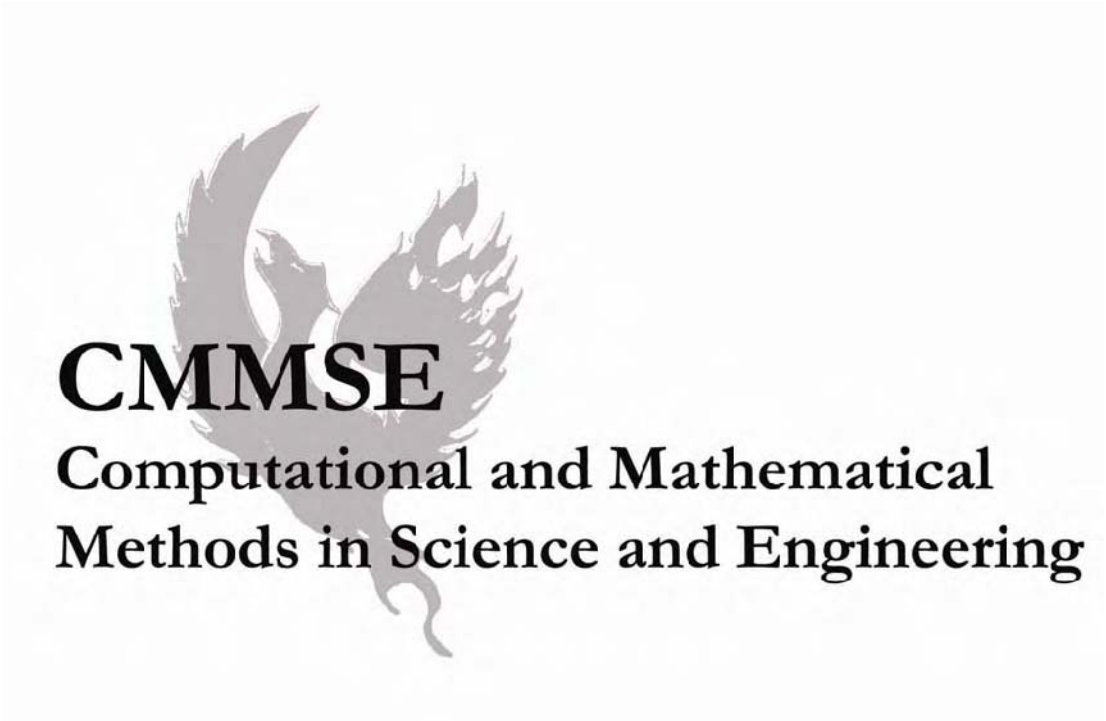
Proceedings of the 2009 International Conference on Computational and Mathematical Methods in Science and Engineering

Gijón (Asturias), Spain

June 30, July 1-3, 2009



Editor:
J. Vigo-Aguiar

Associate Editors:
Pedro Alonso (Spain), Shinnosuke Oharu (Japan)
Ezio Venturino (Italy) and Bruce A. Wade (USA)

ISBN 978-84-612-9727-6

Register Number: 09/9851

Printed on acid-free paper

Volume I

Preface:

We are honoured to bring you this collection of articles and extended abstracts from the *"**9th International Conference on Computational and Mathematical Methods in Science and Engineering"*** (CMMSE 2009), held at Gijón, Asturias, Spain, from June 30 to July 3, 2009. The primary focus of CMMSE is on new ideas and interdisciplinary interaction in rapidly growing fields of computational mathematics, mathematical modelling, and applications. CMMSE 2009 special sessions represent advances in mathematical modelling and engineering, industrial mathematics, computational quantum chemistry, multiscale modelling, high performance computing, mathematical biology, computation for complex networks, mathematical models in artificial intelligence and novel finite difference and hybrid methods for ordinary and partial differential equations. However the use of numerical models to represent different scientific and engineering situations is limited and leads to others fields like qualitative reasoning or the mathematical theory of artificial intelligence presented in these volumes, too.

Today the resolution of scientific problems is unthinkable without High performance computing techniques. This year we have the pleasure to work with the Spanish Network CAPAP-H "High Performance Computing on Heterogeneous Parallel Architectures." We would like to give a special mention to José Ranilla and Enrique S. Quintana-Ortí for their fabulous and very organized work.

The more human side of the conference is the special session realized in occasion on the 70 birthday of Prof. Giampietro Allasia, Università degli Studi di Torino, Italy. Ezio Venturino has written a short bibliography about Giampietro included into these volumes. We have learned much from him, and we will continue doing it.

We would like to thank the plenary speakers for their excellent contributions in research and leadership in their respective fields. We express our gratitude to special session organizers and to all members of the Scientific Committee, who have been a very important part of the conference, and, of course, to all participants.

These four volumes contain all the results of the conference. For a question of style, volumes I, II and III contains the articles written in LaTeX and volume IV contains the articles written in Word and abstracts.

We cordially welcome all participants. We hope you enjoy this conference.

Gijón, Asturias, Spain, June 30, 2009

J. Vigo-Aguiar, Pedro Alonso, Shin Oharu, Ezio Venturino, Bruce Wade

Acknowledgements:

We would like to express our gratitude to our sponsors:

- Ayuntamiento de Gijón,
- CajAstur,
- Departamento de Matemáticas de la Universidad de Oviedo,
- Ministerio de Ciencia e Innovación,
- Principado de Asturias (P.C.T.I.),
- Universidad de Oviedo.

Finally, we would like to thank all of the local organizers:

- L.F. Bayón, R. Cortina, I. Díaz, R. Gallego, J. Ranilla, P.M. Suárez (Universidad de Oviedo) and J. Delgado (Universidad de Zaragoza) for his support to make possible the conference.

- M.T. Bustos, A. Fernández (Universidad de Salamanca) for his invaluable help in the LaTeX and database preparation

CMMSE 2009 Plenary Speakers :

- Giampritero Allasia , Università degli Studi di Torino. Italy

- Tony Drummond, Lawrence Berkeley National Laboratory, USA

- Ian Hamilton., Wilfrid Laurier University, Canada

- S. Jin, University of Wisconsin Madison, USA

- Shinnosuke Oharu, Chuo University, Tokyo, Japan

- Ezio Venturino, Dipartimento di Matematica, University Torino, Italy

The organization of the "Minisymposium on High Performance Computing applied to Computational Problems in Science and Engineering" was part of the activities of the Spanish network "CAPAP-H: Red de computación de altas prestaciones sobre arquitecturas paralelas heterogéneas", supported by the Spanish "Ministerio de Innovación y Ciencia" (Acciones Complementarias TIN2007-29664-E).

Contents:

Volume I

 ISBN: 978-84-612-9727-6

Contents:

Volume II

Contents:

Volume III

Contents:

Volume IV

Abstracts

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Improvements on binary coding using parallel computing

P. Abascal Fuentes[1], David García Quintas[2] and Jorge Jiménez Meana[1]

[1] *Department of Mathematics, University of Oviedo*

[2] *Department: BE/CO, CERN*

emails: abascal@uniovi.es, David.Garcia.Quintas@cern.ch, meana@uniovi.es

Abstract

The error-correcting codes have many applications in fields related to communications.

This paper tackles some partition algorithms to optimize the data encoding. These algorithms are based on sliding windows and they allow a parallel implementation.

We analyze them and we will expound a comparative study between the different partition methods that we propose.

Key words: Code theory, parallel computation

1 Introduction

Codes were invented and designed to correct errors on noisy communication channels and the messages sent through this kind of channels must be sent as quickly and reliably as possible. These kind of channels include satellite pictures, telephone messages via glass fibre light, compact disc audio system.

At first, the error correcting codes were presented to detect and correct errors that could happen in noisy communication channels.

Over time, the error correcting codes have had other applications than those that originated it, and in Cryptography, they have been used to describe both public and private key cryptosystems, see [6], and to solve the problem of the secret sharing schemes, see [7] or [1].

Therefore we are interested in optimizing the involved algorithms. Consequently, we try to optimize by minimizing efficiency of matrix operations as the size of the messages to be encrypted is usually large.

In the section 2, we expose the basic rudiments about the error correcting codes, and, in the section 3, we will present some division techniques of messages that will enable us to apply parallel computing algorithms to improve their performance.

In the section 4, we will expound a comparative study between the different partition methods that we propose, and, finally, we will draw some the conclusions on the work done.

2 Error correcting codes

We make no attempt to present any complete Coding Theory, we will only provide some basic concepts and results about linear codes. For more details see [8].

2.1 Linear codes

Definition 1. *Let $q = p^t$ where p is a prime number and $t \in \mathbb{N}$. A $[n,k]$ linear code $\mathcal{C}$ over the finite field $\mathbb{F}_q$ is a k-dimensional linear subspace of $\mathbb{F}_q^n$. $c \in \mathcal{C}$ are called codewords.*

n is called the length, k the dimension and $n-k$ the redundancy of the code.

A linear code $\mathcal{C}$ is often described by a so-called generator matrix G then $\mathcal{C} = \left\{\overline{a}G / \overline{a} \in \mathbb{F}_q^k\right\}$, so encoding is a multiplication by G.

Definition 2. *Let $\mathcal{C}$ be a $[n,k]$ linear code over $\mathbb{F}_q$, we define $d_{\mathcal{C}} = \min\{d(\overline{u},\overline{v}) / \overline{u},\overline{v} \in \mathcal{C}\ \overline{u} \neq \overline{v}\}$ where $d(\overline{u},\overline{v})$ is the Hamming distance, $d_{\mathcal{C}}$ is called the minimum distance or simply the distance of the code $\mathcal{C}$.*

Proposition 1. *A code with minimum distance d can correct $\frac{1}{2}[d-1]$ errors. If d is even the code can simultaneously correct $\frac{1}{2}[d-2]$ errors and detect $\frac{d}{2}$ errors.*

3 Partitioning the message

In this section we are going to describe different methods of partitioning the original message in order to use parallel computing.

The partition will be made in blocks which we will name “windows”. It is distinguished two kinds of partitions: windows of fixed length and windows of bounded length.

3.1 Methods of partition in blocks of fixed length

3.1.1 Partitioning by columns

In this case, it is made a partition of the matrix in blocks of columns, the coded message is obtained by the yuxtaposition of the components calculated by each CPU.

So, the coded message is obtained by the algorithm 3.1

algorithm 3.1 Partitioning by columns of G

```
1: function ByColumnsOfG(G,m,b)
Require: G ∈ M_{r,c}, m ∈ M_{1,r}, b = LengthOfBlocks
Ensure: m' ∈ M_{1,c}                                   ▷ m' is m coded with G
2:   m' ← ()
3:   α ← 0...r−1                                       ▷ All the rows
4:   for i ∈ k·b ∀k ∈ ℕ/k·b ≤ c−b do
5:     β ← i...i+(b−1)                                 ▷ block i of b columns
6:     subG ← G[α,β]                                   ▷ subG ∈ M_{r,b}
7:     m' ← (m', m × subG)
8:   end for
9:   if c mod b > 0 then
10:    β ← (c−(c mod b))...c−1                         ▷ The rest of columns
11:    subG ← G[α,β]                                   ▷ subG ∈ M_{r,c mod b}
12:    m' ← (m', m × subG)
13:  end if
14:  return m'
15: end function
```

 ISBN: 978-84-612-9727-6

3.1.2 Partitioning by rows

This method is based in the partition of the matrix in blocks of rows and the message in blocks of components. All the blocks have a fixed length and each block (matrix and message) must be multiply by a CPU.

algorithm 3.2 Partitioning by rows of G

function BYROWSOFG(G,m,b)
Require: $G \in \mathcal{M}_{r,c}$, $m \in \mathcal{M}_{1,r}$, $b =$ LengthOfBlocks
Ensure: $\vec{m}' \in \mathcal{M}_{1,c}$ ▷ $\vec{m}'$ is m coded with G
  $\vec{m}' \leftarrow (0 \overset{c}{\ldots} 0)$
  $\beta \leftarrow 0 \ldots c-1$ ▷ All the columns
  for $i \in k \cdot b \, \forall k \in \mathbb{N} / k \cdot b \leq r - b$ **do**
    $\alpha \leftarrow i \ldots i + (b-1)$ ▷ block i of b rows
    $subG \leftarrow G[\alpha, \beta]$ ▷ $subG \in \mathcal{M}_{b,c}$
    $subM \leftarrow m[1, \alpha]$ ▷ $subM \in \mathcal{M}_{1,b}$
    $\vec{m}' \leftarrow \vec{m}' + subM \times subG$
  end for
  if $r \bmod b > 0$ **then**
    $\alpha \leftarrow (r - (r \bmod b)) \ldots r-1$ ▷ The rest of columns
    $subG \leftarrow G[\alpha, \beta]$ ▷ $subG \in \mathcal{M}_{r \bmod b, c}$
    $subM \leftarrow m[1, \alpha]$ ▷ $subM \in \mathcal{M}_{1, r \bmod b}$
    $\vec{m}' \leftarrow \vec{m}' + subM \times subG$
  end if
  return $\vec{m}'$
end function

We can notice that, in this method, it's possible to take advantage of the fact that the message has null blocks, then the number of operations is reduced.

3.2 Sliding windows methods

Now, the objective is to minimize the number of operations, so we have different strategies. We can divide the original message in null blocks and no null blocks, this blocks are denominated "Windows". In this case, the length of the windows it isn't fixed, only bounded.

The number of windows is variable, depends on the length of the original message we wish to code; paying attention to this, it is possible to design different interesting strategies.

algorithm 3.3 General method for calculating the windows

function GETWINDOWS(alg,m,b)
Require: función $alg\colon \{0,1\}^b \to \mathbb{N}$, $m \in \mathcal{M}_{1,r}$, $b \in \mathbb{N}$.
Ensure: $W = \{(b_0, e_0), \ldots, (b_n, e_n)\}$ ▷ The set of valid windows for (alg, m, b)
  $W \leftarrow ()$
  $i \leftarrow 0$
  while $i <$ length(m) **do**
    if $m[1,i] = 1$ **then** ▷ The beginning of the window
      $j = alg(m[1, i \ldots i+b-1])$
      $W \leftarrow (W, (i, i+j-1))$ ▷ That is, append the ordered pair to W
      $i \leftarrow i + j$
    else
      $i \leftarrow i + 1$
    end if
  end while
  return W
end function

Windowed Encoding Once we have a set of valid windows, the coded messages is obtained by application of the algorithm 3.4.

algorithm 3.4 Windowed Encoding

```
1: function WINDOWEDENCODING(W,m,G)
Require: W = {(b_0, e_0), ..., (b_n, e_n)}, m ∈ M_{1,r}, G = M_{r,c}.
Ensure: m' ∈ M_{1,c}                                   ▷ m' is m coded with G
2:     m' ← (0 .^c. 0)
3:     β ← 0...c−1                                     ▷ All the columns
4:     for w ∈ W do                                    ▷ For each window w = (b,e) of W
5:         α ← w_0 ... w_1
6:         subG ← G[α, β]                              ▷ subG ∈ M_{w_1−w_0+1,c}
7:         subM ← m[1, α]                              ▷ subM ∈ M_{1,w_1−w_0+1}
8:         m' ← m' + (subM × subG)
9:     end for
10:    return m'
11: end function
```

3.2.1 Partitioning by non null windows of fixed length

With this strategy the non null blocks have always the same length b, the algorithm covers the original message from the left to the right obtaining windows with 1 in the first component.

Notice that it could have consecutive non null windows but, never adjacent null windows.

This algorithm is described by means of the function 3.5. This one would be the argument introduced in the algorithm GetWindows 3.3

algorithm 3.5 Non null fixed length windows

```
1: function FIXEDWINDOWS(block)
Require: block ∈ {0,1}^k with k ≤ b
Ensure: l                                              ▷ is the ending index of the valid window
2:     l ← length(block)                               ▷ that is, b nearly always
3:     return l
4: end function
```

3.2.2 Partitioning by non null bounded length windows

With this variant the length of the non null blocks is bounded by a parameter b.

This strategy let blocks beginning and ending by 1 whose length is at most b.

The implementation could be the described one in the algorithm 3.6.

algorithm 3.6 Non null bounded length windows

```
1: function BOUNDEDWINDOWS(block)
Require: block ∈ {0,1}^k with k ≤ b
Ensure: l                                              ▷ is the ending index of the valid window
2:     l ← length(block)                               ▷ that is, b nearly always
3:     for i ∈ {0,...,l−1} do
4:         if block_{−i−1} = 1 then                    ▷ It is always satisfied by some i
5:             return l − i
6:         end if
7:     end for
8: end function
```

3.2.3 All ones windows methods

This method propose the partition of the message in windows formed by ones with bounded length and null windows of arbitrary length.

The original message is covered from the left to the right, producing null windows and all ones windows.

The algorithm 3.7 implement this strategy.

algorithm 3.7 All ones bounded windows

```
1: function BOUNDEDALL1SWINDOWS(block)
Require: block ∈ {0,1}^k with k ≤ b
Ensure: l                                              ▷ is the ending index of the valid window
2:     l ← length(block)                               ▷ that is, b nearly always
3:     for i ∈ {1,...,l−1} do
4:         if block_i = 0 then
5:             return i
6:         end if
7:     end for
8:     return l                                        ▷ that is, if ∄i/block_i = 0
9: end function
```

3.3 Implementation details

3.3.1 Parallelization

As it was showed in the section 3, one of the advantages derived from the use of windows is to be able of process them in parallel.

If we consider the code algorithm 3.4, it is easy to see that the principle which guide the parallelization will be the *decomposition based in data*, see [5][1], in which each window define an independent task (thread).

Concreting, the partition of the data – and consequently the derived tasks– depend on the windows and the corresponding rows of the generator matrix G: for each windows extended from the index i to j ($i > j$) of the original message m, the same rows of G are considered.

Compare the algorithm 3.8 with the new one 3.4. In this second paralleled version, it is observed how each task stores the partial results (line 10) which are accumulated before in $\vec{v}'$ in order to obtain the final solution.

algorithm 3.8 *Parallel* coding by means of windows

```
1: function PARALLELWINDOWEDENCODING(W,m,G)
Require: W = {(b_0,e_0),...,(b_n,e_n)}, m ∈ M_{1,r}, G = M_{r,c}.
Ensure: m' ∈ M_{1,c}                                   ▷ m' is m coded with G
2:     m' ← (0 .c. 0)
3:     β ← 0...c−1                                     ▷ All the columns
4:     partialCs ∈ M_{n+1,c}                           ▷ So many partial results as windows
5:     omp parallel for
6:     for w ∈ W do                                    ▷ For each window w = (b,e) of W
7:         α ← w_0...w_1
8:         subG ← G[α,β]                               ▷ subG ∈ M_{w_1−w_0+1,c}
9:         subM ← m[1,α]                               ▷ subM ∈ M_{1,w_1−w_0+1}
10:        partialCs_w ← (subM × subG)                 ▷ partialCs_w: result of window w
11:    end for
12:    for partialC ∈ partialCs do                     ▷ that is, adding each partial result
13:        m' ← m' + partialC
14:    end for
15:    return m'
16: end function
```

[1] page 34 y followings

Given this easy general scheme, the behavior of the coding parallel algorithm will depend on the structure and numbers of windows W given.

4 Comparing the methods

4.1 Methodology

The following variables play an important role on comparing the three methods of partitioning:

- The size of the code, that is, the size of the problem
- The partitioning method used.
- Sequential or parallel execution.

It is important to emphasize the last point due to the partitioning of the message into windows is an optimization of the «classic» method of the matrix multiplication.

For doing the comparatives, we have used pseudorandom messages and generator matrices, avoiding any slant in the binary distribution of the message.

On evaluating and comparing the methods between them, different criteria will be used. On one hand, on respect to parallel execution, the methods must be compare even between themselves because of the numbers of task can vary. These things guide us to the concept of *speedup* and *efficiency*. The first one is defined like the quotient between the sequential execution time and the parallel execution time. The efficiency is defined simply like the quotient between the speedup and the number of tasks, that is, a normalized speedup. As this metrics compare an algorithm with itself on varying the number of parallel tasks, temporal results are included in order to be able to compare the methods between themselves.

In the case of sequential comparing of the methods –section 4.2.1–, the speedup is only considered in the classic sense of the term, as the quotient between the old one and the new one. This metric is used to compare the methods between themselves and between them, resulting this «sequential speedup» proportional to the temporal performance in all the methods.

About the *efficiency* calculus. The *efficiency* is defined as

$$E(P) = \frac{\text{speedup}\,(P)}{P} = \frac{T(1)}{P \cdot T(P)}$$

with P the number of execution units, $T(1)$ is the runtime of *the best* sequential algorithm and $T(P)$ is the runtime of the parallel algorithm using P execution units. However, by convenience, it is used to consider $T(1)$ as the runtime of the parallel algorithm using only one execution unit.

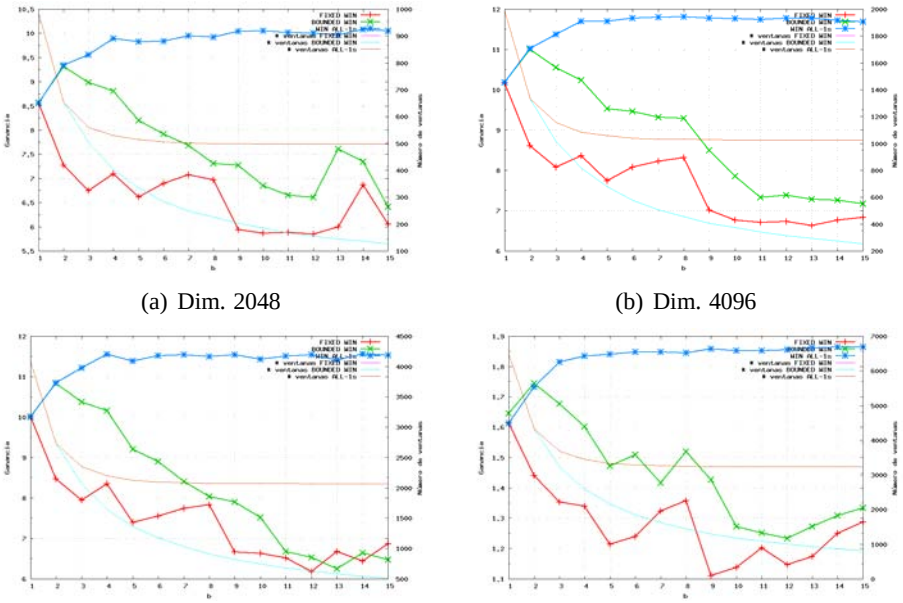
In this case, the only difference with respect to a sequential method is the use of the temporal storage into the loop (see algorithm 3.8, line 10). It isn't appreciated remarkable differences and, therefore, we have taken $T(1)$ like the runtime of the parallel algorithm using one thread.

4.2 Results

4.2.1 Using Partitioning into Windows

How do the proposed window-partitioning methods compare to the naïve approach based on multiplying the message by the code generator matrix? Algorithms `FIXED` and `BOUNDED` are analogous to this trivial calculation if a bound b equal to the dimension is considered. Note that this would be the case if the system only has one execution unit. On the other hand, the number of windows for `ALL-1s` is limited by the number of consecutive 1 bits: the occurrence in the message of a single 0 bit (except at the beginning or the end), results in two windows, regardless of b's value.

The behavior of the three algorithms has been studied for varying values of b and code dimensions. Figure 1 depicts the results for 2048, 4096, 8192 and 13000 code dimensions [2].



(a) Dim. 2048 (b) Dim. 4096

(c) Dim. 8192 (d) Dim. 13000

Figure 1: Sequential execution for varying values of b

4.2.2 Paralelization

The parallel execution of tasks results, in principle, in performance gains if enough physical execution units are available. Likewise, as the number of these tasks grows, their size decreases, enabling the system to store more of them in the cache at the same time. However, if the number

[2] Maybe the reader was expecting 16384 instead of 13000. This is not the case given the excessive memory requirements of the former. Nevertheless, this has no effect whatsoever on the resuls, merely breaks the powers-of-two sequence

 ISBN: 978-84-612-9727-6

tasks being executed is greater than the available parallel resources, performace will decrease, influenced as well by the task management overhead.

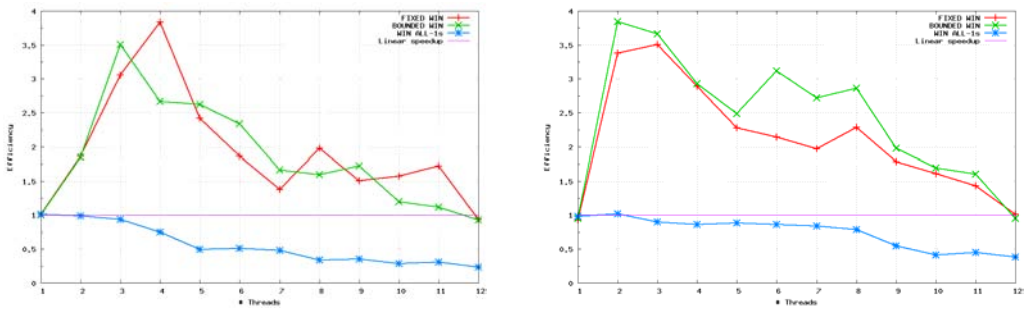
Two “forces” can be identified from this discussion, with increasing number of tasks running in parallel: the task competition for the physical execution units –that drives the performance down– and the higher portion of the tasks that is able to fit in the cache –driving the performance up–. Several examples that follow this model are shown in the following points.

Results are presented as efficiency vs. code dimension plots for each of the test systems considered.

Code of Dimension 2048**.** Figure 2 shows the results for this case. As for the sequential runs, algorithms `FIXED` and `BOUNDED` are closely related, unlike `ALL-1s`. The two former algorithms obtain an efficiency value above 1 for the whole range of number of tasks considered. That is to say, superlinear speedup. This is quite a remarkable result, where the effects of the previosly mentioned “forces” (see 4.2.2) can be seen into action. Namely, the one pushing performance up by taking advantage of the caches. Even when the execution units are forced into running more than one task, the fact that they are able to complete them fast enough, compensates for this overload: in figure 2(a) efficiency stays above 1 for more than four tasks for algorithms `FIXED` and `BOUNDED`. Clearly, as the number of tasks ramps up, the overhead due to the increase in management costs plus the higher number of tasks each physical executing unit has to handle ultimately brings the efficiency down.

On the other hand, `ALL-1s` hardly breaks this superlinear barrier. But this does not mean *anything* when it comes to comparing methods. Nor did it mean anything in the previous case between `FIXED` and `BOUNDED`: efficiency compares methods *solely against themselves*. In particular, against a sequential version of themselves.

Regarding this method’s results, it presents a typical behavior: efficiency progressively declining as the number of tasks increases over the number of available resources. In this case, the cache does not play a main role for the performance as the tasks were already small enough in the beginning.



(a) Dual Dual-core: 4 cores (b) Dual Quad-core: 8 cores

Figure 2: Codes of dimension 2048

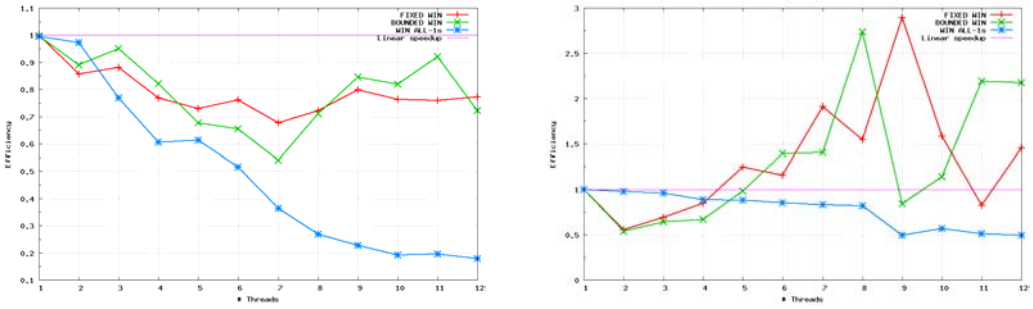
Code of dimension 4096**.** The results are the same than the previous one.

