the ranges we used exhibit similar growth behavior in the remaining (third) parameter. In Figure 5, we present only a set of representative charts. These experimental results verify the theoretical $O((k+s)n^2)$ time complexity, and it shows that our algorithm is very practical for multiplying two input matrices M_1 and M_2, where rows in M_1 and columns in M_2 (separately) form clusters with small radii.

We note a possible improvement of time complexity of our algorithm for $k = o(n)$. Feder and Greene [7] has improved the time complexity of Gonzalez's algorithm [9] to $O(n \log k)$ from $O(kn)$ (not including the time

A)



B)



C)

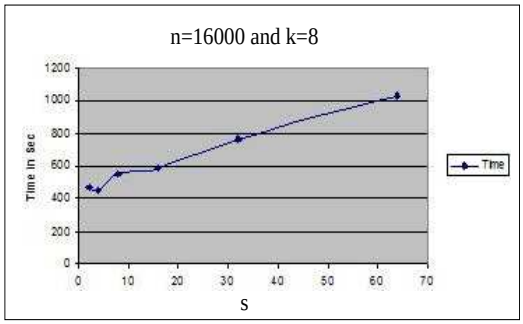


Fig. 5

REPRESENTATIVE CHARTS FOR EXECUTION TIME OF OUR ALGORITHM VERSUS SIZE n, NUMBER OF CLUSTERS k, AND RADIUS s SEPARATELY (WHEN TWO OF THE THREE PARAMETERS ARE FIXED). ALL CHARTS OBTAINED BY FIXING TWO PARAMETERS TO ALL POSSIBLE VALUES IN THE RANGES WE USED EXHIBIT SIMILAR GROWTH BEHAVIOR IN THE REMAINING (THIRD) PARAMETER. PART (A): $k = 16$, $s = 32$, AND THE SIZE n VARIES; PART(B): $n = 16000$, $s = 64$, AND k VARIES; PART (C): $n = 16000$, $k = 8$, AND s VARIES

for distance computations). Feder and Greene [7] achieve this by a sophisticated algorithm that runs in two phases each of which is similar to the Gonzalez' algorithm [9]. The first phase yields a crude solution which is refined in the second phase. This algorithm has the same factor 2 approximation performance guarantee. We can use this algorithm in solving the k-clustering steps in our algorithm. In our case, including the distance computations, each of these steps takes $O(n^2 \log k)$ time. With this modification, our algorithm's worst case total time complexity becomes $O((\log k + s)n^2 + k^2 n)$ time.

4. Discussion and Future Work

We have designed our clustering-based algorithm for multiplying matrices of size $n \times n$ whose rows and columns form k ($\leq n$) clusters with small radius s. In our tests, we assume number of clusters up to $\log n$, and radius up to 64.

We note that our clustering-based matrix multiplication algorithm can use other clustering algorithms with different objectives. The objective affects the total cost of generating all entries (nodes) in the product matrix, and therefore, it needs to be carefully chosen. We ask the following related question:

> **Research Problem**: Are there clustering algorithms (exact or approximate, and other than algorithms in [9], [7]) feasible for preprocessing input matrices (possibly for special cases of matrices, and possibly with different clustering objective) with which we obtain fast algorithms using the clustering-based approach we propose in this section?

We note that there are many clustering algorithms with different objectives (see for example [10] and [8]), which can be used in our approach. We also note that an approximation algorithm that guarantees a constant ratio of the optimum is sufficient for our purposes.

We have introduced a general approach for multiplying two matrices that involve a graph representation, distance definition, and clustering. This approach gives rise to other interesting computational problems for future research.

5. Concluding Remarks

We have proposed a clustering-based algorithm for matrix multiplication. Our algorithm specializes for matrices which form small number of clusters $k \leq n$ (for input matrices of size $n \times n$) with small radius s. The complete vector products are performed for pairs of cluster heads. Other entries are obtained from these by performing additional operations. The algorithm can be implemented to run in time $O((\log k + s)n^2 + k^2 n)$, where k is the given number of clusters, and s is the minimum of the maximum radius in all clusters in an optimal clustering. This work is our attempt for achieving fast algorithms for matrix multiplication for which asymptotically fastest algorithms on general matrices use numerical techniques. If near optimal (close to $\Theta(n^2)$) numerical algorithms exist, we believe that similarly efficient algorithms can be obtained by non-numerical approaches for at least special cases of matrices. This work confirms that matrices whose rows and columns form a small number of clusters with small radius can be multiplied very fast using clustering.

Authors' Contributions

Abdullah N. Arslan developed the main ideas for the matrix multiplication algorithms presented in this paper, analyzed complexity results, and did technical writing of the paper. Arvind Childri implemented these algorithms. In these implementations, Arvind Chidri developed several optimization ideas that improved the performance in practice. Authors created test scenarios together. Arvind Chidri ran the tests, generated the test results, and the figures.

References

[1] P. K. Agarwal and C. M. Procopiuc. Exact and Approximation Algorithms for Clustering (Extended Abstract). *SODA*, pp. 658-667, 1998

[2] A. N. Arslan and A. Chidri. An efficient multiplication algorithm for thin matrices and for matrices with similar rows and columns. *Proceedings of the 2010 International Conference on Scientific Computing(CSC 2010)*, Las Vegas, Nevada, July 12-15, 2010, pp. 147-152, CREA Press, ISBN: 1-60132-137-6, 2010

[3] A. Björklund and A. Lingas. Fast boolean matrix multiplication for highly clustered data. *WADS 2001, LNCS 2125*, pp. 258-163, 2001

[4] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans. Group-theoretic Algorithms for Matrix Multiplication. *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, pp. 379-388, 23-25 October 2005, Pittsburgh, PA

[5] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251-280, 1990

[6] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. *In Proc. 24th Nat. Conf. ACM*, pages 157-172, 1969

[7] T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. *STOC*, pp. 434-444, 1988

[8] L. Gasieniec, J. Jansson, and A. Lingas. Approximation algorithms for Hamming clustering problems. *Journal of Discrete Algorithms*, 2, 289–301, 2004

[9] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38, 2-3, 293-306, 1985

[10] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, Vol. 33, No. 3, pp. 533-550, 1986

[11] C.-P. Jeannerod. LSP matrix decomposition revisited. *Research Report No 2006-28*, INRIA, 2006

[12] S. Robinson. Toward an optimal algorithm for matrix multiplication. *SIAM News*, Volume 38, Number 9, November 20, 2005 (available online: http://www.siam.org/news/news.php?id=174)

[13] V. Strassen. Gaussian Elimination is not Optimal. *Numer. Math.*, 13, pp. 354-356, 1969

On Convergence Properties of One-Dimensional Cellular Automata with *Majority* Cell Update Rule

Predrag T. Tošić, Shankar N. V. Raju
Department of Computer Science, University of Houston, Houston, Texas, USA
ptosic@uh.edu, snvishna@mail.uh.edu

Abstract— *We are interested in simple* cellular automata *(CA) and their computational and dynamical properties. In our past and ongoing work, we have been investigating (i) asymptotic dynamics of various types of CA and (ii) different communication models for CA. In this paper, we specifically focus on the convergence properties of a very simple kind of* totalistic *CA, namely, those defined on one-dimensional arrays where each cell or node updates according to the Boolean Majority function: the new state of a cell becomes 1 if and only if a simple majority of its inputs are currently in state 1, and it becomes 0 otherwise. We have observed in our prior work that such CA tend to have relatively simple asymptotic dynamics: a short transient chain followed by convergence to a "fixed point". We now provide solid statistical evidence for these conjectures, based on our recent extensive computer simulations of Majority 1-D CA. In particular, we study the convergence properties of such CA for two communication models: one is the classical, parallel CA model with perfectly synchronous cell updates, and the other are CA whose cells update sequentially, one at a time; we consider two variants of such sequential update regimes. We simulate CA whose sizes range up to 1,000 cells, and demonstrate very fast (in particular, sublinear), and very slowly decreasing with an increase in the total number of cells, speeds of convergence. Finally, we draw conclusions based on our extensive simulations and outline some interesting questions to be considered in the future work.*

Keywords: models for parallel and distributed computing, cellular automata, simple threshold Boolean functions, asymptotic dynamics, convergence properties

1. Introduction and Motivation

Cellular automata (CA) were originally introduced as an abstract mathematical model of biological systems capable of self-reproduction [13]. CA have been extensively studied in many different domains, especially in the context of modeling and simulation of complex physical, biological, social and socio-technical systems and their collective dynamics; see, e.g., [6], [7], [21], [25], [28], [29], [30]. However, CA have also been viewed as an abstraction of *massively parallel computers* [5], [16], [23]. While most of the previous research in computer and computational sciences on CA and similar models have used these models as abstractions of parallel *hardware architectures*, in our prior and ongoing work we have viewed these discrete dynamical system models as useful abstractions of *open distributed environments* at the *software* level [16], [17]. In particular, we view CA and related Boolean network automata as formal models of *autonomously executing local processes* that are reactive, persistent, and coupled to and interacting with one another. Even when the individual processes are rather simple, their mutual interaction and synergy may, in general, potentially yield a highly complex and difficult to predict *long-term global* behavior [17], [18], [24], [25].

This short paper has two main purposes. On the one hand, we experimentally investigate and validate several conjectures about the overall dynamics and hence possible computations of Majority CA, that were based on mainly analytical and conceptual considerations (but, in several cases not rigorously mathematically proven by either ourselves or, as far as we know, other researchers); see e.g. [21], [23], [24], [25]. On the other hand, our extensive simulations and statistical analysis of simulation results also provide some novel insights into the overall properties of Majority CA dynamics, with implications for various biological, social, socio-technical and computational systems and phenomena that can be modeled as such cellular automata.

We have established elsewhere [20], [21] that the number of possible distinct asymptotic dynamics of Majority CA grows *exponentially* with the number of cells. Consequently, already for the number of cells of the order of hundreds, exhaustive simulations of all possible dynamics is computationally infeasible. We therefore undertake a statistical experimental study, where we randomly sample initial configurations, then evolve a Majority Cellular Automaton (abbreviated as MAJ CA) from such a random initial configuration, then statistically analyze the obtained results with the focus on the speed of convergence to a *fixed point*.

The rest of the paper is organized as follows. We provide formal definitions of CA models and cell update rules of interest in the next section. We briefly review the most relevant prior art. We then summarize and discuss the statistics of our simulation experiments on MAJ CA (with both parallel and sequential cell update regimes) with up to 1,000 cells, and correlate these experimental findings with our prior theoretical results and conjectures about the

MAJ CA convergence properties. Finally, we draw some conclusions and outline several directions for our ongoing and possible future work.

2. Cellular Automata Basics

We formally define classical CA in two steps: we first introduce the notion of a cellular space, and then define a cellular automaton over an appropriate cellular space. We then define the sequential version of the "default" (that is, parallel) CA.

Definition 2.1: A *Cellular Space*, Γ, is an ordered pair (G, Q), where

– G is a regular undirected Cayley graph that may be finite or infinite, with each node labeled with a distinct integer; and

– Q is a finite set of states that has at least two elements, one of which is the special *quiescent state*, denoted by 0.

We denote the set of integer labels of the nodes in Γ by L. That is, L may be equal to, or be a proper subset of, the set of all integers.

Definition 2.2: A *Cellular Automaton* $\boldsymbol{A}$ is an ordered triple (Γ, N, M), where

– Γ is a *cellular space*;

– N is a *fundamental neighborhood*; and

– M is a *finite state machine* whose input alphabet is $Q^{|N|}$, and the local transition function (update rule) for each node is of the form $\delta : Q^{|N|+1} \rightarrow Q$ for *CA with memory*, and $\delta : Q^{|N|} \rightarrow Q$ for *memoryless CA*.

The fundamental neighborhood N specifies which nearby nodes provide inputs to the update rule of a given node. The local transition rule δ specifies how each node updates its state (that is, value), based on its current state, and the current states of its neighbors in N. By composing together the application of the local transition rule to each of the CA's nodes, we obtain *the global map* on the set of global configurations of a CA.

We note that we use the terms *node* and *cell* interchangeably throughout the paper to refer to the elementary single computational unit of a CA.

Definition 2.3: A *Sequential Cellular Automaton* (SCA) $\boldsymbol{S}$ is an ordered quadruple (Γ, N, M, s), where Γ, N and M are as in *Definition 2.2*, and s is an arbitrary sequence, finite or infinite, all of whose elements are drawn from the set L of integers used in labeling the vertices of Γ. The sequence s is specifying the sequential ordering according to which an SCA's nodes update their states, one at a time.

We now adopt a *(discrete) dynamical system* view of CA in order to be able to meaningfully discuss their *dynamics*. A *phase space* of a dynamical system is a directed graph where the vertices are the *global configurations* (or *global states*) of the system, and directed edges correspond to direct transitions from one global state to another. One can define the fundamental types of *global configurations* that a CA can find itself in. These different types of configurations relate to key properties of *asymptotic dynamics* of CA (or other similar models when viewed as discrete dynamical systems). We have been investigating configuration space properties of parallel and sequential CA as they capture *qualitatively distinct* types of possible dynamics of systems abstracted as those various types of CA [17], [21], [22], [26].

We define the fundamental types of dynamical system configurations for parallel CA. These definitions are also applicable to *finite* SCA whose sequential update orderings are required to be *permutations*. In this paper, we only consider such, permutation ordering based, SCA; for a discussion on how to modify the definitions of fundamental types of configurations below in order to make them applicable to SCA with more general sequential update orderings, see [23], [24], [25].

Our classification is based on answering the following question: starting from a given global CA configuration, can this CA return back to that same configuration after a finite number of computational steps?

Definition 2.4: A *fixed point (FP)* is a configuration in the phase space of a CA such that, once the CA reaches this configuration, it stays there forever. A (temporal) *cycle configuration (CC)* is a configuration that, once reached, will be revisited infinitely often with a fixed, finite temporal period of 2 or greater. A *transient configuration (TC)* is a configuration that, once reached, is never going to be revisited again.

In particular, a FP is a special, degenerate case of a recurrent state with period 1. Due to deterministic evolution, any configuration of a classical, parallel CA or a permutation-based sequential CA necessarily has to be exactly one of three: a FP, a "proper" CC, or a TC.

Among various cell update rules for CA, *totalistic* rules based on Boolean functions that are *symmetric* with respect to all of their inputs have been researched particularly extensively (e.g., [27], [28], [29]). Totalistic Boolean update rules, in general, need not be *monotone*. The restricted ones that however are monotone (in addition to being symmetric), have been argued in our prior work to be amenable to mathematical analysis, where many (but not all) interesting properties of the possible resulting dynamics can be explicitly analytically proved. We call such Boolean update rules that are both monotone and symmetric *simple threshold* update rules (see below for definitions). Examples of simple threshold update rules include Boolean AND and OR functions, as well as functions such as "update to 1 if and only if at least 3 out of 7 current inputs are 1", and so on. Arguably the most interesting such rule, in terms of the corresponding CA's dynamics, is the Majority update rule: a cell updates to 1 if and only if a *simple majority* of its inputs (that is, relevant neighboring cells) are currently in state 1.

We next formally define *(simple) linear threshold functions* and the corresponding types of (S)CA. We then focus on the Majority update rule.

Definition 2.5: A *Boolean-valued linear threshold function* of m inputs, $x_1, ..., x_m$, is any function of the form

$$f(x_1, ..., x_m) = \begin{cases} 1, & \text{if } \sum_i w_i \cdot x_i \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where θ is an appropriate *threshold constant*, and $w_1, ..., w_m$ are arbitrary (but fixed) real numbers called *weights*.

A *threshold cellular automaton* (threshold (S)CA) is a (parallel or sequential) cellular automaton where δ is a *Boolean-valued linear threshold function*.

Definition 2.6: A *simple* threshold (S)CA is a cellular automaton whose local update rule δ is a monotone symmetric Boolean threshold function.

As already observed, the most interesting simple threshold CA are those with *Majority* (abbreviated as MAJ) function as the cell update rule [23], [24], [25]. We have extensively analytically studied Majority CA in our prior work, and proven a number of interesting phase space properties for sequential and parallel CA with $\delta = MAJ$ [21], [22], [25]. We have also made some conjectures about overall phase space structures in general, and rates of convergence in particular, in our prior work.

In the present paper, we summarize out simulation results on MAJ CA of small to intermediate sizes (up to 1000 cells) and provide strong statistical evidence that most of our conjectures dating back to the early and mid 2000s were correct. We also share some interesting findings that were not necessarily in accordance with our expectations based on prior mathematical and conceptual analysis of MAJ CA. Therefore, the present paper, whose results are mostly of experimental nature, in essence complements our prior, mostly analytical, investigation of MAJ CA in parallel and sequential settings.

3. Related Work

Various models of *cellular* and *network automata* have been studied in a broad variety of scientific disciplines and research areas, from unconventional models for parallel and distributed computing (e.g., [5], [12], [25]), to complex dynamical systems in physics [6], [7], to theoretical biology [10], [11]. Various sequential and asynchronous variants of CA have also been relatively extensively studied, and in particular, compared and contrasted in various ways with the classical, parallel CA [3], [4], [9], [21], [24], [25].

Computational aspects of the classical Cellular Automata have been investigated in various contexts. Prior to the 1980s, most of the theoretical work dealt with infinite CA and the fundamental (un)decidability results about the global CA properties. Systematic study of a broad variety of computational aspects of CA defined on finite cellular spaces, from topological to formal language theoretic to computational complexity theoretic, was prompted in the 1980s by the seminal work of S. Wolfram [27], [28], [29]. Among other issues, Wolfram addressed the fundamental characteristics of CA in terms of their computational expressiveness and universality. He also offered the first broadly accepted classification of all CA into four qualitatively distinct classes in terms of the structural complexity of the possible computations or, equivalently, dynamics.

Classical CA are a parallel computational model that is characterized by *perfect synchrony* of the parallel node updates. This perfect synchrony implies, in effect, *logical simultaneity*, and is hard to justify on either physics or computer science grounds. By allowing the nodes to update one at a time, one arrives at a sequential version of CA, called *Sequential Cellular Automata* (SCA), and sequential versions of the corresponding more general graph automata [20], [24], [25]. One interesting general question in this context is, what are the differences in asymptotic dynamics of CA with a given local update rule, as a function of the underlying inter-cell communication model? We have analytically addressed this question in our prior work [21], [22], [24], [25]. The present paper complements the theoretical results in our earlier research and offers experimental and statistical comparison-and-contrast, based on extensive simulations, of parallel and sequential CA convergence properties when the cells update according to the Majority rule.

Stable configurations, also known as *fixed points* (FPs), of CA and other discrete dynamical system models have been extensively studied in the literature. Some classical computational problems about FPs include computational hardness (or easiness) of (i) determining the FP existence, (ii) enumerating (exactly or approximately) their number, (iii) determining their reachability from various starting configurations, and (iv) determining the worst-case or average-case convergence to a FP. Our focus in the present paper is on *asymptotic dynamics* of Majority CA, and we contribute to the better understanding of the problem (iv) above, in the context of what are the average or expected convergence speeds to a FP for this restricted type of totalistic Boolean CA. Some references that address various aspects related to FPs in cellular automata, more general graph automata and discrete Hopfield networks include [1], [2], [11], [14], [15], [18], [20], [21], [24], [25].

4. Convergence Properties of Parallel and Sequential Majority CA

We now focus on the main goals of this paper: investigating the convergence properties of *Majority* (MAJ) *one-dimensional* (1D) CA defined on finite cellular spaces, in both parallel and sequential settings. The two main input parameters (beside whether the cells update sequentially or in parallel) are the total number of cells, n, and the Majority update rule radius, r. We assume CA *with memory* throughout; hence, the next state of a cell $c[i]$ depends on the cell's own current state, as well as states of its neighbors:

r neighbors to the left and r neighbors to the right from position i, for the total of $2r + 1$ inputs. Throughout, we assume MAJ CA with *circular boundary conditions*; that is, the underlying cellular spaces in what follows are always *rings* defined on an appropriate number of cells. So, the nearest left neighbor of cell $c[1]$ is cell $c[n]$, the 2nd nearest left neighbor is $c[n-1]$, etc.

Our first specific objective is to determine the implications of the underlying communication model (parallel vs. sequential) on the speed of convergence from a random initial configuration to a fixed point. While (in the case of parallel MAJ CA) it is also possible that (depending on the initial configuration) a temporal two-cycle, as opposed to a FP, is reached [23], [24], [21], we have excluded those (very rare) cases and sampled statistics solely for the "typical" behavior, which is convergence to a FP. (This behavior is typical for parallel MAJ CA, and indeed the only possible for arbitrary sequential CA, as our prior work cited above establishes.) Based on prior theoretical investigations, as well as what is known from scientific large-scale parallel computing (e.g., Gauss-Seidel vs. Jacobi methods in numerical linear algebra), we expected that MAJ SCA, everything else being equal, would converge to a FP faster than corresponding MAJ parallel CA with the same parameter values n and r. This prediction has been corroborated via our simulations, as will be discussed in more detail below.

Our second specific objective is to investigate how the speed of convergence, for a given number of cells, changes with the rule update radius r; we have experimented with the rule radii ranging from $r = 1$ to $r = 5$. For the Majority update rule, cell $c[i]$ updates to 1 if and only if at least $r+1$ of its current updates are in state 1 (since $r + 1$ constitutes a simple majority of $2r + 1$ input values).

To understand the dependence of speed of convergence as a function of rule radius r, one needs to first understand what the typical FP configuration looks like. In our prior work, we have classified MAJ CA FPs into three categories [23], [24]. While there are "atypical" FP configurations that are characterized by particular *spatial symmetries* (whose details depend on n, r and the boundary conditions), a typical fixed point is made of alternating "stable blocks" of consecutive 0s and consecutive 1s, where the size of the minimal stable block depends on r. The extreme cases are special configurations 0^n and 1^n, that are made of all zeros and of all ones, respectively; but (for nontrivial sizes n) the most common FPs are made of some number of smaller stable blocks, where blocks of 0s and blocks of 1s alternate with each other. To illustrate our point here, some examples of such "typical" FPs for MAJ CA on $n = 10$ nodes and with rule radius $r = 1$ (and assuming circular boundary conditions) include 0001111000, 0111001110 and 1110011000.

It is easy to see that, given $r \geq 1$, any block of consecutive $r + 1$ or more zeros is stable, and likewise with blocks of $r + 1$ or more consecutive ones. Hence, for example, a subconfiguration made of two consecutive cells $c[i]c[i+1] = 00$ is stable for $r = 1$, but it is not stable for $r \geq 2$. Consequently, given a fixed number of cells n, (i) the number of FPs of MAJ CA with n nodes monotonically decreases as r increases, and (ii) for the parallel cell update regime, the expected convergence time (number of iterations) to a FP *decreases* as r increases. We have analytically established property (i) in our prior work; we experimentally validate property (ii), at least for the parallel and certain sequential cell update regimes, in the present paper (see below).

We summarize our simulation framework, and then discuss our results in the context of the two main objectives that we have described above. Three sets of simulations were done: one for parallel cell updates and two for different kinds of sequential updates. In one sequential case, the cells update, in each iteration, according to a *fixed* ordering: first node $c[1]$, then $c[2]$, etc., all the way to $c[n]$. Once $c[n]$ has updated, the global iteration of SCA configuration update is done, and the next iteration begins with $c[1]$ updating first, etc. In the second sequential case, at the beginning of each iteration, a random permutation of integers $[1..n]$ is generated, and then the cells are updated according to that permutation. Since a random permutation is generated before each iteration, in general, (i) the node update ordering will differ from one iteration to the next, and (ii) there is no way to "tweak" the speed of convergence based on the node update orderings, as those are (pseudo-)random and hence unpredictable from one global iteration to the next. We shall see shortly that these two different kinds of sequential node update orderings lead to considerably different convergence behaviors.

For a given n, the initial configuration is selected at random, where each cell's initial value is selected randomly and equi-probably to be 0 or 1. This implies, in particular, that (i) the initial cell values are i.i.d random variables and (ii) that a "typical" initial configuration (for large enough n) is going to have roughly the same number of 0s and 1s. Once a random initial configuration is selected, the parallel or sequential CA is evolved fully deterministically (modulo, in the *Sequential2* case, the random choice of node update ordering before each iteration, as explained above).

To obtain statistically significant results, for each scenario, and each combination of values of n and r, a total of 1,000 simulation runs were performed and recorded. Each run starts from a fresh randomly generated initial configuration (hence, for $n \geq 20$, it is extremely unlikely that the same initial configuration will be "hit" twice). The convergence rates, that is, the average number of global iterations until an FP is reached in each scenario, are then determined as averages (arithmetic means) over those 1,000 runs. We have also computed standard deviations for each choice of parameter values (not captured in the plots below, but we plan to use these results in our ongoing and future work).

We note that our Java-based Majority CA simulator is easy to use and has a simple but nice user interface. A snapshot of the simulator's GUI is given in *Figure 0* below. We intend to make a version of the simulator freely available to the scientific community in the near future.

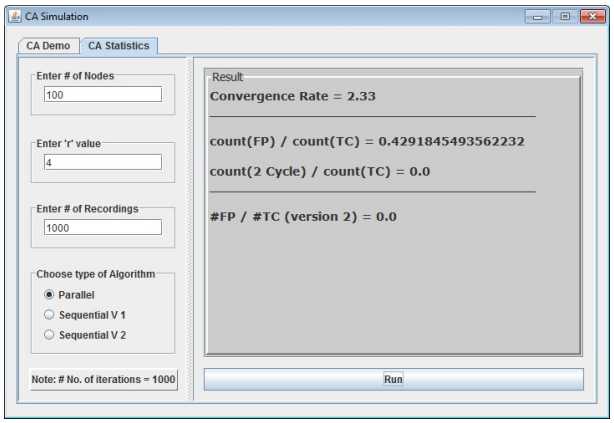


Figure 0: A snapshot of the GUI of our Majority CA simulator.

Next, we briefly summarize predictions of what kind of convergence behavior we expected to see. We will then present our results, followed by discussion where our predictions turned out to be correct, and where they did not – and what are possible explanations for the experimental results we have obtained.

- In both parallel and sequential cases, a typical dynamic evolution of a MAJ CA is a short transient chain followed by reaching a FP (that is, a fairly fast convergence to a FP); if our simulations are done correctly, this is what the statistics we get should corroborate.
- For the same values of n and r, either sequential regime should lead to a faster convergence than the parallel cell update regime.
- For a given n, the speed of convergence should (very slightly) slow down with an increase in r.
- For a given r, the average number of iterations until convergence should very slowly (more specifically, sub-linearly) increase with increase in n.
- For $n \geq 20$ (meaning: one million or more possible configurations), "hitting" probabilistically unlikely events, such as a temporal two-cycle, or reaching the same FP multiple times, should be observed very seldom or never.

One aspect where our intuition turned out to be misleading (and where our prior analytical work on the Majority CA with various communication models shed no light, one way or the other), is the question of whether considerable differences in convergence behaviors between the two sequential models under consideration are to be expected. Since, for each run, new random initial configuration is generated, the particular choice of fixed permutation in method *Sequential1* (we simply chose the permutation $(1, 2, ..., n)$) is expected to be immaterial – it is just as good (or bad) as any other fixed ordering of sequential node updates. Therefore, the question boils down to, how does updating according to an *arbitrary but fixed* sequential ordering compare with updating according to a new (hence, in general, different) random update ordering from one global iteration to the next? Our initial prediction was that there should be no major differences in convergence behaviors; this, however, turns out to be false (see *Figures 2 and 3* below), and we are still seeking explanation of this experimentally observed phenomenon.

In *Figures 1 - 3*, the total number of nodes of a MAJ CA, n, is captured along the x-axis, whereas the y-axis captures the *average* number of iterations until convergence to an FP, from 1,000 randomly generated initial configurations and given the values of n and r.

Figure 1 provides the summary of the speed of convergence statistics for parallel Majority CA:

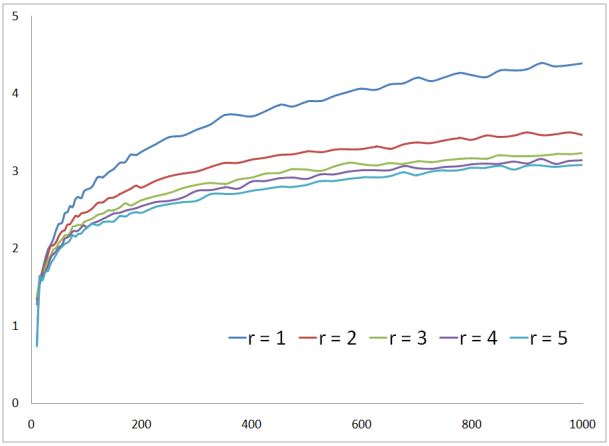


Figure 1: Convergence rates of parallel MAJ CA on up to $n = 1,000$ nodes.

The general behavior pattern – a fairly short transient tail followed by an FP – has certainly been confirmed by our simulations. Also, as predicted, (i) convergence rates are very slowly growing, concave functions of the number of cells n, and (ii) for the same n, larger rule radius r generally implies faster convergence. However, this dependence of convergence rate on r is most pronounced for $r = 1$ as contrasted with $r \geq 2$; but, for $r = 4$ and $r = 5$, the convergence rates already appear practically statistically indistinguishable. Based on the underlying mathematics, we would expect to see some separation of convergence speeds among different values of $r \geq 4$ only once the number of cells n is considerably larger than relatively modest $n = 1000$, the max. number of cells captured by our simulations.

What is the convergence behavior of sequential MAJ CA? The summary of our simulations, for SCA (i) *Sequential1* updating of cells according to the fixed ordering $(1, 2, ..., n)$ and (ii) *Sequential2* updating where a new permutation ordering is randomly generated before each iteration, is captured in Figures 2 and 3, respectively:

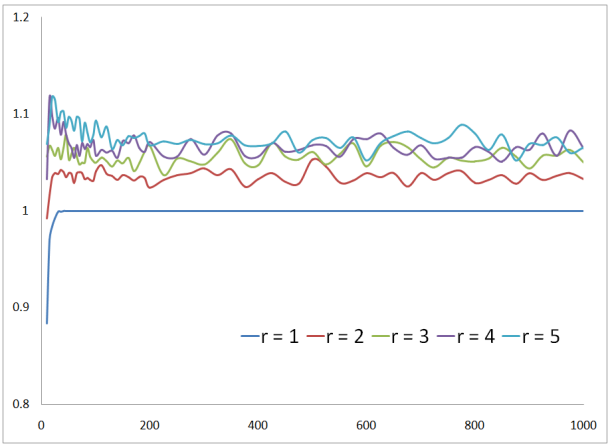


Figure 2: Convergence rates of sequential MAJ CA whose cells update according to the fixed permutation $(1, 2, ..., n)$, and on cellular spaces with up to $n = 1,000$ nodes.

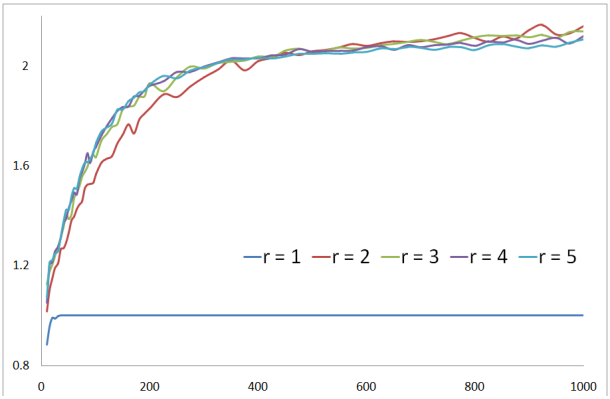


Figure 3: Convergence rates of sequential MAJ CA with random update ordering on each iteration, and on cellular spaces with up to $n = 1,000$ nodes.

When the sequential ordering is generated at random prior to each iteration, we observe exactly what we expected – very similar shapes of the convergence curves to those in the parallel case, except that now the convergence takes place faster. In particular, even for $n = 1000$, and for the values of r we have considered, it (on average) takes only slightly more than 2 global iterations to reach a FP. In contrast, for parallel MAJ CA on $n = 1000$ nodes, the average number of iterations until convergence to a FP is around or below 3 for $r \geq 3$, slightly below 3.5 for $r = 2$, and greater than 4 (but still well below 5) for $r = 1$. Also, the "spread" of convergence rates as a function of r is much wider in the parallel case than in the random sequential case.

One interesting observation about all possible sequential orderings of node updates is the following

Lemma: When $r = 1$, a 1D MAJ CA whose cells update according to any (deterministic or random or mixed) sequential update ordering is guaranteed to converge to a FP in at most one iteration.

Proof: If the initial configuration is an FP, then the convergence takes place in 0 steps. Else, since the cells update sequentially, the temporal cycles aren't possible [24], [25], and therefore the only remaining case is that the initial configuration is transient. As such, it must have one or more cells that are unstable (otherwise, this configuration would be a FP). Since $r = 1$, the only unstable cells are those that aren't a part of a stable block of size 2 or greater – that is, the central cells in subconfigurations 010 and 101. Consider $c[i-1]c[i]c[i+1] = 010$. If the leftmost cell gets to update before the central cell and it changes its value to 1, then the stable block $c[i-1]c[i] = 11$ has formed, and the central cell will never be able to change its state again. Similar analysis applies should the rightmost cell get to update before the central cell, and it updates to 1. The remaining cases are (i) when the central cell gets to update first and (ii) when neither the leftmost nor the rightmost cell changes its value from 0 to 1 before the central cell gets its opportunity to update. Either way, the central cell will necessarily update to 0, and hence the stable block $c[i-1]c[i]c[i+1] = 000$ will form. The other unstable type of subconfiguration, 101, is analyzed analogously. Since these are the only possible *minimal* (with respect to the "is a substring of" partial ordering) unstable subconfigurations, and in all possible scenarios after at most one update of each cell in each such subconfiguration a stable block is created, it follows that, starting from any TC, and for any sequence of node updating (as long as it is a *permutation* [21], [25]), a fixed point will be reached in a single global iteration.

Notice that in *Figures 2-3*, for $r = 1$ and relatively small values of n, the average number of iterations is strictly below 1. The reason is that, for small n, there is a considerable probability that a fixed point will be selected as an initial configuration. Whenever the initial configuration is an FP, the number of iterations is zero, whereas in other cases it is exactly one, thereby resulting in an average of less than 1. Since the statistical dominance of TCs over FPs grows rapidly as n increases [23], [25], and the total number of configurations grows exponentially with n, chance of "hitting" an FP among 1,000 randomly generated initial configurations rapidly approaches zero as n grows. This is why the convergence rate for $r = 1$ in both sequential plots very rapidly levels at 1.

Lastly, one result that we truly did not expect, and still do not have any convincing explanation for: the shapes of the convergence curves for *Sequential1*, where the cells update according to a fixed permutation, and when $r \geq 2$. We have re-run the simulations and obtained similar results and very similarly shaped curves in each case. We are investigating what is the reason for this, rather counterintuitive and unexpected to us, convergence behavior.

5. Summary and Future Work

We investigate one-dimensional cellular automata with the Majority cell update rule. We perform a detailed experimental study of the convergence behavior of such CA

for several different update rule radii, and for both parallel and sequential cell update regimes. Our simulations have generally validated a number of theoretical predictions, including (i) that Majority CA tend to converge to a fixed point very rapidly, where, on average, this convergence rate is clearly a sub-linear[1] function of the number of cells, (ii) that, everything else being equal, the sequential cell updates lead to faster convergence than the parallel cell updates, and (iii) in parallel MAJ CA, the convergence rates tend to get (a little) faster as the rule radius r grows.

However, we have also obtained convergence behavior for the sequential update ordering according to a *fixed* permutation that we cannot explain. Investigating and fully explaining this strange behavior is one of our imminent tasks. In our near-future work, we will investigate CA convergence behavior for much greater numbers of cells n, as well as study occurrences of statistically unlikely patterns, such as the temporal cycles in the parallel case.

Further down the road, we plan to expand the simulator's capabilities so that other cell update rules can be simulated and analyzed. We also intend to first expand the functionality of our simulator and then undertake a similar experimental study to capture when the cell updates take place (i) according to more general sequential regimes than the ones discussed in this paper and (ii) in a genuinely asynchronous manner, as discussed in [21], [22], [23], [25]. We believe that such systematic study will provide valuable insights into the implications of the underlying communication model on the resulting dynamics of various technical, physical and biological systems that can be modeled by cellular automata.

References

[1] Z. Agur. "Fixed Points of Majority Rule Cellular Automata with Applications to Plasticity and Precision of Immune Systems", *Complex Systems* (2), pp. 351 - 357, 1988

[2] C. L. Barrett, H. B. Hunt, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, P. T. Tosic. "Gardens of Eden and Fixed Points in Sequential Dynamical Systems", *Discrete Mathematics and Theoretical Computer Science* (DMTCS), spec. ed. Proc. AA DM–CCG, pp. 95–110, 2001

[3] R. Cori, Y. Metivier and W. Zielonka. "Asynchronous Mappings and Asynchronous Cellular Automata", *Information and Computation*, vol. 106(2), pp. 159Ű202, 1993

[4] N. Fates, E. Thierry, M. Morvan, N. Schabanel. "Fully asynchronous behavior of doublequiescent elementary cellular automata", in *Theoretical Computer Science* , vol. 362 (1-3), pp. 1 Ű 16, 2006

[5] M. Garzon. *"Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks"*, Springer, 1995

[6] E. Goles, S. Martinez. *"Neural and Automata Networks: Dynamical behavior and Applications"*, Math. and Its Applications series (vol. 58), Kluwer, 1990

[7] E. Goles, S. Martinez (eds.). *"Cellular Automata and Complex Systems"*, Nonlinear Phenomena and Complex Systems series, Kluwer, 1999

[8] H. Gutowitz (ed.). *"Cellular Automata: Theory and Experiment"*, The MIT Press / North-Holland, 1991

[9] T. E. Ingerson and R. L. Buvel. "Structure in asynchronous cellular automata", *Physica D: Nonlinear Phenomena*, vol. 10 (1-2), January 1984

[10] S. A. Kauffman. "Metabolic stability and epigenesis in randomly connected nets", *Journal of Theoretical Biology*, vol. 22, pp. 437 – 467, 1969

[11] S. A. Kauffman. "Emergent properties in random complex automata", *Physica D: Nonlinear Phenomena*, vol. 10 (1-2), 1984

[12] M. Mitchell. "Computation in Cellular Automata: A Selected Review", in T. Gramms, S. Bornholdt, M. Gross, M. Mitchell, T. Pellizzari (eds.): *"Nonstandard Computation"*, pp. 95–140, Weinheim: VCH Verlagsgesellschaft, 1998

[13] J. von Neumann. *"Theory of Self-Reproducing Automata"*, (edited and completed by A. W. Burks), Univ. of Illinois Press, Urbana, 1966

[14] P. Orponen. "Computing with truly asynchronous threshold logic networks", *Theoretical Computer Science* (TCS), vol. 174 (1-2), pp. 123 - 136, 1997

[15] K. Sutner. "Computation theory of cellular automata", Proc. MFCS'98 Satellite Workshop on CA, Brno, Czech Rep., 1998

[16] P. Tosic. "A Perspective on the Future of Massively Parallel Computing: Fine-Grain vs. Coarse-Grain Parallel Models", Proc. ACM Computing Frontiers (CF'04), Ischia, Italy, 2004

[17] P. Tosic. "Cellular Automata for Distributed Computing: Models of Agent Interaction and Their Implications", IEEE Int'l Conf. on Systems, Man and Cybernetics (SMC'05), Waikoloa, The Big Island of Hawaii, USA, 2005

[18] P. Tosic. "On the Complexity of Counting Fixed Points and Gardens of Eden in Sequential Dynamical Systems on Planar Bipartite Graphs", *Int'l Journal on Foundations of Computer Science*, vol. 17 (5), pp. 1179–1203, World Scientific, 2006

[19] P. Tosic. "On Modeling and Analyzing Sparsely Networked Large-Scale Multi-agent Systems with Cellular and Graph Automata", *Proc. Int'l Conf. on Computational Science* (ICCS-06), Springer's LNCS, vol. 3993, pp. 272–280, Reading, England, 2006

[20] P. Tosic. "On the complexity of enumerating possible dynamics of sparsely connected Boolean network automata with simple update rules", *Discrete Math. & Theoretical Computer Science* (DMTCS), Proc. AI (Automata 2010: 16th Int'l Workshop on CA & DCS), pp. 125–144, 2010

[21] P. Tosic. "Cellular Automata Communication Models: Comparative Analysis of Parallel, Sequential and Asynchronous CA with Simple Threshold Update Rules", *Int'l J. of Natural Computing Research* (IJNCR), Spec. Issue on CA, vol. 1 (3), pp. 66–84, IGI-Global, July – Sept., 2010

[22] P. Tosic. "On Modeling Large-Scale Multi-Agent Systems with Parallel, Sequential and Genuinely Asynchronous Cellular Automata", *Acta Physica Polonica B*, Proc. Supplement, vol. 4 (2), pp. 217–236, 2011

[23] P. Tosic, G. Agha. "Concurrency vs. Sequential Interleavings in 1-D Threshold Cellular Automata", APDCM Workshop, Proc. of the 18th IEEE Int'l Parallel & Distributed Processing Symposium, Santa Fe, New Mexico, USA, 2004

[24] P. Tosic, G. Agha. "Characterizing Configuration Spaces of Simple Threshold Cellular Automata", Proc. of the 6th Int'l Conference on Cellular Automata for Research and Industry (ACRI'04), Amsterdam, The Netherlands, Springer's LNCS series, vol. 3305, pp. 861–870, 2004

[25] P. Tosic, G. Agha. "Parallel vs. Sequential Threshold Cellular Automata: Comparison and Contrast", Proc. of the First European Conference on Complex Systems (ECCS'05), paper # 251; Euro. Complex Systems Soc., Paris, France, 2005

[26] P. Tosic, G. Agha. "On Computational Complexity of Predicting Dynamical Evolution of Large Agent Ensembles", *Proc. of the 3rd European Workshop on Multiagent Systems* (EUMAS'05), pp. 415–426, Flemish Academy of Sciences, Brussels, Belgium, December 2005

[27] S. Wolfram. "Computation theory of cellular automata", *Communications in Mathematical Physics*, vol. 96, 1984

[28] S. Wolfram. "Twenty problems in the theory of CA", *Physica Scripta* vol. 9, 1985

[29] S. Wolfram (ed.). *"Theory and applications of CA"*, World Scientific, Singapore, 1986

[30] S. Wolfram. *"Cellular Automata and Complexity (collected papers)"*, Addison-Wesley, 1994

[1] We suspect it actually might even be *sub-logarithmic*, but that assertion would require either a mathematical proof or at least more experimental evidence based on simulations on the CA sizes much larger than $n = 1000$.