 ISBN: 978-84-612-9727-6

Code of dimension 8192**.** For the system based on the E5160 CPU, with a total of 8 MB of cache among the two cores, efficiency is always below one. This is due to the sizes of the tasks for this dimension. Nonetheless, because tasks progressively decrease in size with their number, a "bounce effect" can be observed, even when the number of tasks exceeds the number of available physical execution units. Case in point, algorithms `FIXED` and `BOUNDED` from seven to eight tasks.

For the system based on the E5410 –24 MB total cache–, this phenomenon happens earlier, from two to three tasks. This is consistent with the fact that tasks do not need to be as small as in the previous case, as more cache memory is available.

The main factor responsible for `ALL-1s`' performance for the least powerful system is not the lesser number of execution units: even when the number of tasks matche the number of these units, four, efficiency is at a modest 0.6. Once again, the preponderant actor is the cache memory. This idea is further reinforced when contrasted with results from the more powerful system: resuls for `ALL-1s` are good, with an efficiency of 0.81 for eight tasks, the number of physical execution units availables. Moreover, taken individually, the E5160 units are more powerful that those of the E5410 –3.00 GHz vs. 2.33 GHz– for an almost identical architecture. This difference in power is worthless if there is no data in the caches to take advantage of it.



(a) Dual Dual-core: 4 cores (b) Dual Quad-core: 8 cores

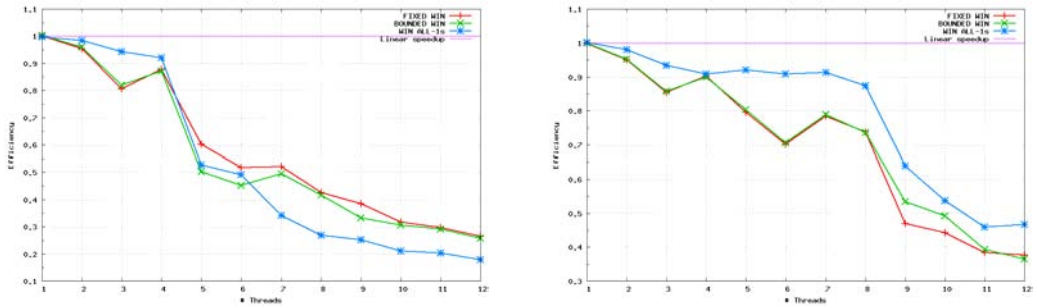
Figure 3: Codes of dimension 8192

Code of Dimension 13000**.** The largest dimension considered, amounts to a problem size of $\approx$ 165 MB. The efects mentioned previosly for dimension 8192 become more noticiable. Considering the plots presented in figure 4, it is visible how the only force supporting high efficiency values is the number of physical execution units: as soon as the amount of tasks exceeds this number, plots drop without the "bounce effect" present in other cases, where the ever decreasing size of the tasks balanced for it.

4.3 Comments on the Results

4.3.1 Sequential Execution

The best both sequential and parallel execution times are attained by `ALL-1s`. Nonetheless, considering a distributed memory environment, the large number of windows this algorithm



(a) Dual Dual-core: 4 cores (b) Dual Quad-core: 8 cores

Figure 4: Codes of dimension 13000

generates would result in an excesive communication overhead, with a small payload per window ratio. In such scenario, it would be better to resource to either `FIXED` or `BOUNDED` in order to split up the problem into chunks to be distributed to the processing nodes. Presumably, these nodes would locally run `ALL-1s` over these chunks. This hybrid aproach combines the strengths of both types of algorithms: data communication –distribution– that maximizes the payload while the ultimate node-wise computation performs as few operations as possible.

4.3.2 Parallel Execution

Despite the fact that partitioning into windows is beneficial even for sequential runs, the main motivation behind it is parallelization. As stated in 3.3.1, the idea of window is easily paralelizable, resulting in both remarkable performance results and a simple implementation.

It is important to stress once again what the results presented in section 4.2.2 mean: efficiency measures an algorithm *only* against itself, not against others. That `FIXED` and `BOUNDED` throw quite espectacular results for some dimensions does *not* imply that they are any better than `ALL-1s`. In fact, the latter outperforms the former two in all cases. That `ALL-1s`' efficiency is not as large as that of the other two algorithms is just a consequence of its sequential implementation already exploiting the system resources properly, limiting the room for improvement when parallelized. Despite this, efficiency values of $0.8 - 0.9$ are notably good, reflecting the method's excelent scalability.

5 Conclusions

The role played by the caches on the performance is the most important result. This aparently obvious fact is largely ignored mainly due to the numerous layers of abstraction commonly present. A careful study of the cache behavior is an unavoidable requirement when trying to attain an optimal performance.

This is more important now than ever, given the current trend towards parallel computing. On one hand, overall computation time is limited by the slowest subtask, rendering knowledge about the individual execution units properties paramount. On the other hand, the process of

parallelization involves the identification of the data that each of the subtasks will need access to. This data will usually be a subset of the total, making it more likely to fit completely in the cache available to the execution unit responsible for its associated task. Precisely this fact has been presented on this paper. In sum, not only extra computing power is made available, but it is also likely that huge gains –even superlinear speedups as we have had in some cases– are drawn from the extensive use of the caches.

Acknowledgements

The research reported on in this paper has been partially supported by Project FEDER-MEC-MTM2007-61193.

References

[1] P. ABASCAL AND J. TENA, *Algoritmos de búsqueda de un código corrector de errores realizando una estructura de acceso para compartir secretos*, V Reunión Española de Criptología y Seguridad de la Información, pp. 279-288, ISBN: 84-8497-820-6, 1998.

[2] INTEL CORP., *Quad-Core Intel Xeon Processor 5100 Series*, August, 2007, http://download.intel.com/design/Xeon/datashts/31335503.pdf

[3] INTEL CORP., *Quad-Core Intel Xeon Processor 5400 Series*, August, 2008, http://download.intel.com/design/xeon/datashts/318589.pdf

[4] B. KERNIGHAN AND D. RITCHIE, *The C Programming Language*, Prentice Hall, 1988.

[5] T. G. MATTSON, B. A. SANDERS AND B. L. MASSINGILL, *Patterns for Parallel Programming*, Addison-Wesley, 2004, Software Patterns Series.

[6] R. J. MCELIECE, *A public-key cryptosystem based on algebraic coding theory.*, DSN Prog. Rep., Jet Prop. Lab., California Inst. Technol., Pasadena, CA, (1987), 114–116.

[7] R. J. MCELIECE AND D. V. SARWATE, *On Sharing Secrets and Reed-Solomon Codes*, Comm ACM, (1981), 583–584

[8] N. J. A. SLOANE AND F. J. MACWILLIAMS, *The Theory of error-correcting codes*, North-Holland, 1988.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Crossover Operators for Permutations Equivalence between Position and Order Based Crossover

F. Aguado[1], J. M. Molinelli[1], G. Pérez[1] and C. Vidal[1]

[1] *Department of Computer Science, University of A Coruña, Spain.*

emails: aguado@udc.es, molineli@udc.es, gperez@udc.es, eicovima@udc.es

Abstract

In the context of genetic algorithms, the use of permutation based representations has worked out more conveniently than the classical binary encoding for some scheduling and combinatorial problems such as the Traveling Salesman Problem. In [1] we implemented in Coq several genetic operators proposed in [4, 9] to deal with the chromosomes of problems where the individuals are encoded as permutations; in these cases we specifically implemented the so called operators `pbx` and `obx`. In [2], we define with an axiomatic implementation two new operators `gen_pbx` and `gen_obx` which generalize the previous ones.

In this paper we formally specify the relation between these operators when restricted to the case of permutations without repetition. We also propose a new crossover operator which actually combines the genetic material from both parents in each child. Experimental results confirm that the use of one or another crossover makes no significant difference.

Key words: genetic algorithms, Coq, functional programming.

1 Introduction

Evolutive computation includes some heuristic techniques more or less based on Darwin's idea that the genetic component of better adapted individuals has a greater chance of being transmitted from parents to their descendants. These evolutive techniques include genetic algorithms (GAs) which have been successfully applied in various fields: optimization, pattern recognition, classification systems, etc. Although there are many GAs in the literature, they all show a process of codification or representation, of evaluation and reproduction. The way by which individuals will be represented is selected in the first one; binary encoding is the most commonly used. Evaluation is used to measure the fitness of each individual through a quality function. In reproduction the individuals are modified to improve the mean evaluation of the population in the next generation.

In some problems of combinatorial optimization, such as the Traveling Salesman Problem, the use of permutations instead of binary encoding is a better choice considering the nature of the problem. However, this decision conditions the selection of the genetic operators (mutation and crossover) that will take part in the reproduction process. If permutations are used to represent the population elements, the classical crossover consisting in cutting the chains at a certain point and exchanging the resulting parts between the progenitors could produce elements not present in the population. To solve this problem, several crossover operators (PBX, OBX) specifically designed to adapt to this special encoding representation are introduced in [3, 9].

In some previous works ([1, 2]), we formalized the definitions of these operators and carried out verified implementations of these using the formal testing system Coq ([10]) and the functional programming languages Ocaml and Haskell. With Coq we can define functions and predicates, formulate theorems and software specifications, develop interactive proofs for these theorems and certify these proofs. The Coq system allows to automatically extract functional programs from the algorithmic content of the proofs.

The crossover operators of [4, 9] were initially proposed for the crossover of permutations where the elements are not repeated. However, in some scheduling problems generally there are permutations with (possibly) repeated elements, and so, we find that the original procedure of the PBX and OBX crossovers allow more than one generalization. Basically, this is because in the crossover of two permutations we select some elements in one that are left out in the other. In this elimination process various selections can be made when the elements to leave out appear more than once in the lists.

In the work [2] we were able to generalize the operators defined in [1] using an axiomatic implementation based on the properties that characterize them. Moreover, these new functions verify the same specifications as those managed in [1] which are simply a particular case of those implemented previously. Then, when we restricted to the case of permutations without repetition, we recover the operators implemented in [1].

In this work, we prove that for this type of permutations, the two crossover operators based on order and position are closely related. Specifically, for each pair of chromosomes p and q, the crossover `gen_pbx` with an ℓ pattern of p and q is the crossover `gen_obx` of p and q with another pattern that depends on ℓ and, on p and q. Moreover, since the original idea of the crossover operator was to combine genetic material from both parents in each child, we propose a new crossover operator denoted by `gen_POBX` which in fact makes use, in the second child, of the genetic material from the first parent that was not used in the first child.

2 Background

Let D be a finite, non-empty and non-unary set. The chromosomes will be represented as finite chains (lists) of D elements, for which the type `list D` of Coq will be used.

 ISBN: 978-84-612-9727-6

To characterize the permutations in `list D`, the predicate `permutation` found in the Coq standard libraries will be used (here we will write $p \approx q$ for `permutation p q`). Chains of bits (`list bit` in Coq) will be used to represent the patterns that determine each crossover operator, `bit` being a set of two elements (here denoted as 0 and 1). Similarly, we recursively define the function `ext` that, given a p list and an ℓ crossover pattern, recovers the list obtained when extracting from p the alleles corresponding to the positions where the ℓ pattern has *ones*.

```
Fixpoint ext (l: (list bit)) (p: (list D)) {struct l}: (list D):=
    match l with
    | nil => nil
    | a :: t => match a with
                |zero => match p with
                         |nil => nil
                         | _ :: lt => ext t lt
                         end
                |one  => match p with
                         |nil => nil
                         | h :: lt => h :: ext t lt
                         end
                end
    end.
```

In addition to the properties of this function obtained in [1] we now prove others that will be used in the proof of the results in later sections; among these we highlight the following:

Lemma 2.1 *Let $p \in$ list D and $\ell \in$ list bit. If length$(p) \leq$ length(ℓ), then* $(\texttt{ext}\ \ell\ p) + + (\texttt{ext}\ \overline{\ell}\ p) \approx p$.

```
Lemma ext_total: forall (s: (list D)) (l: list bit),
   length s <= length l -> permutation ((ext l s) ++ (ext (complement l) s)) s.
```

The main concept is that of "inclusion" of lists. We will use the implementation ($\preceq$) of [2], based on the concepts of permutation and concatenation, included in the Coq standard libraries.

Definition 2.2 *Let $s, t \in$ list D. Denote that t* is included in *s, and it will be represented by $t \preceq s$, if $r \in$ list D exists, such that, $t + + r \approx s$.*

```
Definition inclusion (t s: (list D)) := {r:(list D) | (permutation (t ++ r) s)}.
```

Lemma 2.3 *For any $p \in$ list D and $\ell \in$ list bit we verify that*

$$(\texttt{ext}\ \ell\ p) \preceq p.$$

Lemma 2.4 *Let $p, q \in$ list D. Denote that:*

1. *$p \approx q$ if, and only if, $p \preceq q$ and $q \preceq p$.*

 ISBN: 978-84-612-9727-6

2. If $p \preceq q$ and $q \approx q'$, then $p \preceq q'$.

3. If $q \approx q'$ and $q \preceq p$, then $q' \preceq p$.

A function that obtains a difference between two lists s and t is also needed. The definition of this function (`diff`) is axiomatically made from the also axiomatic definition of `pos` ([2]). Given two lists s and t, the aim was for `pos` s t to be a list of *zeros* and *ones* having the same length as s (Axiom 1) and that, if $t \preceq s$, for `pos` s t to have *ones* at s places where the t elements are (Axiom 2).

Definition 2.5 *If $s, t \in$ list D, then* (`pos` s t) $\in$ *list bit verifies:*

Axiom 1 *length*(`pos` s t) = *length*(s)

Axiom 2 *if $t \preceq s$, then* (`ext` (`pos` s t) s) $\approx t$

```
Variable pos: (list D) -> (list D) -> (list bit).
Axiom axpos1: forall (s t: (list D)), length (pos s t) =  length s.
Axiom axpos2: forall (s t: (list D)), inclusion t s -> permutation (ext
(pos s t) s) t.
```

It was proven that, if $t \preceq s$, the number of *ones* of `pos` s t coincides with the length of t.

Now we implement the function `diff`.

Definition 2.6 *Let $s, t \in$ list D, we define* `diff` s t = `ext` $\overline{(pos\ s\ t)}$ s.

```
Definition diff (s t: (list D)) := ext (complement (pos s t)) s.
```

The following result shows *a fundamental property* of the previous function.

Teorema 2.7 *Let $s, t \in$ list D such that $t \preceq s$, then*

$$t + +\ \texttt{diff}\ s\ t \approx s.$$

```
Lemma perm_diff: forall (s t: (list D)),
inclusion t s -> permutation s (t ++ (diff s t)).
```

3 Position and Order Based Crossover

In [1] we extended the original definitions (for permutations without repetition) of [9] for crossover operators based on position and order to the case of any permutation. Later, in [2], we generalized the previous definitions. In the implementation proposed, the function `subs` was used that, given two lists p and q and an ℓ crossover pattern, recovers the list obtained, substituting in the list p the elements at locations marked with 1 in ℓ for the elements of q from left to right. The implementation in Coq of the function `subs` is the following:

 ISBN: 978-84-612-9727-6

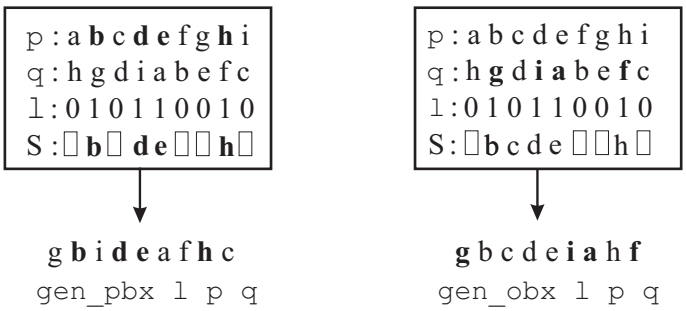


Figure 1: The crossover operators based on position and order for permutations without repetition

```
Fixpoint subs (l: (list bit))(p q: (list D)){struct p}: (list D):=
    match p with
    | nil => nil
    | hp :: tp => match q with
                  | nil => p
                  | hq :: tq  => match l with
                                 | nil => p
                                 | hl :: tl => match hl with
                                               |zero => hp :: (subs tl tp q)
                                               |one  => hq :: (subs tl tp tq)
                                               end
                                 end
                  end
    end.
```

Given two chromosomes P and Q, with $P \approx Q$ and crossover pattern ℓ, in the position based crossover (`gen_pbx`), the idea is to respect the positions of the P alleles indicated with the ones of ℓ and to complete the rest of the chromosome using the missing alleles, keeping the relative order they show in Q. Then, in the order based crossover (`gen_obx`), the aim is to keep the relative order of the alleles indicated with the ℓ pattern in Q and to complete the chromosome using the rest of the alleles at the position occupied in P. Figure 1 shows with an example how each of the operators for permutations without repetition works.

Definition 3.1 *Let $p, q \in$ list D and $\ell \in$ list bit, we define*

$$\texttt{gen\_pbx}\ \ell\ p\ q = \texttt{subs}\ \overline{\ell}\ p\ (\texttt{diff}\ q\ (\texttt{ext}\ \ell\ p))$$
$$\texttt{gen\_obx}\ \ell\ p\ q = \texttt{subs}\ (\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q))\ p\ (\texttt{ext}\ \ell\ q).$$

```
Definition gen_pbx (l: (list bit)) (p q: (list D)) :=
subs (complement l) p (diff q (ext l p))

Definition gen_obx (l: (list bit)) (p q: (list D)) :=
subs (pos p (ext l q)) p (ext l q).
```

Now we summarize in a theorem the results proved in [2] used to verify that the implementations proposed satisfy the specifications required.

Teorema 3.2 *Let $p, q \in list\ D$, with $p \approx q$, and let $\ell \in list\ bit$. Then, we have*

1. $\texttt{ext}\ \ell\ (\texttt{gen\_pbx}\ \ell\ p\ q) = \texttt{ext}\ \ell\ p$
2. $\texttt{ext}\ \overline{\ell}\ (\texttt{gen\_pbx}\ \ell\ p\ q) = \texttt{diff}\ q\ (\texttt{ext}\ \ell\ p)$, *si* $length(\ell) = length(p)$
3. $\texttt{ext}\ (\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q))(\texttt{gen\_obx}\ \ell\ p\ q) = \texttt{ext}\ \ell\ q$
4. $\texttt{ext}\ (\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q))(\texttt{gen\_obx}\ \ell\ p\ q) = \texttt{diff}\ p\ (\texttt{ext}\ \ell\ q)$
5. $\texttt{gen\_obx}\ \ell\ p\ q \approx p$
6. $\texttt{gen\_pbx}\ \ell\ p\ q \approx p$

4 Equivalence between gen_pbx and gen_obx for permutations without repetition

In the paper [2] we showed that there is only one possible implementation for the functions `gen_pbx` and `gen_obx` when only permutations without repetition are considered; this implementation naturally, coincides with that carried out in [1].
Now we will see that in this case of permutations without repetition, given two chromosomes, the crossover based on an order of particular pattern corresponds with the crossover based on the position having another pattern (that depends on the two chromosomes and on the initial pattern).

We will denote as *listnr* D the set of lists of D elements where there are no repeated elements. We implement in Coq this type of lists as follows:

```
Inductive listnr: list D -> Prop :=
  listnr_nil: listnr nil
  | listnr_cons: forall a l, ~ In a l -> listnr l -> listnr (a :: l).
```

The following lemmas will be used in the proof of Theorem 4.6.

Lemma 4.1 *Let $q \in listnr\ D$ and $\ell \in list\ bit$ such that $length(q) = length(\ell)$. Then, we have:*

1. $\texttt{pos}\ \ q\ (\texttt{ext}\ \ell\ q) = \ell$.
2. $\texttt{ext}\ \overline{\ell}\ q = \texttt{diff}\ q\ (\texttt{ext}\ \ell\ q)$.

```
Lemma pos_ext: forall (l: (list bit)) (q: (list D)),
   listnr q -> length l = length q -> pos q (ext l q) = l.
Lemma ext_compl_diff: forall (l: (list bit)) (q: (list D)),
   listnr q -> length l = length q ->
    (ext (complement l) q) = (diff q (ext l q)).
```

Lemma 4.2 *Let $s, t \in list\ D$ such that $t \in listnr\ D$ and $s \preceq t$, then*

$$\texttt{pos}\ t\ s = \overline{\texttt{pos}\ t\ (\texttt{diff}\ t\ s)}.$$

```
Lemma pos_complement: forall (t s: (list D)),
   listnr t -> inclusion s t -> pos t s = complement (pos t (diff t s)).
```

Lemma 4.3 *Let $p, x, y \in$ list D such that $p \in$ listnr D, $x \approx y$ and $x \preceq p$, then* $\texttt{pos}\ p\ x = \texttt{pos}\ p\ y$.

```
Lemma perm_inclusion_pos2: forall (p x y: (list D)),
   listnr p -> permutation x y -> inclusion x p -> pos p x = pos p y.
```

Lemma 4.4 *Let $p, q, r \in$ list D. If $p \approx q$ and $r \preceq p$, then*

$$(\texttt{diff}\ p\ r) \approx (\texttt{diff}\ q\ r).$$

```
Lemma perm_comp_diff: forall (p q r: (list D)),
   permutation p q -> inclusion r p -> permutation (difference p r) (difference q r).
```

Lemma 4.5 *Let $p, q \in$ listnr D and $\ell \in$ list bit such that $p \approx q$ and $length(q) = length(\ell)$. Then,*

1. $\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q) = \overline{\texttt{pos}\ p\ (\texttt{ext}\ \overline{\ell}\ q)}$.

2. $\texttt{pos}\ q\ (\texttt{ext}\ (\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q))\ p) = \ell$.

```
Lemma gen_ob_eq_pb_l: forall (l: (list bit)) (p q: (list D)),
   listnr q -> length l = length q -> permutation p q ->
    (pos p (ext l q)) = (complement (pos p (ext (complement l) q))).
Lemma gen_ob_eq_pb_r: forall (l: (list bit)) (p q: (list D)),
   listnr q -> length l = length q -> permutation p q ->
    (pos q (ext (pos p (ext l q)) p)) = l.
```

1. Applying Lemmas 4.1, 4.4 4.3 and 4.2, we obtain:
   $\underline{\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q)} = \texttt{pos}\ p\ \texttt{diff}\ q\ (\texttt{ext}\ \overline{\ell}\ q)) = \texttt{pos}\ p\ (\texttt{diff}\ p\ (\texttt{ext}\ \overline{\ell}\ q)) = \overline{\texttt{pos}\ p\ (\texttt{ext}\ \overline{\ell}\ q)}$.

2. Applying the previous section, the definition of `diff` and Lemmas 4.4, 4.3 and 4.1, we obtain:
   $\texttt{pos}\ q\ (\texttt{ext}\ (\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q))\ p) = \texttt{pos}\ q\ (\texttt{ext}\ (\overline{\texttt{pos}\ p\ (\texttt{ext}\ \overline{\ell}\ q)})\ p) = \texttt{pos}\ q\ (\texttt{diff}\ p\ (\texttt{ext}\ \overline{\ell}\ q)) = \texttt{pos}\ q\ (\texttt{diff}\ q\ (\texttt{ext}\ \overline{\ell}\ q)) = \texttt{pos}\ q\ (\texttt{ext}\ \ell\ q) = \ell$

The following theorem proves that, if we fix a pair of chromosomes, the map which changes the pattern ℓ of one crossover (`gen_obx`) to the pattern of the other crossover (`gen_pbx`) is bijective.

For each $n \in \mathbb{N}$, we will denote

$$(\texttt{list bit})_n = \{\ell \in (\texttt{list bit})\ ;\ length(\ell) = n\}.$$

Teorema 4.6 *Let $p, q \in listnr\ D$ and $\ell \in list\ bit$ such that $p \approx q$ and $length(q) = length(\ell)$. The map:*

$$f_{p,q} : (\texttt{list bit})_n \to (\texttt{list bit})_n$$

defined by $f_{p,q}(\ell) = \texttt{pos}\ p\ (\texttt{ext}\ \bar{\ell}\ q)$ is bijective with inverse

$$f_{p,q}^{-1}(\ell) = \texttt{pos}\ q\ (\texttt{ext}\ \bar{\ell}\ p) = f_{q,p}(\ell).$$

```
Theorem g_inv_f: forall (l: (list bit)) (p q: (list D)),
  listnr q -> length l = length q -> permutation p q ->
   g_pq p q (f_pq p q l) = l.
Theorem f_inv_g: forall (l: (list bit)) (p q: (list D)),
  listnr q -> length l = length q -> permutation p q ->
   f_pq p q (g_pq p q l) = l.
```

Moreover, for each pattern ℓ, we have that:

$$\begin{aligned} \texttt{gen\_obx}\ \ell\ p\ q &= \texttt{gen\_pbx}\ f_{p,q}(\ell)\ p\ q \\ \texttt{gen\_pbx}\ \ell\ p\ q &= \texttt{gen\_obx}\ f_{p,q}^{-1}(\ell)\ p\ q \end{aligned}$$

```
Theorem gen_ob_eq_pb: forall (l: (list bit)) (p q: (list D)),
  listnr q -> length l = length q -> permutation p q ->
   (gen_ob_cross l p q) = (gen_pb_cross (f_pq p q l) p q).
Theorem gen_pb_eq_ob: forall (l: (list bit)) (p q: (list D)),
  listnr q -> length l = length q -> permutation p q ->
   (gen_pb_cross l p q) = (gen_ob_cross (g_pq p q l) p q).
```

Firstly, we can apply the lemma 4.5 and state that

$$\texttt{pos}\ q\ (\texttt{ext}\ f_{p,q}(\ell)\ p) = \bar{\ell}.$$

Furthermore:

$$\begin{aligned} \texttt{gen\_obx}\ \ell\ p\ q &= \texttt{subs}\ (\texttt{pos}\ p\ (\texttt{ext}\ \ell\ q))\ p\ (\texttt{ext}\ \ell\ q) \\ \texttt{subs}\ (\overline{\texttt{pos}\ p\ (\texttt{ext}\ \bar{\ell}\ q)})\ p\ (\texttt{ext}\ \ell\ q) &= \texttt{subs}\ \overline{f_{p,q}(\ell)}\ p\ (\texttt{ext}\ \bar{\bar{\ell}}\ q) \\ \texttt{subs}\ \overline{f_{p,q}(\ell)}\ p\ (\texttt{ext}\ (\overline{\texttt{pos}\ q\ (\texttt{ext}\ f_{p,q}(\ell)\ p)}))\ q &= \texttt{subs}\ \overline{f_{p,q}(\ell)}\ p\ (\texttt{diff}\ q\ (\texttt{ext}\ f_{p,q}(\ell)\ p)) \\ &= \texttt{gen\_pbx}\ f_{p,q}(\ell)\ p\ q \end{aligned}$$

In order to show that $f_{p,q}$ is bijective, note that:

$$\begin{aligned} f_{p,q}(\overline{f_{p,q}^{-1}(\ell)}) &= f_{p,q}(\texttt{pos}\ q\ (\texttt{ext}\ \bar{\ell}\ p)) \\ \texttt{pos}\ p\ (\texttt{ext}\ (\overline{\texttt{pos}\ q\ (\texttt{ext}\ \bar{\ell}\ p)}))\ q &= \texttt{pos}\ p\ (\texttt{ext}(\texttt{pos}\ q\ (\texttt{ext}\ \ell\ p)))\ q \\ &= \ell. \end{aligned}$$

Reciprocally, we prove that $f_{p,q}^{-1}(f_{p,q}(\ell)) = \ell$. Finally, the only thing left is taking into account that:

$$\texttt{gen\_pbx}\ \ell\ p\ q\ =\ \texttt{gen\_pbx}\ (f_{p,q}(f_{p,q}^{-1}(\ell))\ p\ q\ = \texttt{gen\_obx}\ f_{p,q}^{-1}(\ell)\ p\ q$$

 ISBN: 978-84-612-9727-6

5 Position and Order Crossover Revisited

In a genetic algorithm, crossover involves combining elements from two parent chromosomes into (usually) two child chromosomes. The child chromosomes for the position and the order crossover operators proposed in [4] and generalized in [2] are:

$$\texttt{gen\_PBX}\ \ell\ p\ q\ = (\texttt{gen\_pbx}\ \ell\ p\ q, \texttt{gen\_pbx}\ \ell\ q\ p)$$

$$\texttt{gen\_OBX}\ \ell\ p\ q\ = (\texttt{gen\_obx}\ \ell\ p\ q, \texttt{gen\_obx}\ \ell\ q\ p)$$

where p and q are any pair of parent chromosomes and ℓ is any binary pattern.