Stochastic mixed integer second-order cone programming: A new modeling tool for stochastic mixed integer optimization

Baha' M. Alzalg[1] **and K. A. Ariyawansa**[2]
[1,2]Department of Mathematics, Washington State University, Pullman, WA 99164-3113, USA
[1]This material is part of the doctoral dissertation of this author in preparation at Washington State University
[2]The work of this author was supported in part by the US Army Research Office under Award W911NF-08-1-0530

Abstract— *In deterministic mixed integer second-order cone programs (DMISCOPs) we minimize a linear objective function over the intersection of an affine set and a product of second-order (Lorentz) cones, and an additional constraint that requires a subset of the variables attain integers values. We refer to them as* deterministic *mixed integer second-order cone programs since data defining them are deterministic. Stochastic programs have been studied since 1950âĂŹs as a tool to handle optimization problems that involve uncertainty in data. In this paper, we introduce a new modeling tool for stochastic mixed integer optimization to handle uncertainty in data defining DMISOCPs by introducing two-stage stochastic mixed integer second-order cone programs (SMISCOPs) (with recourse). An application of class of problems will be described.*

Keywords: Stochastic programming; Mixed integer programming; Recourse; Second-order cone programming

1. Introduction

In deterministic mixed integer second-order cone programs (DMISCOPs) [6], a linear objective function is minimized over the intersection of an affine set and a product of second-order (Lorentz) cones, and an additional constraint that requires a subset of the variables attain integers values. We refer to them as *deterministic* mixed integer second-order cone programs since data defining them are deterministic. Deterministic 0-1 second-order cone programs (0-1DSCOPs) [6] are DMISCOPs but the variables that must take integer values are restricted to be binary.

In some applications we cannot specify the model entirely because it depends on information which is not available at the time of formulation but will only be determined at some point in the future. Stochastic programs have been studied since 1950âĂŹs to find optimal decisions in problems with uncertainty in data. See [5], [20], [4], [10], [13] and references contained therein. In particular, two-stage stochastic mixed integer linear programs (SMILPs) have been formulated to handle uncertainty in data defining mixed integer linear programs [16]. Some algorithm have been developed recently for solving SMILPs (see for example [15], [14]).

In this paper, we propose a new class of optimization problems to handle uncertainty in data defining DMISOCPs by introducing two-stage stochastic mixed integer second-order cone programs (SMISCOPs) (with recourse). We also describe an application of this new class of problems in stochastic mixed integer optimization,

1.1 Notations

We begin by introducing some notations that we use in the sequel. The notations in this part follows that of Alizadeh and Goldfarb [1] and Todd [18].

Let $\mathbb{R}^{m\times n}$ and $\mathbb{R}^{n\vee n}$ denote the vector spaces of real $m\times n$ matrices and real symmetric $n\times n$ matrices respectively. For $U, V \in \mathbb{R}^{n\vee n}$, we write $U \succeq 0$ ($U \succ 0$) to mean that U is positive semidefinite (positive definite), and $U \succeq V$ or $V \preceq U$ to mean that $U - V \succeq 0$.

We use "," for adjoining vectors and matrices in a row, and use ";" for adjoining them in a column. So, for example, if $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{z}$ are vectors, the following are equivalent:

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix} = (\boldsymbol{x}^{\mathsf{T}}, \boldsymbol{y}^{\mathsf{T}}, \boldsymbol{z}^{\mathsf{T}})^{\mathsf{T}} = (\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{z}).$$

If $\mathcal{A} \subseteq \mathbb{R}^k$ and $\mathcal{B} \subseteq \mathbb{R}^l$, then the Cartesian product of $\mathcal{A}\times\mathcal{B} := \{(\boldsymbol{x};\boldsymbol{y}) : \boldsymbol{x} \in \mathcal{A} \text{ and } \boldsymbol{y} \in \mathcal{B}\}$.

For each vector $\boldsymbol{x} \in \mathbb{R}^k$ indexed from 0, we write $\bar{\boldsymbol{x}}$ for the sub-vector consisting of entries 1 through $k-1$; therefore $\boldsymbol{x} = (x_0; \bar{\boldsymbol{x}})$.

The second-order cone (also known as the quadratic, Lorentz, or the ice-cream cone) of dimension n is defined as $\mathcal{Q}_n := \{\boldsymbol{x} = (x_0; \bar{\boldsymbol{x}}) \in \mathbb{R}\times\mathbb{R}^{n-1} : x_0 \geq ||\bar{\boldsymbol{x}}||\}$ where $||\cdot||$ denotes the Euclidean norm. It is well known that the cone $\mathcal{Q}_2$ is convex, pointed, closed and with a nonempty interior.

We write $\boldsymbol{x} \succeq \mathbf{0}$ to mean that $\boldsymbol{x} \in \mathcal{Q}_n$, and $\boldsymbol{x} \succeq_{\langle n_1,n_2,\cdots,n_r\rangle} \mathbf{0}$ to mean that $\boldsymbol{x} \in \mathcal{Q}_{n_1}\times\mathcal{Q}_{n_2}\times\cdots\times\mathcal{Q}_{n_r}$. For simplicity, we write $\boldsymbol{x} \succeq_{\langle n_1,n_2,\cdots,n_r\rangle} \mathbf{0}$ as $\boldsymbol{x} \succeq_r \mathbf{0}$ when $n_1, n_2, \cdots, n_r$ are known from the context. We also write $\boldsymbol{x} \succeq_r \boldsymbol{y}$ or $\boldsymbol{y} \preceq_r \boldsymbol{x}$ to mean that $\boldsymbol{x} - \boldsymbol{y} \succeq_r \mathbf{0}$.

It is immediately seen that, for every vector $\boldsymbol{x} \in \mathbb{R}^n$ where $n = \sum_{i=1}^r n_i$, $\boldsymbol{x} \succeq_r \mathbf{0}$ if and only if $\boldsymbol{x}$ is partitioned conformally as $\boldsymbol{x} = (\boldsymbol{x_1}; \boldsymbol{x_2}; \cdots; \boldsymbol{x_r})$ and $\boldsymbol{x_i} \succeq \mathbf{0}$ for $i = 1, 2, \cdots, r$.

2. Definitions of SMISOCP with Recourse

An SMISOCP with recourse in primal standard form is defined based on deterministic data $A \in \mathbb{R}^{m_1 \times n_1}, \boldsymbol{b} \in \mathbb{R}^{m_1}$ and $\boldsymbol{c} \in \mathbb{R}^{n_1}$ and random data $T \in \mathbb{R}^{m_2 \times n_1}, W \in \mathbb{R}^{m_2 \times n_2}, \boldsymbol{h} \in \mathbb{R}^{m_2}$ and $\boldsymbol{d} \in \mathbb{R}^{n_2}$ whose realizations depend on an underlying outcome ω in an event space Ω with a known probability function $\mathbb{P}$. Given this data, a two-stage SMISOCP with recourse in *primal standard form* is

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + \mathbb{E}\left[Q(\boldsymbol{x},\omega)\right] \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \succeq_{r_1} \boldsymbol{0} \\ & x_k \in [\alpha_k, \beta_k] \bigcap \mathbb{Z},\ k \in \Gamma \end{array} \tag{1}$$

where r_1 divides $n_1, \Gamma \subset \{1, 2, \cdots, n_1\}$, the first-stage decision variable $\boldsymbol{x} \in \mathbb{R}^{n_1}$ has some of its components x_k ($k \in \Gamma$) with integer values and bounded by $\alpha_k, \beta_k \in \mathbb{R}$, and $Q(\boldsymbol{x}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \boldsymbol{d}(\omega)^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & T(w)\boldsymbol{x} + W(\omega)\boldsymbol{y} = \boldsymbol{h}(\omega) \\ & \boldsymbol{y} \succeq_{r_2} \boldsymbol{0} \\ & y_l \in [\gamma_l, \delta_l] \bigcap \mathbb{Z},\ l \in \Lambda \end{array} \tag{2}$$

where r_2 divides $n_2, \Lambda \subset \{1, 2, \cdots, n_2\}$, the second-stage decision variable $\boldsymbol{y} \in \mathbb{R}^{n_2}$ has some of its components y_l ($l \in \Lambda$) with integer values and bounded by $\gamma_l, \delta_l \in \mathbb{R}$, and

$$\mathbb{E}[Q(\boldsymbol{x},\omega)] := \int_{\Omega} Q(\boldsymbol{x},\omega) P(d\omega).$$

This class of optimization problems may be termed as *stochastic mixed integer second-order cone programs* (SMISOCPs) with recourse. If the integrality constraints in (1) and (2) are restricted to be binary, then we get the problem

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + \mathbb{E}\left[Q(\boldsymbol{x},\omega)\right] \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \succeq_{r_1} \boldsymbol{0} \\ & x_k \in \{0, 1\},\ k \in \Gamma \end{array} \tag{3}$$

where r_1 divides $n_1, \Gamma \subset \{1, 2, \cdots, n_1\}$, the first-stage decision variable $\boldsymbol{x} \in \mathbb{R}^{n_1}$ has some of its components x_k ($k \in \Gamma$) with integer values and bounded by $\alpha_k, \beta_k \in \mathbb{R}$, and $Q(\boldsymbol{x}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \boldsymbol{d}(\omega)^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & T(w)\boldsymbol{x} + W(\omega)\boldsymbol{y} = \boldsymbol{h}(\omega) \\ & \boldsymbol{y} \succeq_{r_2} \boldsymbol{0} \\ & y_l \in \{0, 1\},\ l \in \Lambda \end{array} \tag{4}$$

where r_2 divides $n_2, \Lambda \subset \{1, 2, \cdots, n_2\}$, the first-stage decision variable $\boldsymbol{y} \in \mathbb{R}^{n_2}$ has some of its components y_l ($l \in \Lambda$) with integer values and bounded by $\gamma_l, \delta_l \in \mathbb{R}$, and

$$\mathbb{E}[Q(\boldsymbol{x},\omega)] := \int_{\Omega} Q(\boldsymbol{x},\omega) P(d\omega).$$

This class of optimization problems may be termed as *stochastic 0-1 second-order cone programs* (0-1SSOCPs) with recourse.

3. Two general classes of problems can be cast as SMISOCPs

In this section we describe two general classes of problems that can be cast as MISSOCPs.

3.1 Stochastic mixed integer linear programs

If $r_1 = n_1$, then $x_i \in \mathcal{Q}_2^1 = \{t \in \mathbb{R} : t \geq 0\}$ for each $i = 1, 2, \cdots, n_1$. Thus the constraint $\boldsymbol{x} \succeq_{n_1} \boldsymbol{0}$ means the same as $\boldsymbol{x} \geq \boldsymbol{0}$, i.e., $\boldsymbol{x}$ lies in the nonnegative orthant of $\mathbb{R}^{n_1}$. Similarly, if $n_2 = r_2$ in (2), then $\boldsymbol{y}$ lies in the nonnegative orthant of $\mathbb{R}^{n_2}$. Thus, when both $n_1 = r_1$ in (1) and $n_2 = r_2$ in (2), then the SMISOCP problem (1, 2) reduces to the problem

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + \mathbb{E}\left[Q(\boldsymbol{x},\omega)\right] \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & x_k \in [\alpha_k, \beta_k] \bigcap \mathbb{Z},\ k \in \Gamma \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

where $\Gamma \subset \{1, 2, \cdots, n_1\}$, the first-stage decision variable $\boldsymbol{x} \in \mathbb{R}^{n_1}$ has some of its components x_k ($k \in \Gamma$) with integer values and bounded by $\alpha_k, \beta_k \in \mathbb{R}$, and $Q(\boldsymbol{x}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \boldsymbol{d}(\omega)^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & T(w)\boldsymbol{x} + W(\omega)\boldsymbol{y} = \boldsymbol{h}(\omega) \\ & y_l \in [\gamma_l, \delta_l] \bigcap \mathbb{Z},\ l \in \Lambda \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{array}$$

where $\Lambda \subset \{1, 2, \cdots, n_2\}$, the second-stage decision variable $\boldsymbol{y} \in \mathbb{R}^{n_2}$ has some of its components y_l ($l \in \Lambda$) with integer values and bounded by $\gamma_l, \delta_l \in \mathbb{R}$, and

$$\mathbb{E}[Q(\boldsymbol{x},\omega)] := \int_{\Omega} Q(\boldsymbol{x},\omega) P(d\omega).$$

Thus, SMILP problems can be cast as SMISOCP problems.

3.2 Stochastic mixed integer quadratic programs

Stochastic quadratic programs (SMIQPs) can also be cast as SMISOCPs. To demonstrate this, recall that a two-stage SMIQP (with recourse) is defined based on deterministic data $C \in \mathbb{R}^{n_1 \vee\ n_1}, C \succ 0, \boldsymbol{c} \in \mathbb{R}^{n_1}, A \in \mathbb{R}^{m_1 \times n_1}$ and $\boldsymbol{b} \in \mathbb{R}^{m_1}$; and random data $H \in \mathbb{R}^{n_2 \vee\ n_2}, H \succ 0, \boldsymbol{d} \in \mathbb{R}^{n_2}, T \in \mathbb{R}^{m_2 \times n_1}, W \in \mathbb{R}^{m_2 \times n_2}$, and $\boldsymbol{h} \in \mathbb{R}^{m_2}$ whose realizations depend on an underlying outcome in an event space Ω with

a known probability function $\mathbb{P}$. Given this data, an SMIQP with recourse is

$$\begin{array}{ll} \min & q_1(\boldsymbol{x},\omega) = \boldsymbol{x}^\mathsf{T} C\boldsymbol{x} + \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + \mathbb{E}[Q(\boldsymbol{x},\omega)] \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & x_k \in \{0,1\},\ k \in \Gamma \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array} \tag{5}$$

where $\Gamma \subset \{1, 2, \cdots, n_1\}$, the first-stage decision variable $\boldsymbol{x} \in \mathbb{R}^{n_1}$ has some of its components x_k ($k \in \Gamma$) with integer values and bounded by $\alpha_k, \beta_k \in \mathbb{R}$, and $Q(\boldsymbol{x},\omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & q_2(\boldsymbol{y},\omega) = \boldsymbol{y}^\mathsf{T} H(\omega)\boldsymbol{y} + \boldsymbol{d}(\omega)^\mathsf{T}\boldsymbol{y} \\ \text{s.t.} & T(\omega)\boldsymbol{x} + W(\omega)\boldsymbol{y} = \boldsymbol{h}(\omega) \\ & y_l \in [\gamma_l, \delta_l] \bigcap \mathbb{Z},\ l \in \Lambda \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{array} \tag{6}$$

where $\Lambda \subset \{1, 2, \cdots, n_2\}$, the second-stage decision variable $\boldsymbol{y} \in \mathbb{R}^{n_2}$ has some of its components y_l ($l \in \Lambda$) with integer values and bounded by $\gamma_l, \delta_l \in \mathbb{R}$, and

$$\mathbb{E}[Q(\boldsymbol{x},\omega)] := \int_\Omega Q(\boldsymbol{x},\omega) P(d\omega).$$

Observe that the objective function of (5) can be written as (see [1])

$$q_1(\boldsymbol{x}_1,\omega) = ||\bar{\boldsymbol{u}}||^2 + \mathbb{E}[Q(\boldsymbol{x},\omega)] - \frac{1}{4}\boldsymbol{c}^\mathsf{T} C^{-1}\boldsymbol{c},$$

where

$$\bar{\boldsymbol{u}} = C^{1/2}\boldsymbol{x} + \frac{1}{2}C^{-1/2}\boldsymbol{c}.$$

Similarly, the objective function of (6) can be written as

$$q_2(\boldsymbol{y},\omega) = ||\bar{\boldsymbol{v}}||^2 - \frac{1}{4}\ \boldsymbol{d}(\omega)^\mathsf{T} H(\omega)^{-1}\boldsymbol{d}\ (\omega)$$

where

$$\bar{\boldsymbol{v}} = H(\omega)^{1/2}\boldsymbol{y} + \frac{1}{2}H(\omega)^{-1/2}\boldsymbol{d}(\omega).$$

Thus, problem (5, 6) can be transformed into the SMISOCP:

$$\begin{array}{ll} \min & u_0 \\ \text{s.t.} & \bar{\boldsymbol{u}} - C^{1/2}\boldsymbol{x} = \frac{1}{2}\ C^{-1/2}\boldsymbol{c} \\ & A\boldsymbol{x} = \boldsymbol{b} \\ & x_k \in [\alpha_k, \beta_k] \bigcap \mathbb{Z},\ k \in \Gamma \\ & (u_0; \bar{\boldsymbol{u}}) \succeq \boldsymbol{0} \\ & \boldsymbol{x} \geq 0 \end{array} \tag{7}$$

where $Q(\boldsymbol{x},\omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & v_0 \\ \text{s.t.} & \bar{\boldsymbol{v}} - H(\omega)^{1/2}\boldsymbol{y} = \frac{1}{2}\ H(\omega)^{-1/2}\boldsymbol{d}(\omega) \\ & T(\omega)\boldsymbol{x} + W(\omega)\boldsymbol{y} = \boldsymbol{h}(\omega) \\ & y_l \in [\gamma_l, \delta_l] \bigcap \mathbb{Z},\ l \in \Lambda \\ & (u_0; \bar{\boldsymbol{v}}) \succeq \boldsymbol{0} \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{array} \tag{8}$$

where

$$\mathbb{E}[Q(\boldsymbol{x},\omega)] := \int_\Omega Q(\boldsymbol{x},\omega) P(d\omega).$$

Note that the SMIQP problem (5, 6) and the SMISOCP problem (7, 8) will have the same minimization, but their optimal objective values are equal up to constants. More precisely, the difference between the optimal objective values of (6, 8) would be $-\frac{1}{2}\boldsymbol{d}(\omega)^\mathsf{T} H(\omega)^{-1}\ \boldsymbol{d}(\omega)$. Similarly, the optimal objective values of (5, 6) and (7, 8) will differ by

$$-\frac{1}{2}\ \boldsymbol{c}^\mathsf{T} C^{-1}\boldsymbol{c} - \frac{1}{2}\int_\Omega \left(\boldsymbol{d}(\omega)^\mathsf{T}\ H(\omega)^{-1}\boldsymbol{d}(\omega)\right) P(d\omega).$$

It is interesting to note that we can use the transformation described in this part to formulate an SMISOCP model for capital budgeting problems with a mean-variance objective described in [2]. In [2] the authors ignored the financing structure and considered a simple assumption that all given projects have a *fixed* available budget, and then, in order to fit their approach for deriving cutting planes, they transformed the problem from its model in 0-1DQP into a 0-1DSOCP model. But we believe that it is much closer to the reality to assume that we have a *random* budget for the projects. Consequently, it is more convenient to consider the stochastic version of this problem and hence to transform it from the resulting 0-1SQP model into a 0-1SSOCP model.

4. An application: Stochastic discrete Euclidean facility location problems

In *facility location problems* (FLPs) we are interested in choosing a location to build a new facility or locations to build multiple new facilities so that an appropriate measure of distance from the new facilities to existing facilities is minimized. FLPs arise locating airports, regional campuses, wireless communications towers, etc. The following are some ways of classifying FLPs (see also [17]):

- We can classify FLPs based on the number of new facilities in the following sense: if we add only one new facility then we get a problem known as a *single facility location problem* (SFLP), while if we add multiple new facilities instead of adding only one, then we get more a general problem known as a *multiple facility location problem* (MFLP).
- Another way of classification is based on the distance measure used in the model between the facilities. If we use the Euclidean distance then these problems are called *Euclidean facility location problems* (EFLPs), if we use the rectilinear distance then these problems are called *rectilinear facility location problems* (RFLPs).
- When the new facilities can be placed any place in solution space, the problem is called a *continuous facility location problem* (CFLP), but usually the investor needs the new facilities to be placed within *specific locations* (called nodes) and not in any place in the solution space. In this case the problem is called a *discrete facility location problem* (DFLP).
- In some applications, the locations of existing facilities cannot be fully specified because the locations of some

of them depend on information not available at the time when decision needs to be made but will only be available at a later point in time. In this case, we are interested in *stochastic facility location problems*. When the locations of all old facilities are fully specified, FLPs are called *deterministic facility location problems*.

FLPs have seen a great deal of recent research activity. For further details, consult the book of Tompkins and *et al.* [17]. In particular, deterministic Euclidean facility location problems are often cited as an application of deterministic second-order cone programs (see for example [19] and [11]). In this subsection, we consider (both single and multiple) stochastic discrete Euclidean facility location problems when, in particular, some of the variables are restricted to be integer variables.

4.1 Stochastic discrete Euclidean single facility location problem

In deterministic Euclidean single facility location problems, we are interested in choosing a location to build a new facility among existing facilities so that this location minimizes the sum of a weighted distance to all existing facilities.

Assume that we are given r existing facilities represented by the fixed points $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_r$ in $\mathbb{R}^n$, and we plan to place a new facility represented by $\boldsymbol{x}$ so that we minimize the weighted sum of the distances between $\boldsymbol{x}$ and each of the points $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_r$. This leads us to the problem

$$\min \quad \sum_{i=1}^{r} w_i \, ||\boldsymbol{x} - \boldsymbol{a}_i||$$

or, alternatively, to the problem

$$\begin{array}{ll} \min & \sum_{i=1}^{r} w_i \, t_i \\ \text{s.t.} & (t_1; \boldsymbol{x} - \boldsymbol{a}_1; \cdots; t_r; \boldsymbol{x} - \boldsymbol{a}_r) \succeq_r \boldsymbol{0} \end{array}$$

where w_i is the weight associated with the ith existing facility and the new facility for $i = 1, 2, \ldots, r$.

Before we describe the stochastic version of this generic application, we indicate a more concrete version of it. Assume that we have a new city with many suburbs and we want to build a hospital for treating the residents of this city. Some people live in the city at the present time. As the city expands, many houses in new suburbs need to be built and the locations of these suburbs will be known in the future in different sides of the city. Our goal is to find the best location of this hospital so that it can serve the current suburbs and the new ones. This location must be determined at the current time and before information about the locations of the new suburbs become available.

Generally speaking, let $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{r_1}$ be fixed points in $\mathbb{R}^n$ representing the coordinates of r_1 existing fixed facilities and $\tilde{\boldsymbol{a}}_1(\omega), \tilde{\boldsymbol{a}}_2(\omega), \cdots, \tilde{\boldsymbol{a}}_{r_2}(\omega)$ be random points in $\mathbb{R}^n$ representing the coordinates of r_2 random facilities who realizations depends on an underlying outcome ω in an event space Ω with a known probability function $\mathbb{P}$.

Suppose that at present we do not know the realizations of r_2 random facilities, and that at some point in time in future the realizations of these r_2 random facilities become known.

Our goal is to locate a new facility $\boldsymbol{x}$ that minimizes the weighted sums of the distance between the new facility and each one of the existing fixed facilities and also minimizes the expected weighted sums of the distance between the new facility and the realization of each one of the random facilities. Note that this decision needs to be made before the realizations of the r_2 random facilities become available. We consider the discrete version of the problem by assuming that the new facility needs to be placed within specific locations and not in any place in 2- or 3- (or higher) dimensional space. Let the points $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k \in \mathbb{R}^n$ represent these specific locations. So, we add the constraint $\boldsymbol{x} \in \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$. Clearly, the above constraint can be replaced by the following linear and binary constraints:

$$\begin{array}{l} \boldsymbol{x} = \boldsymbol{v}_1 \, y_1 + \boldsymbol{v}_2 \, y_2 + \cdots + \boldsymbol{v}_k \, y_k, \\ y_1 + y_2 + \cdots + y_k = 1, \text{ and} \\ (y_1, y_2, \cdots, y_k) \in \{0,1\}^k. \end{array}$$

This leads us to the following 0-1SSOCP model:

$$\begin{array}{ll} \min & \sum_{i=1}^{r_1} w_i \, t_i + \mathbb{E}\,[Q(\boldsymbol{x}; \boldsymbol{y}, \omega)] \\ \text{s.t.} & (t_1; \boldsymbol{x} - \boldsymbol{a}_1; \cdots; t_{r_1}; \boldsymbol{x} - \boldsymbol{a}_{r_1}) \succeq_{r_1} \boldsymbol{0} \\ & \boldsymbol{x} = \boldsymbol{v}_1 \, y_1 + \boldsymbol{v}_2 \, y_2 + \cdots + \boldsymbol{v}_k \, y_k \\ & \boldsymbol{1}^{\mathsf{T}} \, \boldsymbol{y} = 1, \; \boldsymbol{y} \in \{0,1\}^k \end{array}$$

where $Q(\boldsymbol{x}; \boldsymbol{y}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \sum_{j=1}^{r_2} \tilde{w}_j(\omega) \, \tilde{t}_j \\ \text{s.t.} & (\tilde{t}_1; \boldsymbol{x} - \tilde{\boldsymbol{a}}_1(\omega); \cdots; \tilde{t}_{r_2}; \boldsymbol{x} - \tilde{\boldsymbol{a}}_{r_2}(\omega)) \succeq_{r_2} \boldsymbol{0} \\ & \boldsymbol{x} = \boldsymbol{v}_1 \, y_1 + \boldsymbol{v}_2 \, y_2 + \cdots + \boldsymbol{v}_k \, y_k \\ & \boldsymbol{1}^{\mathsf{T}} \, \boldsymbol{y} = 1, \; \boldsymbol{y} \in \{0,1\}^k \end{array}$$

and

$$\mathbb{E}[Q(\boldsymbol{x}; \boldsymbol{y}, \omega)] := \int_{\Omega} Q(\boldsymbol{x}; \boldsymbol{y}, \omega) P(d\omega).$$

where w_i is the weight associated with the ith existing facility and the new facility for $i = 1, 2, \ldots, r_1$ and $\tilde{w}_j(\omega)$ is the weight associated with the jth random existing facility and the new facility for $j = 1, 2, \ldots, r_2$.

Sometimes we may need the specific points have to attain integer values. In most cities of the world that were planned, streets are laid out on a grid plan, so that city is subdivided into small numbered blocks that are square or rectangular. Figure 2 shows the blocks of Chicago in 1857. In this case, usually the investor needs the new facility to be placed on of the corners of the city blocks. Thus, let us assume that the variable $\boldsymbol{x}$ lies in the hyperrectangle $\Xi^n := \{\boldsymbol{x} : \boldsymbol{\zeta} \leq \boldsymbol{x} \leq \boldsymbol{\eta}, \boldsymbol{\zeta} \in \mathbb{R}^n, \boldsymbol{\eta} \in \mathbb{R}^n\}$ and has to attain specific points

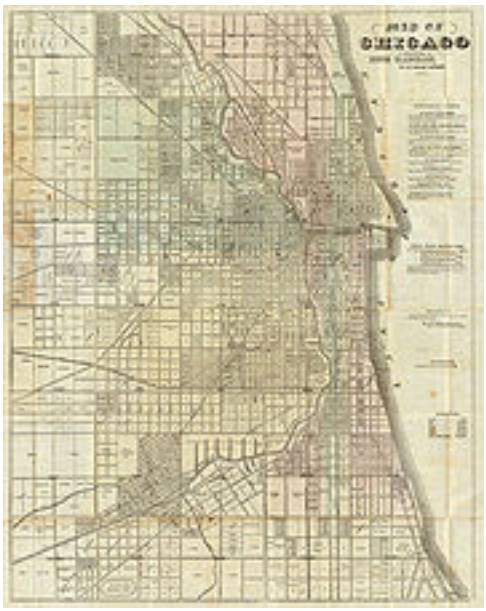


Fig. 1: The regular pattern of square or rectangular city blocks is very common among American cities. This map shows the blocks of Chicago in 1857.

in the grid $\Xi^n \bigcap \mathbb{Z}^n$. Then we simply solve the following SMISOCP model:

$$\begin{array}{ll} \min & \sum_{i=1}^{r_1} w_i\ t_i + \mathbb{E}\ [Q(\boldsymbol{x}, \omega)] \\ \text{s.t.} & (t_1; \boldsymbol{x} - \boldsymbol{a}_1; \cdots; t_{r_1}; \boldsymbol{x} - \boldsymbol{a}_{r_1}) \succeq_{r_1} \mathbf{0} \\ & \boldsymbol{x} \in \Xi^n \bigcap \mathbb{Z}^n \end{array}$$

where $Q(\boldsymbol{x}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \sum_{j=1}^{r_2} \tilde{w}_j(\omega)\ \tilde{t}_j \\ \text{s.t.} & (\tilde{t}_1; \boldsymbol{x} - \tilde{\boldsymbol{a}}_1(\omega); \cdots; \tilde{t}_{r_2}; \boldsymbol{x} - \tilde{\boldsymbol{a}}_{r_2}(\omega)) \succeq_{r_2} \mathbf{0} \\ & \boldsymbol{x} \in \Xi^n \bigcap \mathbb{Z}^n \end{array}$$

and

$$\mathbb{E}[Q(\boldsymbol{x}, \omega)] := \int_\Omega Q(\boldsymbol{x}, \omega) P(d\omega).$$

4.2 Stochastic discrete Euclidean multiple facility location problem

If we consider the concrete model described in §4, and suppose that we want to build three hospitals for this city, or build a hospital, a university, and a fire station then we get a multiple facility version of the model. Generally, in order to be precise only the latest information of the random facilities is used. This may require an increasing or decreasing of the number of the new facilities after the latest information about the random facilities become available. For simplicity, let us assume that the number of new facilities was previously known and fixed and we add m new facilities, namely $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m \in \mathbb{R}^n$, instead of adding only one. We also assume that the variables $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ need to be placed within specific locations represented by the points $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k \in \mathbb{R}^n$.

We have two cases depending whether or not there is an interaction among the new facilities in the underlying model. If there is no interaction between the new facilities, we are just concerned in minimizing the weighted sums of the distance between each one of the new facilities on one hand and each one of the fixed facilities and the realization of each one of the random facilities. In other words, we solve the following 0-1SSOCP model:

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_1} w_{ij}\ t_{ij} + \mathbb{E}\ [Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega)] \\ \text{s.t.} & (t_{1j}; \boldsymbol{x}_j - \boldsymbol{a}_1; \cdots; t_{r_1 j}; \boldsymbol{x}_j - \boldsymbol{a}_{r_1}) \succeq_{r_1} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \boldsymbol{x}_j = \boldsymbol{v}_1\ y_1 + \boldsymbol{v}_2\ y_2 + \cdots + \boldsymbol{v}_k\ y_k \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \mathbf{1}^{\mathsf{T}}\ \boldsymbol{y} = 1,\ \boldsymbol{y} \in \{0, 1\}^k, \end{array}$$

where $Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_2} \tilde{w}_{ij}(\omega)\ \tilde{t}_{ij} \\ \text{s.t.} & (\tilde{t}_{1j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_1(\omega); \cdots; \tilde{t}_{r_2 j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_{r_2}(\omega)) \succeq_{r_2} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \boldsymbol{x}_j = \boldsymbol{v}_1\ y_1 + \boldsymbol{v}_2\ y_2 + \cdots + \boldsymbol{v}_k\ y_k \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \mathbf{1}^{\mathsf{T}}\ \boldsymbol{y} = 1,\ \boldsymbol{y} \in \{0, 1\}^k, \end{array}$$

and

$$\mathbb{E}[Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega)] := \int_\Omega Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega) P(d\omega).$$

where w_{ij} is the weight associated with the ith existing facility and the jth new facility for $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, r_1$, and $\tilde{w}_{ij}(\omega)$ is the weight associated with the ith random existing facility and the jth new facility for $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, r_2$.

If interaction exists among the new facilities, then, in addition to the above requirements, we need to minimize the sum of the Euclidean distances between each pair of the new facilities. In this case, we are interested in a model of the form:

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_1} w_{ij} t_{ij} + \sum_{j=2}^{m} \sum_{j'=1}^{j-1} \hat{w}_{jj'} \hat{t}_{jj'} \\ & + \mathbb{E}\ [Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega)] \\ \text{s.t.} & (t_{1j}; \boldsymbol{x}_j - \boldsymbol{a}_1; \cdots; t_{r_1 j}; \boldsymbol{x}_j - \boldsymbol{a}_{r_1}) \succeq_{r_1} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & (\hat{t}_{j(j+1)}; \boldsymbol{x}_j - \boldsymbol{x}_{j+1}; \cdots; \hat{t}_{jm}; \boldsymbol{x}_j - \boldsymbol{x}_m) \succeq_{(m-j)} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m-1 \\ & \boldsymbol{x}_j = \boldsymbol{v}_1\ y_1 + \boldsymbol{v}_2\ y_2 + \cdots + \boldsymbol{v}_k\ y_k \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \mathbf{1}^{\mathsf{T}}\ \boldsymbol{y} = 1,\ \boldsymbol{y} \in \{0, 1\}^k, \end{array}$$

where $Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_2} \tilde{w}_{ij}(\omega)\, \tilde{t}_{ij} + \sum_{j=2}^{m} \sum_{j'=1}^{j-1} \hat{w}_{jj'} \hat{t}_{jj'} \\ \text{s.t.} & (\tilde{t}_{1j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_{\boldsymbol{1}}(\omega); \cdots; \tilde{t}_{r_2 j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_{\boldsymbol{r_2}}(\omega)) \succeq_{r_2} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & (\hat{t}_{j(j+1)}; \boldsymbol{x}_j - \boldsymbol{x}_{j+1}; \cdots; \hat{t}_{jm}; \boldsymbol{x}_j - \boldsymbol{x}_m) \succeq_{(m-j)} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m-1 \\ & \boldsymbol{x}_j = \boldsymbol{v}_1\, y_1 + \boldsymbol{v}_2\, y_2 + \cdots + \boldsymbol{v}_k\, y_k \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \mathbf{1}^{\mathsf{T}}\, \boldsymbol{y} = 1, \ \boldsymbol{y} \in \{0, 1\}^k, \end{array}$$

and

$$\mathbb{E}[Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega)] := \int_{\Omega} Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m; \boldsymbol{y}, \omega) P(d\omega).$$

where $\hat{w}_{jj'}$ is the weight associated with the new facilities j' and j for $j' = 1, 2, \ldots, j-1$ and $j = 2, 3, \ldots, m$.

If we need some specific points have to attain integer values, then for each $k \in \Delta \subset \{1, 2, \cdots, m\}$, we assume that the variable $\boldsymbol{x}_k$ lies in the hyperrectangle $\Xi_k^n \equiv \{\boldsymbol{x}_k : \boldsymbol{\zeta}_k \leq \boldsymbol{x}_k \leq \boldsymbol{\eta}_k, \boldsymbol{\zeta}_k \in \mathbb{R}^n, \boldsymbol{\eta}_k \in \mathbb{R}^n\}$ and has to be integer-valued, i.e. $\boldsymbol{x}_k$ must be in the grid $\Xi_k^n \bigcap \mathbb{Z}^n$.

Thus, if there is no interaction between the new facilities, we solve the following SMISOCP model:

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_1} w_{ij}\, t_{ij} + \mathbb{E}\, [Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_{r_1}, \omega)] \\ \text{s.t.} & (t_{1j}; \boldsymbol{x}_j - \boldsymbol{a}_1; \cdots; t_{r_1 j}; \boldsymbol{x}_j - \boldsymbol{a}_{r_1}) \succeq_{r_1} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \boldsymbol{x}_k \in \Xi_k^n \bigcap \mathbb{Z}^n, \ k \in \Delta, \end{array}$$

where $Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_2} \tilde{w}_{ij}(\omega)\, \tilde{t}_{ij} \\ \text{s.t.} & (\tilde{t}_{1j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_1(\omega); \cdots; \tilde{t}_{r_2 j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_{r_2}(\omega)) \succeq_{r_2} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & \boldsymbol{x}_k \in \Xi_k^n \bigcap \mathbb{Z}^n, \ k \in \Delta, \end{array}$$

and

$$\mathbb{E}[Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m, \omega)] := \int_{\Omega} Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m, \omega) P(d\omega).$$

If interaction exists among the new facilities, then we are interested in a model of the form:

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_1} w_{ij} t_{ij} + \sum_{j=2}^{m} \sum_{j'=1}^{j-1} \hat{w}_{jj'} \hat{t}_{jj'} \\ & + \mathbb{E}\, [Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_{r_1}, \omega)] \\ \text{s.t.} & (t_{1j}; \boldsymbol{x}_j - \boldsymbol{a}_1; \cdots; t_{r_1 j}; \boldsymbol{x}_j - \boldsymbol{a}_{r_1}) \succeq_{r_1} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & (\hat{t}_{j(j+1)}; \boldsymbol{x}_j - \boldsymbol{x}_{j+1}; \cdots; \hat{t}_{jm}; \boldsymbol{x}_j - \boldsymbol{x}_m) \succeq_{(m-j)} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m-1 \\ & \boldsymbol{x}_k \in \Xi_k^n \bigcap \mathbb{Z}^n, \ k \in \Delta, \end{array}$$

where $Q(\boldsymbol{x}_1; \cdots; \boldsymbol{x}_m, \omega)$ is the minimum of the problem

$$\begin{array}{ll} \min & \sum_{j=1}^{m} \sum_{i=1}^{r_2} \tilde{w}_{ij}(\omega)\, \tilde{t}_{ij} \\ & + \sum_{j=2}^{m} \sum_{j'=1}^{j-1} \hat{w}_{jj'} \hat{t}_{jj'} \\ \text{s.t.} & (\tilde{t}_{1j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_{\boldsymbol{1}}(\omega); \cdots; \tilde{t}_{r_2 j}; \boldsymbol{x}_j - \tilde{\boldsymbol{a}}_{\boldsymbol{r_2}}(\omega)) \succeq_{r_2} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m \\ & (\hat{t}_{j(j+1)}; \boldsymbol{x}_j - \boldsymbol{x}_{j+1}; \cdots; \hat{t}_{jm}; \boldsymbol{x}_j - \boldsymbol{x}_m) \succeq_{(m-j)} \mathbf{0} \\ & \text{where } \ j = 1, 2, \cdots, m-1 \\ & \boldsymbol{x}_k \in \Xi_k^n \bigcap \mathbb{Z}^n, \ k \in \Delta, \end{array}$$

5. Future research directions

In this paper we introduced a new class of problems for stochastic mixed integer programming that may be referred as stochastic mixed integer second-order cone programs with recourse. Stochastic mixed integer second-order cone programs generalize both stochastic mixed integer linear programs and stochastic mixed integer quadratic programs. Our development is indeed significant in value, because it gives us a new methodology to cover those applications that cannot be captured by stochastic mixed integer linear and quadratic programs. In terms of modeling, beyond the application described in §4, it would be interesting to investigate other applications of this new class of optimization problems. For example, in [8] Fampa and Maculan proposed a deterministic mixed integer second-order cone programming formulation of the Euclidean Steiner tree problem (in which the set of nodes in the connection is *fixed* over time). Based on this formulation, we can describe a stochastic mixed integer second-order cone programming formulation of a related problem called *dynamic* Euclidean Steiner tree problem (where the set of nodes in the connection changes over time) proposed by Imase and Waxman in [9] and motivated by multipoint routing in communication networks.

It is useful to develop algorithm for SMISOCPs. A forthcoming paper will focus on developing a decomposition-based branch-and-bound algorithm for solving this new class of problems by extending the work of Sherali and Zhu [15] (see also [14]).

References

[1] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Program.*, Ser. B 95:3–51, 2003.

[2] A. Atamtürk, V. Narayanan. Conic mixed-integer rounding cuts. *Math. Program.*, Ser. A 122:1–20, 2010.

[3] J. R. Birge. Stochastic programming computation and applications. *INFORMS Journal on Computing*, 9:111–133, 1997.

[4] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, NY, USA, 1997.

[5] M. A. H. Dempster. *Stochastic Programming*. Academic Press, London, UK, 1980.

[6] S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universität Darmstadt, 2009.

[7] Y. Ermoliev and R. J-B. Wets. *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, Berlin, Germany, 1988.

[8] M. Fampa, N. Maculan. A new relaxation in conic form for the Euclidian Steiner tree problem in $\mathbb{R}^n$. *RAIRO Operations Research*, 35:383–394, 2001.

[9] M. Imase and B. M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.

[10] P. Kall and S. Wallace. *Stochastic Programming*. Wiley, New York, NY, USA, 1994.

[11] Y-J. Kuo, H. Mittelmann. Interior point methods for second-order cone programming and OR applications. *Computational Optimization and Applications*, 28: 255–285, 2004.

[12] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, 284:193–228, 1998.

[13] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Boston, MA, USA, 1995.

[14] S. Sen, H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Math. Program.*, Ser. A 106:203–223, 2006.

[15] H. D. Sherali, X. Zhu. On solving discrete two-stage stochastic programs having mixed-integer first-and second-stage variables. *Math. Program.*, Ser. B 108:597–616, 2006.

[16] R. Schultz, L. Stougie, and M. H. van der Vlerk. Two-stage stochastic integer programming: a survey. *Statistica Neerlandica.*, 50(3): 404-416, 1996.

[17] J. A. Tompkins, J. A. White and Y. A. Bozer, J. M. Tanchoco *Facilities planning*. 3rd edn. Wiley, Chichester, 2003.

[18] M. J. Todd. Semidefinite optimization. *ACTA Numerica*, 10: 515–560, 2001.

[19] R. J. Vanderbei and H. Yurittan. Using LOQO to solve second-order cone programming problems. Report SOR 98-09, Princeton University, Princeton, USA, 1998.

[20] D. W. Walkup and R. J-B. Wets. Stochastic programs with recourse II: On the continuity of the objective. *SIAM J. App. Math.*, 17: 98–103, 1969.

Least Squares Digital Differentiators (LSDD) The 2-D Subclass

Abdulwahab A. Abokhodair

Department of Earth Sciences, KFUPM, Dhahran, 31261, Saudi Arabia

Abstract – *In this paper, we present results of our investigation of the 2-D LSDD subclass of digital differentiation filters. We discuss methods of their generation from least-squares normal equations, and examine their properties in the space and frequency domains. In addition to ease of generation and implementation via convolution, the filters are shown to have many desirable properties. They are low pass linear phase, stable filters with narrow transition band attenuating rapidly at high frequencies. More important is that the filters are separable implying considerable computational time saving and except for an integer scaling- factor, their coefficients are all integers thus reducing the risk of cumulative round-off errors. The filters may be generated for any order derivative with arbitrary length to suite any desired sampling frequency. Unlike ordinary full-band DD that typically amplify high frequency noise, the low-pass nature of LSDD renders them noise suppressant with very low noise amplification factor. Moreover, the filters are actually multitasking, performing data smoothing and differentiation simultaneously. Their only drawback is that they are not maximally linear near the DC frequency.*

Keywords: Data processing; Filtering; Digital differentiation; Image processing

1. Introduction

Digital differentiation is at the core of many new technologies for source imaging, analyses and interpretation of geoscience data. Compared to the measured signal, gradients of the signal have greater spatial resolution, better definition of lateral boundaries, added depth discrimination and filtering properties, and better structural indicators. As the power and graphics capabilities of modern computers continue to increase, new and more powerful gradient-based imaging and interpretation technologies continue to emerge. Such techniques include high resolution detection of geologic boundaries [1,2]; Werner deconvolution for source depthing [3-5]; Euler deconvolution and its extended form for the calculations of physical property contrasts, dip information, location and depth of [6,7]; analytic, enhanced analytic signal and local wave numbers for source characterization and imaging [8-10].

The success of these new technologies made numerical computation of spatial gradients a basic step in processing geoscience data. Finite difference methods often used to estimate numerical derivatives suffer from a major draw back, namely noise amplification; hence, it is suited only for theoretical data uncontaminated with errors. This paper is a sequel to an earlier one presented in CSC'10 in which we discussed the spectral properties and filtering performance of the 1-D subclass of LSDD filters. In this paper we investigate the 2-D subclass of these filters and focus on first and second derivative operators of short length as required in most applications.