The original idea of the biologically inspired crossover operator was to combine the genetic material from both parents into the two children. So, to define the second child, we should consider the genetic material from both parents not used in the first one (child). If we analyze the definition of `gen_PBX`, we notice that the first child $\texttt{gen\_pbx}\ \ell\ p\ q$ has the alleles of $\texttt{ext}\ \ell\ p$ in the positions where the pattern ℓ has ones. The rest of the alleles ($\texttt{diff}\ p\ (\texttt{ext}\ \ell\ p)$) appear in this first child following the order they have in q. If we want the second child to inherit the genetic material not used from both p and q, we expect the alleles of $\texttt{ext}\ \ell\ p$ to maintain the positions and order they have in q. The rest of the alleles must appear in the order they have in p. Because of this, we propose a new crossover operator `gen_POBX` defined by:

$$\texttt{gen\_POBX}\ \ell\ p\ q\ = (\texttt{gen\_pbx}\ \ell\ p\ q, \texttt{gen\_obx}\ \bar{\ell}\ q\ p).$$

If we denote by $\ell^* = \texttt{pos}\ q\ (\texttt{ext}\ \ell\ p)$, we have to verify

1. $\texttt{ext}\ \ell\ p = \texttt{ext}\ \ell\ (\texttt{gen\_pbx}\ \ell\ p\ q)$
2. $\texttt{ext}\ \bar{\ell^*}\ q = \texttt{ext}\ \bar{\ell}\ (\texttt{gen\_pbx}\ \ell\ p\ q)$
3. $\texttt{ext}\ \ell^*\ q = \texttt{ext}\ \ell^*\ (\texttt{gen\_obx}\ \bar{\ell}\ q\ p)$
4. $\texttt{ext}\ \bar{\ell}\ p = \texttt{ext}\ \bar{\ell^*}\ (\texttt{gen\_obx}\ \bar{\ell}\ q\ p)$

The first property corresponds to the first statement of Theorem 3.2. The second one also follows from 3.2 and the fact that

$$\texttt{diff}\ q\ (\texttt{ext}\ \ell\ p) = \texttt{ext}\ \overline{\texttt{pos}\ q\ (\texttt{ext}\ \ell\ p)}\ q.$$

With regard to the second child, first note (see Theorem 4.6) that:

$$\texttt{gen\_obx}\ \bar{\ell}\ q\ p = \texttt{gen\_pbx}\ f_{q,p}(\bar{\ell})\ q\ p = \texttt{gen\_pbx}\ \ell^*\ q\ p.$$

Now, and again applying Theorem 3.2, it holds that:

$$\texttt{ext}\ \ell^*\ (\texttt{gen\_obx}\ \bar{\ell}\ q\ p) = \texttt{ext}\ \ell^*\ (\texttt{gen\_pbx}\ \bar{\ell^*}\ q\ p) = \texttt{ext}\ \ell^*\ q.$$

Finally, and taking into account Lemma 4.5:

$$\begin{aligned}
\texttt{ext}\ \bar{\ell^*}\ (\texttt{gen_obx}\ \bar{\ell}\ q\ p) &= \texttt{ext}\ \bar{\ell^*}\ (\texttt{gen_pbx}\ \bar{\ell^*}\ q\ p) &= \\
&= \texttt{diff}\ p\ (\texttt{ext}\ \ell^*\ q) &= \\
&= \texttt{ext}\ \overline{\texttt{pos}\ p\ (\texttt{ext}\ \ell^*\ q)}\ p &= \\
&= \texttt{ext}\ \bar{\ell}\ p.
\end{aligned}$$

 ISBN: 978-84-612-9727-6

Acknowledgements

This work has been partially supported by the project INCITE08-PXIB105159PR (Xunta de Galicia-Spain).

References

[1] F. Aguado, J. L. Doncel, J. M. Molinelli, G. Pérez, C. Vidal, A. Vieites, *Certified Genetic Algorithms: Crossover Operators for Permutations*, Lecture Notes in Computer Science **4739**(2007) 282–289.

[2] F. Aguado, J. M. Molinelli, G. Pérez, C. Vidal, A. Vieites, *Generalización de los cruces basados en el orden y en la posición. Una implementación verificada*, Proceedings of CLEI 2007 (ISBN: 978-9968-9678-9-1) (2007).

[3] L. Davis, *Applying Adaptive Algorithms to Epistatic Domains*, IJCAI85, Proceedings of the Ninth International Joint Conference on Artificial Intelligence (1985) 162–164.

[4] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[5] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[6] A. K. De Jong, *An Analysis of the Behavior of a class of Genetic Adaptive Systems*, PhD Thesis, University of Michigan, 1975.

[7] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Massachusetts, 1996.

[8] C. Paulin-Mhoring, *Extraction de programmes dans le calcul des constructions*, Thése de doctorat, Université de Paris VII, 1989.

[9] G. Syswerda, *Schedule Optimization using Genetic Algorithms*, in [4], chapter 21 (1985) 332–349.

[10] INRIA, *The Coq proof assistant*, http://coq.inria.fr.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Numerical integration schemes for the discretization of BIEs related to wave propagation problems

Alessandra Aimi[1], Mauro Diligenti[1] and Chiara Guardasoni[2]

[1] *Department of Mathematics, University of Parma - Italy*

[2] *Department of Mathematics, University of Milano - Italy*

emails: alessandra.aimi@unipr.it, mauro.diligenti@unipr.it, chiara.guardasoni@unimi.it

Abstract

We consider wave propagation problems under vanishing initial and mixed boundary conditions reformulated in terms of boundary integral equations with retarded potential. The integral problem is then set in a weak form based on a natural energy identity satisfied by the differential problem and recently proposed. The related energetic Galerkin boundary element method used in the discretization phase, after a double integration in time variables, has to deal with weakly singular, singular and hypersingular double integrals in space variables. Efficient quadrature schemes for the numerical evaluation of these integrals are here proposed.

Key words: wave propagation, boundary integral equation, energetic Galerkin boundary element method, numerical integration
MSC 2000: 65N38, 65D30

1 Introduction

Time-dependent problems that are frequently modelled by hyperbolic partial differential equations can be dealt with the boundary integral equations (BIEs) method. The transformation of the problem to a BIE follows the same well-known method for elliptic boundary value problems. Boundary element methods (BEMs) have been successfully applied in the discretization phase. In principle, both the frequency-domain and time-domain BEM can be used for hyperbolic boundary value problems. Anyway, the consideration of the time-domain (transient) problem yields directly the unknown time-dependent quantities. In this case, the representation formula in terms of single layer and double layer potentials uses the fundamental solution of the hyperbolic partial differential equation and jump relations, giving rise to retarded boundary integral equations. Usual numerical discretization procedures include collocation techniques

and Laplace-Fourier methods coupled with Galerkin boundary elements in space. The convolution quadrature method for the time discretization has been developed in [10]. It provides a straightforward way to obtain a stable time stepping scheme using the Laplace transform of the kernel function. The application of Galerkin boundary elements in both space and time has been implemented by several authors but in this direction only the weak formulation due to Ha Duong [9] furnishes genuine convergence results. The only drawback of the method is that it has stability constants growing exponentially in time, as stated in [5].
Recently [3], we have considered 2D Dirichlet or Neumann problems for a temporally homogeneous (normalized) scalar wave equation in the time interval $[0,T]$, reformulated as a boundary integral equation with retarded potential. Special attention was devoted to a natural energy identity related to the differential problem, that leads to a space-time weak formulation for the BIEs, having, under suitable constraint, precise continuity and coerciveness properties and consequently the possibility to be discretized by unconditionally stable schemes with well-behaved stability constants even for large times.
The related energetic Galerkin boundary element method used in the discretization phase, after a double analytic integration in time variables, has to deal with weakly singular, singular and hypersingular double integrals in space variables. Efficient quadrature schemes for the numerical evaluation of these types of integrals are here proposed, referring to interior wave propagation problems with vanishing initial and mixed boundary conditions.

2 Model problem and its boundary integral weak formulation

Here we will consider a mixed boundary value problem for the wave equation in a polygonal domain $\Omega \subset \mathbb{R}^2$ with boundary Γ:

$$u_{tt} - \triangle u = 0, \qquad \mathbf{x} \in \Omega, t \in (0,T) \tag{1}$$
$$u(\mathbf{x},0) = u_t(\mathbf{x},0) = 0, \qquad \mathbf{x} \in \Omega \tag{2}$$
$$u(\mathbf{x},t) = \bar{u}(\mathbf{x},t), \qquad (\mathbf{x},t) \in \Sigma_T^u := \Gamma_u \times [0,T] \tag{3}$$
$$p(\mathbf{x},t) := \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x},t) = \bar{p}(\mathbf{x},t), \qquad (\mathbf{x},t) \in \Sigma_T^p := \Gamma_p \times [0,T] \tag{4}$$

where $\mathbf{n}$ is the unit outward normal vector of Γ, $\bar{\Gamma} = \bar{\Gamma}_u \cup \bar{\Gamma}_p$, $\Gamma_u \cap \Gamma_p = \emptyset$ and $\bar{u}$, $\bar{p}$ are given boundary data of Dirichlet an Neumann type, respectively. Let us consider the boundary integral representation of the solution of (1)-(4):

$$u(\mathbf{x},t) = \int_\Gamma \int_0^t \left[G(r, t-\tau) p(\boldsymbol{\xi},\tau) - \frac{\partial G}{\partial \mathbf{n}_{\boldsymbol{\xi}}}(r, t-\tau) u(\boldsymbol{\xi},\tau) \right] d\tau \, d\gamma_{\boldsymbol{\xi}}, \qquad \mathbf{x} \in \Omega, t \in (0,T), \tag{5}$$

 ISBN: 978-84-612-9727-6

where $r = \|\mathbf{r}\|_2 = \|\mathbf{x} - \boldsymbol{\xi}\|_2$ and

$$G(r, t-\tau) = \frac{1}{2\pi} \frac{H[t-\tau-r]}{[(t-\tau)^2 - r^2]^{\frac{1}{2}}} \tag{6}$$

is the forward fundamental solution of the two dimensional wave operator, with $H[\cdot]$ the Heaviside function. With a limiting process for $\mathbf{x}$ tending to Γ we obtain the space-time BIE (see [5])

$$\frac{1}{2}\, u(\mathbf{x}, t) = \int_\Gamma \int_0^t G(r, t-\tau) p(\boldsymbol{\xi}, \tau)\, d\tau\, d\gamma_{\boldsymbol{\xi}} - \int_\Gamma \int_0^t \frac{\partial G}{\partial \mathbf{n}_{\boldsymbol{\xi}}}(r, t-\tau) u(\boldsymbol{\xi}, \tau)\, d\tau\, d\gamma_{\boldsymbol{\xi}}\,, \tag{7}$$

which can be written, with obvious meaning of notation, in the compact form

$$\frac{1}{2}\, u(\mathbf{x}, t) = (Vp)(\mathbf{x}, t) - (Ku)(\mathbf{x}, t). \tag{8}$$

The BIE (8) is generally used to solve Dirichlet problems but can be employed for mixed problems too. However, in this last case, one can consider a second space-time BIE, obtainable from (5), performing the normal derivative with respect to $\mathbf{n}_{\mathbf{x}}$ and operating a limiting process for $\mathbf{x}$ tending to Γ:

$$\frac{1}{2}\, p(\mathbf{x}, t) = \int_\Gamma \int_0^t \frac{\partial G}{\partial \mathbf{n}_{\mathbf{x}}}(r, t-\tau) p(\boldsymbol{\xi}, \tau)\, d\tau\, d\gamma_{\boldsymbol{\xi}} - \int_\Gamma \int_0^t \frac{\partial^2 G}{\partial \mathbf{n}_{\mathbf{x}}\, \partial \mathbf{n}_{\boldsymbol{\xi}}}(r, t-\tau) u(\boldsymbol{\xi}, \tau)\, d\tau\, d\gamma_{\boldsymbol{\xi}}, \tag{9}$$

which can be written in the compact form

$$\frac{1}{2}\, p(\mathbf{x}, t) = (K'p)(\mathbf{x}, t) - (Du)(\mathbf{x}, t). \tag{10}$$

Hence, a mixed boundary value wave propagation problem can be rewritten as a system of two BIEs

$$\begin{cases} \frac{1}{2}\, \bar{u}(\mathbf{x}, t) = (Vp)(\mathbf{x}, t) - (Ku)(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_u, \;\; t \in (0, T), \\ \\ -\frac{1}{2}\, \bar{p}(\mathbf{x}, t) = -(K'p)(\mathbf{x}, t) + (Du)(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_p, \;\; t \in (0, T), \end{cases} \tag{11}$$

in the boundary unknowns the functions $p(\mathbf{x}, t)$ and $u(\mathbf{x}, t)$ on Γ_u, Γ_p, respectively. Note that the operator K' is the adjoint of the Cauchy singular operator K, which can be expressed as

$$Ku(\mathbf{x}, t) = \int_\Gamma \frac{\partial r}{\partial \mathbf{n}_{\boldsymbol{\xi}}} \int_0^t G(r, t-\tau) \left[u_t(\boldsymbol{\xi}, \tau) + \frac{u(\boldsymbol{\xi}, \tau)}{(t-\tau+r)} \right] d\tau\, d\gamma_{\boldsymbol{\xi}}\,, \tag{12}$$

while the hypersingular integral operator D can be equivalently expressed in the following way

$$\begin{aligned} Du(\mathbf{x}, t) = & \int_\Gamma \frac{\partial^2 r}{\partial \mathbf{n}_{\mathbf{x}} \partial \mathbf{n}_{\boldsymbol{\xi}}} \int_0^t G(r, t-\tau) \left[u_t(\boldsymbol{\xi}, \tau) + \frac{u(\boldsymbol{\xi}, \tau)}{(t-\tau+r)} \right] d\tau\, d\gamma_{\boldsymbol{\xi}} \\ & + \int_\Gamma \frac{\partial r}{\partial \mathbf{n}_{\mathbf{x}}} \frac{\partial r}{\partial \mathbf{n}_{\boldsymbol{\xi}}} \int_0^t G(r, t-\tau) \left[u_{tt}(\boldsymbol{\xi}, \tau) + 2\frac{u_t(\boldsymbol{\xi}, \tau)}{(t-\tau+r)} + 3\frac{u(\boldsymbol{\xi}, \tau)}{(t-\tau+r)^2} \right] d\tau\, d\gamma_{\boldsymbol{\xi}}. \end{aligned} \tag{13}$$

 ISBN: 978-84-612-9727-6

Further, in the right hand side of (11) one has to suitably insert boundary data. Then, the energetic weak formulation of the system (11) is defined (see [2, 3]) as

$$\begin{cases} < \frac{1}{2}\,\bar{u}_t, \psi >_{L^2(\Sigma_T^u)} = < (Vp)_t, \psi >_{L^2(\Sigma_T^u)} - < (Ku)_t, \psi >_{L^2(\Sigma_T^u)} \\ - < \frac{1}{2}\,\bar{p}, \eta_t >_{L^2(\Sigma_T^p)} = - < K'p, \eta_t >_{L^2(\Sigma_T^p)} + < Du, \eta_t >_{L^2(\Sigma_T^p)} \end{cases} \tag{14}$$

where ψ and η are suitable test functions, belonging to the same functional space of p, u, respectively.

2.1 Galerkin BEM discretization

For time discretization we consider a uniform decomposition of the time interval $[0, T]$ with time step $\Delta\, t = T/N_{\Delta t}\,, N_{\Delta t} \in \mathbb{N}^+$, generated by the $N_{\Delta t} + 1$ instants

$$t_k = k\,\Delta\, t, \quad k = 0, \cdots, N_{\Delta t}\,,$$

and we choose temporally piecewise constant shape functions for the approximation of p and piecewise linear shape functions for the approximation of u, although higher degree shape functions can be used. Note that, for this particular choice, temporal shape functions, for $k = 0, \cdots, N_{\Delta t} - 1$, will be defined as

$$v_k^p(t) = H[t - t_k] - H[t - t_{k+1}]$$

for the approximation of p, or as

$$v_k^u(t) = R(t - t_k) - 2\,R(t - t_{k+1}) + R(t - t_{k+2}),$$

for the approximation of u, where $R(t - t_k) = \frac{t - t_k}{\Delta\, t}\, H[t - t_k]$ is the ramp function.
For the space discretization, we employ a Galerkin boundary element method. We consider a boundary mesh on Γ_u constituted by M_u straight elements $\{e_1^u, \cdots, e_{M_u}^u\}$, with $2l_i^u := length(e_i^u)$, $l^u = \max_i\{2l_i^u\}$, $e_i^u \cap e_j^u = \emptyset$ if $i \neq j$ and such that $\bigcup_{i=1}^{M_u} \overline{e}_i^u = \overline{\Gamma}_u$. The same is done for the Neumann part of the boundary Γ_p with obvious change of notation. Let $l = \max\{l^u, l^p\}$. The functional background compels one to choose spatially shape functions belonging to $L^2(\Gamma_u)$ for the approximation of p and to $H_0^1(\Gamma_p)$ for the approximation of u. Hence, having defined $\mathcal{P}_{d_i}$ the space of algebraic polynomials of degree d_i, we consider, respectively, the space of piecewise polynomial functions

$$X_{-1,l} := \{w^p(\mathbf{x}) \in L^2(\Gamma_u) \,:\, w^p_{|e_i^u} \in \mathcal{P}_{d_i}\,, \forall\, e_i^u \subset \Gamma_u\,\}\,; \tag{15}$$

and the space of continuous piecewise polynomial functions

$$X_{0,l} := \{w^u(\mathbf{x}) \in C^0(\Gamma_p) \,:\, w^u_{|e_j^p} \in \mathcal{P}_{d_j}\,, \forall\, e_j^p \subset \Gamma_p\,\}\,. \tag{16}$$

Hence, denoted with M_l^u, M_l^p the number of unknowns on Γ_u and Γ_p, respectively, and having introduced the standard piecewise polynomial boundary element basis functions

 ISBN: 978-84-612-9727-6

$w_j^p(\mathbf{x})$, $j = 1, \cdots, M_l^p$, in $X_{-1,l}$ and $w_j^u(\mathbf{x})$, $j = 1, \cdots, M_l^u$ in $X_{0,l}$, the approximate solutions of the problem at hand will be expressed as

$$\tilde{p}(\mathbf{x},t) := \sum_{k=0}^{N_{\Delta t}-1} \sum_{j=1}^{M_l^p} \alpha_{pj}^{(k)}\, w_j^p(\mathbf{x})\, v_k^p(t) \qquad \text{and} \qquad \tilde{u}(\mathbf{x},t) := \sum_{k=0}^{N_{\Delta t}-1} \sum_{j=1}^{M_l^u} \alpha_{uj}^{(k)}\, w_j^u(\mathbf{x})\, v_k^u(t). \tag{17}$$

The Galerkin BEM discretization coming from energetic weak formulation produces the linear system (see [2])

$$\mathbb{E}\,\boldsymbol{\alpha} = \mathbf{b}\,, \tag{18}$$

where matrix $\mathbb{E}$ has a block lower triangular Toeplitz structure, since its elements depend on the difference $t_h - t_k$ and in particular they vanish if $t_h \le t_k$. Each block has dimension $M_l := M_l^p + M_l^u$. If we indicate with $\mathbb{E}^{(\ell)}$ the block obtained when $t_h - t_k = (\ell+1)\,\Delta t$, $\ell = 0, \ldots, N_{\Delta t} - 1$, the linear system can be written as

$$\begin{pmatrix} \mathbb{E}^{(0)} & 0 & 0 & \cdots & 0 \\ \mathbb{E}^{(1)} & \mathbb{E}^{(0)} & 0 & \cdots & 0 \\ \mathbb{E}^{(2)} & \mathbb{E}^{(1)} & \mathbb{E}^{(0)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \\ \mathbb{E}^{(N_{\Delta t}-1)} & \mathbb{E}^{(N_{\Delta t}-2)} & \cdots & \mathbb{E}^{(1)} & \mathbb{E}^{(0)} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{(0)} \\ \boldsymbol{\alpha}^{(1)} \\ \boldsymbol{\alpha}^{(2)} \\ \vdots \\ \boldsymbol{\alpha}^{(N_{\Delta t}-1)} \end{pmatrix} = \begin{pmatrix} \mathbf{b}^{(0)} \\ \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \\ \vdots \\ \mathbf{b}^{(N_{\Delta t}-1)} \end{pmatrix} \tag{19}$$

where

$$\boldsymbol{\alpha}^{(\ell)} = \left(\alpha_j^{(\ell)}\right) \qquad \text{and} \qquad \mathbf{b}^{(\ell)} = \left(b_j^{(\ell)}\right), \quad \text{with} \quad \ell = 0, \ldots, N_{\Delta t} - 1;\ j = 1, \ldots, M_l. \tag{20}$$

Note that each block has a 2×2 block sub-structure of the type

$$\mathbb{E}^{(\ell)} = \begin{bmatrix} \mathbb{E}_{uu}^{(\ell)} & \mathbb{E}_{up}^{(\ell)} \\ \mathbb{E}_{pu}^{(\ell)} & \mathbb{E}_{pp}^{(\ell)} \end{bmatrix} \tag{21}$$

where diagonal sub-blocks have dimensions M_l^p, M_l^u, respectively, $\mathbb{E}_{pu}^{(\ell)} = (\mathbb{E}_{up}^{(\ell)})^\top$ and the unknowns are organized as follows

$$\boldsymbol{\alpha}^{(\ell)} = \left(\alpha_{p1}^{(\ell)}, \cdots, \alpha_{pM_l^p}^{(\ell)}, \alpha_{u1}^{(\ell)}, \cdots, \alpha_{uM_l^u}^{(\ell)}\right)^\top.$$

The solution of (19) is obtained with a block forward substitution, i.e. at every time instant $t_\ell = (\ell+1)\,\Delta t$, $\ell = 0, \cdots, N_{\Delta t} - 1$, one computes

$$\mathbf{z}^{(\ell)} = \mathbf{b}^{(\ell)} - \sum_{j=1}^{\ell} \mathbb{E}^{(j)} \boldsymbol{\alpha}^{(\ell-j)}$$

and then solves the reduced linear system:

$$\mathbb{E}^{(0)} \boldsymbol{\alpha}^{(\ell)} = \mathbf{z}^{(\ell)}\,. \tag{22}$$

 ISBN: 978-84-612-9727-6

Procedure (22) is a time-marching technique, where the only matrix to be inverted is the positive definite $\mathbb{E}^{(0)}$ diagonal block, while all the other blocks are used to update at every time step the right-hand side. Owing to this procedure we can construct and store only the blocks $\mathbb{E}^{(0)}, \cdots, \mathbb{E}^{(N_{\Delta t}-1)}$ with a considerable reduction of computational cost and memory requirement.

Having set $\Delta_{hk} = t_h - t_k$, the matrix elements in blocks of the type $\mathbb{E}^{(\ell)}_{uu}$, after a double analytic integration in the time variables, are of the form

$$\sum_{\alpha,\beta=0}^{1} (-1)^{\alpha+\beta} \int_{\Gamma_u} w_i^p(\mathbf{x}) \int_{\Gamma_u} H[\Delta_{h+\alpha,k+\beta} - r]\, \mathcal{V}(r, t_{h+\alpha}, t_{k+\beta}) w_j^p(\boldsymbol{\xi})\, d\gamma_{\boldsymbol{\xi}}\, d\gamma_{\mathbf{x}}, \tag{23}$$

where

$$\mathcal{V}(r, t_h, t_k) = \frac{1}{2\pi} \left[\log\left(\Delta_{hk} + \sqrt{\Delta_{hk}^2 - r^2} \right) - \log\, r \right]\, ; \tag{24}$$

matrix elements in blocks of the type $\mathbb{E}^{(\ell)}_{up}$, after a double analytic integration in the time variables, are of the form

$$\sum_{\alpha,\beta,\delta=0}^{1} (-1)^{\alpha+\beta+\delta} \int_{\Gamma_u} w_i^p(\mathbf{x}) \int_{\Gamma_p} H[\Delta_{h+\alpha,k+\beta+\delta} - r]\, \mathcal{K}(r, t_{h+\alpha}, t_{k+\beta+\delta})\, w_j^u(\boldsymbol{\xi})\, d\gamma_{\boldsymbol{\xi}}\, d\gamma_{\mathbf{x}}, \tag{25}$$

where

$$\mathcal{K}(r, t_h, t_k) = \frac{1}{2\pi\, \Delta\, t}\, \frac{\mathbf{r} \cdot \mathbf{n}_{\boldsymbol{\xi}}}{r^2}\, \sqrt{\Delta_{hk}^2 - r^2}\, ; \tag{26}$$

matrix elements in blocks of the type $\mathbb{E}^{(\ell)}_{pp}$, after a double analytic integration in the time variables, are of the form

$$\sum_{\alpha,\beta,\delta=0}^{1} (-1)^{\alpha+\beta+\delta} \int_{\Gamma_p} w_i^u(\mathbf{x}) \int_{\Gamma_p} H[\Delta_{h+\alpha,k+\beta+\delta} - r]\, \mathcal{D}(r, t_{h+\alpha}, t_{k+\beta+\delta})\, w_j^u(\boldsymbol{\xi})\, d\gamma_{\boldsymbol{\xi}}\, d\gamma_{\mathbf{x}}, \tag{27}$$

where

$$\begin{aligned} \mathcal{D}(r, t_h, t_k) \;=\; & \frac{1}{2\pi\, (\Delta\, t)^2} \Big\{ \frac{\mathbf{r} \cdot \mathbf{n}_{\mathbf{x}}\;\; \mathbf{r} \cdot \mathbf{n}_{\boldsymbol{\xi}}}{r^2}\; \frac{\Delta_{hk} \sqrt{\Delta_{hk}^2 - r^2}}{r^2} + \\ & \frac{(\mathbf{n}_{\mathbf{x}} \cdot \mathbf{n}_{\boldsymbol{\xi}})}{2} \Big[\log(\Delta_{hk} + \sqrt{\Delta_{hk}^2 - r^2}) - \log r - \frac{\Delta_{hk}\, \sqrt{\Delta_{hk}^2 - r^2}}{r^2} \Big] \Big\}. \end{aligned} \tag{28}$$

We will refer to one of the double integrals in (23), (25) or (27), in the sequel indicated by

$$\int_{\Gamma} w_i(\mathbf{x}) \int_{\Gamma} H[\Delta_{hk} - r]\, \mathcal{S}(r, t_h, t_k)\, w_j(\boldsymbol{\xi})\, d\gamma_{\boldsymbol{\xi}}\, d\gamma_{\mathbf{x}}\, , \tag{29}$$

where $\mathcal{S}$ represents one of the kernels (24), (26) or (28) and where we have dropped redundant apices u, p in the notation, being clear which parts of the boundary and which test and shape functions are involved in the double integration, in relation to the

 ISBN: 978-84-612-9727-6

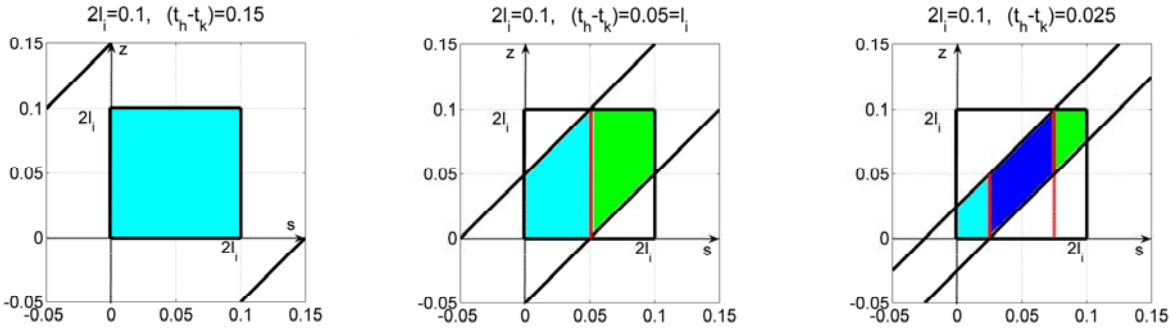


Figure 1: Double integration domain (coincident elements) for different values of Δ_{hk}.

fixed kernel. In the following, this simplification will be operated whenever possible. Using the standard element by element technique, the evaluation of every double integral of the form (29) is reduced to the assembling of local contributions of the type

$$\int_{e_i} \widetilde{w}_i^{(d_i)}(\mathbf{x}) \int_{e_j} H[\Delta_{hk} - r]\, \mathcal{S}(r, t_h, t_k)\, \widetilde{w}_j^{(d_j)}(\boldsymbol{\xi})\, d\gamma_{\boldsymbol{\xi}}\, d\gamma_{\mathbf{x}}\,, \tag{30}$$

where $\widetilde{w}_i^{(d_i)}(\mathbf{x})$ defines one of the local lagrangian basis function in the space variable of degree d_i defined over the element e_i of the boundary mesh.
Looking at (24), (26) and (28), we observe space singularities of type $\log r$, $O(r^{-1})$ and $O(r^{-2})$ as $r \to 0$, which are typical of $2D$ static weakly singular, singular and hypersingular kernels. Hence, we are particularly interested in the efficient evaluation of double integrals of type (30) when $e_i \equiv e_j$ and when $e_i\,, e_j$ are consecutive. Further, we remark that when the kernel is hypersingular and $e_i \equiv e_j$ we have to define both the inner and the outer integrals as Hadamard finite parts, while if e_i and e_j are consecutive, only the outer integral must be understood in the finite part sense. The correct interpretation of double integrals is the key point for any efficient numerical approach based on element by element technique (see [1]).

3 Numerical integration issues

Looking at the double integral (30), we find the Heaviside function $H[\Delta_{hk} - r]$ and the function $\sqrt{\Delta_{hk}^2 - r^2}$ in the kernel $\mathcal{S}(r, t_h, t_k)$, which are responsible for different type of troubles, that here we will illustrate for the case of coincident elements $e_i \equiv e_j$ of length $2\,l_i$, where $r = |s - z|$ in the local variables of integration.
In figure 1 we show the double integration domain represented by the intersection between the square $[0, 2\,l_i] \times [0, 2\,l_i]$ and the strip $|s - z| < \Delta_{hk}$ where the Heaviside function is not trivial, for different values of $\Delta_{hk} = t_h - t_k$.

The numerical quadrature in the outer variable of integration s has been optimally performed subdividing, when necessary, the outer interval of integration, as shown in

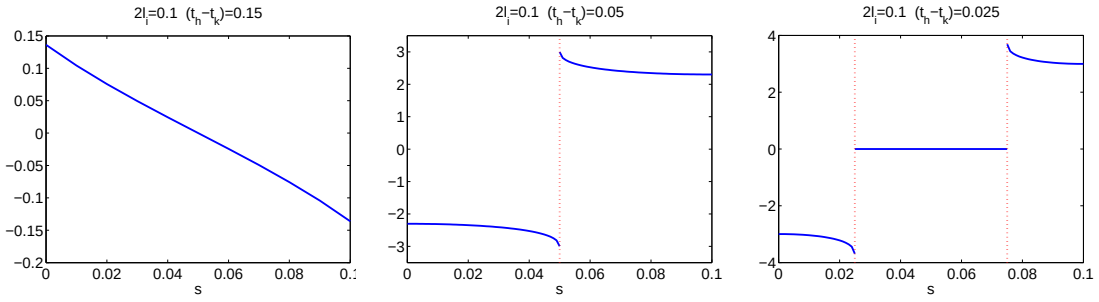


Figure 2: Behavior of outer integrand function derivative for different values of Δ_{hk}.

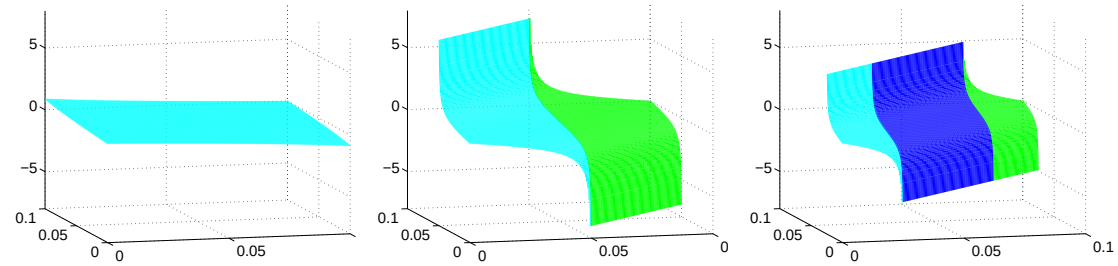


Figure 3: Behavior of $\frac{\partial}{\partial z}\sqrt{\Delta_{hk}^2 - |s-z|^2}$ for $\Delta_{hk} = 0.15, 0.05, 0.025$.

the same figure. In fact, the derivative with respect to s of the outer integrand function, after the inner integration, presents jumps in correspondence to the subdivision points, as depicted in figure 2 for the function

$$\frac{d}{d\,s}\int_0^{2\,l_i} H[\Delta_{hk} - |s-z|]\,\log(\Delta_{hk} + \sqrt{\Delta_{hk}^2 - |s-z|^2})\,\tilde{w}_j^{(0)}(z)\,dz\,.$$

Without this subdivision, one should use a lot of nodes in the gaussian quadrature formula to achieve the single precision accuracy.
Then, after subdividing, when necessary, the outer interval of integration in suitable subdomains, the inner numerical integration is still difficult, due to the presence of the square root function in the kernel. In fact, the argument of $\sqrt{\Delta_{hk}^2 - |s-z|^2}$ is always positive but it can assume very small values and in the limit for the argument tending to zero the derivative of the square root with respect to the inner variable of integration z becomes unbounded. This behavior happens along the oblique boundary of the double integration domain, as shown in figure 3 for different values of Δ_{hk}, and produces a bad performance, for instance in the evaluation of the integral

$$\int_0^{2\,l_i} \tilde{w}_i^{(0)}(s)\int_0^{2\,l_i} H[\Delta_{hk} - |s-z|]\,\log(\Delta_{hk} + \sqrt{\Delta_{hk}^2 - |s-z|^2})\,\tilde{w}_j^{(0)}(z)\,dz\,ds\,, \quad (31)$$

even of classical Gauss-Legendre quadrature formula, in the sense that one should use a lot of quadrature nodes to achieve the single precision accuracy. To overcome this

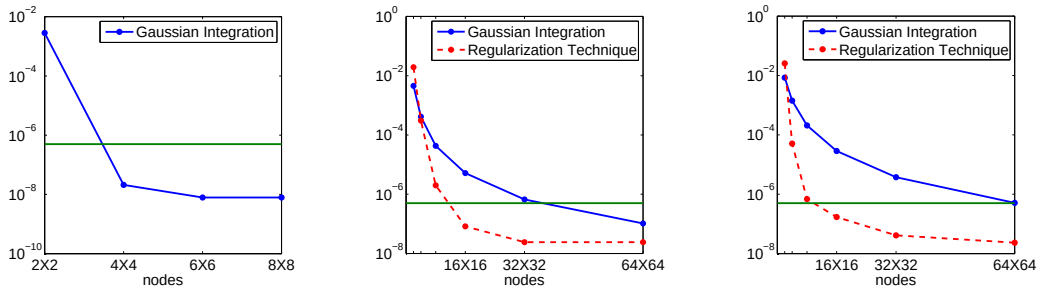


Figure 4: Computational cost of Gaussian quadrature and regularization procedure, for $\Delta_{hk} = 0.15, 0.05, 0.025$.

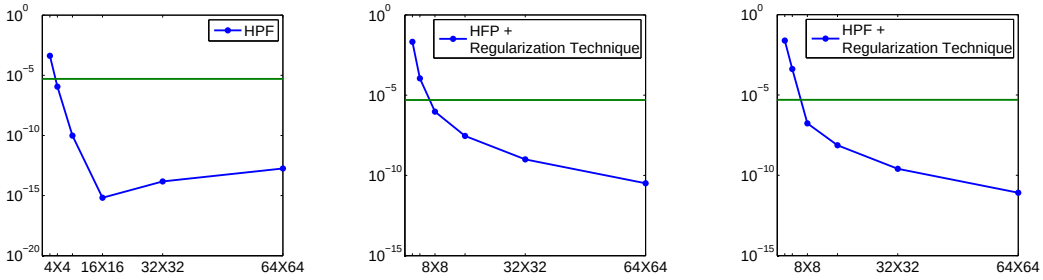


Figure 5: Computational cost of HFP quadrature coupled with regularization procedure, for $\Delta_{hk} = 0.15, 0.05, 0.025$.

difficulty, we have considered the regularization procedure introduced in [11], which suitably pushes the Gaussian nodes towards the end-points of the inner interval of integration and modify the Gaussian weights in order to regularize integrand functions with mild boundary "singularities".
In figure 4, we show the computational costs of Gaussian quadrature formula and regularization procedure in relation to the achievement of the single precision accuracy (horizontal line) in the evaluation of the double integral (31) for $\Delta_{hk} = 0.15, 0.05, 0.025$. The numerical treatment of weak singularities, strong singularities and hypersingularities, respectively of type $\log r$, $O(r^{-1})$ or $O(r^{-2})$ as $r \to 0$, has been operated through the quadrature schemes proposed in [1] and widely used in the context of Galerkin BEM related to BIEs coming from elliptic problems. In figure 5 we show the performance of the Hadamard Finite Part quadrature formula coupled, if necessary, with the regularization technique cited above, with respect to the single precision accuracy in the numerical evaluation of the hypersingular integral

$$\int_0^{2\,l_i} \tilde{w}_i^{(1)}(s) \int_0^{2\,l_i} H[\Delta_{hk} - |s-z|] \, \frac{\Delta_{hk}\sqrt{\Delta_{hk}^2 - |s-z|^2}}{|s-z|^2} \, \tilde{w}_j^{(1)}(z)\, dz\, ds\,,$$

for $\Delta_{hk} = 0.15, 0.05, 0.025$.

The above considerations have to be done, in the element by element technique, also for

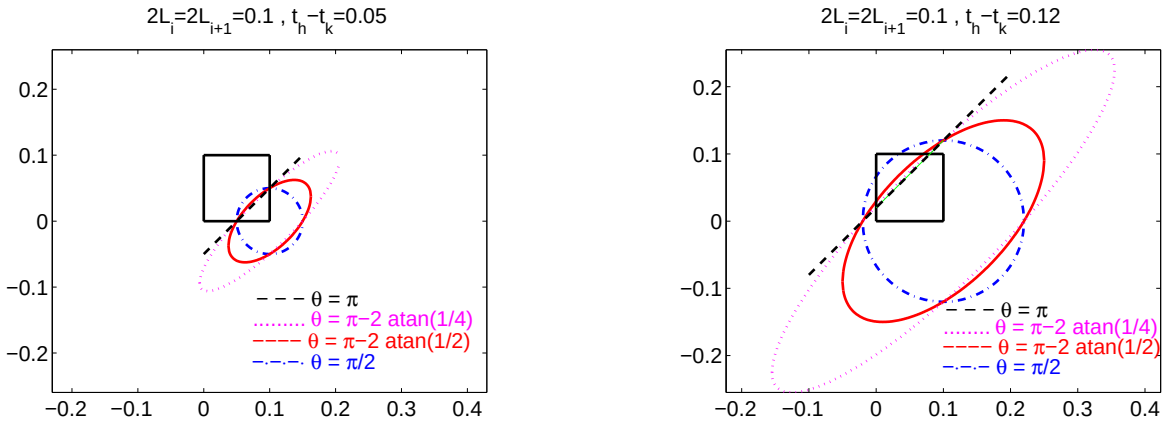


Figure 6: Double integration domain (contiguous elements) for different values of Δ_{hk} and different angles θ.

the other geometrical disposition of the double integration mesh elements e_i, e_j. In particular, if we consider contiguous elements e_i, e_{i+1}, with length respectively $2\,l_i$, $2\,l_{i+1}$, forming an angle θ, with $0 < \theta < \pi$, the double integration domain is constituted by the intersection between the rectangle $[0, 2\,l_i] \times [0, 2\,l_{i+1}]$ and the 2D domain $r^2 - \Delta_{hk}^2 < 0$, where the Heaviside function is not trivial, that for not aligned elements is an ellipsis centered in the unique singularity point $(2\,l_i, 0)$. The directions of the two axes of the ellipsis are $(1, 1)$ and $(-1, 1)$ with semi-length respectively $(1 + \cos(\theta))^{-1/2}$ and $(1 - \cos(\theta))^{-1/2}$. In figure 6, we show various types of intersections, i.e. double integration domains, for different values of Δ_{hk} and different angles θ between contiguous elements.

A complete illustration of the efficient numerical integration schemes we have used for the discretization of weakly, strongly and hypersingular BIEs related to wave propagation problems and which represent a valid alternative to those proposed in [8, 12], will be the subject of a paper which is currently in preparation [4].

4 Numerical results

To validate the presented discretization approach, we consider a standard benchmark (see for instance [6, 7]), involving a strip Ω of unit height, unbounded in horizontal direction, fixed in the inferior part where the Dirichlet boundary datum $\bar{u} = 0$ is assigned, and subject to a uniform traction $\bar{p} = H[t]$ in its superior part, as shown in figure 7. A finite portion of the strip is taken into account, in such a way that vertical dimension of the resulting rectangle is five times the other one. On the "cut" sides of the domain the equilibrium condition $\bar{p} = 0$ has been assigned. In order to apply energetic Galerkin BEM, we have introduced on Γ a uniform mesh with 48 elements ($l = 0.05$) and we have used in spatial variable constant shape function for the approximation of

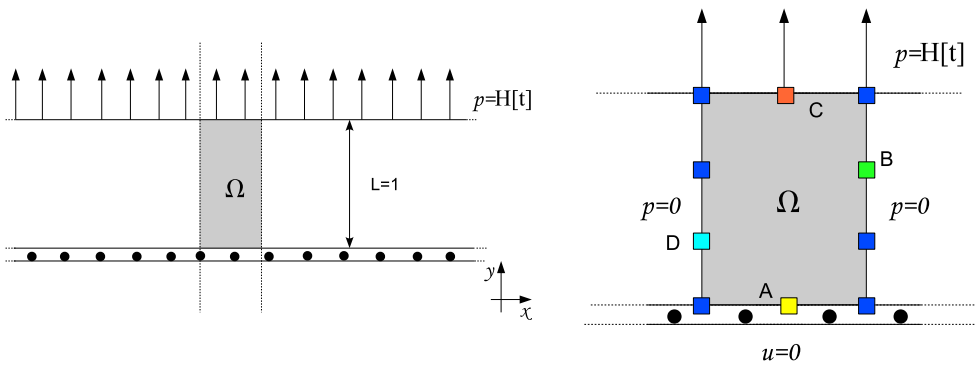


Figure 7: Domain and mixed boundary conditions for the test problem.

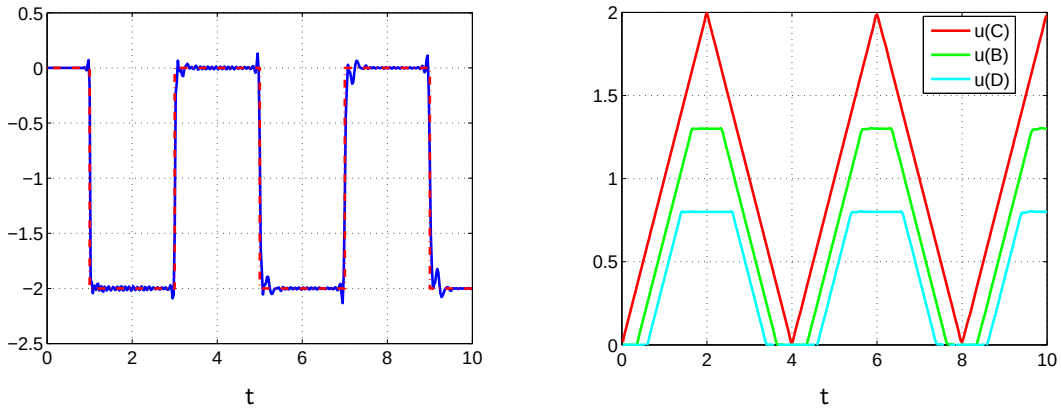


Figure 8: Approximate solution.

p and linear shape functions for the approximation of u. The time interval of analysis $[0, 10]$ has been discretized with different time steps. In figure 8 we show the recovered numerical solution obtained with $\Delta t = 0.025$. In particular traction in the point A, $p(A, t)$ is shown on the left, together with the corresponding analytical solution, while displacement in the points B, C, D, respectively $u(B, t)$, $u(C, t)$, $u(D, t)$ are shown on the right: here the three curves overlap with their respective analytical solutions. Note that the oscillations in the graph of $p(A, t)$ are due to the difficulty of approximating the jump discontinuities of the analytical solution; anyway, the obtained numerical solution is substantially better with respect to those found in literature, which present much more instabilities [6].

Acknowledgements

This work has been partially supported by italian Ministero dell'Istruzione, dell'Unversità e della Ricerca (MIUR) under contract PRIN 2007JL35WY_002.

References

[1] A. Aimi, M. Diligenti, G. Monegato, *New Numerical Integration Schemes for Applications of Galerkin BEM to 2D Problems*, Int. J. Numer. Meth. Engng. **40** (1997) 1977–1999.

[2] A. Aimi, M. Diligenti, *A new space-time energetic formulation for wave propagation analysis in layered media by BEMs*, Int. J. Numer. Meth. Engng. **75** (2008) 1102–1132.

[3] A. Aimi, M. Diligenti, C. Guardasoni, I. Mazzieri, S. Panizzi, *An energy approach to space-time Galerkin BEM for wave propagation problems*, Int. J. Numer. Meth. Engng., to appear.

[4] A. Aimi, M. Diligenti, C. Guardasoni, *Efficient numerical integration schemes for the discretization of (hyper) singular BIEs related to wave propagation problems*, in preparation.

[5] M. Costabel, *Time-dependent problems with the boundary integral method*, in: Encyclopedia of Computational Mechanics, Stein E, de Borst R., Hughes TJR (eds), Wiley, 2004.

[6] A. Frangi, G. Novati, *On the numerical stability of time-domain elastodynamic analyses by BEM*, Comput. Methods Appl. Mech. Engrg. **173** (1999) 403–417.

[7] A. Frangi, *"Causal" shape functions in the time domain boundary element method*, Comp. Mech. **25** (2000) 533–541.

[8] R. Gallego, J. Dominguez, *Hypersingular BEM for transient elastodynamics*, Int. J. Numer. Meth. Engng. **39** (1996) 1681–1705.

[9] T. Ha Duong, *On retarded potential boundary integral equations and their discretization*, in: Topics in computational wave propagation. Direct and inverse problems, Davies P, Duncan D, Martin P, Rynne B (eds), Springer-Verlag, Berlin, 2003.

[10] C. Lubich, *On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations*, Numer. Math. **67** (1994) 365–389.

[11] G. Monegato, L. Scuderi, *Numerical integration of functions with boundary singularities*, J. Comput. Appl. Math. **112** (1999) 201–214.

[12] Ch. Zhang, *A 2D hypersingular time-domain traction BEM for transient elastodynamic crack analysis*, Wave motion **35** (2002) 17–40.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Mimicking spatial effects in predator-prey models with group defense

Valerio Ajraldi[1] **and Ezio Venturino**[1]

[1] *Dipartimento di Matematica "Giuseppe Peano", Università di Torino, Italia*

emails: valerio_85@hotmail.com, ezio.venturino@unito.it

Abstract

In this paper we consider predator-prey models of interacting populations. We specifically consider prey populations which live together in herds. The existence of Hopf bifurcations is shown, which is a distinctive feature of this model in comparison with other classical population models of the same nature.

Key words: predator-prey, limit cycles, bifurcations, group defense
MSC 2000: AMS 92D25, 92D50

1 Introduction

In this paper we consider predator-prey models of interacting populations. We specifically consider prey populations which live together in herds.

In the literature, an idea for this type of interactions and group defense has been developed in [2], in which the model reads

$$\frac{dx}{dt} = xg(x) - p(x)y \ , \quad \frac{dy}{dt} = yf(y) + q(x)y \ ,$$

where x and y denote respectively prey and predators. The functions f and g represent the reproduction terms, while p and q describe the hunting process. The distinctive feature on group defense lies in the assumption that $q' < 0$ and $p' < 0$ both for $x > M$ for a suitable $M > 0$. In other words, the larger the prey population is, the smaller the success of hunting and the corresponding return rate are for predators. In our investigation we will depart from these assumptions, by looking instead at a situation in which the shape of the herd matters, not just the numbers. In spite of the fact that the word shape reminds of space, and therefore implies the use of sophisticated mathematical tools as partial differential equations, we are able to model the situation via the use of ordinary differential equations. The resulting dynamical system captures the essence of the species interactions and leads to the discovery of peculiar features that are not present in the corresponding classical predator-prey systems. Furthermore the system's behavior can be completely analyzed and reduced to the easy evaluation, in terms of the model parameters, of just one single entity.

2 The model

Consider two populations, the prey R and the predators F, which live in a common environment. We assume that the prey have a highly socialized behavior, in that they live together in herds, and when hunted by the predators, their behavior consists in a common defense, for which they surround the weaker individuals at the center of their herd. Therefore the attacking predators in general find difficult to enter in this defensive group. It follows that in general if the hunt is successful, only the individuals who are on the borderline of the defensive setting are harmed. Since the total prey population is R and occupies a certain area A on the ground, the individuals who for defensive purposes occupy the outermost positions are proportional to the length of the perimeter of the patch A, which is proportional then to the square root of R. The usual mass action law that is assumed for the interactions leading to loss of prey and gain for predators should in this case then be replaced by a Gompertz-like response, with the very specific exponent $\frac{1}{2}$ as remarked above, namely $F\sqrt{R}$. A similar reasoning has been applied in [1] to model poison release through the surface of a three-dimensional toxic plankton patch. Assuming furthermore logistic growth for the prey and the latter to be the only food source for the predators, the new predator-prey model then reads then

$$\begin{aligned} \frac{d}{dt}R(t) &= r(1-\frac{R(t)}{K})R(t) - a\sqrt{R(t)}F(t) \ , \\ \frac{d}{dt}F(t) &= -mF(t) + ae\sqrt{R(t)}F(t) \ . \end{aligned} \tag{1}$$

The first equation describes the prey evolution, the first term on the right hand side expresses logistic growth with r being the net reproduction rate and K the carrying capacity of the environment; the second term instead models the hunting process they are subject to by predators. The same term, scaled by the conversion coefficient e, appears also in the second equation, in which predators die out with mortality rate m in absence of their only food sources, namely the prey R.

All the model parameters are assumed to be nonnegative.

3 System's equilibria

In the long run, the system will evolve toward the equilibria, namely the points in the phase plane in which there is no change in the two populations. They are thus obtained by setting to zero the right hand sides of (1). We obtain therefore the points

$$P_1 = (0,0) \ , \quad P_2 = (K,0) \ , \quad P_3 = \left(\frac{m^2}{e^2a^2}, \frac{rm(Ke^2a^2-m^2)}{a^4Ke^3}\right) \ .$$

Feasibility is obvious in the first two cases, while for P_3 the nonnegativity of the predator popululation must be ensured. The latter gives $Ke^2a^2 - m^2 > 0$ i.e. the condition

$$\frac{Ke^2a^2}{m^2} > 1 \ . \tag{2}$$

 ISBN: 978-84-612-9727-6

4 Stability

Let us consider the Jacobian of the system (1)

$$J \equiv \begin{pmatrix} -\frac{2rR}{K} + r - \frac{aF}{2\sqrt{R}} & -a\sqrt{R} \\ \\ \frac{aeF}{2\sqrt{R}} & -m + ae\sqrt{R} \end{pmatrix} \tag{3}$$

with the characteristic equation $L_2 l^2 + L_1 l + L_0 = 0$ where the coefficients are explicitly given by

$$L_2 = 2K \ , \quad L_1 = \frac{1}{\sqrt{R}} \left(2K\sqrt{R}m - 2r\sqrt{R}K + aFK - 2KRea + 4rR^{\frac{3}{2}} \right) \ ,$$
$$L_0 = \frac{1}{\sqrt{R}} \left(4rR^{\frac{3}{2}}m - 2r\sqrt{R}Km + 2rRKea - 4rR^2ea + aFKm \right) \ .$$

Stability analysis of P_1

The Jacobian in this case is degenerate, but the instability of the origin can be easily seen, by observing that in its neighborhood the linearized equations stemming from (1) are

$$\frac{d}{dt}R(t) = rR(t) \ , \quad \frac{d}{dt}F(t) = -mF \ .$$

Therefore along the R axis $F = 0$, the trajectories move away from the equilibrium, showing its instability.

Stability analysis of P_2

The Jacobian matrix in this case is upper triangular, from which the eigenvalues are easily found to be $\lambda_1 = -r$ e $\lambda_2 = -m + ae\sqrt{K}$.