Filter Generation

LSDD 2-D filters have their roots in the principle of ordinary least squares (OLS) fitting of polynomial surfaces to a subset of equally spaced data. The key point is that the fitting process is performed only once and subsequent smoothing and differentiation of the entire data set is accomplished via a convolution kernels derived from the normal equations. As is well known, the problem of fitting data $\mathbf{z_0}$ to a polynomial of degree *d* in (*x, y*) leads to the so-called normal equations solution given by:

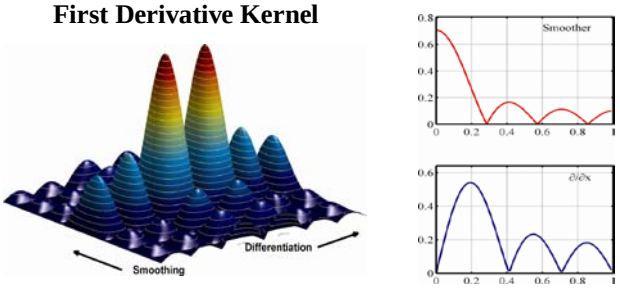
$$\hat{\mathbf{C}} = \left[\left(\mathbf{V}^T \mathbf{V} \right)^{-1} \mathbf{V}^T \right] \mathbf{z_0} = \mathbf{H} \mathbf{z_0}, \qquad (1)$$

where $\mathbf{V}$ is the basic matrix whose columns correspond to the bases vectors [**x**, **y**]. The matrix H contains the LSDD filters row-wise up to derivative order d.

Separability Property

The 2-D operator kernels are separable; i.e. they can be written as the outer product of two vectors called projection vectors. This property is their primary advantage since it implies a substantial reduction in computational complexity of the filtering operations. However, we also use this characteristic of the kernels to investigate their spectral properties and filtering performance. It turns out that these filters are multitasking, smoothing in one direction while performing differentiation in the perpendicular direction (Figure 1).

First Derivative Kernel



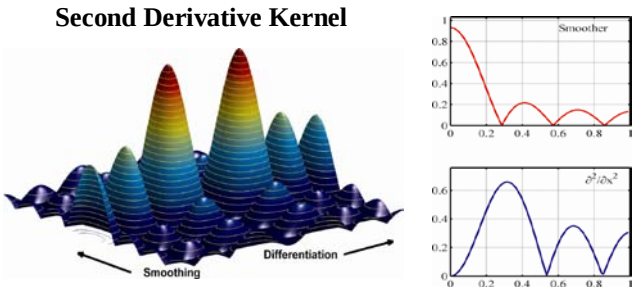


Figure 1: Frequency response of first and second derivative filters (Left panel) and their corresponding projection vectors (Right panel).

Generation of separable kernels follows the pattern shown in Figure 2. For a given differentiation order *k*, polynomials of degree *d* and *d+1* produce identical separable kernels and only first degree polynomials produce a separable smoothing filter. Therefore, only even-degree polynomials are needed to generate unique differentiation filter kernels of all orders. Moreover, since in practice only first and second derivatives are required, then polynomials of degree $d = 2$ are sufficient for routine applications.

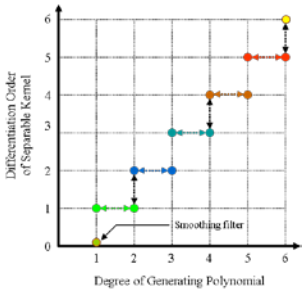


Figure 2: Generation pattern of LSDD separable differentiation kernels.

Spectral Properties

The filters are linear phase as indicated by the antisymmetry of the coefficients of the first derivative operator and the symmetry of the second derivative operators. Moreover, the filters are band-limited with cut-off frequency that is dependent on filter size (hw). However, unlike their 1-D counterparts, the 2-D filters are not maximally flat near the dc frequency. Figure 3 compares the magnitude response for several width (hw) 2-D operators with the response of their 1-D equivalents. Whereas the 1-D response curves maintain tangency with the curves of the ideal filters (IDD), the curves for 2-D operators deviate appreciably from the IDD curves. Thus the 2-D LSDD do not approximate the ideal filters in any frequency range, with the deviation from ideal behavior increasing with increasing filter half-width (hw). This result seems to cast doubt on the accuracy of their output.

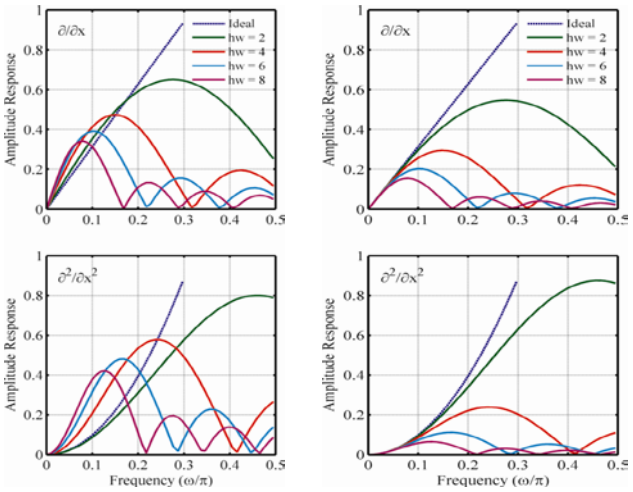


Figure 3: Comparison of the magnitude response of 2-D (left panel) and 1-D LSDD (right panel) operators of different length (hw) with the IDD response.

The Mean Square Error (MSE)

Writing the frequency response of the band-limited ideal differentiation filters as:

$$H_{id}^{1}(\omega) = i\omega \quad |\omega| \le \alpha\pi$$
$$H_{id}^{2}(\omega) = -\omega^{2} \quad |\omega| \le \alpha\pi \qquad (2)$$

for first and second derivatives respectively. Then the mean square error may be defined as:

$$E(\alpha) = \frac{1}{\pi}\int_{0}^{\pi} |H_{id}(\omega) - H_{ls}(\omega)|^{2} d\omega \,, \qquad (3)$$

where $H_{ls}(\omega)$ is the response of the LSDD filter. Using Parseval's relation, this equation may be written in terms of impulse response as:

$$E(\alpha) = \frac{1}{\pi}\sum_{n=-\infty}^{\infty} |h_{id}(n) - h_{ls}(n)|^{2} \qquad (4)$$

The variations of MSE with α is depicted in Figure 4 for different length operators. As the figure shows, the error increase with increasing cut-off frequency being smallest in the range $0 < \alpha < 0.1$. Moreover, the error decreases with increasing filter length.

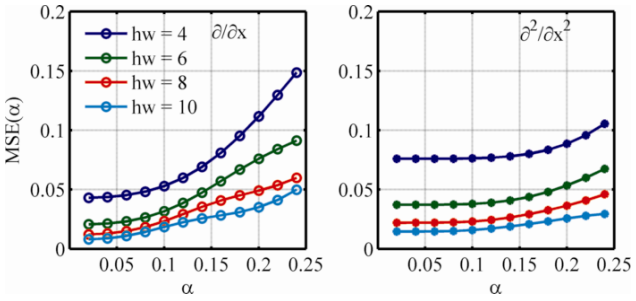


Figure 4: Mean square error (MSE) of LSDD operators of different lengths (hw).

Noise Reduction

The performance of filters in the presence of random errors (noise) is measured by means of the noise reduction ratio (NRR) which is given by [11]:

$$\sigma_y^2 = \sum_{n=-hw}^{hw} |h(n)|^2 \sigma_\varepsilon^2 \qquad (5)$$

σ_ε *and* σ_y are the standard deviations of the errors in input and output respectively. The NRR of the LSDD first and second derivative operators are shown in Figure 5. where it is seen that NRR decrease quite rapidly with increasing filter length.

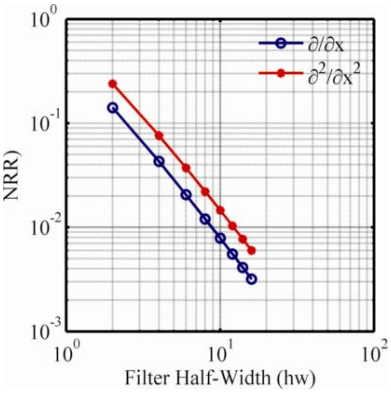


Figure 5: NRR of first and second derivative operators.

Conclusion

Aside from the non-linear behavior near the dc frequency, the 2D LSDD filters combine a number of attractive properties for routine applications. These include separability which implies a considerable reduction in computational complexity especially when processing large data set sets as is typical in geoscience applications. In addition, the filters are linear phase, self-damping and highly stable. Moreover, unlike the often used finite difference operators which amplify errors, the LSDD filters have extremely low NRR which implies a rapid attenuation of noise in the high -frequency band.

References

[1] S. K. Hsu, J. C. Sibuet and C. T. Shyu, "High-resolution detection of geological boundaries from potential-field anomalies: An enhanced analytic signal technique", Geophysics, 61 (1996), 373–386.

[2] R. J. Blakely and R. W. Simpson, "Approximating edges of source bodies from magnetic or gravity anomalies", Geophysics, 51 (1986), p. 1494.

[3] R. O. Hansen, " 3D multiple-source Werner deconvolution for magnetic data", Geophysics, 70 (2005), p. 45.

[4] M. F. Mushayandebvu, P. van Driel, A. B. Reid, and J. D. Fairhead, " Magnetic Source Parameters of Two-Dimensional Structures Using Extended Euler Deconvolution", Geophysics, 66 (2001), p. 814.

[5] M. E. Gimenez, M. P. Martinez, T. J. F. Ruíz, and F. L. Klinger, " Gravity Characterization of the La Rioja Valley Basin, Argentina", Geophysics, 74 (2009), p. 83 .

[6] J. Zhang, "An Analysis of the Accuracy of Magnetic Source-Body Geometry Determined from the 3-D Analytic Signal", Geophysics, 66 (2001), p. 579 .

[7] K. Davis, Y. Li, and M. Nabighian, "Automatic Detection of UXO Magnetic Anomalies Using Extended Euler Deconvolution", Geophysics, 75 (2010), p. 13.

[8] M. Fedi and G. Florio, " Detection of Potential Fields Source Boundaries By Enhanced Horizontal Derivative Method", Geophysical Prospecting, 49 (2001), p. 40

[9] R. S. Smith, A. Salem and J. Lemieux, "An Enhanced Method For Source Parameter Imaging of Magnetic Data Collected for Mineral Exploration", Geophysical Prospecting, 53 (2005), p. 655.

[10] Wen-Bin Doo, Shu-Kun Hsu and Yi-Ching Yeh, " A Derivative-Based Interpretation Approach to Estimating Source Parameters of Simple 2D Magnetic Sources from Euler Deconvolution, The Analytic-Signal Method and Analytical Expressions of the Anomalies", Geophysical Prospecting, 55 (2007), p. 255.

[11] S.J. Orfanidis, "Introduction to Signal Processing, Prentice Hall", Englewood Cliffs, NJ, 1996.

Linux Scheduler Performance for Beowulf Compute Nodes

Ronald Marsh, Michael Aguilar
Computer Science Department, University of North Dakota, Grand Forks, North Dakota, USA

Abstract - *In the last decade, the use of the Beowulf Cluster concept for High Performance Computing and Cloud Computing has exploded. In addition, the Linux operating system used by many of these computing platforms has also greatly advanced. Understanding the effects of the underlying kernel scheduler on the computational performance of compute nodes is one of the main concerns in customizing a Linux operating system. The overlying batch scheduling, message passing, and network communications are highly affected by subtle changes in kernel scheduler customizations. In this paper, we explore the computational performance of two types of Beowulf compute nodes, one using the O(1) scheduler and one using the new Completely Fair Scheduler, in various types of operating modes to determine an optimum scheduler and configuration.*

Keywords: High Performance Computing, Real-Time Kernel Scheduling, Compute Nodes, Complete Fair Scheduler, O(1) Scheduler.

1 Introduction

Over the last few years there has been a great deal of debate within the Linux community of how kernel scheduling should be done [1]. In each case, the goal of the kernel scheduler has been to follow a 'fairness' doctrine, with each process being given some consideration for an equal share of CPU time.

Used in many flavors of Enterprise Linux kernels, the O(1) scheduler allows the kernel to allocate CPU time by iterating though an 'active run queue' giving each process a chance to execute [2]. Once a process has received its share of run-time, the process is moved to an expired queue to wait for its next period of run time. Each priority level is given two run queues, an active queue and an expired queue. While working its way down the priority levels, the scheduler traverses each active queue in its entirety; it then swaps the expired queue for the active queue.

Another kernel scheduler used in many recent Enterprise Linux kernels, the Completely Fair Scheduler (CFS) is more concerned with the amount of time the processor has already spent running processes [3]. For instance, after running a process for a certain time 'quantum', the process is put back into a binary tree. The process placement on the tree is determined by a 'Virtual Run Time'. The Virtual Run Time is calculated by weighting each process' run-time used already by the process' static priority. When a process Virtual Run Time is larger, the process priority is lowered. If a process hasn't been run for a long time its corresponding Virtual Run Time will be smaller and the process will be automatically moved to a higher priority leaf on the tree.

It has become common practice to base custom Linux High Performance Computing operating systems on different types of Enterprise Linux operating systems. Many System Administrators and Computer Scientists will install these Enterprise Linux operating systems and then modify them for use in High Performance computing clusters. Organizations, such as Lawrence-Livermore, with their Chaos operating system [4], have modified 'off-the-shelf' Enterprise Linux operating systems by costumizing the kernel and the services installed in each portion of the system. Most of these changes have been made in IP connection services and file storage drivers (ie. Lustre and Hadoop) [5], [6]. However, it must be noted that much of the effort in changing the kernel has been in areas that are subservient to the kernel scheduler. Since the kernel scheduler is responsible for determining when a process will run, the kernel scheduler bears the ultimate responsibility for how long it takes the process to execute.

The purpose of this paper is to put the focus back onto the kernel scheduler, in an effort to find better performing configurations. Another goal is to determine what scheduler properties seem to enhance the processing performance the most. Since we are looking at prioritizing the processing of calculations on the compute nodes, and since both the O(1) scheduler and the Completely Fair Scheduler are included with most Enterprise Linux kernels, we will focus on the processing performance of only these schedulers. Normally, these schedulers are operated in "sched_other" mode with an average nice (priority) level of 0.

2 Test Set Up

2.1 System Model

A large share of the computational research at the University of North Dakota is performed on a Beowulf cluster nicknamed Shale. Shale's operating system is Red Hat Enterprise Linux version 5.5. Moab [7] is installed along with the Torque PBS Scheduler in the head node, to allow versatility in scheduling of tasks. Each Compute node operates in a diskless configuration, via NFS mount and PXE boot.

2.2 Test Setup

The test computations were done on part of the Shale cluster. The images used were based on Red Hat Enterprise Linux 5.5 (2.6.18-238 kernel with the O(1) scheduler). The Completely Fair Scheduler was installed with a 2.6.35.6 kernel. Mindful of the fact that many different runs would be required to gather data and realizing that a longer run-time would reduce the likelihood that small anomalies in individual, localized processes would affect the results, an approximate run-time of around 15 minutes was chosen. The NAS Parallel Benchmarks-Scalar Pentadiagonal Solve Benchmark [8] was used to force the CPU cores on each Compute node to process matrix type non-linear differential equations. The 15 minute run-time goal was achieved by setting the matrix to solve equations for each processor at 400 points, or Class B.

3 Parameters Used for Experiments

Three types of test computations were done on each scheduler. In one set of computations sched_other mode, which is the default operating mode of the schedulers, was used. Round-Robin and First-In/First-Out scheduler operating modes were set for the executing processes to run in a real-time operating mode.

In order to test compute node performance, we chose to alter run-time priorities. Priority levels were predetermined, ranging from -19 to 19. Priority levels of -19 hold the highest priority in sched_other mode. Every priority level was represented at least twice. In several cases, many more iterations of the computations were done at the same priority levels, in order to gather multiple data sets.

Real-time test runs were done using sched_fifo and sched_rr. In order to discover if there was an optimum priority level for the real-time runs, all test computations were done with real-time priority levels, ranging from 80-99.

4 Experimental Results

In this section, the results using each type of scheduler algorithm are presented. Three different operating modes were used, the default operation of sched_other mode and two real-time modes with sched_rr and sched_fifo. In each case, priority levels were changed for the process to determine the effects on the computations times.

Figure 1 shows the calculation time in minutes for a large number of test runs of the Scalar Pentadiagonal benchmark on the compute nodes using the Completely Fair Scheduler in sched_other mode. Process priority levels were varied by adjusting the nice parameters, from the default of 0. For these experiments, a mean calculation time of 15.800 minutes was found with a standard deviation of .167 minutes.

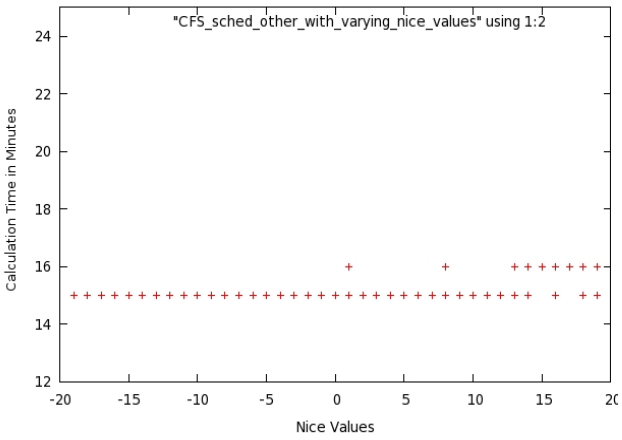


Figure 1. Calculation time in minutes for benchmark while running on the Completely Fair Scheduler in sched_other mode with varying nice values.

Figure 2 shows the calculation time in minutes for a large number of test runs of the Scalar Pentadiagonal benchmark for the compute nodes in sched_other mode using the O(1) scheduler. Again, process priority levels were varied by adjusting the nice parameters for each test run of the benchmark. For these experiments, a mean calculation time of 16.727 minutes was found with a standard deviation of 2.558 minutes.

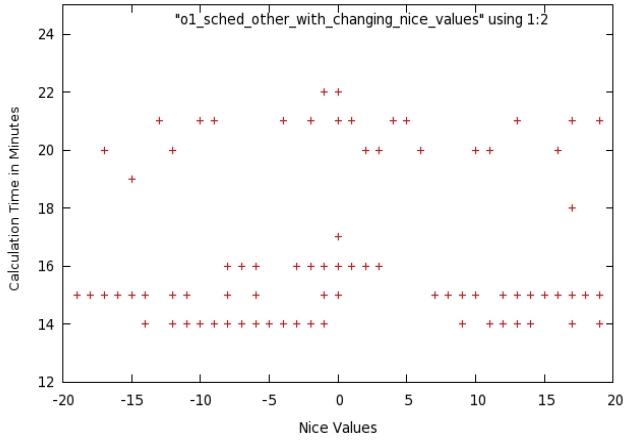


Figure 2. Calculation time in minutes for benchmark while running on the O(1) scheduler in sched_other mode with varying nice values.

As figure 1 and figure 2 show, the computation time in minutes is nearly the same for every nice level. This shows there is little improvement in the benchmark computation times by varying nice levels.

Figure 3 shows the calculation time in minutes for a large number of test runs of the Scalar Pentadiagonal benchmark for the compute nodes in real-time Round-Robin mode, for the Completely Fair Scheduler. Process priority levels were varied by adjusting the real-time process parameters for each test run of the benchmark. For these experiments, a mean calculation time of 13.679 minutes was found with a standard deviation of .971 minutes.

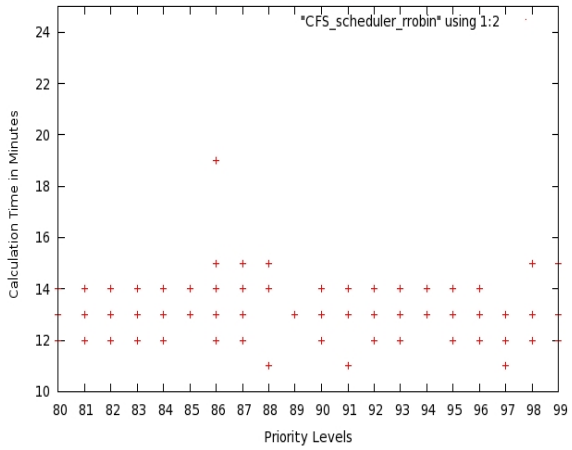


Figure3. Calculation time in minutes for benchmark while running on the Completely Fair Scheduler in real-time/Round-Robin mode with different process priorities.

Figure 4 shows the calculation time in minutes for a large number of test runs of the Scalar Pentadiagonal benchmark for the compute nodes in real-time Round-Robin mode, for the O(1) scheduler. Process priority levels were varied by adjusting the real-time process parameters for each test run of the benchmark. For these experiments, a mean calculation time of 17.019 minutes was found with a standard deviation of 2.392 minutes.

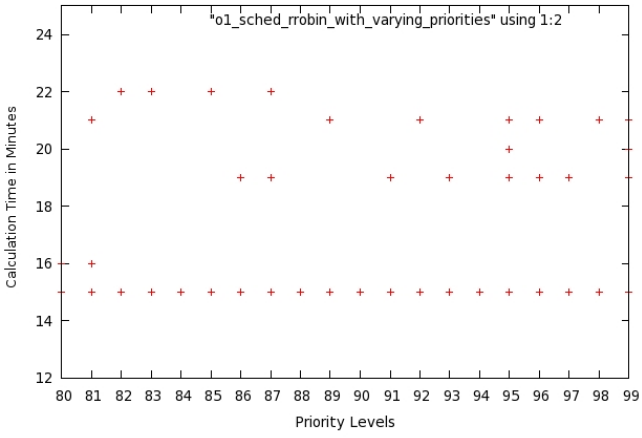


Figure 4. Calculation time in minutes for benchmark while running on the O(1) scheduler in real-time/Round-Robin mode with different process priorities.

While in real-time Round-Robin mode, the mean computation time for the Completely Fair Scheduler dropped. The mean O(1) scheduler computation time actually increased to over 17 minutes. In addition, while there was an increase in the standard deviation for computation times for the benchmark calculations on the Complete Fair Scheduler, there was a greater standard deviation in computation times for the O(1) scheduler with the deviation being almost 2.5 minutes. Again, little improvement in the computation times was found by varying the real-time priority levels.