It follows then that the predator-free equilibrium P_2 is stable if $-m + ae\sqrt{K} < 0$ i.e. for

$$\frac{a^2e^2K}{m^2} < 1 \ . \tag{4}$$

Stability analysis of P_3

In this case the Jacobian matrix assumes the form

$$J_3 \equiv \begin{pmatrix} \frac{r(-3m^2+Ke^2a^2)}{2Ke^2a^2} & -\frac{m}{e} \\ \\ \frac{r(Ke^2a^2-m^2)}{2ea^2K} & 0 \end{pmatrix} \tag{5}$$

with the characteristic equation

$$2a^2e^2K\lambda^2 + (3rm^2 - rKe^2a^2)\lambda + rmKe^2a^2 - rm^3 = 0 \ . \tag{6}$$

 ISBN: 978-84-612-9727-6

The eigenvalues are then found to be

$$\lambda_1 = \frac{B + \sqrt{\Delta}}{4Ke^2a^2}, \quad \lambda_2 = \frac{B - \sqrt{\Delta}}{4Ke^2a^2} \ ,$$

where

$$B = -3m^2r + rKe^2a^2 \ , \quad \Delta = B^2 + 8Ke^2a^2rm^3 - 8K^2e^4a^4rm \ .$$

But since from (2) we have

$$B^2 - \Delta = 8Ke^2a^2rm(Ke^2a^2 - m^2) \geq 0$$

it follows that $|B|$ is larger than the real part of $\sqrt{\Delta}$ so that P_3 is stable if and only if $-3m^2 + Ke^2a^2 < 0$. The same conclusion can be obtained using the Routh-Hurwitz condition, namely

$$\frac{Ke^2a^2}{m^2} < 3 \ . \tag{7}$$

Note that for $\frac{Ke^2a^2}{m^2} = 3$ it follows $B = 0$ so that the eigenvalues become pure imaginary. In this case then a Hopf bifurcation arises.

5 Summary of the system's behavior

The previous analysis shows the occurrence of an important parameter, namely

$$\rho = \frac{Ke^2a^2}{m^2} \ , \tag{8}$$

for assessing the ultimate shape of the system's trajectories. Recall once again that the origin, P_1, is always unstable. The findings for the equilibria P_2 e P_3 give several cases, which can be summarized in the following Table 1.

Condition	P_2	P_3	bifurcation
$\rho < 1$	asymptotically stable	infeasible	
$\rho = 1$			transcritical $P_3 = P_2$
$1 < \rho < 3$	instabile	asymptotically stable	
$\rho = 3$	unstable		Hopf
$\rho > 3$	unstable	unstable	

Table 1: Equilibria of system (1)

 ISBN: 978-84-612-9727-6

It remains then to determine the behavior for the case of $\rho > 3$. To this end it is better to investigate the phase plane, by drawing the nullclines. The latter for $\frac{dR}{dt} = 0$ are the two curves

$$R = 0, \quad F = \frac{r\sqrt{R}}{a}\left(1 - \frac{R}{K}\right) , \tag{9}$$

while for $\frac{dF}{dt} = 0$ we find

$$F = 0, \quad R = \frac{m^2}{a^2e^2} . \tag{10}$$

We investigate now the phase plane plots and the ω-limit sets of the dynamical system as function of the newly introduced parameter ρ.

The case $\rho < 1$

Here we know from Table 1 that the coexistence equilibrium is not feasible, while P_2, the predator-free equilibrium is asymptotically stable. The nullclines in fact intersect for $F < 0$, and it is easily seen that the system's trajectories are bounded. In fact the set $S = \{(R, F) : R \leq R^*, F \leq F^*\}$, with $R^* = \frac{m^2}{a^2e^2}$, $F^* \geq F^+$, where the last value is obtained from the maximum value of the nullcline $\frac{dR}{dt} = 0$, namely $F^+(\frac{K}{3}) = \frac{2r\sqrt{K}}{3a}$, is a positively invariant set for the system (1). In fact on the vertical line $R = R^*$, the trajectories enter into the set S since they are oriented down and to the left. On the horizontal line $F = F^*$ trajectories move similarly, down and to the left. Therefore the ultimate system behavior is confined to this set. Since P_2 is the only equilibrium within this set, all the system's trajectories must approach it, therefore rendering it in this case globally asymptotically stable.

In Figure 1 we plot the phase plane diagram for the situation in which the parameters assume the values $a = e = r = 1$, $m = K = 2$. Figure 2 contains instead the two solutions as functions of time and the computed trajectory in the phase plane, for the same parameter values.

Note that as ρ grows toward the value $\rho = 1$ the vertical nullcline moves to the left, thereby approaching the equilibrium P_2. Thus when the latter is reached, the two nullclines intersect at the point $P_2 \equiv P_3$. Past this value, the coexistence equilibrium given by the intersection P_3 becomes feasible, and P_2 loses its stability. Hence P_2 and P_3 by meeting when $\rho = 1$ they interchange their stability properties, thus showing as claimed that for $\rho = 1$ a transcritical bifurcation occurs, see Figure 3.

The case $1 < \rho < 3$

For $1 < \rho < 3$ it follows that $\frac{K}{3} < \frac{m^2}{a^2e^2} < K$ and therefore the nullclines $F = \frac{r\sqrt{R}}{a}(1-\frac{R}{K})$ and $R = \frac{m^2}{a^2e^2}$ intersect at the point P_3, which now has become feasible.

In spite of this, we note that the boundedness result obtained for $\rho < 1$ continues to hold here as well. In fact, the proof of this statement is deferred to the next subsection for $\rho > 3$, as it is not enough here to take in the phase plane any vertical line $R = \widehat{R}$, with $\widehat{R} > K$, and again the horizontal line $f = F^*$. In fact this rectangle $\widehat{S}$, is not

 ISBN: 978-84-612-9727-6

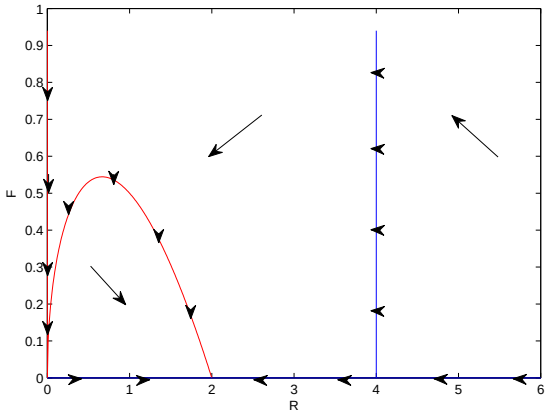


Figure 1: Phase plane diagram for the case $a = e = r = 1$, $m = K = 2$ i.e. for $\rho \equiv \frac{Ke^2a^2}{m^2} = 0.5$

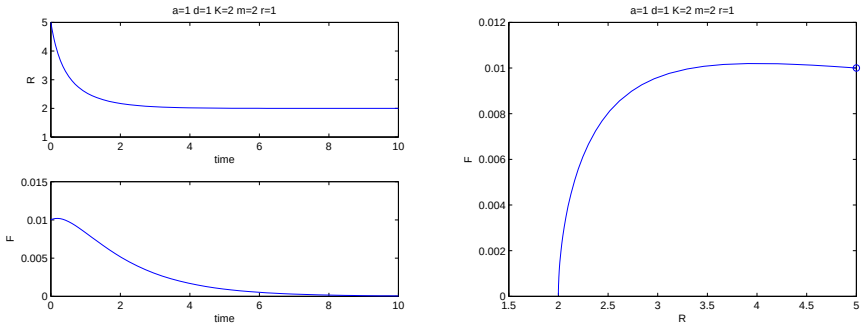


Figure 2: Left: solutions of (1) as functions of time for the paramter values $a = e = r = 1$ $m = K = 2$ i.e. for $\rho \equiv \frac{Ke^2a^2}{m^2} = 0.5$ and initial conditions $= [5; 0.01]$; Right: phase plane computed strajectory.

a positively invariant set for (1), since on the right part of the horizontal line the trajectories lead outside $\widehat{S}$, as it can be easily ascertained inspecting the phase plane diagram of Figure 4, for the parameter values $a = e = m = r = 1$, $K = 2$.

From the stability analysis of Table 1, for which P_3 is stable, in this case we expect the trajectories ultimately to tend to this equilibrium, although this cannot be directly inferred from Figure 4. This is confirmed by the numerical simulations results of Figure 5.

Again when ρ grows the vertical nullcline continues to move to the left, until it crosses the vertex of the parabola representing the second nullcline. At this point we have $\rho = 3$ and the eigenvalues become purely imaginary, giving a Hopf bifurcation. Oscillations then arise, as can be seen in Figures 6 and 7 for different initial conditions. From the latter, the stability of the limit cycle can be observed.

 ISBN: 978-84-612-9727-6

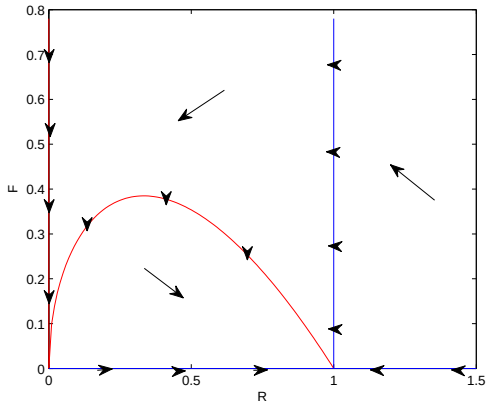


Figure 3: Transcritical bifurcation at $\rho = \frac{Ke^2a^2}{m^2} = 1$ for $a = e = K = m = r = 1$.

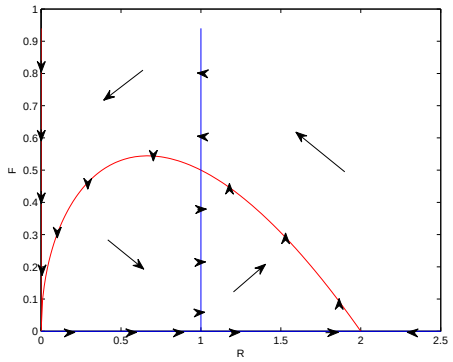


Figure 4: Phase plane diagram for the parameter values $a = e = m = r = 1$, $K = 2$ which correspond to $\rho = \frac{Ke^2a^2}{m^2} = 2$.

The case $\rho > 3$

In this situation Table 1 shows that all the possible system's equilibria become instable. The question is what happens to the system trajectories, what are then its ω-limit sets.

Let us consider the solution of (1) with initial condition (R_0, F_0). We assume at first that $R_0 \geq K$. Our goal is to show that at some instant in time t_2, the prey population will attain the value $R(t_2) = K$. From the second equation (1) it follows then

$$\frac{d}{dt}F(t) \geq m\left(-1 + \frac{ae}{m}\sqrt{K}\right) = m(\rho - 1) \geq m(\sqrt{3} - 1) \geq 0$$

from which $F(t) \geq F_0$. Since $r(1 - \frac{R(t)}{K}) \leq 0$ the first (1) gives $\frac{d}{dt}R(t) \leq -a\sqrt{R(t)}F_0$ so that upon integration,

$$\sqrt{R(t)} \leq -\frac{aF_0t}{2} + \sqrt{R_0}\ . \tag{11}$$

 ISBN: 978-84-612-9727-6

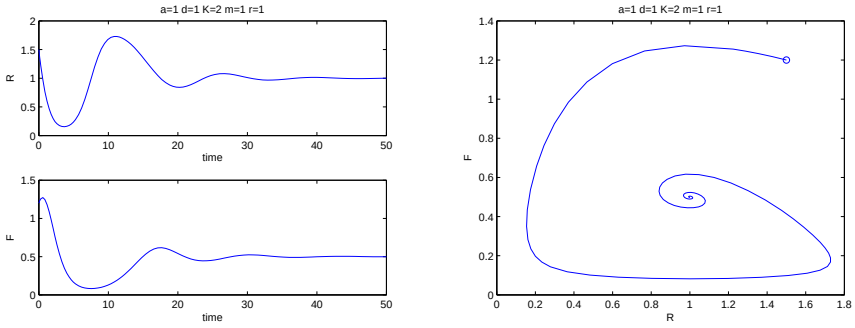


Figure 5: Left: Solutions of (1) as functions of time for the parameter values $a = e = r = m = 1$, $K = 2$, i.e. for $\rho = \frac{Ke^2a^2}{m^2} = 2$, with initial conditions $= [0.5; 0.5]$; Right: the computed trajectories in the phase plane in the same conditions.

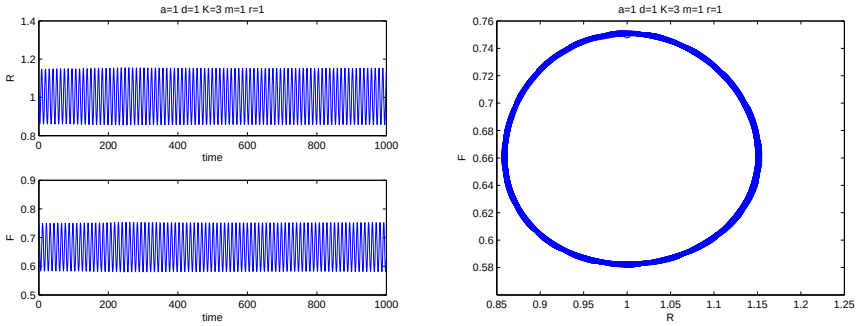


Figure 6: Onset of Hopf bifurcation for $a = e = m = r = 1$, $K = 3$ i.e. $\rho = \frac{Ke^2a^2}{m^2} = 3$ with initial conditions $= [1; 0.7]$; Left: solutions as functions of time; Right: phase plane picture.

Thus as t grows, it follows that $\sqrt{R}$ must decrease from its initial value R_0; this occurs until the inequality $F \geq F_0$ holds true. Let us denote by t_1 the instant in time at which $-\frac{aF_0t}{2} + \sqrt{R_0} = \sqrt{K}$. Solving this equation, we find $t_1 = \frac{2}{aF_0}(\sqrt{R_0} - \sqrt{K})$. Thus at time t_1 it follows from the previous inequality for $R(t)$, (11), that

$$\sqrt{R(t_1)} \leq \sqrt{K} \ . \tag{12}$$

There are now two possibilities. Either in $[0, t_1]$ there exists a t^* such that $R(t^*) < K$, or alternatively for every $t \in [0, t_1]$ we have $R(t) \geq K$. In the former case, by the continuity of the function $R(t)$, there must exist a $t_2 \in [0, t^*]$ such that $R(t_2) = K$. In the second case we must have $\sqrt{R(t_1)} \geq \sqrt{K}$ and therefore combining with the inequality (12) it follows that $\sqrt{R(t_1)} = \sqrt{K}$, i.e. $t_2 = t_1$. If the trajectory starts far enough on the right of the phase plane, we have thus established that the prey population for a suitable time t_2 attains the value K. This shows that any trajectory originating to the right of the vertical nullcline $R = K$ will cross it at some positive height, see Figure 8.

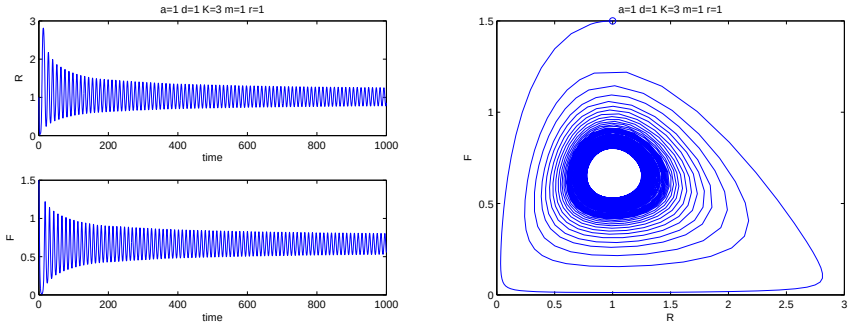


Figure 7: Onset of Hopf bifurcation for the same parameter values as of Figure 7, but with different initial conditions, namely = [1; 1.5]; Left: solutions as functions of time; Right: phase plane picture.

We now consider the trajectory emanating from the initial condition $(R_0, F_0) = \left(K, \frac{2\sqrt{3}r\sqrt{K}}{9a} + c\right)$ where $c > 0$ is chosen so as to satisfy $F_0 \geq F(t_2)$. For $R \geq \frac{m^2}{a^2e^2}$ from the second (1) it follows $\frac{d}{dt}F(t) \geq 0$ so that $F(t) \geq F_0$. The first (1) gives in this case $\frac{d}{dt}R(t) \leq -ac\sqrt{R(t)}$ and therefore integrating, we find

$$\sqrt{R(t)} \leq -\frac{act}{2} + \sqrt{R_0}\ . \tag{13}$$

Again this inequality holds as long as $F(t) \geq F_0$. Let us define the time t_3 at which $-\frac{act_3}{2} + \sqrt{R_0} = \frac{m}{ae}$, i.e. solving

$$t_3 = \frac{2}{ac}\left[\sqrt{R_0} - \frac{m}{ae}\right]\ . \tag{14}$$

Two alternative cases may arise once again, to show that at some time t_5 we have $R(t_5) = \frac{m^2}{a^2e^2}$. Firstly, if $R(t) \geq \frac{m^2}{a^2e^2}$ for all $t \in [0, t_3]$, since at t_3 from (13) it follows $R(t_3) \leq \frac{m^2}{a^2e^2}$, so that combining this last inequality with the former, we find $R(t_3) = \frac{m^2}{a^2e^2}$, i.e. $t_5 = t_3$. On the contrary, if there exists t_4 such that $R < \frac{m^2}{a^2e^2}$ by the continuity of the function $R(t)$, there must exist $t_5 \in [0, t_5]$ for which $R(t_5) = \frac{m^2}{a^2e^2}$. In both cases we have thus shown that trajectories originating from the vertical line $R_0 = K$ must at a certain time t_5 cross the nullcline $R = \frac{m^2}{a^2e^2}$, for which $\frac{dF}{dt} = 0$.

In summary, if the initial condtion is $R_0 > K$ in a finite time $t_2 < \infty$ we find $R(t_2) = K$, if instead $R_0 = K$ and $F_0 = \frac{2}{9a}r\sqrt{3k} + c$ in a finite time $t_5 < \infty$ it follows $R(t_5) = \frac{m^2}{a^2e^2}$. We can now construct a positively invariant set S^*. In fact it suffices to take the vertical line through $R = K$ up to a certain point with height h, then follow the trajectory emanating from (K, h) until it intercepts the vertical nullcline at a point $(\frac{m}{ae}, h^*)$. Finally take the horizontal line $F = h^*$ from this interception up to the vertical axis. The two segments on the coordinate axes complete the set S^*, see Figure 8. To ensure that it contains any possible trajectory of the system, if $R_0 \leq K$ we choose c so that $\frac{2}{9a}r\sqrt{3k} + c \geq F_0$ and the solution is entirely contained in S^*, if instead $R_0 > K$ we choose c so that $F(t_2) \leq \frac{2}{9a}r\sqrt{3k} + c$ so that from time t_2 onwards the trajectory

belongs to a suitable S^*. Thus for every possible initial condition a positively invariant set S^* exists.

A straightforward application of the Poincaré-Bendixson theorem shows that in these conditions a limit cycle must exist, since the coexistence equilibrium in this case is unstable. The simulations show it graphically, see Figure 9.

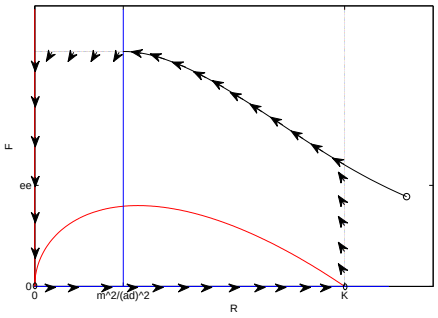


Figure 8: Positively invariant set.

6 Conclusion

We have proposed and analyzed a model of interacting populations, in which the biological relationship is expressed by the hunting of the predators and a defensive group strategy response from the prey. The analysis shows results contrary to what happens in classical predator prey systems. In fact in the latter either neutral type of oscillations are found, like in the Lokta-Volterra model [5, 7], or only globally stable equilibria exist, [3, 4]. In this case limit cycles arise naturally, under suitable conditions on the parameters. The key parameter ρ, (8) has been identified, as the parameter that contains the whole dynamics of the system. For $\rho < 1$ the predators are wiped out of the system, for $1 < \rho < 3$ the two species coexist at stable levels, for $\rho = 3$ there is the onset of limit cycles which remain from there onwards becoming larger and larger in amplitude. Therefore coexistence of the species can in this case also be ensured by stable oscillations, which are triggered by suitably low values of the predator's mortality, or correspondingly larger values of the prey carrying capacity or the hunting rate or the conversion coefficient. The model's behavior bears some similarities with the Holling-Tanner model, although the latter is formulated under a completely different set of assumptions. The bottom line of this investigation effort states then that it is not necessary to introduce Michaelis-Menten type terms or the predators' carrying capacity proportional to the prey amount in order to trigger stable oscillations into a predator-prey system. These arise also rather naturally if the prey live in herds and adopt a group defense strategy.

 ISBN: 978-84-612-9727-6

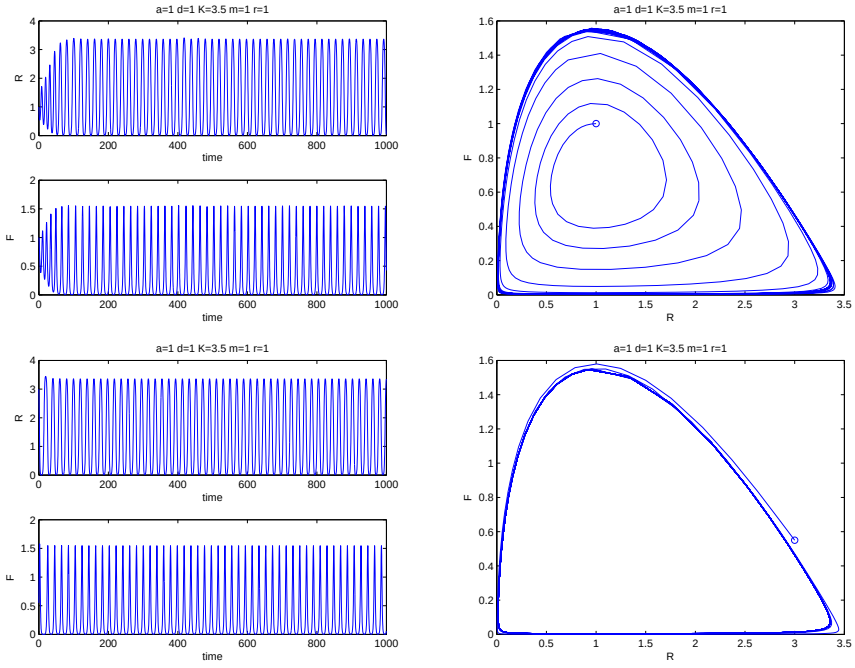


Figure 9: Simulations for the parameter values $a = e = r = m = 1$, $K = 3.5$, which imply $\rho = \frac{Ke^2a^2}{m^2} = 3.5$ with initial conditions $(1; 1)$ inside the limit cycle, top, and $(3; 0.55)$ outside the limit cycle, bottom; Left: solutions as functions of time; Right: phase plane plots showing the limit cycle whose existence is proven by the Poincaré-Bendixson theorem.

References

[1] J. Chattopadhyay, S. Chatterjee, E. Venturino, *Patchy agglomeration as a transition from monospecies to recurrent plankton blooms*, Journal of Theoretical Biology, **253** (2008) 289–295.

[2] H. I. Freedman, G. Wolkowitz, *Predator-prey systems with group defence: the paradox of enrichment revisited*, Bull. Math. Biol. **48** (1986) 493–508.

[3] B. S. Goh, *Stability in models of mutualism*, *The American Naturalist*, **113**, (1979) 261–275.

[4] M. Hirsch, S. Smale, *Differential Equations, Dynamical Systems and Linear Algebra*, Academic Press, New York, 1974.

[5] A. J. Lotka, *Elements of Mathematical Biology*, Dover, New York, 1956.

[6] J. D. Murray, *Mathematical Biology*, Springer Verlag, New York, 1989.

[7] V. Volterra, U. D'Ancona, *La concorrenza vitale tra le specie nell'ambiente marino*, VIIe Congr. int. acqui. et de pêche, Paris, (1931) 1–14.

 ISBN: 978-84-612-9727-6

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Recent advances in the parallel iterative solution of large-scale sparse linear systems

J. I. Aliaga[1], M. Bollhöfer[2], A. F. Martín[1] and E. S. Quintana-Ortí[1]

[1] *Department of Computer Science and Engineering, Universidad Jaume I, Spain*

[2] *Institute of Computational Mathematics, TU-Braunschweig, Germany*

emails: aliaga@icc.uji.es, m.bollhoefer@tu-braunschweig.de, martina@icc.uji.es, quintana@icc.uji.es

Abstract

In this paper we will review the main design issues which are involved in the development of the OpenMP version of the ILUPACK library. The approach to parallelization is based on graph partitioning algorithms to split the computation into concurrent tasks, which are then mapped to the processors for the parallel numerical solution steps of the solver. Experimental results for several PDE-based applications on a shared-memory multiprocessor with up to 16 processors are given.

Key words: Large-scale sparse linear systems, multilevel ILU preconditioning, Krylov subspace methods, task-level parallelism, shared-memory multiprocessors

1 Introduction

Many modern scientific computing applications are driven by the need to solve increasingly larger-scale sparse linear systems that arise from the discretization of mathematical models of physical systems. For these kind of applications, the development for fast and efficient numerical solution techniques becomes crucial.

While sparse direct solvers have proven to be extremely efficient for a large class of application problems, they perform poorly for those where fill-in becomes an issue. For example, direct methods are highly efficient when applied to the numerical solution of PDEs in two spatial dimensions (2D), but for three dimensional problems, they scale poorly with problem size in terms of computational and memory complexity. Even without considering application problems governed by PDEs, systems with several millions of unknowns are now routinely encountered, making the use of iterative methods virtually mandatory.

Among the main classes of iterative methods that are currently being used to solve large-scale sparse linear systems, we focus on on preconditioned Krylov subspace methods. In our approach, preconditioning relies on a multilevel variant of the incomplete

LU (ILU) decomposition that is constructed from the underlying system matrix A in a purely algebraic way, i.e., without specific tailoring to the problem. It is based on the so-called inverse-based approach which lays the foundations for the software package ILUPACK[1]. While not being as efficient as AMG-based solution techniques for PDE problems, multilevel inverse-based ILU preconditioned iterative solvers are general-purpose techniques which lead to a reasonable compromise between robustness and efficiency for a wide range of application problems. Indeed, in [2, 5], ILUPACK-based solvers have been found to be extremely robust, reliable and memory-efficient for the solution of several large-scale challenging application problems.

The large-scale application problems currently faced by ILUPACK makes it evident the need to reduce the time-to-solution employing parallel computing techniques. In this paper, we focus on the main ideas of our approach for the development of a parallel version of the ILUPACK library for the solution of Symmetric Positive Definite (SPD) systems of equations on shared-memory multiprocessors [1]. In the rest of this paper we briefly review some of the aspects that will be considered: the approach followed by ILUPACK for the iterative solution of large-scale sparse linear systems (Section 2), our approach to parallelization (Section 3), and a reduced set of experiments with our parallel solver (Section 4).