Figure 5 shows the calculation time in minutes for a large number of test runs of the Scalar Pentadiagonal benchmark for the compute nodes in real-time First In-First Out mode for the Completely Fair Scheduler. Process priority levels were varied by adjusting the real-time process parameters for each test run of the benchmark. For these experiments, a mean calculation time of 13.457 minutes was found with a standard deviation of .786 minutes.

Further improvement in calculation performance times for the Completely Fair Scheduler running in First In-First Out configuration was found. The average computation times for the test computations dropped to slightly under 13.5 minutes while the standard deviation of the computation times dropped to within 1 minute.

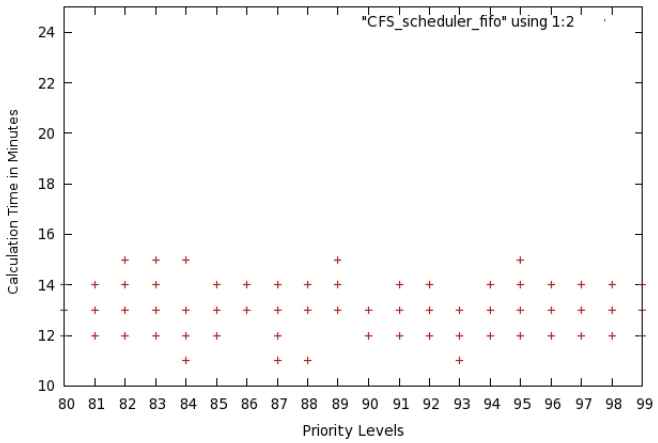


Figure 5. Calculation time in minutes for benchmark while running on the Completely Fair Scheduler in real-time/First In-First Out mode with different process priorities.

Figure 6 shows the calculation time in minutes for a large number of test runs of the Scalar Pentadiagonal benchmark for the compute nodes in real-time First In-First Out mode for the O(1) scheduler. Process priority levels were varied by adjusting the real-time process parameters for each test run of the benchmark. For these experiments, a mean calculation time of 16.538 minutes was found with a standard deviation of 1.826 minutes.

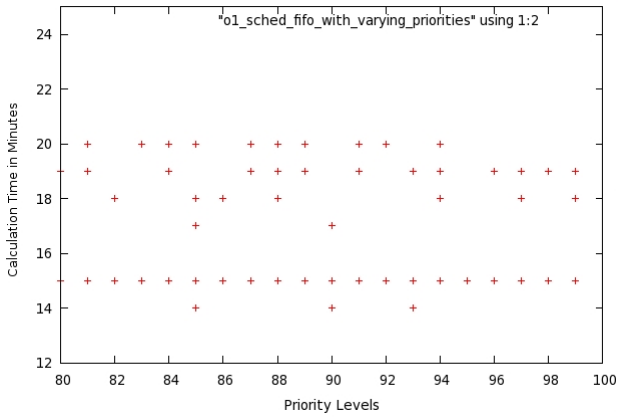


Figure 6. Calculation time in minutes for benchmark while running on the O(1) scheduler in real-time/First In-First Out mode with different process priorities.

The O(1) scheduler real-time First-In/First-Out computation times were slightly better than the O(1) scheduler computation times in the standard sched_other mode. The mean computation time dropped .189 minutes while the standard deviation of the computation time dropped .732 minutes. As with all of the other computation modes, there was no change in the computation times by varying the priority level of the calculation processes.

5 Conclusions

In an effort to find the optimum scheduler and associated operating mode, several tests were run against a fixed, well known, parallel based, benchmark program. The results supported, conclusively, the superiority of the Completely Fair Scheduler when used in a real-time First In-First Out fashion. Operating times were decreased up to 2 minutes for the benchmark program with a standard deviation of varying runs within 1 minute. In addition, it was found that varying the priority levels, whether in real-time operating mode or in the standard batch modes had no effect on the operating performance of the computations when using the Completely Fair Scheduler and the O(1) scheduler.

Overall, comparing the Completely Fair Scheduler with the O(1) scheduler, we find that the Completely Fair Scheduler requires a shorter mean calculation time with a smaller standard deviation in calculation times.

Interestingly, running the benchmark processes on the O(1) scheduler, running in regular sched_other mode gave decreased mean computation times over the O(1) scheduler running in real-time Round-Robin mode. The O(1) scheduler, running in real-time First-In/First-Out mode showed a slight improvement in performance versus the O(1) scheduler in sched_other mode.

6 References

[1] Andrews, J. (2007, July 27). Linux: Linux On CFS vs SD. Message posted to http://kerneltrap.org/node/14008

[2] Bovet, D., & Cesati, M. (2001). Process Scheduling. C. Morris (Ed.), "Understanding the Linux Kernel" (pp. 277-283). O'Reilly & Associates, Inc.

[3] Kumar, A. IBM. (2008, January 8). "Multiprocessing with the Completely Fair Scheduler." Retrieved Jan 15, 2011, From http://www.ibm.com/developerworks/linux/library/l-cfs/index.htm

[4] Linux @ Livermore. "Chaos". From https://computing.llnl.gov/linux/projects.html

[5] Linux@Livermore. "Lustre." From https://computing.llnl.gov/linux/projects.html

[6] White, T. (2009). The Hadoop Distributed File System. M. Loukides (Ed.), "Hadoop The Definitive Guide" (pp. 41-73). O'Reilly Media, Inc.

[7] Cluster Resources, "Moab Cluster Suite". From http://www.clusterresources.com/products/moab-cluster-suite.php

[8] National Aeronautics and Space Administration. "NAS Parallel Benchmarks". From http://www.nas.nasa.gov/Resources/Software/npb.html

Polynomial Transformation Method for Non-Gaussian Noise Environment

Jugalkishore K. Banoth, Pradip Sircar*

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur 208016, India
*Corresponding author; Email address: sircar@iitk.ac.in

Abstract - *Signal processing in non-Gaussian noise environment is addressed in this paper. For many real-life situations, the additive noise process present in the system is found to be dominantly non-Gaussian. The problem of detection and estimation of signals corrupted with non-Gaussian noise is difficult to track mathematically. In this paper, we present a novel approach for optimal detection and estimation of signals in non-Gaussian noise. It is demonstrated that preprocessing of data by the orthogonal polynomial approximation together with the minimum error-variance criterion converts an additive non-Gaussian noise process into an approximation-error process which is close to Gaussian. The Monte Carlo simulations are presented to test the Gaussian hypothesis based on the bicoherence of a sequence. The histogram test and the kurtosis test are carried out to verify the Gaussian hypothesis.*

Keywords: Orthogonal polynomial approximation, Signal detection and estimation, Non-Gaussian noise

1 Introduction

In the signal detection and estimation problems, we often assume that the additive random noise process is Gaussian distributed because this distribution is simple and mathematically tractable, and the assumption makes analytical results possible. However for many real-life situations, the additive noise process is found to be dominantly non-Gaussian. Some examples are the ocean acoustic noise and the urban radio-frequency (RF) noise [1]. The RF receivers designed to perform in white Gaussian noise can not perform satisfactorily when the electromagnetic environment encountered by the receiver system is non-Gaussian in nature [2]. For detection and estimation of radar signals in high clutter environments and similar processing of sonar signals in presence of high reverberation, we need to deal with non-Gaussian noise [1, 2].

There are two existing approaches for solving the problems of detection and estimation of signals in non-Gaussian noise environment. The first approach is to use the robust statistics in lieu of the classical mathematical statistics, and to look for procedures which are consistent or in other words, insensitive to deviations of the noise distribution from the idealized model, i.e., the Gaussian distribution [3]. An optimally robust procedure minimizes the maximum degradation of performance for a preset deviation of the noise distribution. The robust techniques, however, can not provide consistent performance for a noise process with an arbitrary probability density function (PDF).

The second approach to deal with a non-Gaussian noise environment is to use a noise model which is general enough to depict an arbitrary PDF, yet the model retains the desirable simplicity of manipulation as that of a Gaussian PDF. Accordingly, the Gaussian-mixture PDF, the generalized Gaussian PDF, the Middleton class A PDF, and some such PDFs are employed to model non-Gaussian noise [4]. Incidentally, as the noise model is required to be more accurate, the ease of analysis as that of a Gaussian PDF disappears.

In this paper, we present a third approach to deal with a non-Gaussian noise environment, by employing the polynomial transformation method. Preprocessing of data by the orthogonal polynomial approximation (OPA) together with the minimum error-variance criterion (MEC) has an excellent noise-rejection capability [5, 6]. The OPA based transformation was originally proposed to convert non-uniformly sampled data into uniformly sampled data [5]. However, since the transformation provides significant signal enhancement by rejecting the high frequency interference, preprocessing of data may be useful in detection and estimation problems for better accuracy even for uniformly sampled data [6]. Perhaps the most desirable feature of preprocessing the signal samples by the OPA based method is that the statistical distribution of the approximation-error process in the preprocessed data becomes close to Gaussian when the noise process is not necessarily Gaussian distributed [5]. Based on this argument, the maximum likelihood estimator (MLE) can be designed to estimate parameters of a signal corrupted with non-Gaussian noise [7].

In the present work, we take a closer look of the preprocessing of data by the OPA based method, and we test the hypothesis that the approximation-error process in the preprocessed data is Gaussian distributed even when the noise process corrupting the sampled data is non-Gaussian. Several types of tests are applied for testing the hypothesis. We plot the histogram of a given sequence and look for the proverbial bell shape as a simple test for its Gaussian distribution [8]. We compute the kurtosis [9] and apply the Hinich test [10, 11] for validation of the Gaussian hypothesis. We consider the following noise processes for the Monte Carlo simulation: (i) Gaussian, (ii) Laplacian, (iii) Uniform, and (iv) Gamma distributed.

2 Orthogonal Polynomial Transformations

The real-valued discrete-time signal $g[n]$ is to be detected/ estimated utilizing the sampled sequence $x[n]=g[n]+w[n]$, where $w[n]$ is the noise sequence which may not be Gaussian distributed. The sampled data $\{x[n]\}$ are preprocessed by the orthogonal polynomial transformation to obtain the transformed data $\{y[n]\}$ as follows [6],

$$\mathbf{y}=\mathbf{P}\mathbf{Q}^{-1}\mathbf{P}^{T}\mathbf{x} \qquad (1)$$

where

$$\mathbf{x}=\left[x[0],\, x[1],\cdots,\, x[N-1]\right]^{T},$$

$$\mathbf{y}=\left[y[0],\, y[1],\cdots,\, y[N-1]\right]^{T},$$

$$(\mathbf{P})_{ij}=p_j[i];\; i=0,1,\cdots,N-1;\; j=0,1,\cdots,J-1;$$

$$\mathbf{Q}=\mathbf{P}^{T}\mathbf{P}=\mathrm{Diag}\left[\sum_{m=0}^{N-1}p_0^2[m],\sum_{m=0}^{N-1}p_1^2[m],\cdots,\sum_{m=0}^{N-1}p_{J-1}^2[m]\right]$$

The orthogonal polynomials $p_j[n]$ are computed by the recurrence relation given in [5, 12], and the order of approximation J is chosen such that the error-variance is minimum.

The transformed sequence $y[n]$ is given by $y[n]=g[n]+e[n]$, where $e[n]$ is the approximation error. By utilizing the relation between the error sequence $e[n]$ and the noise sequence $w[n]$,

$$e[n]=\sum_{m=0}^{N-1}\xi_{nm}w[m] \qquad (2)$$

where $$\xi_{nm}=\sum_{j=0}^{J-1}\left\{\frac{p_j[n]p_j[m]}{\sum_{i=0}^{N-1}p_j^2[i]}\right\},$$

we can compute the autocorrelation functions (ACFs) of the error process [6], provided the ACFs of the noise process are known. Furthermore, by invoking the central limit theorem when the random variables $w[n]$ are independent with zero mean and identical variance, and the coefficients ξ_{nm} are bounded [5, 13], we can argue that the error process will be close to Gaussian even when the distribution of the noise process is non-Gaussian.

3 Gaussian Hypothesis Testing

The third order cumulant of the noise/ error process $u[n]$ is given by

$$C_{3u}[k_1,k_2]=E\{u[n]u[n+k_1]u[n+k_2]\} \qquad (3)$$

where E is the expectation operator, and the third order spectrum, commonly known as the bispectrum, is defined as the two-dimensional Fourier transform of the third order cumulant,

$$S_{3u}(\omega_1,\omega_2)=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}C_{3u}[k_1,k_2]\exp\{-j(\omega_1k_1+\omega_2k_2)\} \qquad (4)$$

The squared bicoherence $\left|B_{3u}(\omega_1,\omega_2)\right|^2$ is determined as follows

$$\left|B_u(\omega_1,\omega_2)\right|^2=\frac{\left|S_{3u}(\omega_1,\omega_2)\right|^2}{S_{2u}(\omega_1)S_{2u}(\omega_2)S_{2u}(\omega_1+\omega_2)} \qquad (5)$$

where $S_{2u}(\omega)$ is the power spectrum.

The Hinich test is based upon the squared bicoherence at a bifrequency (ω_1,ω_2) being zero for Gaussianity of the underlying sequence. The $|B|^2$ value is averaged over the principal domain [10, 11], and the resulting statistics is central χ^2 distributed under the null hypothesis: $S_{3u}(\omega_1,\omega_2)\equiv 0$. Hence, it is easy to devise a statistical test to determine whether the observed squared bicoherence is consistent with a central χ^2 distribution by computing a probability of false alarm (PFA) value. If the null hypothesis of the bispectrum being zero is not rejected, we then compute the average kurtosis K_u given by [9]

$$K_u = \frac{E\{u^4\}}{\left[E\{u^2\}\right]^2} - 3 \qquad (6)$$

where the value is averaged over each element of the sequence $\{u[n]\}$. The kurtosis test is based on the null hypothesis: $K_u \equiv 0$ for a Gaussian distribution of the underlying process.

4 Simulation Results

The real part of the complex-exponential transient discrete-time signal

$$g[n] = \sum_{i=1}^{3} b_i \exp(j\varphi_i)\exp(s_i nT) \qquad (7)$$

where $b_1 = 1.0, s_1 = -0.2 + j2.0, \varphi_1 = 0;$

$b_2 = 0.5, s_2 = -0.1 + j4.0, \varphi_3 = \pi/4;$

$b_3 = 0.5, s_3 = -0.3 + j1.0, \varphi_3 = \pi/6;$

corrupted with non-Gaussian noise setting the signal-to-noise ratio (SNR) at 10 dB, is sampled at 60 uniformly spaced points with time interval $T = 0.15$.

We present the three cases of Laplacian, Uniform, and Gamma distributed noise environments in this work, beside the Gaussian noise case. In each case, after applying the polynomial transformation we obtain the transformed data, and then, subtracting the transient signal from the transformed data, the error process is separated. The input noise and the output error processes are tested for Gaussianity. Figs. 1− 4 show the bispectrum and the histogram plots.

For the Gaussian noise case, we calculate the bicoherence of the output error and check whether the squared bicoherence is consistent with a central χ^2 distribution by computing the PFA value. The PFA is computed to be 0.9479, which is high, and we cannot reject the null hypothesis. The average kurtosis value for the output error is computed to be −0.1526, whereas the kurtosis value for the input noise is computed to be −0.0857 (theoretical value zero). For the Laplacian case, the PFA for the input noise is 0.396, and the PFA for the output error is 0.9975. Since the PFA of the output error is high, we cannot reject the null hypothesis. The kurtosis values are 2.9359 for the input noise and 0.0148 for the output error. For the Uniform noise case, we compute the PFA for the input noise to be 0.6973 and the PFA for the output error to be 0.9649. The average kurtosis values are computed to be −1.2408 for the input noise and −0.1346 for the output error. For the Gamma distributed noise environment, the PFA for the input noise is 0.7379, and the PFA for the output error is 0.9847. The kurtosis values are 0.6962 for the input noise and 0.1025 for the output error. In all cases, we find that the average kurtosis value of the output error process is near zero, confirming that the error process is close to Gaussian.

5 Concluding Remarks

In this paper, we present a new technique for optimal detection and estimation of signals corrupted with non-Gaussian noise. We preprocess the sampled data by the polynomial transformation method which converts the noise process into an approximation-error process which is Gaussian distributed.

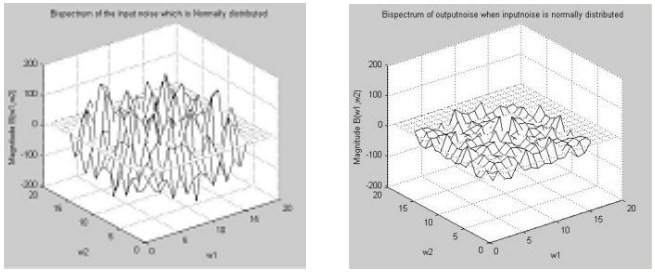


Figure 1(a): The Bispectrum of the input noise (Gaussian) and the output error process

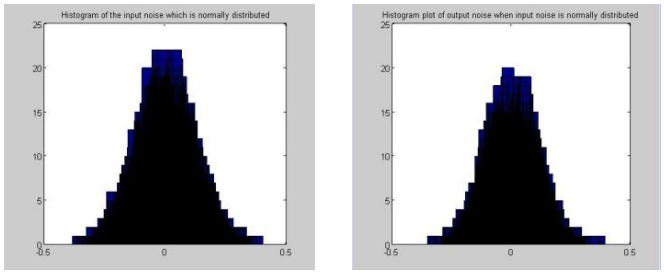


Figure 1(b): The Histogram of the input noise (Gaussian) and the output error process

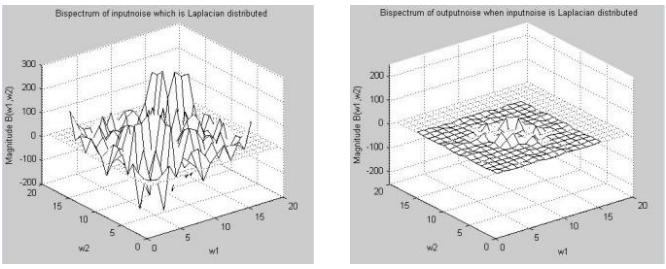


Figure 2(a): The Bispectrum of the input noise (Laplacian) and the output error process

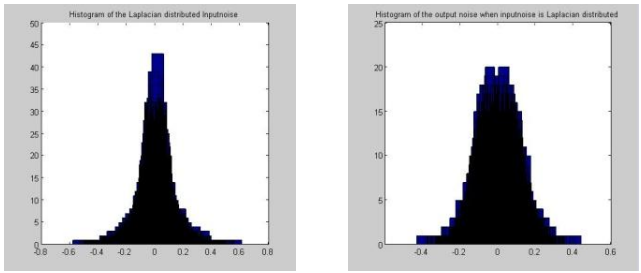


Figure 2(b): The Histogram of the input noise (Laplacian) and the output error process

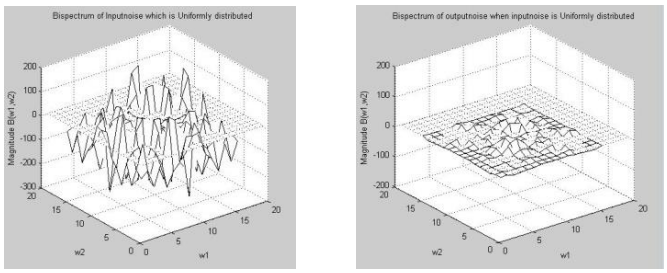


Figure 3(a): The Bispectrum of the input noise (Uniform) and the output error

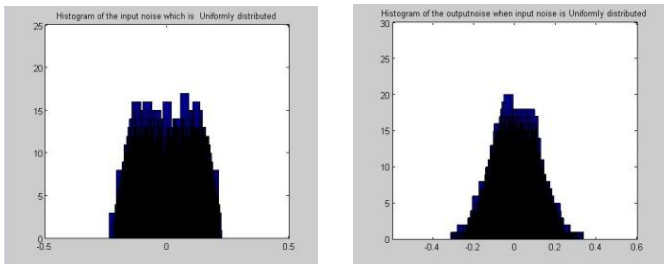


Figure 3(b): The Histogram of the input noise (Uniform) and the output error

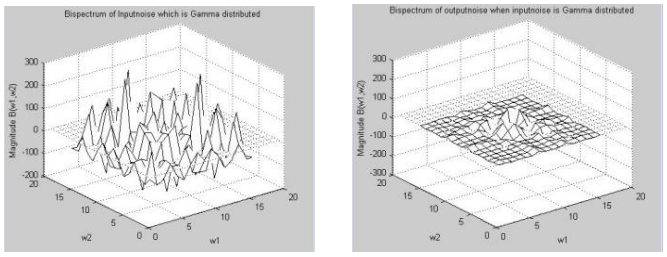


Figure 4(a): The Bispectrum of the input noise (Gamma distributed) and the output error

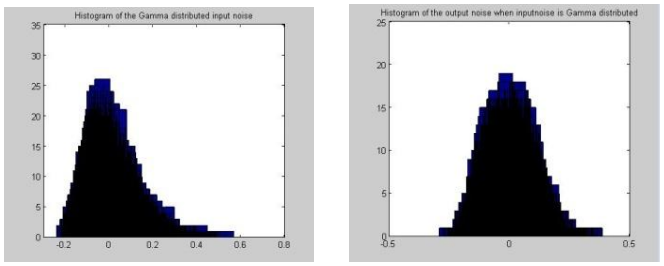


Figure 4(b): The Histogram of the input noise (Gamma distributed) and the output error

6 References

[1] E.J. Wegman and J.G. Smith (Eds.), Statistical Signal Processing, Marcel Dekker, New York, 1984.

[2] E.J. Wegman, S.C. Schwartz and J.B. Thomas (Eds.), Topics in Non-Gaussian Signal Processing, Springer-Verlag, New York, 1989.

[3] P.J. Huber and E.M. Ronchetti, Robust Statistics, John Wiley, Hoboken, N.J., 2009.

[4] S. A. Kassam, Signal Detection in Non-Gaussian Noise, Springer-Verlag, New York, 1988.

[5] P. Sircar and A.C. Ranade, "Nonuniform sampling and study of transient systems response", IEE Proceedings F - Radar and Signal Processing, vol. 139 , no. 1, pp. 49–55, 1992.

[6] M. Ravi Shankar and P. Sircar, "Nonuniform sampling and polynomial transformation method", IEEE Int'l Conf. on Commun. ICC 2002, vol. 3, pp. 1721–1725, 2002.

[7] P. Sircar and S. Mukhopadhyay, "Parameter estimation of transient signal in non-Gaussian noise", IEEE Region 10 Int'l Conf. on EC3 – Energy, Computer, Commun. & Cont. Systems TENCON '91, vol. 3, pp. 236–240, 1991.

[8] D.C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, John Wiley, New York, 1999.

[9] J.A. Cadzow, "Blind deconvolution via cumulant extrema", IEEE Signal Process. Mag., vol. 13, no. 5, pp. 24–42, 1996.

[10] M.J. Hinich, "Testing for Gaussianity and linearity of a stationary time series", J. Time Series Analysis, vol. 3, no. 3, pp. 169–176, 1982.

[11] J.K. Tugnait, "Validation, testing, and noise modeling", In: V.K. Madisetti and D.B. Williams (Eds.), The Digital Signal Processing Handbook, pp. 16.1–16.12, CRC Press, Boca Raton, 1998.

[12] A. Ralston and P. Rabinowitz, A First Course in Numerical Analysis, McGraw-Hill, Singapore, 1988.

[13] W. Feller, An Introduction to Probability Theory and Its Applications, Vol. I & II, John Wiley, New York, 1965.

.

Model and Algorithm for Fractional Delay HPF Design

Jinming Ge
Vaisala Inc
Louisville, CO 80027, USA
jinming.ge@vaisala.com

Abstract – *Implementing fractional delay filters in FPGA needs a cost-effective approach. This paper tries to clarify a misconception in fractional delay filter design, especially the phase delay and group delay behavior of low pass and high pass filters. Starting with the ideal filter frequency response, followed by practical filter simulation, two important delay factors are compared side by side. Mathematical models and algorithms are derived to establish the relation between phase delay and group delay of high pass filters. Some application tips are given at the end of the paper.*

Keywords: algorithm, fractional delay, finite impulse filter, phase delay, group delay.

1 Introduction

The frequency response of an ideal fractional delay filter can be described as:

$$H(\omega) = Ge^{-j\omega D} \qquad 0 \le |\omega| \le \pi \qquad (1)$$

Where G is the gain, D is the delay and ω is the normalized frequency. Its phase response, phase delay and group delay are:

$$\Phi(\omega) = -\omega D \qquad (2)$$

$$pd = -\Phi(\omega) / \omega = D \qquad (3)$$

$$gd = -d\Phi(\omega) / d\omega = D \qquad (4)$$

Those properties imply that an ideal fractional delay filter can be implemented by an all-pass filter. But within limited bandwidth, it can be implemented cost-effectively by a low-pass filter (LPF):

$$H_l(\omega) = Ge^{-j\omega D} \qquad 0 \le |\omega| \le \omega_c \qquad (5)$$

where ω_c is the cutoff frequency. If implemented in high-pass filter (HPF), the frequency response is

$$H_h(\omega) = Ge^{-j\omega D} \qquad \omega_c \le |\omega| \le \pi \qquad (6)$$

Some literatures evaluate the fractional delay filter using phase delay [1, 2], while others using group delay [3, 4, 5]. The confusion or misconception leads to misinterpretations of filter behavior, especially for HPF. To implement a finite impulse response (FIR) filter in FPGA often requires evaluating the tradeoffs between performance and cost. A cost-effective approach would be to have the filter perform within a reasonable bandwidth, instead of an all-pass filter. The fractional delay HPF and LPF are discussed in section 2, the relation between phase delay and group delay, especially for high-pass filter is analyzed and a design algorithm is presented in section 3. Section 4 concludes this paper with application suggestions and further studies.

2 Fractional Delay HPF vs. LPF

Practical implementation of the fractional delay filter will introduce a non-ideal term, or error, Φ_e to the phase response:

$$\Phi(\omega) = -\omega D + \Phi_e \qquad (7)$$

A HPF can be derived by (frequency) transforming a LPF - shifting its frequency response by π:

$$h_h(k) = h_l(k) * cos(k\,\pi),\ k=0,1,\ \ldots N \qquad (8)$$

where N is the order of the filter, h_h and h_l are the impulse response or coefficient of HPF and LPF respectively. The π shift effect to the term is unique for HPF (or BPF, which can be viewed as a composite of HPF and LPF). The following evaluate the filter with all four characteristics: magnitude, phase, phase delay and group delay.

2.1 Symmetric Response of HPF and LPF

It was expected that the HPF and LPF behave symmetrically around the Nyquist (or normalized ω as $\pi/2$), as HPF is just a frequency shift of LPF. The same fractional delay and the same bandwidth are used in this paper to evaluate both filters.

Fig.1 shows the magnitude response of the two filters. It is evident that the HPF and LPF are

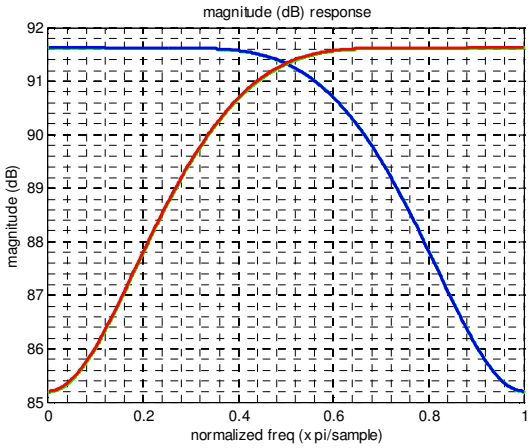


Fig 1. Magnitude Response of LPF (in Blue) and HPF (in Red) with fractional delay 0.1 (sample)

symmetric centering at the Nyquist frequency (0.5 in the figure). Or, view it in another way, rotating the HPF by π will overlap with that of LPF response, reasonably so because the HPF was designed based on shifting LPF by π.

The phase response of the filters is shown in Fig.2. Note that the filter's structural phase (relevant to the order or the length of the filter) is removed from the figure, or the phase calculation is based on the center tap of the filter). While the LPF phase response is linear within the low frequency range, the HPF is linear within the higher frequency range – the reason the HPF was attempted to be used. But it is also worthy to notice that, although the HPF phase

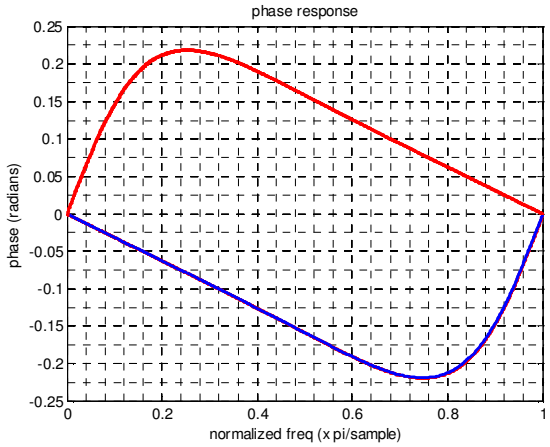


Fig 2. Phase Response of LPF (in Blue) and HPF (in Red) with fractional delay 0.1 (sample)

response is linear within that range, the response is not a simple linear one, instead it becomes a Generalized (or affined) Linear, that is the Φ_e in the equation (7) is not zero, or

$$\Phi_h(\omega) = -\omega D + \Phi_{he} \qquad (9)$$

In which Φ_{he} is the intercept of the general liner equation at the starting frequency where the linear phase response starts for the higher frequency range, instead of zero.

Another symmetric property between HPF and LPF is the group delay, as shown in Fig.3. Both filters produce designated group delay (0.1 in the figure) within designed frequency bandwidth. Although $\Phi_h(\omega)$ has a non-zero term Φ_{he}, the group delay ($d\Phi_h(\omega)/d\omega$) is still a constant within its bandwidth range.

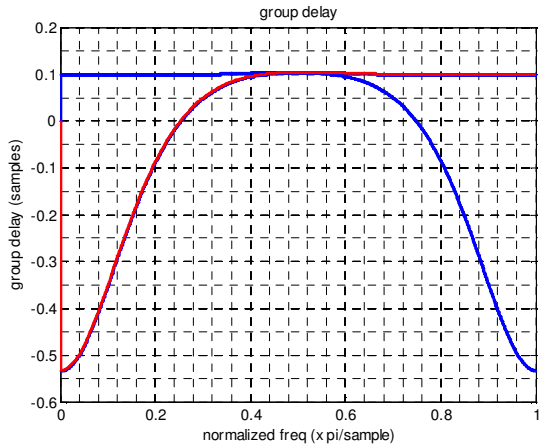


Fig 3. Group Delay of LPF (in Blue) and HPF (in Red) with fractional delay 0.1 (sample)

2.2 Asymmetric Phase Delay of HPF and LPF

As $\Phi_h(\omega)$ has a non-zero term Φ_{he}, the phase delay ($\Phi_h(\omega)/\omega$) is NOT a constant within bandwidth range, in contrast from that of LPF, as shown in Fig.4.

The phase delay of the HPF is far from that of

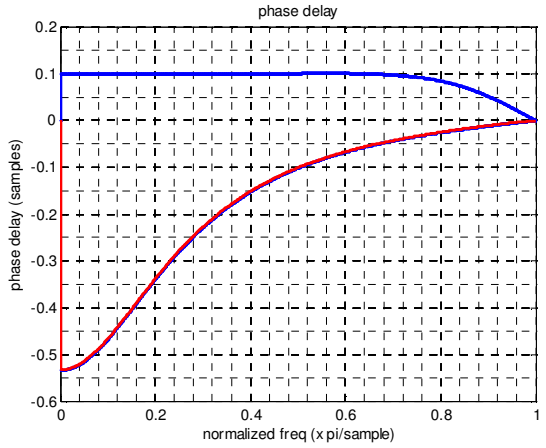


Fig 4. Phase Delay of LPF (in Blue) and HPF (in Red) with fractional delay 0.1 (sample)

an ideal fractional delay filter, even within its linear phase (high) frequency range.

Comparing Fig.4 with Fig.3, the LPF has the same group delay and phase delay, but HPF does not. The HPF has group delay as designated, but its phase delay is far from ideal. This has significant implications to some applications where an HPF is to be used. For example, applications may need to detect the relative delay between two filtered output signals, quite often it is the phase delay detected, which differs from the group delay often assued to be equal to.

3 The Correlation between Phase Delay and Group Delay of HPF

It is desirable for fractional delay HPF to have phase delay equal to group delay. The HPF was designed based on frequency shifting of LPF by π. From Fig. 3, its phase delay can be "corrected" by the following to achieve constant phase delay within its bandwidth:

$$pdh_d = \frac{\phi_{hd}(\omega)}{\omega} = pdh * \frac{\omega}{\pi - \omega} \qquad (10)$$

where

$$pdh = \frac{\phi_h(\omega)}{\omega} \qquad (11)$$
$$0 < |\omega| < \pi$$

pdh_d is the "desired" HPF phase delay and $\Phi_{hd}(\omega)$ is the desired phase response. It is also interesting to

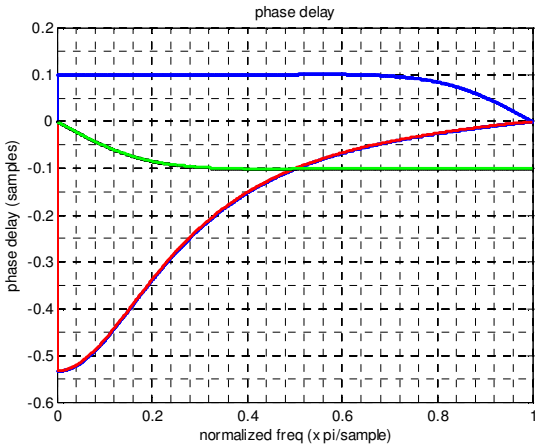


Fig 5. Phase Delay of LPF (in Blue), HPF (in Red) and HPF Symmetric correlated (in Green) with fractional delay 0.1 (sample)

see the "average" phase delay within a specific bandwidth B:

$$pdh_{dav} = (1/B) * \int_{fc-B/2}^{fc+B/2} (pdh * \frac{f}{0.5-f}) df$$

$$= pdh * [\frac{1}{2*B} * \ln \frac{(2*fc-1)-B}{(2*fc-1)+B} - 1] \qquad (12)$$

The "corrected" phase delay comparing with original phase delay is shown in Fig.5. It can be see that the "corrected" phase delay (in green in the figure) now is really symmetric comparing with that of LPF (in red in the figure).

The corrected phase delay shown in Fig.5 can be used to derive the desired phase of HPF, $\Phi_{hd}(\omega)$, as shown in Fig.6. Although it is not symmetric to

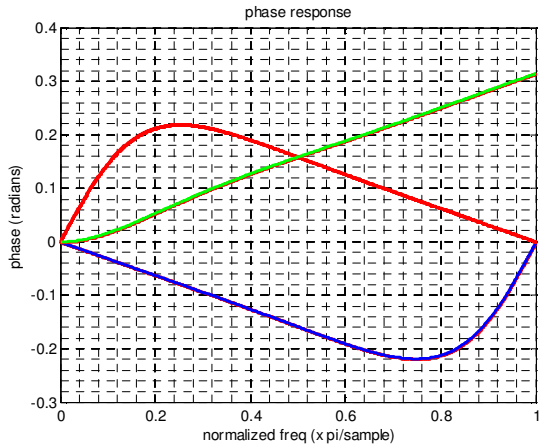


Fig 6. Phase response of LPF (in Blue), HPF (in Red) and HPF Symmetric correlated (in Green) with fractional delay 0.1 (sample)

LPF, as the original one, but it is still linear within its bandwidth.

Since the desired phase response is available, the desired group delay can be derived, as shown in Fig.7. Similar to that of phase response, although the group delay derived from the desired phase delay is not symmetric to that of LPF anymore, it is still a constant within its bandwidth.

Making both phase delay and group delay constant within a reasonable bandwidth for HPF has significant implications, especially in embedded systems or the filter implemented as a

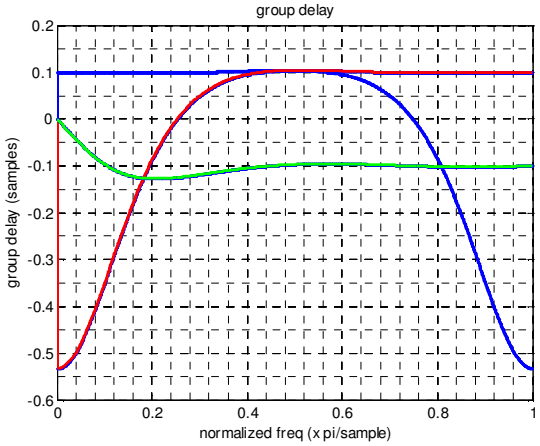


Fig 7. Group Delay of LPF (in Blue), HPF (in Red) and HPF Symmetric correlated (in Green) with fractional delay 0.1 (sample)

reconfigurable module in FPGA, in which very limited hardware resources available. Even though an all-pass fractional delay filter may be preferred for performance reasons at relatively high normalized frequency, it requires significant hardware resources in terms of multipliers, adders, and other DSP blocks.

Phase delay and group delay are assumed, or preferred to be equal in some applications. Fig. 8 shows one scenario, in which the phase delay of the filter's output is detected by a phase-sensitive-detector (PSD), the reconfigurable fractional delay

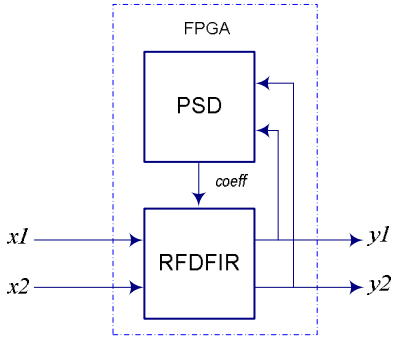


Fig 8. An Example of Application that assuming equal phase delay and group delay

finite impulse response (RFDFIR) filter then need to be reconfigured to cancel the relative delay between the two input signal channels, which

results in outputs without time delay between them, or cancelled. The RFDFIR is used as a precise adjustable delay lines.

4 Discussions and Further Directions

By comparing the four characteristics of fractional delay LPF and HPF, a common misconception of phase delay and group delay is discussed, followed by analysis of the difference and correlation between them in HPF. An algorithm is presented to "correct" the phase delay performance of a common HPF and to achieve both constant phase delay and group delay for the HPF. This approach is useful when cost-effective FDFIR is needed to cover a bandwidth in the higher frequency range.

A formal approach to correlate desired phase response with that of typical HPF may be developed. One possible method is to use frequency sampling [6], since the desired frequency domain behavior is known after the correlation. Quite often, for fractional delay filters, the approach may or may not generate real coefficients for the filter, while implement a quadrature filter needs more DSP blocks. The performance needs to be evaluated further, if the HPF is implemented by only real coefficients for cost saving.

5 References

[1] T. Laakso, V. Valimaki, M. Karjalainen and U. Laine, Splitting the Unit Delay, IEEE Signal Processing Magzine, Jan., 1996.

[2] V. Valimaki, Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters, Nov., 1995

[3] T. Nguyen, T. Laakso and R. Koilpillai, Eigenfilter Approach for the Design of Allpass Filters Approximating a Given Phase Response, Transaction on Signal Processing, Vol. 42, No.9, Sept., 1994.

[4] W. Lu, S. Pei and C. Tseng, Weighted Least-Square Method for the Design of Stable 1-D and 2-D IIR Digital Filters, Trans on Signal Processing, Vol.46, No.1, Jan., 1998.

[5] A. Tkacenko, P. Vaidyanathan and T. Nguyen, On the Eigenfilter Design Method and Its Applications: A Tutorial, Trans on Circuits and Systems –II: Analog and Digital Signal Processing, Vol.50, No.9, Sept., 2003.

[6] J. Proakis, D. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, Prentice-Hall, 1996.

Parallel Computations for Simulating Heat Conduction

M. Zahid Ayar; Kanaan A. Faisal; Bekir Yilbas; Adel Ahmed; Saad Mansour
King Fahd University of Petroleum and Minerals (KFUPM), ICS & ME Departments, Saudi Arabia

Abstract

The world of CPUs is moving towards parallelization. As the number of transistors being placed onto semi-conductor chips increase to be able to make more powerful processing units, a power limit is reached whereby too much leakage of power occurs between very tightly placed transistors on a chip. Due to this limit and two other limits, the way to go around these barriers is by parallelizing the CPUs. Intel has introduced multi-core technologies to achieve parallelization at the CPU level and believes this to be the future of processing units. To use this new available power however, requires a special knowledge of parallel programming. It is like learning to program from the beginning and requires a different way of thinking about programming. In our current study, we have applied parallel programming to solve heat equations using CUDA and OpenMP technologies. This paper will discuss the details of these implementations as well as show the visualization of the results obtained from heat equation simulations for various data. To achieve our results, heat conduction models such as Cattaneo, Two-Equation, and Fourier were solved using traditional single processor techniques and the results were visualized in 2 and 3 dimensions. Then the Fourier model of the heat conduction equation was used in our experiment of parallelization on the CUDA and OpenMP platforms.

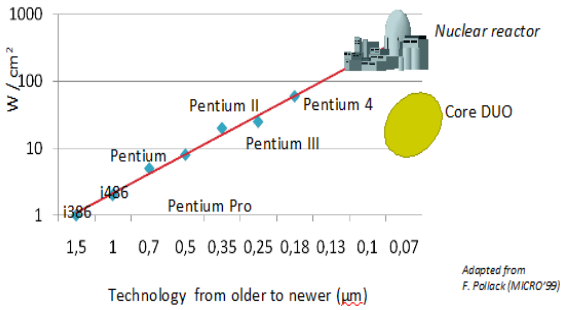
Keywords: Parallel Programming, Numerical Simulation, Visualization, Heat Transfer, Heat Conduction

1. Introduction

Paul Otellini, the President and CEO of Intel said "We are dedicating all of our future product development to multicore designs" and "We believe this is a key inflection point for the industry." Now parallel computing is becoming much more widespread and it gives lots more computing power to its users. To be able to have this power in your hands means that you can now use the computer to make much more realistic simulations that are much better approximations to real life.

Serial computing has doubled in speed many times but it has 3 barriers to future growth: The Memory Wall, Instruction Parallelism Wall, and the Power Wall. The Memory Wall refers to the growing gap between CPU speed and memory access speed. To try to make up for this disparity, the cache size of the CPU has grown so as to reduce the "average memory reference" time. From 1986 to 2000, CPU speed has increased at a rate of 55% annually while memory access speed has only increased 10% annually.