2 Iterative solution of sparse linear systems

The preconditioning approach of ILUPACK relies on the computation of the so-called inverse-based Multilevel Incomplete Cholesky (MIC) decomposition. This multilevel preconditioner is based on a root-free IC Cholesky factorization kernel that, when applied to the coefficient matrix A, produces the following block approximate decomposition

$$P^T A P = \begin{pmatrix} B & F^T \\ F & C \end{pmatrix} = \begin{pmatrix} L_B & 0 \\ L_F & \mathrm{I} \end{pmatrix} \begin{pmatrix} D_B & 0 \\ 0 & S_C \end{pmatrix} \begin{pmatrix} L_B^T & L_F^T \\ 0 & \mathrm{I} \end{pmatrix} + E = L D L^T + E,$$

where P is symmetric permutation which separates the unknowns of the system into fine (B block) and coarse (C block) variables, and E refers to some error matrix which contains those "small" entries which have been dropped during the approximate factorization. The permutation matrix P is constructed in such a way that the norm of the inverse Cholesky factor is bounded, i.e., $\|L^{-1}\| \leq \kappa$, where κ is user-prescribed moderate bound. The whole method is then restarted on the approximate Schur complement S_C, turning this computation into a multilevel approach. The computation is finished when, at a given level, S_C is void or "dense enough" to be handled by a dense Cholesky solver. A theoretical analysis along with numerical observations justifies that this strategy improves the numerical performance of the method as the iterative solution of the preconditioned linear system involves the inverse of the approximate Cholesky factor [2, 3]. The preconditioned system is solved with the Preconditioned Conjugate Gradient (PCG) solver. The major computational kernels of this Krylov

[1] `http://ilupack.tu-bs.de`

 ISBN: 978-84-612-9727-6

subspace method are: multiplication of a sparse matrix with a vector, and application of the MIC preconditioner.

3 Parallel Iterative solution of sparse linear systems

Our approach for the computation of the parallel MIC employs Multilevel Nested Dissection (MLND) partitioning algorithms [4]. These algorithms use a heuristic partitioning strategy which is based on the recursive application of vertex separator-finding algorithms to the adjacency graph of A, G_A. The recursion is applied until the desired number of independent subgraphs are obtained, and the resulting graph exhibits the property that disconnected subgraphs or separators can be eliminated independently. This property is captured in the task dependency tree in Figure 1, which also identifies a certain ordering of the elimination process imposed by the dependencies. As a result of this process, a permuted coefficient matrix $A \rightarrow \Pi^T A \Pi$ is obtained, and because of its structure (see Figure 1), concurrency can be easily exploited for the computation of the preconditioner. For the parallel MIC, essentially the multilevel approach of ILUPACK is applied within the nodes of the task tree, taking care of the dependencies during the computation. In Figure 1, this implies that the four leading diagonal blocks are factorized in parallel, after that, the two diagonal blocks corresponding to the second-level separator nodes still in parallel, and finally the root node. The parallelization of the operations involved in the preconditioned iteration is done conformally with the logical structure built during the computation of the parallel MIC. For details, see [1].

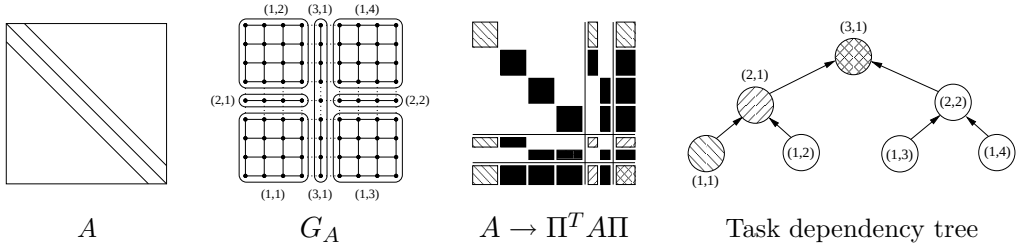


Figure 1: MLND partitioning. From left to right: natural ordering, nested dissection, nested dissection (re)ordering, and task dependency tree.

4 Experimental Results

The test problems which are considered in this section are derived from the linear finite element discretization of the $3D$ elliptic PDE

$$-\nabla \cdot K \nabla u = f, \tag{1}$$

in a 3D computational domain, where $K(x, y, z)$ is chosen with positive random coefficients. The target platform is a SGI Altix 350 CC-NUMA shared-memory multipro-

cessor consisting on 16 Intel Itanium2@1.5 GHz processors sharing 32 GBytes of RAM connected via a SGI NUMAlink network.

Table 1 compares the performance of the sequential algorithm in ILUPACK (results for $p = 1$) with the performance of our parallel solver using $p = 2, 4, 8, 16$ processors, when applied to a discrete version of (1) with $n = 5,413,520$ equations and unknowns, and $nnz = 78,935,174$ nonzero elements. Separate results are provided for the MLND, MIC, and PCG stages. The results for the MLND partitioning step were obtained using ParMETIS parallel partitioning package [4]. Although the execution time for this step is reduced significantly, the computational complexity of the partitioning step increases with p because we are considering a truncated version of the dissection procedure, i.e., a larger value of p translates to additional levels of recursion in the nested dissection of G_A. The table also reports the total number of nonzero elements in the MIC Cholesky triangular factor (in millions), the execution time of the MIC factorization, as well as the number of iteration steps required for convergence and execution time for the PCG solver. As we can observe in Table 1, the computational time is efficiently reduced as p increases, in a practical demonstration that our parallelization approach reveals a high degree of concurrence for a moderate number of processors, while preserving the semantics of the sequential preconditioner in ILUPACK (because the number of iteration steps is only slightly increased).

p	MLND	MIC		PCG	
		mem.	time	steps	time
	[sec]	[nonzeros $\times 10^{-6}$]	[sec]		[sec]
1	0.0	100.7	279.7	60	362.3
2	50.2	100.2	164.3	61	242.6
4	27.2	100.0	76.8	62	106.9
8	15.4	99.2	38.5	62	57.0
16	9.2	98.9	19.8	62	30.2

Table 1: Parallel performance for the OpenMP-version of the ILUPACK library.

References

[1] José I. Aliaga, Matthias Bollhöfer, Alberto F. Martín, and Enrique S. Quintana-Ortí. Exploiting thread-level parallelism in the iterative solution of sparse linear systems. 2008. submitted to Parallel Comput.

[2] Matthias Bollhöfer, Marcus Grote, and Olaf Schenk. Algebraic multilevel preconditioner for the helmholtz equation in heterogeneous media. *SIAM J. Scientific Computing*, 2008. to appear.

[3] Matthias Bollhöfer and Yousef Saad. Multilevel preconditioners constructed from inverse–based ILUs. *SIAM J. Sci. Comput.*, 27(5):1627–1650, 2006. special issue on the 8–th Copper Mountain Conference on Iterative Methods.

[4] George Karypis and Vipin Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *J. Parallel Distrib. Comput.*, 48(1):71–95, 1998.

[5] Olaf Schenk, Matthias Bollhöfer, and Rudolf A. Römer. On large scale diagonalization techniques for the Anderson model of localization. *SIAM Review*, 50:91–112, 2008.

Acknowledgements

This research has been supported by the CICYT projects TIN2005-09037-C02-02 and TIN2008-06570-C04, and the DAAD D-07-13360/*Acciones Integradas Hispano-Alemanas* programme HA2007-0071.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Scattered Multivariate Interpolation by a Class of Spline Functions

Giampietro Allasia[1]

[1] *Department of Mathematics "G. Peano", University of Turin, Italy*

emails: giampietro.allasia@unito.it

Abstract

A sequence of univariate spline functions, which converges to the normal density function, is investigated for application in multivariate scattered interpolation. This aim appears very interesting because the splines have simple analytic expressions and compact supports. A crucial point is to verify if the interpolation matrices to be considered are strictly positive definite.

Key words: spline functions, positive definite basis functions, scattered interpolation, spherically symmetric distributions, radial basis function interpolation

1 Introduction

We consider a sequence of univariate spline functions arising in probability theory, which converges to the normal density function (see e.g. [1]). Now, it is well-known that the Gaussian function is particularly suited for interpolation by radial basis functions. Hence, our aim is to investigate if the considered splines could be used for multivariate scattered interpolation, not necessarily radial. This application appears very interesting because the splines have simple analytic expressions and compact supports. A crucial point is to verify if the interpolation matrices to be considered are strictly positive definite.

We deduce the analytic expressions of the splines and some of their properties, reasoning in the context of probability theory. The strictly positive definiteness of the interpolation matrices for univariate and multivariate interpolation is discussed referring to Bochner's theorem. Formulas for univariate and multivariate interpolation to scattered data are pointed out. The extension by radial simmetry of the considered univariate formula is sketched.

2 A Class of Spline Functions of Probability Theory

Let us consider an infinite sequence $X_1, X_2, \ldots$ of random variables, which are independent and uniformly distributed on $[-a, a],\ a \in \mathbb{R}^+$. The reduced sum of the first n variables

$$S_n^* = \frac{X_1 + X_2 + \cdots + X_n}{a\sqrt{\frac{n}{3}}}$$

satisfies the central limit theorem in the local form (see e.g. [5], pp. 421–425), namely the sequence $f_n^*(x),\ n = 1, 2, \ldots$, of the density functions of the random variables S_n^* converges to the density function of the normal distribution function with expectation 0 and standard deviation 1, i.e.

$$\lim_{n\to\infty} f_n^*(x) = \frac{1}{\sqrt{2\pi}} \exp\Big(\frac{-x^2}{2}\Big), \tag{1}$$

and moreover the convergence is uniform for all $x \in \mathbb{R}$.

To get explicit expressions of the f_n^* it is convenient referring to the sum

$$S_n = a\sqrt{\frac{n}{3}} S_n^* = X_1 + X_2 + \cdots + X_n, \tag{2}$$

whose density function is $f_n(x)$. Since $S_n = X_1 + S_{n-1}$ and X_1 and S_{n-1} are independent, $f_n(x)$ is given by the convolution product

$$f_n(x) = \int_{-\infty}^{+\infty} f_1(u) f_{n-1}(x-u)du = \frac{1}{2a}\int_{-a}^{+a} f_{n-1}(x-u)du,$$

because by definition $f_1(x) = 1/(2a)$ for $-a \le x \le +a$ and $f_1(x) = 0$ elsewhere. Setting $x - u = t$, we get the recursive formula

$$f_n(x) = \int_{x-a}^{x+a} f_{n-1}(t)dt, \qquad n = 2, 3, \ldots \tag{3}$$

From (3) it follows for $-na \le x \le na$ ($f_n(x) = 0$ elsewhere)

$$\begin{aligned} f_n(x) &= \frac{1}{(2a)^n (n-1)!}\Big\{(x+na)^{n-1} - \binom{n}{1}[x+(n-2)a]^{n-1} \\ &+ \binom{n}{2}[x+(n-4)a]^{n-1} - \binom{n}{3}[x+(n-6)a]^{n-1} \\ &+ \binom{n}{4}[x+(n-8)a]^{n-1} - \binom{n}{5}[x+(n-10)a]^{n-1} + \ldots\Big\}, \end{aligned} \tag{4}$$

where the sum is extended to all arguments $x + (n-2k)a,\ k = 0, 1, 2, \ldots$, which are positive. This formula can be proved, together with the property that $f_n(x)$ is an even function, reasoning by induction (see [3], pp. 26–29, and [6], pp. 104–106).

The function $f_n(x)$ is graphically represented by arcs of parabolas of degree $n-1$; the first $n-2$ derivatives of different arcs of parabolas are equal at the ends of successive

 ISBN: 978-84-612-9727-6

intervals, i.e. $f \in C^{n-2}[-na, na]$. From a computational viewpoint it is convenient to evaluate $f_n(x)$ starting from the pieces defined on $[-na, 0]$ and then obtain the pieces on $[0, na]$ by symmetry. Moreover, each piece on $[-na, 0]$ can be obtained by the preceding one by simply adding a term, as it clearly appears from (4).

Considering the connection between the $f_n(x)$ and $f_n^*(x)$, the limit (1) becomes

$$\lim_{n\to\infty} a\sqrt{\frac{n}{3}}\, f_n\left(a\sqrt{\frac{n}{3}}\, x\right) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right).$$

The convergence is initially very fast, so that the curve of $f_4^*(x)$ can hardly be distinguished from the Gaussian curve, but then the convergence becomes too slow (see, e.g., [5], p. 421, and [1], p. 327).

The characteristic function of $X_i,\ i = 1, 2, \ldots$, is

$$\varphi_{X_i}(t) = \frac{\sin(at)}{at}.$$

Since the random variables $X_1, X_2, \ldots, X_n$ are independent, the sum S_n has the characteristic function

$$\varphi_{S_n}(t) = \int_{-\infty}^{+\infty} \exp(itx) f_n(x) dx = \left[\frac{\sin(at)}{at}\right]^n,$$

and, conversely, the density function has the expression

$$f_n(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \exp(-itx) \varphi_{S_n}(t) dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \exp(-itx) \left[\frac{\sin(at)}{at}\right]^n dt. \tag{5}$$

Let us consider the d-dimensional random vector $V_d = (Y_1, Y_2, \ldots, Y_d),\ d \geq 2$, where the variables are independent and have the same distribution as the sum S_n in (2), so that the density function of the vector V_d is $f_V(y_1, y_2, \ldots, y_d) = f_n(y_1) f_n(y_2) \cdots f_n(y_d)$. Then, the characteristic function of V_d is

$$\varphi_{V_d}(t_1, t_2, \ldots, t_d) = \prod_{j=1}^{d} \varphi_{S_n}(t_j) = \prod_{j=1}^{d} \left[\frac{\sin(at_j)}{at_j}\right]^n, \tag{6}$$

and, conversely,

$$f_{V_d}(y_1, y_2, \ldots, y_d) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \exp\left(-i \sum_{j=1}^{d} t_j y_j\right) \prod_{j=1}^{d} \left[\frac{\sin(at_j)}{at_j}\right]^n dt_1 dt_2 \cdots dt_d.$$

3 Positive Definite Basis Functions

Let us consider a set $\mathcal{S}$ of (generally scattered) distinct *data sites* labeled $x_j \in \mathbb{R}^d,\ j = 1, 2, \ldots, N,\ d \geq 1$, and the corresponding *data values* $y_j \in \mathbb{R},\ j = 1, 2, \ldots, N$. We will assume that $\mathcal{S} \subset \Omega$ for some region Ω in $\mathbb{R}^d$. The *scattered data interpolation problem* consists in finding a continuous function $F(x)$ such that $F(x_j) = y_j,\ j = 1, 2, \ldots, N$.

 ISBN: 978-84-612-9727-6

A common approach to solving the scattered data problem is to make the assumption that the function $F(x)$ is a linear combination of certain continuous *basis functions* B_k, i.e.

$$F(x) = \sum_{k=1}^{N} c_k B_k(x), \quad x \in \mathbb{R}^d. \tag{7}$$

Solving the interpolation problem under this assumption leads to a system of linear equations of the form $Ac = y$, where the entries of the *interpolation matrix* A are given by $A_{jk} = B_k(x_j),\ i,k = 1,2,\ldots,N$, and $c = [c_1, c_2, \ldots, c_N]^T, y = [y_1, y_2, \ldots, y_N]^T$. The finite-dimensional linear function space $\mathcal{B} \subset C(\Omega)$ with basis $B_1, B_2, \ldots, B_N$ is a *Haar space* on Ω if $\det A \neq 0$ for any set of distinct $x_1, x_2, \ldots, x_N$ in Ω. Existence of Haar space guarantees invertibility of the interpolation matrix A, i.e. existence and uniqueness of an interpolant of the form (7) to data specified at $x_1, x_2, \ldots, x_N$ from the space $\mathcal{B}$.

In order to characterize classes of basis functions $B_k(x)$ that generate a non-singular system matrix in (7) for any set $\mathcal{S}$ of distinct data sites, it is convenient to consider a particular class of matrices, i.e. *positive definite matrices*. A real symmetric matrix A is called *positive semi-definite* if its associated quadratic form is non negative, i.e.

$$\sum_{j=1}^{N}\sum_{k=1}^{N} c_j c_k A_{jk} \geq 0, \tag{8}$$

for $c = [c_1, c_2, \ldots, c_N]^T \in \mathbb{R}^N$. If the quadratic form (8) is zero only for $c \equiv 0$, then A is called *positive definite*.

Then, in order to obtain data dependent approximation spaces, it is convenient to consider basis functions, which are the shifts of a certain function to the data sites, i.e. $B_k(x) = \Phi(x - x_k)$, so that the interpolation matrix is positive definite. A complex-valued continuous function $\Phi : \mathbb{R}^d \to \mathbb{C}$ is called *positive definite on* $\mathbb{R}^d$ if

$$\sum_{j=1}^{N}\sum_{k=1}^{N} c_j \overline{c_k} \Phi(x_j - x_k) \geq 0, \tag{9}$$

for any N pairwise different points $x_1, x_2, \ldots, x_N$, and $c = [c_1, c_2, \ldots, c_N]^T \in \mathbb{C}^N$. The function Φ is called *strictly positive definite on* $\mathbb{R}^d$ if the quadratic form (9) is zero only for $c \equiv 0$.

The preceding discussion suggests that we should use strictly positive definite functions as basis functions in (7), i.e.

$$F(x) = \sum_{k=1}^{N} c_k \Phi(x - x_k), \quad x \in \mathbb{R}^d.$$

Some of the most important properties of (strictly) positive definite functions are listed in ([2], pp. 29–30).

 ISBN: 978-84-612-9727-6

A celebrate result on positive definite functions is their characterization in terms of Fourier transforms established by S. Bochner, that we present in the following form (see [2], pp. 31–32):

Theorem). *A (complex-valued) function $\Phi \in C(\mathbb{R}^d)$ is positive definite on $\mathbb{R}^d$ if and only if it is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^d$, i.e.*

$$\Phi(x) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbf{R}^d} \exp(-i\langle x, y\rangle) d\mu(y), \quad x \in \mathbb{R}^d,$$

where $\langle x, y\rangle$ is the usual inner product in $\mathbb{R}^d$.

Now we are able to establish some definiteness properties of the functions introduced in Section 2:

(a) From (5) we have for any N pairwise different points $x_1, x_2, \ldots, x_N \in \mathbb{R}$

$$\sum_{k=1}^{N}\sum_{j=1}^{N} c_k\overline{c_j} f_n(x_k - x_j) = \frac{1}{2\pi}\sum_{k=1}^{N}\sum_{j=1}^{N} c_k\overline{c_j} \int_{-\infty}^{+\infty} \exp\,[\,-it(x_k - x_j)]\left[\frac{\sin(at)}{at}\right]^n dt$$
$$= \frac{1}{2\pi}\int_{-\infty}^{+\infty} \left|\sum_{k=1}^{N} c_k \exp(-itx_k)\right|^2 \left[\frac{\sin(at)}{at}\right]^n dt.$$

Since for n even the function

$$\left[\frac{\sin(at)}{at}\right]^n$$

is non-negative and not identically equal to zero, it follows that $f_n(x)$ is strictly positive definite.

(b) Similarly, we have from (6) for any N pairwise different points $y_1, y_2, \ldots, y_N \in \mathbb{R}^d$

$$\sum_{k=1}^{N}\sum_{j=1}^{N} c_k\overline{c_j} f_{V_d}(y_k - y_j) = \frac{1}{(2\pi)^d}\sum_{k=1}^{N}\sum_{j=1}^{N} c_k\overline{c_j} \int_{\mathbf{R}^d} \exp\,[\,-i\langle t, y_k - y_j, \rangle]\varphi_{V_d}(t)dt$$
$$= \frac{1}{(2\pi)^d}\int_{\mathbf{R}^d} \left|\sum_{k=1}^{N} c_k \exp[-i\langle t, y_k\rangle]\right|^2 \prod_{h=1}^{d}\left[\frac{\sin(at_h)}{at_h}\right]^n dt.$$

The last integral is greater than zero for n even. So $f_{V_d}(y)$ is strictly positive definite.

4 Univariate and Multivariate Scattered Interpolation

Univariate interpolation on scattered data can be obtained by the operator

$$F(x) = \sum_{k=1}^{N} c_k f_n(x - x_k), \qquad x, x_k \in \mathbb{R},$$

where $f_n(x)$ is given in (4) and n is even, because in this case $f_n(x)$ is strictly positive defined. The interpolant $F(x)$ is a linear combination of the shifted functions $f_n(x-x_k)$,

 ISBN: 978-84-612-9727-6

which are spline functions with compact supports $[-na - x_k, na - x_k]$ and continuous up to $n-2$ derivatives. Increasing the degree n we do not match the phenomenon of the oscillatory nature typical of high degree polynomials, but using high degree splines does not appear convenient for computation. It it plain that there exist in the univariate setting other interpolation methods simpler and more efficient than the considered one.

The coordinates x and y of a randomly selected point in the square $\{-a \le x \le a, -a \le y \le a\}$ can be considered as independent random variables X and Y, uniformly distributed on the square. The density function of the bivariate random variable (X, Y) is

$$f_1(x, y) = f_1(x) f_1(y) = \begin{cases} \dfrac{1}{4a^2}, & \text{if } -a \le x \le a, -a \le y \le a, \\ 0, & \text{elsewhere,} \end{cases}$$

where $f_1(\cdot)$ is the rectangular density function.
Similarly, we can suppose that the coordinates x and y of a randomly selected point in the square $\{-na \le x \le na, -na \le y \le na\}$ are independent random variable X and Y distributed like S_n with density $f_n(\cdot)$ in (4). The density function of the bivariate random variable (X, Y) is for any n

$$f_n(x, y) = f_n(x) f_n(y).$$

Bivariate interpolation on scattered data can be obtained by the operator

$$F(x, y) = \sum_{k=1}^{N} c_k f_n(x - x_k) f_n(y - y_k), \qquad x, x_k, y, y_k \in \mathbb{R},$$

where n is even, because in this case $f_n(x) f_n(y)$ is strictly positive defined. In general, for dimension $d \ge 2$, we have

$$F(y_1, y_2, \ldots, y_d) = \sum_{k=1}^{N} c_k f_n(y_1 - y_{1k}) f_n(y_2 - y_{2k}) \cdots f_n(y_d - y_{dk}),$$

where n is even, $(y_1, y_2, \ldots, y_d) \in \mathbb{R}^d$, and $(y_{1k}, y_{2k}, \ldots, y_{dk}) \in \mathbb{R}^d$, $k = 1, 2, \ldots, N$, are the data sites. The interpolant $F(y_1, y_2, \ldots, y_d)$ is rather atypical in the sense that it is neither a product formula, like e.g. the Lagrange product formula, nor a radial one. Nevertheless, $F(y_1, y_2, \ldots, y_d)$ deserves to be considered for scattered interpolation, because several tests show that its performance is quite good.

5 Spherically Symmetric Distributions and Radial Basis Interpolation

Changing x into $\|x\| = \sqrt{x^2 + y^2}$ in the function $z = f_n(x)$ in (4), we rotate the curve around its symmetry axis. In this way we get the spherically symmetric function

$$\tilde{g}_n(x, y) = \frac{1}{(2a)^n (n-1)!} \sum_{k=0}^{n} (-1)^k \binom{n}{k} \left[\sqrt{x^2 + y^2} + (n - 2k)a \right]_+^{n-1},$$

 ISBN: 978-84-612-9727-6

which depends on $r = \sqrt{x^2 + y^2}$ only. Dividing the function $\tilde{g}_n(x, y)$ by the volume γ_n of its rotation solid we obtain a density function $g_n(x, y)$, that is, $g_n(x, y) = \tilde{g}_n(x, y)/\gamma_n$. In a similar way we can consider spherically symmetric functions of $d > 2$ variables, that is $g(y_1, y_2, \ldots, y_d)$ which depend on $(y_1^2 + y_2^2 + \cdots + y_d^2)^{1/2} = r$. The main properties of spherically symmetric distributions are discussed in [4], in particular relations between characteristic functions and density functions. However, the property of strictly positive definiteness of the density function deserves still a deep investigation, since Fourier tranforms are not easy to compute. The first numerical tests are promising, but the topic must be deepened also in comparison with other compactly supported radial basis functions(see e.g. [7], pp. 119–132).

References

[1] G. Allasia, *Approssimazione della funzione di distribuzione normale mediante funzioni spline*, Statistica (anno XLI) **2** (1981), 325–332.

[2] G. E. Fasshauer, *Meshfree approximation methods with Matlab*, World Scientific Publishing, Singapore, 2007.

[3] W. Feller, *An introduction to the probability theory and its applications*, vol. II, 2nd ed., Wiley, New York, 1971.

[4] A. M. Mathai, *An introduction to geometrical probability*, Gordon and Breach, Amsterdam, 1999.

[5] A. Rényi, *Calcul des probabilités*, Dunod, Paris, 1966.

[6] B. L. Waerden (van der), *Mathematical statistics*, Springer, Berlin, 1969.

[7] H. Wendland, *Scattered data approximation*, Cambridge University Press, Cambridge, 2005.

 ISBN: 978-84-612-9727-6

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Two interpolation formulas on irregularly distributed data

Giampietro Allasia[1] **and Cesare Bracco**[1]

[1] *Department of Mathematics "G. Peano", University of Turin, Italy*

emails: giampietro.allasia@unito.it, cesare.bracco@unito.it

Abstract

Two linear interpolation operators on irregularly distributed data are examined and compared: the first is polynomial, while the second is a cardinal basis interpolant obtained by modifying the first. The two formulas can be generalized to an inner product space setting.

Key words: multivariate interpolation, cardinal basis functions, scattered data

1 Introduction

Let $z_1, ..., z_N$ be N distinct elements of Ω, a subset of $\mathbb{R}^m$, with associated elements $c_1, ..., c_N$ belonging to $\mathbb{R}^n$. We consider the *Lagrange interpolation problem from* $\mathbb{R}^m$ *into* $\mathbb{R}^n$ consisting of finding a function $F : \mathbb{R}^m \to \mathbb{R}^n$ such that

$$F(z_i) = c_i, \qquad i = 1, 2, ..., N. \tag{1}$$

If there isn't any additional requirement about the solution, such as its uniqueness, then a suitable F is

$$F(z) = \sum_{i=1}^{N} l_i(z)\ c_i, \qquad z \in \mathbb{R}^m, \tag{2}$$

where

$$l_i(z) = \prod_{k=1, k\neq i}^{N} \frac{(z - z_k, z_i - z_k)}{(z_i - z_k, z_i - z_k)}, \qquad l_i(z_j) = \delta_{ij}, \qquad i, j = 1, ..., N, \tag{3}$$

and (w, z), for any $w, z \in \mathbb{R}^m$, is the usual inner product in $\mathbb{R}^m$. Note that for a given $z \in \Omega$, each $l_i(z)$ takes a scalar value in $\mathbb{R}$, whereas $c_i \in \mathbb{R}^n$, $i = 1, 2, ..., N$, and so $F(z) \in \mathbb{R}^n$. In the case $m = 2$ and $n = 1$, this interpolant was studied by Berezin and Zhidkov (see [2], pp. 170–171). Since we did not succeeded in finding previous presentations, we will refer to it as the *Berezin-Zhidkov interpolant*.

Starting from (3) we can construct the cardinal basis functions

$$g_i(z) = \prod_{k=1,k\neq i}^{N} \frac{|(z-z_k, z_i-z_k)|^p}{\sum_{h=1}^{N}\prod_{k=1,k\neq h}^{N}|(z-z_k, z_h-z_k)|^p}, \qquad p>0,\ i=1,...,N,$$

which satisfy

$$g_i(z_j)=\delta_{ij}, \qquad g_i(z)\geq 0, \qquad \sum_{i=1}^{N} g_i(z)=1, \qquad i,j=1,...,N,\ z\in\Omega.$$

Then we can define the interpolant

$$\Phi(z)=\sum_{i=1}^{N} g_i(z)\ c_i, \qquad z\in\mathbb{R}^m, \tag{5}$$

which is a solution to (1) as well.