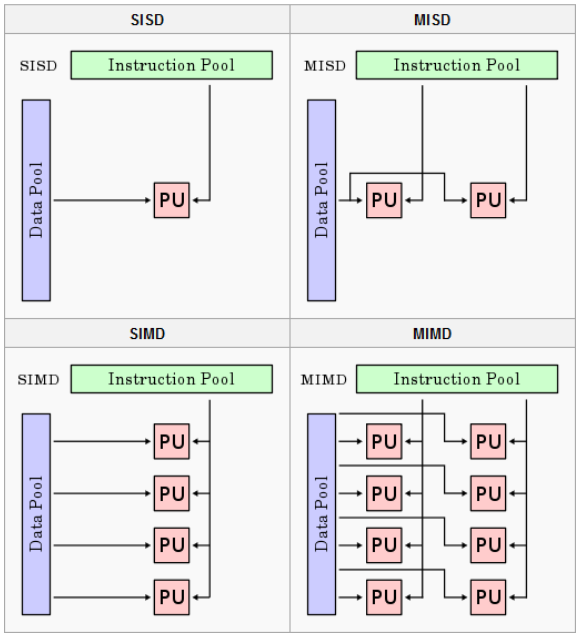
The Instruction Level Parallelism wall refers to the limit that guessing and loading of most probably needed instructions by the CPU in anticipation of future instruction usage before the process becomes too complex and slows the CPU down unnecessarily. The Power Wall is the barrier furnished by the limit of transistor packing in a small area. As more and more transistors in CPUs are packed closer and closer, the power leakage is increased so much so that now the Watts generated per centimeter square of a Intel Core Duo processor is the same as that provided by a nuclear reactor.



CPUs currently are very powerful and they are not overwhelmed by data that they are not able to process in time. On the contrary, they are going through data starvation whereby not enough data is reaching to them to be processed. This is due to other bottlenecks in the system. So to make much better use of the computer system, parallel computing needs to be used.

Non-GPU parallel programming support is provided by the following software:

- MPI (original goal was distributed memory systems, now it is very effective as a shared memory system as well).
- OpenMP
- Threads (on operating systems)
- Parallel libraries
- Intel's Task Parallel Library
- Microsoft's Task Parallel Library
- SWARM (Gtech)
- Charm++ (UIUC) … growing http://charm.cs.uiuc.edu/
- STAPL (Standard Template Adaptive Parallel Library) B. Stroustrup, Texas A&M.. undergoing effort



The diagram above shows Flynn's Taxonomy of computer architectures. SISD is the original Von Neumann architecture. CUDA architecture is SIMD. The Top500 supercomputers in the world all fall into the category of MIMD (Multiple Instruction Multiple Data).

2. Numerical Simulation using Traditional Serial Programming

To be able to compute the values of mathematical functions utilizing the current computer technology, it is necessary to first discretize those functions by utilizing numerical techniques. The process of discretizing a function is an art and a science because there is no one systematic way to discretize all possible analytical functions. These discretized versions of the analytical functions are approximations of the analytical function. Once you have a discretized function, you can then proceed to compute all the required approximate values that the original analytical function models (such as temperature values of a metallic plate over a defined time period). From the computed values you can then proceed to visualize the computed values using various visualization techniques in 2 or 3 dimensions.

There are three equations that may be used to model the temperature variation in a metallic substrate, they are: Cattaneo (hyperbolic), Two-Equation (hyperbolic, parabolic), and the Fourier Equation (parabolic). From this work, one of the aims is to find the differences between the Cattaneo and Two-Equation heat conduction models in producing temperature values at different time and space scales and find out which exact terms in the respective equations are causing those differences. The benefit of this work is that we can determine which of the given models more closely predicts the real-life values and then use this model to predict the correct values of temperatures at other points of the material and at other times during heating, without the cost of doing the actual heating experiments for the material to obtain the temperature values.

Laser heating of metallic surfaces is involved with the deposition of laser energy into the substrate material through the absorption process. The prior knowledge on temperature rise in the irradiated region is important from the practical applications in industry. In this case, the proper setting of the laser parameters provides the desired temperature distribution in the heated region. However, experimental measurement of temperature distribution inside the metallic substrate is difficult and is involved with expensive equipment. Therefore, predictions through model studies provide temperature distribution with less cost and in short time duration. However, determining these temperature values for metals requires expensive laboratory equipment and execution. If there is a way in which we can produce an accurate model for the behavior of the metal in terms of temperature variation under different heating conditions, it would be a much more economical way to know the required values of temperature for manufacturing or material usage. These three equations (Cattaneo, Two-Equation, and Fourier) can be used to determine the discrete temperature of the metal at it's different points as a result of heating it. By using these equations, one can get the values of the temperature at various points of the metal at different times after heating. Each computation to determine a temperature value requires enormous amounts of computation involving partial differential equations containing two to three terms.

2.1 Discretization Process

Using the Finite Difference method for discretizing a continuous function, there are three basic methods to proceed in order to transform a continuous analytical function into discretized form:

a) Forward Difference

$$\frac{dy}{dx}\Big|_{x=x_2} \quad \frac{y_3 - y_2}{\Delta x}$$

b) Backward Difference

$$\frac{dy}{dx}\Big|_{x=x_2} \quad \frac{y_2 - y_1}{\Delta x}$$

c) Central Difference

$$\frac{dy}{dx}\Big|_{x=x_i} \quad \frac{y_{i+1} - y_{i-1}}{2\Delta\Delta}$$

Using these techniques, numerical simulation libraries written in the C# language were developed to compute the temperature values of a given metal according to its specific thermal diffusivity property over a specified period of time. The temperature values were calculated based on the three different temperature models, namely, the Fourier equation, Cattaneo model, and the Two-equation. For the Fourier equation, the C# library was developed to support the calculation of temperature values of a 1 dimensional object, 2-dimensional object, and a 3-dimensional object. For the Cattaneo model, temperature values of a 1-dimensional and 2-dimensional object may be calculated, and for the Two-equation, the library supports temperature values computation of a 1-dimensional object.

The Fourier equation, better known as the Fourier Heat Conduction equation is one of the most famous equations used to describe heat distribution in a given region over time [18]. It is the partial differential equation shown below:

$$\frac{\partial u}{\partial t} - k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = 0$$

The partial differential equation shown above contains 3 terms for describing space (x,y, and z) and 1 for time. We can discretize the Fourier equation shown above using the finite difference numerical simulation techniques mentioned previously.

After discretizing this equation we get:

$$T_{i,j}^{p} = a_e T_{i+1,j}^{p} + a_w T_{i-1,j}^{p} + a_n T_{i,j+1}^{p} + a_s T_{i,j-1}^{p} + \frac{T_{i,j}^{p-1}}{\alpha \Delta t}$$

where $T_{i,j}^{p}$ stands for the temperature at location *(i,j)* on a 2-dimensional metallic plate at time step *p*. The Cattaneo model for temperature distribution is:

$$\tau \frac{\partial^2 T}{\partial t^2} + \frac{\partial T}{\partial t} = \alpha\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right]$$

The discretized form of the Cattaneo equation is:

$$T_{i,j}^{p} = \frac{a_e\left(T_{i+1,j}^{p} + T_{i-1,j}^{p}\right) + a_n\left(T_{i,j+1}^{p} + T_{i,j-1}^{p}\right) + T_{i,j}^{p-1}\left[\frac{1}{\Delta t} + \frac{2\tau}{\Delta t}\right] - T_{i,j}^{p-2}\left(\frac{\tau}{2\Delta\Delta}\right)}{a}$$

For the two-equation model, we have a set of two equations which are:

$$C_e \frac{\partial T_e}{\partial t} = k \frac{\partial^2 T_e}{\partial x^2} - G\left[T_e - T_l\right] + S_0$$

and

$$C_l \frac{\partial T_l}{\partial t} = G\left[T_e - T_l\right]$$

Discretizing those two equations gives us:

$$T_{ei}^{p} = \frac{a_e k\left[T_{ei+1}^{p} + T_{ei-1}^{p}\right] + S_o + G T_{li}^{p} + \frac{C_e}{\Delta t} T_{ei}^{p-1}}{a}$$

and

$$T_{li}^{p} = \frac{G T_{li}^{p} + \frac{C_e}{\Delta t} T_{li}^{p-1}}{\frac{C_e}{\Delta t} + G}$$

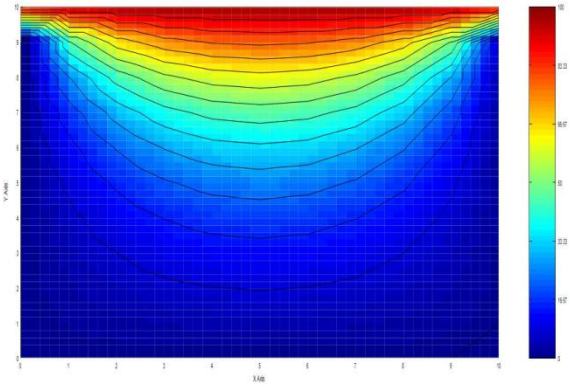
From the work above, a scientific library has been produced that can solve the 3 equations mentioned above. The required data to solve temperature distribution for each of the 3 models is given below in the form of the signatures of the constructors of the objects that represent the different temperature distribution models:

```
public FourierHeatConduction3DTransient ( float startX , float endX , int numberOfXRegions , float startY , float endY , int numberOfYRegions , float startZ , float endZ , int numberOfZRegions , float temperatureAtStartX , float temperatureAtEndX , float temperatureAtStartY , float temperatureAtEndY , float temperatureAtStartZ , float temperatureAtEndZ , float initialTemperature , int numberOfTimeRegions )
```

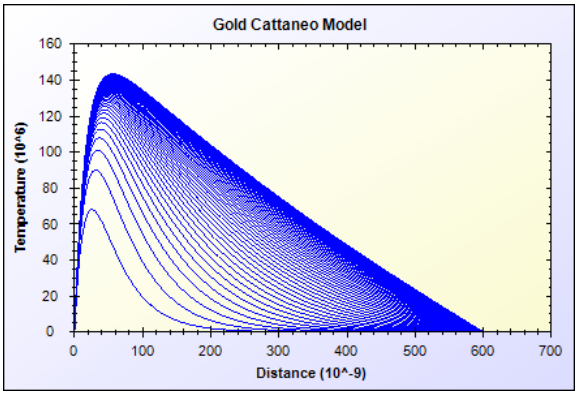
```
public CattaneoHeatConduction2D ( float
startX , float endX , int numberOfXRegions ,
float startY , float endY , int
numberOfYRegions , float temperatureAtStartX
, float temperatureAtEndX , float
temperatureAtStartY , float
temperatureAtEndY , float initialTemperature
, int numberOfTimeRegions )


public TwoEquationHeatConduction1D ( float
startX , float endX , int numberOfXRegions ,
float temperatureAtStartX , float
temperatureAtEndX , float initialTemperature
, float startTime , float endTime , int
numberOfTimeRegions , Material thisMaterial
)
```

2.2 Nelements Knowledge Visualization Library

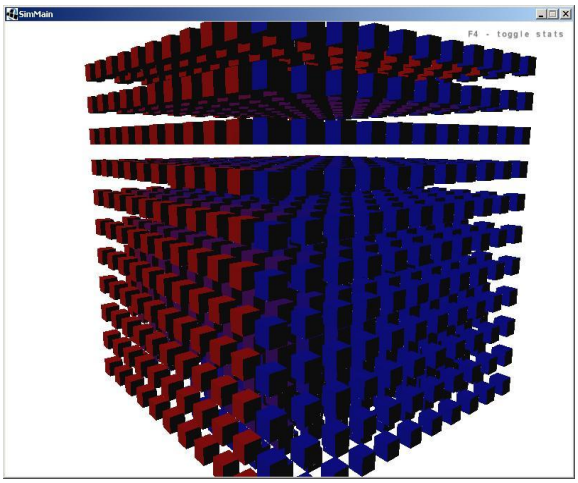
A visualization library has also been developed to display the computed values from the numerical simulation library. This visualization library is called Nelements Knowledge Visualization and it supports 2D contour maps which can be static or animated. Also, functionality has been developed to be able to easily plot function values at variable ranges. Below are some of the results obtained by using the visualization library.



A 2d contour map generated using data from the Fourier 2d steady state simulation. This is a simulation of heating a metallic plate from the top at 100 degrees Celsius and all the other sides of the plate are 0 degrees Celsius. A legend is also provided on the right of the output to give information about color to temperature value mapping.



The 2d function plot above shows the temperature over distance values for a 1d gold metal rod calculated using the Cattaneo model. The separate lines represent the temperature distribution over the rod at different time steps.



The snapshot above shows the visualization of a 3d metallic box and the color-coded temperature distribution based on the Fourier heat conduction model. This 3d visualization component has been developed by Dr. Adel Ahmed using the Java Monkey Engine.

3. Simulations done by Parallel Programming

High performance computing uses supercomputers to solve computationally expensive tasks for standard PCs. Computationally expensive in this context can mean computational tasks which may take a standard PC a very long time to do such as a day, a week, or more. A standard PC usually has one processor whereas a high-performance computing system is made up of a number of nodes

each with one or more processors and each containing its own memory. The nodes can be connected to each other by a high-speed network connection.

High performance computing makes use of parallel computation to be able to achieve high-levels of performance. In parallel computing a given program is split up into many sub-programs and then they are all run together in parallel to compute the values required. These sub-programs can be called threads and in a high-performance computing cluster of nodes, each node can get a thread or more to work on. For these threads to be able to work together, there needs to be well-organized communication between them.

3.1 CUDA

In the CUDA parallel environment, the computation needs to be designed by means of organizing the threads to be run into blocks on a grid. The block can be 1D, 2D, or 3D. While the grid containing the blocks can be 1D or 2D. In the case of simulating the heat equation in 3-dimensions, 3-dimensional blocks of size 5x5x9 have been used on a grid size of 2x2 to compute the discrete values of a computational matrix with 10x10x10 node elements. The code for defining the sizes is:

```
// DIVIDE_INTO(x/y) for integers, used to determine # of blocks/warps etc.
#define DIVIDE_INTO(x,y) (((x) + (y) - 1)/(y))
// I3D to index into a linear memory space from a 3D array index
#define I3D(ni, nj, i, j, k) ((i) + (ni)*(j) + (ni)*(nj)*(k)) //newcode
// Block size in the i, j, and k directions
#define NI_TILE 5
#define NJ_TILE 5
#define NK_TILE 9
```

The kernel used to compute the temperature values with the Fourier equation is given below:

```
// kernel to update temperatures - GPU version (not using shared mem)
__global__ void step_kernel_gpu(int ni,
                                int nj,int nk,
                                float tfac,
                                float *temp_in,
                                float *temp_out) {
int i, j, k, ti, tj, tk , i000, im100, ip100, i0m10, i0p10, i00m1, i00p1;
        float d2tdx2, d2tdy2, d2tdz2;
        // find i and j indices of this thread
        ti = threadIdx.x;
        tj = threadIdx.y;
        tk = threadIdx.z;
        i = blockIdx.x*(NI_TILE) + ti;
        j = blockIdx.y*(NJ_TILE) + tj;
        k = blockIdx.z*(NK_TILE) + tk;
// find indices into linear memory for central point and neighbours
    i000 = I3D(ni,nj, i, j,k);
    im100 = I3D(ni,nj, i-1, j,k);
    ip100 = I3D(ni,nj, i+1, j,k);
    i0m10 = I3D(ni,nj, i, j-1,k);
    i0p10 = I3D(ni,nj, i, j+1,k);
    i00m1 = I3D(ni,nj, i, j,k-1);
    i00p1 = I3D(ni,nj, i, j,k+1);
        if (i > 0 && i < ni-1 && j > 0 && j < nj-1 && k > 0 && k < nk-1) {
                // evaluate derivatives
                d2tdx2 = temp_in[im100] - 2*temp_in[i000] + temp_in[ip100];
                d2tdy2 = temp_in[i0m10] - 2*temp_in[i000] + temp_in[i0p10];
                d2tdz2 = temp_in[i00m1] - 2*temp_in[i000] + temp_in[i00p1];
                // update temperatures
                temp_out[i000] = temp_in[i000] + tfac*(d2tdx2 + d2tdy2 + d2tdz2);
        }
}
```

3.2 OpenMP

Two standards for doing parallel computing on a cluster computing platform are MPI and OpenMP. Message passing is a technique developed to allow multiple processes running concurrently to communicate with one another by passing and receiving messages from each other. Message passing can be accomplished in a distributed memory system or in a shared memory setting. Distributed memory systems are made up of a number of separate nodes linked together in a network. They are massively parallel machines. Shared memory systems are supercomputers that have a great computational power on a single

workstation that contains large numbers of cores to be able to execute large parallel tasks. MPI is a standard message passing library for distributed memory systems that was first prepared in May 1994 called MPI-1. People were so excited about parallel computing during those days that the first implementations of the MPI-1 standard were released only about a year later in June 1995 – the most popular one being Argonne's MPICH. In those days, there were many supercomputing companies and they also produced their own implementations of the MPI-1 standard. The second MPI standard called MPI-2 was completed in 1998, this time the enthusiasm wasn't as high as before so the first implementation of this standard came about after about 4 years in November 2002. There are a number of MPI-2 implementations including MPICH2, Intel MPI, HP-MPI, Microsoft MPI, MPAVICH, Open MPI (open source MPI), and others. Among these different implementations you can find implementations that work in Windows and Linux environments.

OpenMP is an API that supports shared-memory parallel programming in contrast to distributed memory parallel programming. The Architectural Board for OpenMP first published its standard based on FORTRAN 1.0 in October 1997. One year later, in October 1998, the standard based on C/C++ was published. In 2000, the second version of the OpenMP FORTRAN specification was written. Version 2.0 of the OpenMP C/C++ specification was completed in 2002. The next version of the OpenMP specification, which is version 2.5 covered both FORTRAN and C/C++, and that was released in 2005. On May 2008, version 3.0 of OpenMP was released having some new features among which include the task concept and the task construct.

Below you can see a code snippet for solving the 2D Heat Equation using OpenMP:

```
#pragma omp parallel for shared(solution,cur_gen,next_gen,diff_constant) private(i,j)
  for (i = 1; i <= RESN; i++)
    for (j = 1; j <= RESN; j++)
      solution[next_gen][i][j] = solution[cur_gen][i][j] +
       (solution[cur_gen][i + 1][j] +
        solution[cur_gen][i - 1][j] +
        solution[cur_gen][i][j + 1] +
        solution[cur_gen][i][j - 1] -
        4.0 * solution[cur_gen][i][j]) *
       diff_constant;
```

Solving the 2d Heat equation using MPI:

```
  for (it = 1; it <= TIME_STEPS; it++)
{
   if (neighbor1 != NONE)
    {
         MPI_Send(&u[iz][offset][0], NYPROB, MPI_FLOAT, neighbor1,
          NGHBOR2, MPI_COMM_WORLD);
      source = neighbor1;
      message_tag = NGHBOR1;
      MPI_Recv(&u[iz][offset-1][0], NYPROB, MPI_FLOAT, source,
           message_tag, MPI_COMM_WORLD, &status);
    }
   if (neighbor2 != NONE)
    {
      MPI_Send(&u[iz][offset+number_rows-1][0], NYPROB, MPI_FLOAT, neighbo
           NGHBOR1, MPI_COMM_WORLD);
      source = neighbor2;
      message_tag = NGHBOR2;
      MPI_Recv(&u[iz][offset+number_rows][0], NYPROB, MPI_FLOAT, source, m
           MPI_COMM_WORLD, &status);
    }
   /* Now call update to update the value of grid points */
   update(start,end,NYPROB,&u[iz][0][0],&u[1-iz][0][0]);
   iz = 1 - iz;
}
   /* Finally, send my portion of final results back to master */
   MPI_Send(&offset, 1, MPI_INT, MASTER, DONE, MPI_COMM_WORLD);
   MPI_Send(&number_rows, 1, MPI_INT, MASTER, DONE, MPI_COMM_WORLD);
   MPI_Send(&u[iz][offset][0], number_rows*NYPROB, MPI_FLOAT, MASTER, DONE,
        MPI_COMM_WORLD);
}
```

4. Conclusion

As a result of these simulations involving the three models Cattaneo, Two-Equation, and Fourier, we have found that the two models Cattaneo and Two-Equation start to differ in their temperature values after a time range between 10^{-14} seconds and 10^{-9} seconds. For the space variable, the two models differ at 10^{-9} meters to 10^{-8} meters and 10^{-11} meters. These results are helpful to decide which of the models are suitable to be used in certain simulations of temperature distribution under different circumstances.

Acknowledgements

The authors would like to acknowledge King Fahd University of Petroleum and Minerals (KFUPM) for their support and thank KFUPM-ITC for providing support in learning and using the KFUPM-HPC center at the ITC.

5. References

[1] Using OpenMP by Chapman, Jost, Van Der Pas

[2] CUDA Programming by John Seland

[3] NVIDIA Tutorial CUDA , NVIDIA Developer Technology by Cyril Zeller

[4] OpenMP wikipedia article : http://en.wikipedia.org/wiki/OpenMP

[5] Relativistic heat conduction wikipedia article:

http://en.wikipedia.org/wiki/Relativistic_heat_conduction

[6] Improved formulation of electron kinetic theory approach for laser ultra-short-pulse heating by Bekir Sami Yilbas

[7] The History of MPI : http://beige.ucs.indiana.edu/I590/node54.html

[8] HP-MPI : http://www.hp.com/go/mpi

[9] MPICH2 : http://www.mcs.anl.gov/research/projects/mpich2/

[10] MPAVICH – MPI over Infiniband and iWARP : http://mvapich.cse.ohio-state.edu/

[11] MPICH-MX and MPICH2-MX software : http://www.myri.com/scs/download-mpichmx.html

[12] Microsoft MPI : http://msdn.microsoft.com/en-us/library/bb524831%28VS.85%29.aspx

[13] Intel MPI : http://www.intel.com/go/mpi/

[14] CUDA Zone – What is CUDA : http://www.nvidia.com/object/cuda_what_is.html

[15] JCublas : http://javagl.de/jcuda/jcublas/JCublas.html

[16] GASS – CUDA.NET : http://www.gass-ltd.co.il/en/products/cuda.net/

[17] Heat Equation - http://en.wikipedia.org/wiki/Heat_equation

[18] Dan Negrut's High Performance Computing Slides - http://sbel.wisc.edu/Courses/ME964/2008/