2 Comparisons

The Berezin-Zhidkov interpolant is a polynomial belonging to Π_{N-1}^m, the space of polynomials of total degree $N-1$ in m variables, whose dimension does not match the number of interpolation conditions for $m>1$, since

$$\dim(\Pi_{N-1}^m)=\binom{N-1+m}{m}>N.$$

This means that F is not the unique solution of the interpolation problem in Π_{N-1}^m, and that it is not the polynomial interpolant of least degree, whose determination is a complicated problem (see, e.g., [3]).
If $m=n=1$ or if the nodes $z_1,...,z_N$ are on the same straight line, $F(z)$ reduces to the Lagrange interpolation polynomial. However, it is not possible to generalize all the well-known properties of the univariate Lagrange interpolation to the Berezin-Zhidkov interpolant, since the fundamental relation

$$\sum_{i=1}^{N} l_i(z)=1$$

does not hold in general.

Given a function $f:\mathbb{R}^m\to\mathbb{R}^n$ to be interpolated by (2), the error can be bounded as follows

$$\|f(z)-F(z)\|\leq M(1+\Lambda_N), \qquad z\in\Omega,$$

where

$$\Lambda_N=\max_{z\in\Omega}\lambda_N(z), \quad \lambda_N(z)=\sum_{i=1}^{N}|l_i(z)| \quad \text{and} \quad M=\max_{z\in\Omega}\|f(z)\|.$$

 ISBN: 978-84-612-9727-6

Since Λ_N can take, likewise the usual Lebesgue constant, larger and larger values as N increases, the approximation error is often very large, as showed by numerical tests.

On the contrary, $\Phi(z)$ enjoys the good approximation properties of the cardinal basis interpolants (see, e.g., [1]). In particular, we have

$$\|f(z) - F(z)\| \leq \max_i \|f(z) - c_i\| \leq \max_i \max_{z\in\Omega} \|f(z) - c_i\|, \qquad z \in \Omega, \tag{6}$$

if f is continuous and Ω is a compact set. This is sufficient to assure the uniform boundedness of the error and so a better approximation performance if compared to the Berezin-Zhidkov interpolant. However, to get an accurate approximation of the underlying function f, it is necessary a further modification of Φ. In fact, as N increases, the cardinal basis functions $g_i(z)$ corresponding to the nodes close to the boundaries of the convex hull of the nodes assume much larger values than the other ones in a large part of Ω. In order to avoid the dominance of a few cardinal functions over the other ones, and so to get good approximation results, a solution is to use a localizing scheme, that is, when evaluating Φ at $z \in \Omega$, to consider only the nodes close to z. More precisely, given $M \leq N$, generally small, we consider

$$\hat{\Phi}(z) = \sum_{i=1}^{M} c_{h_i} \frac{\prod_{j=1, j\neq i}^{M} |(z - z_{h_j}, z_{h_i} - z_{h_j})|^p}{\sum_{k=1}^{M} \prod_{j=1, j\neq k}^{M} |(z - z_{h_j}, z_{h_k} - z_{h_j})|^p},$$

where z_{h_j}, $j = 1, 2, ..., M$, are, for each $z \in \Omega$, the M nearest nodes to z.
Finally, we can note that, if p is an even integer, Φ and the localized version $\hat{\Phi}$ are rational interpolants.

3 Extensions to inner product spaces

It is very interesting to note that both of the interpolants can be generalized to an inner product space setting. In fact, considering an inner product space X on a field K with inner product $(\cdot, \cdot)$ in place of $\mathbb{R}^m$ and a Banach space Y on the same field K in place of $\mathbb{R}^n$, the formulas (2) and (5) define two interpolants $F_{ext}(z)$ and $\Phi_{ext}(z)$ satisfying the interpolation property. Actually, F_{ext} coincides with the interpolant in Hilbert spaces presented in [4]. Let us consider the following

Definition 1. *Let X be a Hilbert space on the field K. A polynomial of degree N from X into K is given by*

$$P(x) = L_0 + L_1 x + ... + L_N x^N$$

where L_k is a k-linear operator from X into K, for $k = 1, ..., N$, and $L_k x^k$ stands for $L_k(x, x, ..., x)$.

 ISBN: 978-84-612-9727-6

Then, for $i = 1, ..., N$ the operator from X^{N-1} into K

$$M_i(x_1, ..., x_{N-1}) \equiv \prod_{k=1}^{i-1} \frac{(x_k - z_k, z_i - z_k)}{(z_i - z_k, z_i - z_k)} \cdot \prod_{k=i+1}^{N} \frac{(x_{k-1} - z_k, z_i - z_k)}{(z_i - z_k, z_i - z_k)},$$

where $z_1, ..., z_N$ are the interpolation points, is an $(N-1)$-linear operator on X. Being

$$l_i(z) = M_i(z, z, ..., z), \qquad i = 1, ..., N,$$

the interpolant F_{ext} is a polynomial in the sense of Definition 1. Similarly, if p is an even integer, Φ_{ext} is a ratio of polynomials. We observe that both F_{ext} and Φ_{ext}, as interpolation operators, are linear.
As shown by numerical tests, the approximation properties of F_{ext} are unsatisfactory, since the error behaviour is the same described in the Euclidean case, i.e. from $\mathbb{R}^m$ into $\mathbb{R}^n$. For instance, let $X = \mathcal{C}[-\pi, \pi]$ with inner product

$$(f, g) = \int_{-\pi}^{\pi} f(t)g(t)dt, \qquad f, g \in \mathcal{C}[-\pi, \pi],$$

and $Y = \mathbb{R}$. Let

$$\Omega = \{\alpha \cos(t) + \beta \cos(2t), \alpha, \beta \in [0, 2]\},$$

and let the nodes be $\alpha_i \cos(t) + \beta_i \cos(2t)$, with (α_i, β_i) taken on a regular $r \times r$ grid over the square $[1, 1 + 1/20] \times [1, 1 + 1/20]$ $(r = 6, 7, 8, 9, 10, 11)$. Let the interpolated function be

$$f(z) = \int_{-\pi}^{\pi} tz(t)dt.$$

The interpolant has been evaluated at 169 points belonging to Ω, obtained taking again the coefficients (α, β) on a regular 13×13 grid over the square $[1, 1+1/20] \times [1, 1+1/20]$. The results are reported in Table 1.

nodes	RMSE	nodes	RMSE
$5\times$ 5	$4.056 \cdot 10$	8×8	$1.809 \cdot 10^3$
$6\times$ 6	$1.458 \cdot 10^2$	9×9	$4.018 \cdot 10^5$
$7\times$ 7	$3.456 \cdot 10^2$	10×10	$3.505 \cdot 10^6$

Table1. *Evaluation of test operator by F_{ext} in $\mathcal{C}[-\pi, \pi]$*

On the contrary Φ_{ext} gives good approximation results, assured by the extension of the estimate (6) to the inner space setting. The dependence of the error from the point distribution can be better showed by the inequality

$$\|f(z) - \Phi_{ext}(z)\| \leq \omega[f](\max_i \|z - z_i\|),$$

where

$$\omega[f](\delta) = \sup_{z_1, z_2 \in \Omega} \{\|f(z_1) - f(z_2)\|, \|z_1 - z_2\| \leq \delta\}$$

is the modulus of continuity of f. In the case $X = \mathbb{R}^m$ and $Y = \mathbb{R}^n$, it is easy to deduce convergence properties from this inequality, taking the nodes on finer and finer grids over the domain Ω. On the contrary, when X is a general inner product space, it is less clear how to consider "grids"over Ω, suitable to get convergence.

The numerical experiments involving the interpolant $\hat{\Phi}_{ext}$ show very promising results, which suggest a possible application to the theory of non-linear system modelling, and in particular system identification (see, e.g., [5]).

References

[1] G. Allasia, *A class of interpolating positive linear operators: theoretical and computational aspects*, in S.P. Singh (ed.) *Approximation Theory, Wavelets and Applications*, Kluwer, Dordrecht, 1995, 1–36.

[2] I.,S. Berezin and N. P. Zhidkov, *Computing methods*, two vols., Pergamon Press, Oxford, 1965; transl. of *Metody vychislenii*, Fizmatgiz, Moscow, 1959.

[3] M. Gasca and T. Sauer, *Polynomial interpolation in several variables*, Adv. Comp. Math., **12** (2000), 377–410.

[4] P. M. Prenter, *Lagrange and Hermite interpolation in Banach spaces*, Journal Approximation Theory, **4** (1971), 419–432.

[5] A. Torokhti and P. Howlett, *Computational methods for modelling of non-linear systems*, Elsevier, Amsterdam, 2007.

 ISBN: 978-84-612-9727-6

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Neville Elimination and Multi-core Systems: OpenMP vs MPI

P. Alonso[1], R. Cortina[2], F.J. Martínez-Zaldívar[3] and J. Ranilla[2]

[1] *Departamento de Matemáticas, Universidad de Oviedo, Spain*

[2] *Departamento de Informática, Universidad de Oviedo, Spain*

[3] *Departamento de Comunicaciones, Universidad Politécnica de Valencia, Spain*

emails: palonso@uniovi.es, raquel@uniovi.es, fjmartin@dcom.upv.es, ranilla@uniovi.es

Abstract

This paper describes several parallel algorithmic variations of the Neville elimination. This elimination solves a system of linear equations making zeros in a matrix column by adding to each row an adequate multiple of the preceding one. The parallel algorithms are run and compared on different multi-core platforms using two parallel programming techniques: MPI and OpenMP.

Key words: Neville, Multi-core, OpenMP, MPI

1 Introduction

A multi-core processor combines two or more independent cores or CPUs in a single die or integrated circuit, sharing the external bus interface and some cache memory, [11]. This combination usually represents the most extreme form of tightly-coupled multiprocessing. Many actual multi-core processors has two, four, eight or even more cores. When the number of cores is higher (several hundreds) the architecture is named many-core processor (e.g. GPUs). These systems are specially suited in applications where there are at least as many running processes or independent process threads as cores. The shared memory programming paradigm can be considered the natural way of programming these systems. Other paradigms, like message passing, can be easily implemented using the shared memory communication mechanisms found in these kind of multiprocessors.

Neville elimination is an alternative procedure to that of Gauss to transform a square matrix A into an upper triangular one. Strictly speaking, Neville elimination makes zeros on an A column adding to each row a multiple of the previous one. It is a better alternative to Gaussian elimination when working with totally positive matrices, sign-regular matrices or other related types of matrices (see [7] and [8]). According to [6] and [9], Neville elimination is considered

to be an interesting alternative to Gauss elimination for certain types of research. Furthermore, there are other works (see [1], [2] and [3]) that show the advantages of the foresaid procedure in the field of High Performance Computing. The mentioned works have been performed considering distributed memory environments.

This paper compares the performance of several parallel implementations of the Neville algorithm devised to solve a system of linear equations on multi-core architectures. We have used implementations of the MPI and OpenMP standards in order to program parallel Neville algorithms with distributed and shared memory schemes.

This paper is organized as follows: first we will show the sequential Neville algorithm; next we will describe two variations of the shared memory parallel algorithm, later some tests will show the performance of these algorithmic variations in several multi-core platforms and finally the conclusions of this work will be stated.

2 Neville algorithm

A system of equations $Ax = b$ is usually solved in two stages ($A = LU$). First, through a series of algebraic manipulations the original system of equations is reduced to an upper-triangular system $Ux = y$. In the second stage, the upper-triangular system is solved by a procedure known as back-substitution.

If A is a square matrix of order n, the Neville elimination procedure consists of $n-1$ successive major steps (see [8] for a detailed and formal introduction), resulting in a sequence of matrices as follows $A = A^{(1)} \longrightarrow A^{(2)} \longrightarrow \cdots \longrightarrow A^{(n)} = U$, where U is an upper-triangular matrix. If A is non-singular, the matrix $A^{(k)} = (a_{i,j}^{(k)})_{1 \leq i,j \leq n}$ has zeros below its main diagonal in the $k-1$ first columns.

Let us consider the case in which Neville elimination can be performed without changing rows. The work presented in [7] shows that row changes are not necessary when the Neville elimination process is applied to a totally positive matrix (a matrix whose minors are non-negative). In order to get $A^{(k+1)}$ from $A^{(k)}$, zeros are obtained in the k^{th} column below the main diagonal, subtracting a multiple of the i^{th} row from the $(i+1)^{\text{th}}$ for $i = n-1, n-2, ..., k$, according to the formula:

$$a_{i,j}^{(k+1)} = \begin{cases} a_{i,j}^{(k)} & \text{if } 1 \leq i \leq k, \\ a_{i,j}^{(k)} - \dfrac{a_{i,k}^{(k)}}{a_{i-1,k}^{(k)}} a_{i-1,j}^{(k)} & \text{if } k+1 \leq i \leq n \text{ and } a_{i-1,k}^{(k)} \neq 0, \\ a_{i,j}^{(k)} & \text{if } k+1 \leq i \leq n \text{ and } a_{i-1,k}^{(k)} = 0. \end{cases} \quad (1)$$

Algorithm 1 shows the Neville iterations to solve a non-singular system of linear equations. $A(i,j)$ denotes the (i,j)-element of the A matrix. When all the iterations finish, A becames upper triangular and x can be got via back substitution from A and b. The information of the method steps can be stored in the matrix.

The inner loop represents a typical `axpy` operation ($y = ax + y$), in which a portion of the i^{th} row is updated with a linear combination of it and the same portion of the $(i-1)^{\text{th}}$ row; and a is the ratio $-A(i,j)/A(i-1,j)$. The cost of this algorithm is $2/3n^3$ flops.

 ISBN: 978-84-612-9727-6

Algorithm 1 Sequential Neville

Require: $\mathbf{A} \in \mathbf{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$
Ensure: processed $\mathbf{A}$ and $\mathbf{b}$ with $\mathbf{A}$ upper triangular and $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

```
for j = 1 : n − 1 do
  for i = n : −1 : j + 1 do
    α = A(i, j)/A(i − 1, j)
    for r = j + 1 : n do
      A(i, r) = A(i, r) − αA(i − 1, r)
    end for
    b(i) = b(i) − αb(i − 1)
    A(i, j) = 0
  end for
end for
```

Figure 1 shows the index runnings and the rows and elements involved in one inner iteration.

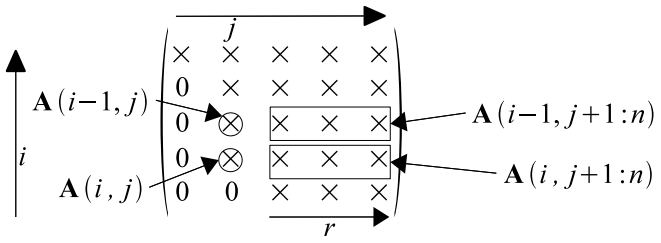


Figure 1: Sequential Neville algorithm

3 OpenMP parallel algorithms

In previous works (see [1], [2] and [3]) we have proposed an organization of the Neville elimination algorithm for computers with the message passing model and we have carried out a general analysis based on upper bounds for the three metrics: execution time, efficiency/speedup and scalability.

The realized work has allowed us to confirm the good performance of Neville's method using parallel computation on single-core systems. For example, in [5] we have verified that using a column-wise cyclic-striped distribution of the data (size of block 1), the reached efficiency is near to one.

Next we describe several parallel algorithmic variations for multi-core platforms. The parallel algorithms are run and compared using two parallel programming techniques: MPI and OpenMP.

3.1 Blocks of contiguous rows

The dependency graph of the parallel algorithm would show that the result in the i^{th} row depends only on it and on the $(i-1)^{\text{th}}$ row for the j^{th} iteration. Hence, we can group the rows in blocks of consecutive rows and apply the `axpy` operation sequentially in the block. The (not yet updated) row of highest index in a block will be needed (without updating) by the lowest index row of the next block, hence it should be saved in a buffer before its updating. It will be necessary to synchronize the threads in order to avoid race conditions in the last row of the blocks. The block workload may be assigned to a thread.

The configuration of every block of contiguous rows may be static or dynamic. If we use the static configuration, there will be load unbalancing because the workload associated to the first rows will finish earlier (as the j index increases, the number of rows involved in the computation decreases, as can be observed in the Neville **Algorithm 1**), hence the associated threads will be idle. One way to avoid this load unbalancing is to reconfigure the blocks of contiguous rows every j-iteration, dividing the number of the active rows among the threads.

Hence, this first version can be derived directly from the sequential version (with the proper additional controls over the buffer of the non yet updated row, and the row indexes associated to the thread).

Figure 2 shows the division of the workload in form of blocks of consecutive rows, the dependency of the first row of a block with the last row of the upper block, and the way the matrix lower triangle is symbolically zeroed.

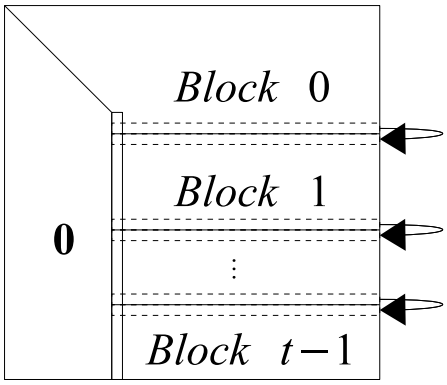


Figure 2: OpenMP block of rows parallel Neville algorithm

3.2 Blocks of contiguous columns

The updating of the rows may be done on a column basis beginning at the bottom and finishing at the top of every column in an independent way. It can be easily accomplished using the **Algorithm 2**.

The columns ranged by every private copy of the j index (columns processed by a thread) depend on the optional scheduling parameter of the `#pragma` directive, [4]. Figure 3 shows the independence among the columns ranged by the j index.

Algorithm 2 Block of columns OpenMP version.

Require: $\mathbf{A} \in \mathbf{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$
Ensure: processed $\mathbf{A}$ and $\mathbf{b}$ with $\mathbf{A}$ upper triangular and $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

```
for c = 0 : N − 2 do
  #pragma omp parallel for private(i,j,alfa) schedule(...)
  for j = c + 1 : N − 1 do
    for i = N − 1 : −1 : c + 1 do
      α = A(i, c)/A(i − 1, c)
      A(i, j) = A(i, j) − αA(i − 1, j)
      if (j == N − 1) then
        b(i) = b(i) − αb(i − 1)
      end if
    end for
  end for
end for
```

4 Experimental results

4.1 Test platforms

We have tested the MPI and the OpenMP versions on the next multi-core platforms:

- HP ProLiant BL465 G5 dual processors AMD Opteron(TM) 2356 Quad-CoreProcessor, 2.3 GHz, 512 kB cache, MPI: HP-MPI V2.2 (it complies fully with the MPI-1.2 standard and provides full MPI-2 functionality).
- DELL PowerEdge 2950 III dual processors Intel(R) Xeon(R) CPU E5420, 2.50 GHz, 6 MB cache, MPI: MPICH-1.2.7.p1.

4.2 Efficiencies

To compare the behavior of OpenMP and MPI we use the Efficiency metric obtained from the total execution wall-time of the sequential and parallel algorithms presented in this work. Afterwards, the implementation made in [5] for columm-wise cyclic striped distribution using MPI was tested. The obtained efficiencies are shown from Figure 4 to Figure 9.

We can observe a superlinearity behavior in all the variants due to cache effect. In general, the OpenMP implementations do not get a high efficiency for high size problems because of the non exploitation of locality. The row blocks based parallel algoritm needs to synchronize the threads to avoid the race condition over the shared buffer. Finally, the power of two problem sizes improve the efficiency (in a special way in the HP Proliant —this machine is a pure CC-NUMA—)

The efficiencies behaviour of the MPI versions are better and more stable (in fact, the HP-MPI library is highly optimized for HP servers). Here, the increment of the number of cores does not penalize so much the obtained efficiency.

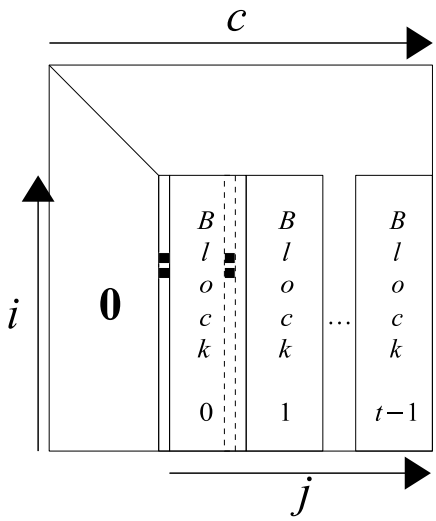


Figure 3: OpenMP block of columns parallel Neville algorithm

5 Conclusions

In general, the parallel efficiency is poorer in the OpenMP versions than in the MPI versions. The main reason is the difficulty to exploit the cache data locality in this type of algorithms. Also, the competition for the common resources increases as the number of cores increases, hence appearing bottlenecks that avoid a good efficiency. In OpenMP, only inside some problem size interval it is possible to obtain a good (superlinear) efficiency due to the cache effect over the data; eventually, it can be observed a better behavior for power of two sizes due to similiar reasons. Obviously, there is an important performance difference between the HP-MPI implementation and the MPICH implementation of the MPI standard in favor of the former.

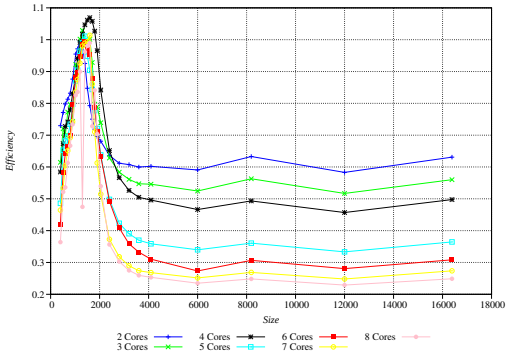


Figure 4: Efficiency of row-blocks OpenMP parallel algorithm in Dell PE

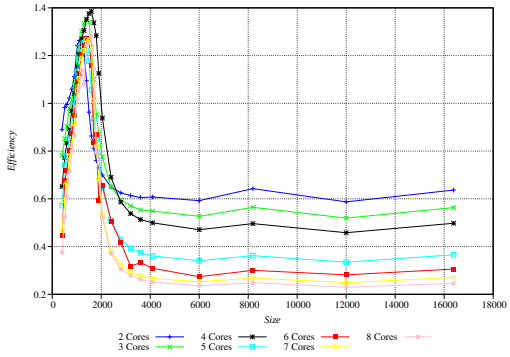


Figure 5: Efficiency of column-blocks OpenMP parallel algorithm in Dell PE

 ISBN: 978-84-612-9727-6

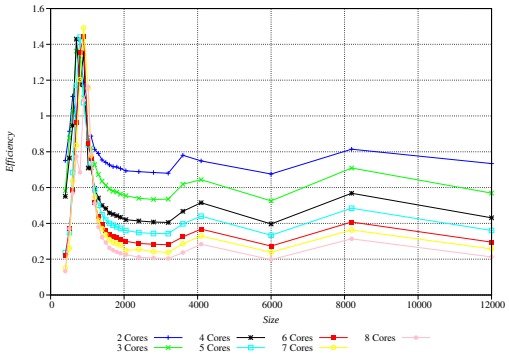


Figure 6: Efficiency of row-blocks OpenMP parallel algorithm in HP Proliat

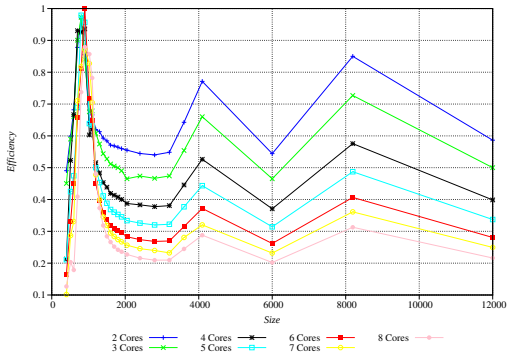


Figure 7: Efficiency of column-blocks OpenMP parallel algorithm in HP Proliat

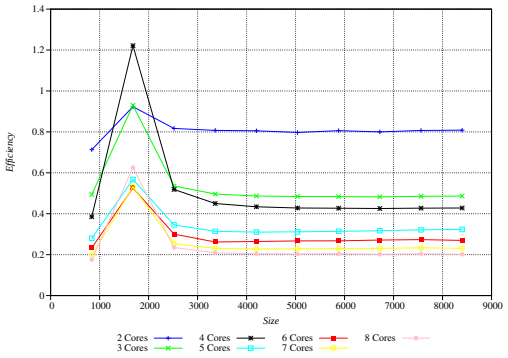


Figure 8: Efficiency of column-blocks MPI parallel algorithm in Dell PE

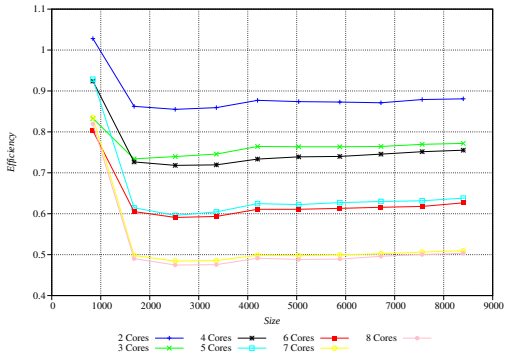


Figure 9: Efficiency of column-blocks HP-MPI parallel algorithm in HP Proliat

 ISBN: 978-84-612-9727-6

Acknowledgements

This work has been supported by MEC-TIN2007-61273 and the project 20080811 of the Generalitat Valenciana.

References

[1] P. ALONSO, R. CORTINA, I. DÍAZ AND J. RANILLA, *Neville Elimination: A study of the efficiency using Checkerboard Partitioning*, *Linear Algebra and its Applications* **393** (2004), 3–14.

[2] P. ALONSO, R. CORTINA, I. DÍAZ AND J. RANILLA, *Scalability of Neville elimination using checkerboard partitioning*, *International Journal of Computer Mathematics* **85(3-4)** (2008), 309–317.

[3] P. ALONSO, R. CORTINA, I. DÍAZ AND J. RANILLA, *Blocking Neville elimination algorithm for exploiting cache memories*, *Applied Mathematics and Computation* **209(1)** (2009), 2–9.

[4] R. CHANDRA ET AL., *Parallel Programming in OpenMP*, Morgan Kaufmann Publishers (2001).

[5] R. CORTINA, *El método de Neville: un enfoque basado en Computación de Altas Prestaciones*, Ph. D. Thesis, Univ. of Oviedo, Spain, 2008.

[6] J. DEMMEL AND P. KOEV, *The Accurate and Efficient Solution of a Totally Positive Generalized Vandermonde Linear System*, SIAM J. Matrix Anal. Appl. **27** (2005), 142–152.

[7] M. GASCA AND J. M. PEÑA, *Total positivity and Neville elimination*, Linear Algebra Appl. **165** (1992), 25–44.

[8] M. GASCA AND J. M. PEÑA, *A matricial description of Neville elimination with applications to total positivity*, *Linear Algebra Appl.* **202** (1994), 33–45.

[9] L. GEMIGNANI, *Neville Elimination for Rank-Structured Matrices*, *Linear Algebra Appl.* **428(4)**(2008), 978–991.

[10] M.D. HILL AND M.R. MARTY, *Amdahl's Law in the Multicore Era* , *IEEE Computer*, July (2008).

[11] INTEL, *Intel Multi-Core Processor Architecture Developer Backgrounder*, White paper, (2005).

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009 30 June, 1–3 July 2009.

Growth Factors of Pivoting Strategies Associated to Neville Elimination

Pedro Alonso[1], Jorge Delgado[2], Rafael Gallego[1] and Juan Manuel Peña[2]

[1] *Departamento de Matemáticas, Universidad de Oviedo, Spain*

[2] *Departamento de Matemática Aplicada, Universidad de Zaragoza, Spain*

emails: palonso@uniovi.es, jorgedel@unizar.es, rgallego@uniovi.es, jmpena@unizar.es

Abstract

Neville elimination is a direct method for the resolution of linear systems of equations alternative to Gaussian elimination, with advantages for some classes of matrices and in the context of pivoting strategies for parallel implementations. We analyze several pivoting strategies for Neville elimination, including pairwise pivoting. Several definitions of growth factors for Neville elimination are compared. Bounds for the growth factors of these pivoting strategies are provided. Moreover, an approximation of the average normalized growth factor associated with these pivoting strategies is computed and analyzed for random matrices. Comparisons with other pivoting strategies for other elimination procedures are also presented.

Key words: Neville elimination, pivoting, growth factor, total positivity
MSC 2000: 65F05, 65G05

1 Introduction

The usual direct method to solve a linear system of equations $Ax = b$ is Gaussian elimination. Neville elimination is an alternative procedure to Gaussian elimination to transform a square matrix A into an upper triangular matrix U, and it has advantages for some classes of matrices and in the context of pivoting strategies for parallel implementations. Neville elimination makes zeros in a column of the matrix A by adding to each row a multiple of the previous one. Here we only give a brief description of this procedure (for a detailed and formal introduction we refer to [9]). If $A \in \mathbb{R}^{n\times n}$, the Neville elimination procedure consists of at most $n-1$ steps:

$$A = A^{(1)} \to \widetilde{A}^{(1)} \to A^{(2)} \to \widetilde{A}^{(2)} \to \cdots \to A^{(n)} = \widetilde{A}^{(n)} = U.$$

 ISBN: 978-84-612-9727-6

Partial pivoting for Neville elimination was already introduced in [10]. Pivoting strategies for parallel implementations closely related to Neville elimination have been used frequently and are usually called pairwise or neighbor pivoting (see [17], [18] and [19]).

On the one hand, $\widetilde{A}^{(t)}$ can be obtained from the matrix $A^{(t)}$ through an adequate pivoting strategy, so that the rows with a zero entry in column t are the final rows and

$$\widetilde{a}_{it}^{(t)} = 0, \quad i \geq t \quad \Rightarrow \quad \widetilde{a}_{ht}^{(t)} = 0, \quad \forall\, h \geq i.$$

On the other hand, $A^{(t+1)}$ is obtained from $\widetilde{A}^{(t)}$ making zeros in the column t below the main diagonal by adding an adequate multiple of the ith row to the $(i+1)$th for $i = n-1, n-2, \ldots, t$. If A is nonsingular, the matrix $A^{(t)}$ has zeros below its main diagonal in the first $t-1$ columns. It has been proved that this process is very useful with totally positive matrices, sign-regular matrices and other related types of matrices (see [8] and [9]).

A real matrix is called totally positive (TP) if all its minors are nonnegative. TP matrices arise in a natural way in many areas of Mathematics, Statistics, Economics, etc. (see [5]). In particular, their application to Approximation Theory and Computer Aided Geometric Design (CAGD) is of great interest. For example, coefficient matrices of interpolation or least square problems with a lot of representations in CAGD (the Bernstein basis, the B-spline basis, etc.) are TP. Some recent applications of such kind of matrices to CAGD can be found in [13] and [16]. For applications of TP matrices to other fields see [8]. In [7], [9] and [11] it has been proved that Neville elimination is a very useful alternative to Gaussian elimination when working with TP matrices.

In addition, there are some studies that prove the high performance computing of Neville elimination for any nonsingular matrix (see [4]). In [3] the backward error of Neville elimination has also been analyzed. In [1] we give a sufficient condition that ensures the convergence of iterative refinement using Neville elimination for a system $A\,x = b$ with A any nonsingular matrix in $\mathbb{R}^{n\times n}$, and then we apply it to the case where A is TP. Other applications and a study of the stability have been presented in [2].

2 Pivoting Strategies

We have already mentioned some advantages of Neville elimination for some classes of matrices. In general, Gaussian elimination and, in particular, Gaussian elimination with partial pivoting is the usual direct method for solving linear systems. However, as recalled in [14], in a serial implementation of Gaussian elimination with partial pivoting, the exchange of rows is recorded as an index permutation. In contrast, in parallel computations, physical exchange of elements of the matrix between processors will actually be required to implement that pivoting strategy, and so the communication cost associated with this exchange may be a limiting factor in the global efficiency of the parallel implementation, particularly if bidirectional communication is not possible. For Gaussian elimination, Onaga and Takechi introduced in [15] a strategy for reducing communications in parallel implementations. On the other hand, pairwise pivoting strategies (which can be closely related to Neville elimination) were considered for

 ISBN: 978-84-612-9727-6

parallel implementations (see [17], [18] and [19]) and even were mentioned in the earliest days of scientific computing by Wilkinson in [20].

An error analysis of a variant of pairwise pivoting closely related to Gaussian elimination was presented in [17]. In addition, numerical experiments for random matrices using neighbor pivoting (as it was called the variant of pairwise pivoting closely related to Neville elimination) were presented in [19].

3 Growth Factor

Let us now recall that the growth factor is an indicator of the numerical stability of a numerical algorithm and it measures the size of intermediate and final quantities relative to initial data. Given an $n \times n$ nonsingular matrix, we can mention the classical growth factor of A, used by Wilkinson,

$$\rho_n(A) := \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|},$$

where the matrices $A^{(k)} = (a_{ij}^{(k)})_{1 \leq i,j \leq n}$ are the intermediate matrices of the elimination procedure (see also [12]). In this paper, several definitions of growth factors for Neville elimination are compared. Theoretical bounds for the growth factors of several pivoting strategies related to Neville elimination are provided. These theoretical bounds do not allow us to distinguish among the considered pivoting strategies for Neville elimination, and in the case of the classical growth factor introduced by Wilkinson, we obtain the same bound as that of partial pivoting: 2^{n-1} for an $n \times n$ matrix.

Finally, we analyze numerical experiments for the average normalized growth factor (see [19] and [6]) for random $n \times n$ matrices. We use Neville elimination with partial pivoting and the other mentioned strategies related to Neville elimination, which present better behavior than partial pivoting in spite of satisfying similar theoretical bounds. We also compare the results of these numerical experiments with those obtained with some strategies related to Gaussian elimination.

Acknowledgments

This work has been partially supported by the Spanish Research Grant MTM2006-03388 and under MEC and FEDER Grant TIN2007-61273.

References

[1] P. Alonso, J. Delgado, R. Gallego and J. M. Peña, *Iterative Refinement for Neville Elimination*, Int. J. Comput. Math. **86(2)** (2009) 341–353.

[2] P. Alonso, J. Delgado, R. Gallego and J. M. Peña, *Neville elimination: an efficient algorithm with application to Chemistry*, to appear in Journal of Mathematical Chemistry.

[3] P. Alonso, M. Gasca and J. M. Peña, *Backward Error Analysis of Neville Elimination*, Appl. Numer. Math. **23** (1997) 193–204.

[4] P. Alonso, R. Cortina, I. Díaz and J. Ranilla, *Neville Elimination: a Study of the Efficiency Using Checkerboard Partitioning*, Linear Algebra Appl. **393** (2004) 3–14.

[5] T. Ando, *Totally Positive Matrices*, Linear Algebra Appl. **90** (1987) 165–219.

[6] V. Cortés and J. M. Peña, *Growth factor and expected growth factor of some pivoting strategies*, J. Comp. Appl. Math. **202** (2007) 292–303.

[7] J. Demmel and P. Koev, *The Accurate and Efficient Solution of a Totally Positive Generalized Vandermonde Linear System*, SIAM J. Matrix Anal. Appl. **27** (2005) 142–152.

[8] M. Gasca and and C. A. Micchelli, eds., *Total Positivity and its Applications*, Kluwer Academic Publishers, Boston, 1996.

[9] M. Gasca and J. M. Peña, *Total positivity and Neville elimination*, Linear Algebra Appl. **165** (1992) 25–44.

[10] M. Gasca and J. M. Peña, *Scaled pivoting in Gauss and Neville elimination for totally positive systems*, Appl. Numer. Math. **13** (1993) 345–356.

[11] M. Gassó and J. R. Torregrosa, *A Totally Positive Factorization of Rectangular Matrices by the Neville elimination*, SIAM J. Matrix Anal. Appl. **25** (2004) 986–994.

[12] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[13] H. Lin, H. Bao and G. Wang, *Totally positive bases and progressive iteration approximation*, Computers and Mathematics with Applications **50** (2005) 575–586.

[14] J. L. Mead, R. A. Renaut and B. D. Welfert, *Stability of a pivoting strategy for parallel Gaussian elimination*, BIT **41** (2001) 633–639.

[15] K. Onaga and T. Takechi, *A wavefront algorithm for LU decomposition of a partitioned matrix on VLSI processor arrays*, J. Par. Dist. Comput. **3** (1986) 158–182.

[16] J. M. Peña, *Shape preserving representations in Computer Aided-Geometric Design*, Nova Science Publishers, Inc., New York, 1999.

[17] D. C. Sorensen, *Analysis of pairwise pivoting in Gaussian elimination*, IEEE Trans. Comput. **C-34** (1985) 274–278.

[18] A. Tiskin, *Communication-efficient parallel generic pairwise elimination*, Future Generation Computer Systems **23** (2007) 179–188.

[19] L. N. Trefethen and R. S. Schreiber, *Average case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl. **11** (1990) 335–360.

[20] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendom Press, Oxford, 1965.

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009 30 June, 1–3 July 2009.

Reduced order models of industrial fluid-thermal problems

D. Alonso[1], L.S. Lorente[1], A. Velázquez[1] and J.M. Vega[2]

[1] *Aerospace Propulsion and Fluid Dynamics Department, School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, 28040 Madrid, Spain*

[2] *Applied Mathematics Department, School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, 28040 Madrid, Spain*

emails: diego.alonso.fernandez@upm.es, luissantiago.lorente@upm.es, angel.velazquez@upm.es, josemanuel.vega@upm.es

Abstract

A method is presented to generate reduced order models (ROMs) in Fluid Dynamics problems. The method is based on the expansion of the flow variables on a Proper Orthogonal Decomposition (POD) basis, obtained from a limited number of snapshots calculated via Computational Fluid Dynamics (CFD). Then, the POD-mode amplitudes are calculated as minimizers of a properly defined overall residual of the equations and boundary conditions. The residual can be defined using only a limited number of points in the flow field, which can be scattered either all over the whole computational domain or over a smaller projection window. The process is both computationally efficient (reconstructed flow fields require less than 1 % of the time needed to compute a full CFD solution) and flexible (the projection window can avoid regions of large localized CFD errors). Also, the number and distribution of snapshots and the effect of CFD errors are briefly discussed, to conclude that the method is numerically robust, since the results are largely insensitive to the definition of the residual, to CFD errors, and to the CFD method itself, which may contain artificial stabilizing terms. Thus, the method is amenable for practical engineering applications.

Key words: Reduced order model, Proper Orthogonal Decomposition, Incompressible nonisothermal flow

1 Introduction

The interest in the development of ROMs is twofold. The scientific interest in solving complex non linear multi-parametric problems in a computationally efficient way, using a limited number of modes is obvious. Second, the industrial interest focuses on the

generation of fast methods to solve problems of practical engineering interest. Computational efficiency is becoming crucial nowadays to improve design cycles, saving both design cost and time to market.

Many industrial applications involve complex Fluid Dynamics problems whose direct numerical simulation is well beyond present computer capability. Turbulence models and other related simplifications still require spatial discretization with thousands to millions of mesh points and some artificial stabilizing terms to avoid numerical instability. Steady state situations are of interest either because the industrially relevant solutions are genuinely steady, as in the example considered below, or because of turbulence modeling simplifications. The objective of this paper is to present a method, already anticipated in [1] and fully developed in [2], to obtaining ROMs to calculate steady states of multi-parametric fluid problems. The method itself relies as a strategy based on the minimization of a residual rather than on conventional Galerkin projection, which is more appropriate to deal with unsteady solutions and exhibits instabilities associated with mode truncation. The main new ingredient results from the observation that the residual can be calculated using a limited number of mesh points in the fluid domain. The method is checked to conclude that results are largely insensitive to peculiarities of the CFD method (artificial stabilization terms can be ignored in our formulation) and to CFD errors (the method improves CFD results). Results are sensitive instead to the selected snapshots in the parameter plane, which is also illustrated. Thus, the method is both numerically robust and computationally cheap, and thus amenable to engineering applications. Regarding organization of the article, the method is presented in section 2, a test problem is described in section 3, and results are given and discussed in section 4, followed by conclusions, in section 5.

2 ROM description

Our ROM consists of first calculating the relevant POD modes. The flow variables are then expanded as linear combinations of the POD modes. The associated mode amplitudes are calculated minimizing a properly defined residual.

POD decomposition based on a set of CFD calculated snapshots is made in each flow variable as explained in section 2.1. The overall residual to be minimized is explained in section 2.2.

2.1 POD modes for each state variable

We use a CFD code to calculate N_0 solutions of (1), with state variables $q_{j1} \dots q_{jN_0}$ (calculated for various sets of parameter values), which will be called snapshots. Using these, for each state variable we calculate the associated POD modes, denoted as $Q_{j1} \dots Q_{jN_0}$, with $Q_{jk} = \sum_{r=1}^{N_0} \alpha_{jk}^r q_{jr}$, where for each j, α_{jk}^r are the eigenvectors of the covariance matrix R^j , namely $\sum_{r=1} R_{kr}^j \alpha_{js}^r = \lambda_{js} \alpha_{js}^k$, defined as $R_{kr}^j = \langle q_{jk}, q_{jr} \rangle$. Here $\langle , \rangle$ is an appropriately defined inner product.

 ISBN: 978-84-612-9727-6

Now, for each state variable, we define the number of retained modes, N_j,

$$\sqrt{\sum_{s=1}^{N_j} \lambda_{js} \Big/ \sum_{s=1}^{N_0} \lambda_{js}} < \epsilon$$

for some pre-determined error bound ϵ. This means, invoking well known POD formulae, that after truncation to N_j modes, the root mean square error (RMSE) of reconstructing all snapshots in each state variable is bounded by ϵ. Errors are defined here using the norm $\|q\| = \sqrt{\langle q, q \rangle}$. Now, we expand each state variable in terms of its retained modes, as

$$q_j \approx \sum_{k=1}^{N_j} A_{jk} Q_{jk}. \tag{1}$$

Using independent POD modes for each flow variable allows us to simultaneously impose all equations and boundary conditions, which cannot be done in standard Galerkin-like reduced order equations.

2.2 Overall residual

Let us denote as $E_j = 0$ and $BC_j = 0$ the governing partial differential equations and boundary conditions, respectively. The POD-mode amplitudes are calculated minimizing an overall residual of the equations and boundary conditions, defined as

$$H = \sum_{j=1}^{m} \sqrt{\frac{1}{N_E} \sum_{k=1}^{N_E} |E_j(x_k, y_k)|^2} + \sum_{j=1}^{n} \sqrt{\frac{1}{N_{BC}} \sum_{k=1}^{N_{BC}} |BC_j(x_k, y_k)|^2} \tag{2}$$

where the first sum is extended to all mesh points (30,452 points in test test problem below) in the computational domain, and the second sum, to all the mesh points in the boundary of the computational domain

Computational cost is drastically reduced considering in (2) only a limited number of mesh points, scattered either over the whole domain Ω or over a portion of it, which will be called projection window below. As explained in [2] and illustrated below, results are fairly independent of both the projection window and the number of selected mesh points,with only weak limitations. Improvement of computational cost is very significant. The parameter N_{BC} could be also decreased as we did with N_E, but this would not reduce computational time because N_{BC} is usually quite small compared to N_E.

Now, the residual defined above can be minimized using various methods. Genetic Algorithms (GA) exhibit the advantage of being robust, which is convenient in the present paper, where the GA described in [1] is used. Of course, gradient based methods such as steepest descend would provide much faster versions of our ROMs.

 ISBN: 978-84-612-9727-6

3 Problem description

To illustrate the method, we have selected as a test problem the non-isothermal (heated from the lower wall) flow past a backwards facing step in the steady regime (see Fig. 1); see [3, 4] for a more detailed description of the problem and the CFD method.

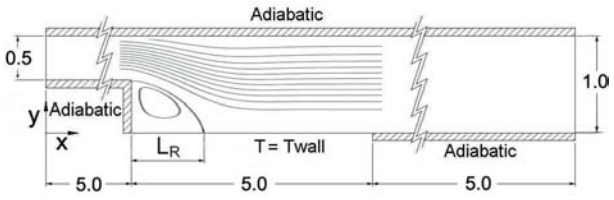


Figure 1: Sketch of the computational domain and the streamlines

This problem is characterized by a recirculation bubble that appears behind the step. We consider temperature dependent viscosity and thermal conductivity to account for the fact that large variations (of the order of 300 % in the former, see reference [5]) occur in the temperature range considered here, from 293 K to 353 K; Reynolds number varies from 50 to 250.

If the width of the channel is large compared to depth, then the flow topology can be considered 2-D except near the lateral walls) because, as shown in references [6, 7], the onset of 3-D instabilities starts at a Reynolds number in the range from 700 to 1000, depending on the remaining parameters. Dimensionless continuity, momentum, and energy equations are the usual ones, see [1, 2]. For numerical reasons, some small fourth order stabilizing terms are added to the equations in the CFD method (as usually done in industrial codes, see [3]), but the exact equations are used to calculate the residual (2).

At the inlet and outlet sections we impose a Pouseuille-like flow (see [8]) and a stress-free condition, respectively. No-slip ($u = v = 0$) is imposed at solid walls and pressure is computed solving an approximate formulation of momentum equations with one sided derivatives (into the flow domain). Wall temperature is prescribed in the region $5 < x < 10$ (see Fig. 1), while the remaining part of the lower wall and the upper wall are both considered to be thermally insulated.

As formulated, the test problem above depends on two nondimensional parameters, the wall temperature and the Reynolds number (Re, based on the outlet height), which are assumed to vary in the intervals [0,1] and [50,250], respectively. On the other hand, three parameters have been selected to compare ROM and CFD solutions: the reattachment length L_R (see Fig. 1), the pressure drop P_D (namely, a measure of the required pump power, which is defined as the difference between the average pressures at the inlet and outlet sections), and the Nusselt number Nu (which is a nondimensional measure of the total heat flux across the non-insulated part of the lower wall).

 ISBN: 978-84-612-9727-6

4 Results

To apply the method, we have calculated 25 snapshots using CFD and have selected five test points, PT1,...,PT5, where velocity, pressure, and temperature fields will be reconstructed. A visual impression of the parametric plane, the computed snapshots, and the selected test points is presented in Fig. 2.

In order to have a reference of the results obtained below, we give in Fig. 3 contours of the local residual of the CFD solution on continuity and x-momentum equations, at point PT3; residuals at the remaining points and the remaining equations are similar. Note that large CFD errors are concentrated near the upper part of the step, which are due to the singularity associated with the 270-corner.

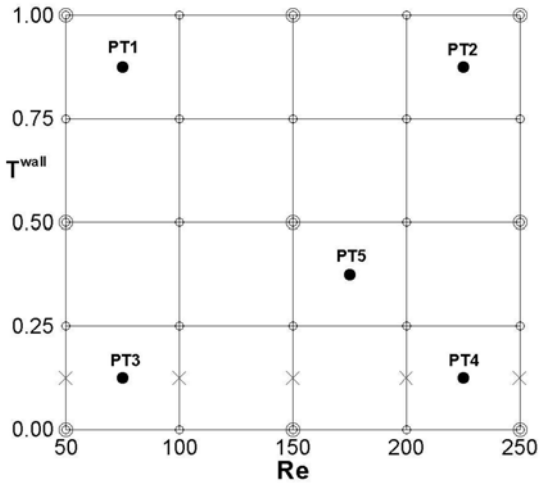


Figure 2: Sketch of the parametric domain

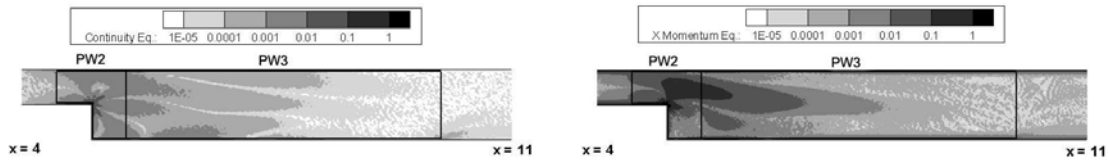


Figure 3: Contours of the local residuals (namely, the absolute value of the left hand side of the equation) of CFD solution at point PT3 in continuity (up) and x-momentum (down) equations

Also for reference, we have applied to this test problem both CFD and the POD plus Interpolation method described in [9, 10]. Results are given in Table 1. Errors

 ISBN: 978-84-612-9727-6

lower than 0.1 % are rounded hereafter to 0.0. Note that POD+interpolation predict the reattachment length with errors smaller than 2% all over the parametric plane. Pressure drop and Nusselt number relative errors deteriorate (relative errors of the order of 10 %) at low Re, at points PT1 and PT3, which is due to the fact that dependence of pressure drop on Re is strongly nonlinear at low Re.

	CFD results			POD+I (in %)		
	L_R	P_D	Nu	L_R	P_D	Nu
PT1	1.26	3.59	4.48	1.5	8.6	0.6
PT2	2.76	0.98	6.15	0.0	1.0	0.1
PT3	1.16	3.71	0.54	0.0	8.6	16.7
PT4	2.62	1.01	0.90	0.8	0.0	1.1
PT5	2.24	1.38	2.47	0.0	2.2	0.8

Table 1: CFD-results and relative errors resulting from POD plus interpolation (POD+I) over the six first modes

With these results in mind, the object of the remaining of this section is to illustrate the effect on the results of the various parameters appearing in the method described in section 2. Summarizing the ROM details, which have been chosen after some preliminary calibration, we retain 6 POD modes in each flow variable (which leads to a total number of 6x4=24 mode amplitudes) from the set of 25 snapshots, introduce the expansion (1) into the equations and boundary conditions, and minimize the overall residual (2) using a Genetic Algorithm (see [1] for more details).

4.1 The projection window

Two projection windows, PW1 and PW2, are considered (see Fig. 3) both to calculate the POD modes and the overall residual. PW1 and PW2 contain 11,074 and 1,849 mesh points, respectively.

Results using these two projection windows are given in Table 2 and show that POD plus interpolation results in Table 1 are clearly improved by the ROM. In fact using the whole computational domain provides errors that are within 3% in all cases except for the Nusselt number at point PT3, which is calculated with a 11.1% error (a further analysis of this point is given in section 4.3 below). Also, accuracy degrades only slightly when the projection window PW1 is used, which highly decreases computational time: each computation only requires 10 minutes in a desktop personal computer. Results are consistently better in case PW1 than in case PW2. This is due to the fact that the projection window PW1 contains at least a part of the recirculation bubble and thus bears more information about the flow topology than window PW2; also, PW1 excludes the region where CFD errors are larger, see Fig. 3. Thus, all calculations below are made in projection window PW1. Finally, it is remarkable that using a projection window, we are obtaining the solution in the whole computational domain minimizing the residual only in a small part of it.

 ISBN: 978-84-612-9727-6

	Whole domain			PW1			PW2		
	L_R	P_D	Nu	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	1.4	1.1	1.6	2.2	1.3	6.3	1.7	0.9
PT2	0.7	1.0	0.1	0.0	1.0	0.0	2.2	13.3	0.2
PT3	0.0	0.8	11.1	1.8	1.1	20.3	5.2	1.9	18.5
PT4	2.3	0.0	1.1	2.3	1.0	3.3	2.3	4.0	1.1
PT5	0.0	2.1	0.8	0.0	1.4	0.8	5.4	8.7	0.4

Table 2: Relative errors (in %) at test points obtained using all mesh nodespoints in the computational domain and in the projection windows PW1 and PW2. We retain 6 POD modes in each in each flow variable obtained from 25 snapshots

4.2 Number of mesh points inside the projection window

As an additional step to save computational time, we now check whether the number of mesh points in the projection window can still be decreased, considering the following numbers of equispaced mesh points in projection window PW1: 84, 51, and 26. Results are shown in Table 3. Comparison with Table 2 indicates that, excluding again Nusselt number at point PT3, no loss of accuracy results from decreasing the number of mesh points in the projection window. Moreover, the number of mesh points needed by our ROM is really small. Actually, it suffices that this number be somewhat larger than the number of unknowns in the problem, namely the number of POD mode amplitudes (i.e., 6 x 4 = 24 in the present case). The computational cost is consistently decreased, since, e.g., 53, 42, and 34 seconds are enough to compute each case using 84, 51, and 26 mesh points, respectively.

	84 points			51 points			26 points		
	L_R	P_D	Nu	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	0.3	0.9	0.0	1.9	0.9	1.6	2.5	0.7
PT2	0.7	1.0	0.0	0.7	1.0	0.0	0.0	2.0	0.0
PT3	1.7	1.0	18.5	1.7	0.3	16.7	1.8	0.3	13.0
PT4	2.3	1.0	1.1	2.3	1.0	0.0	2.3	1.0	2.2
PT5	0.0	1.4	0.8	0.0	2.2	1.2	0.0	1.4	0.8

Table 3: As in Table 2, but minimizing the residual in the indicated number of mesh points in the projection window PW1

4.3 Improving CFD results with errors: analysis of points PT1 and PT3

Let us now concentrate in the ROM calculation at test points PT3. A first attempt to improve these results consists of adding five new snapshots, at those points denoted

with crosses in Fig. 2, namely at $T^{wall} = 0.125$ and $Re = 50$, 100, 150, 200, and 250. Modes are selected as in Table 3 and results are given in Table 4, up. Comparison with Table 3 shows that prediction of Nusselt number at point PT3 has been only slightly improved. At this point, a doubt arises on the precision of CFD results at PT3, which leads us to plot in Fig. 4 (solid line) the CFD values of Nusselt versus Reynolds number at $T^{wall} = 0.125$. Such plot shows a non decreasing slope, which is suspicious. CFD errors may be due to the fourth order stabilizing term in the energy equation, which has been added to the CFD code to avoid numerical instability and is especially dangerous at this low temperature. Thus, we recalculate the CFD temperature profiles integrating only the energy equation, without any stabilizing term, using the velocity field provided by the former CFD calculation. Results are plotted with dashed lines in Fig. 4 and confirm that former CFD results were wrong, as suspected. When comparing with this corrected CFD results we obtain the errors given in Table 4, down. Note that they are quite good.

	ROM			POD+I		
	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	1.1	0.9	1.6	7.8	0.9
PT2	0.7	0.0	0.2	0.7	0.0	0.2
PT3	0.0	0.8	11.3	1.7	6.2	11.3
PT4	0.0	1.0	3.3	0.0	1.0	3.3
PT5	1.1	1.0	1.7	3.3	0.5	1.5
PT3	1.7	1.1	5.3	3.4	1.1	5.3

Table 4: Up: relative errors (in %) resulting from two sets of snapshots (30 snaps), using (6,8,4,8)-POD-modes in each flow variable in the projection window PW1 with 84 points. Down: PT3 relative errors after recalculation of CFD

All these mean that the ROM exhibits unexpected advantages. Namely, all results on the ROM in Table 4, down, have been obtained using the original CFD-calculated snapshots, which exhibit errors at low temperature due to the unphysical stabilization terms. In spite of that, ROM results are quite good and, what is more important, ROM improve CFD results. This has obvious consequences when devising industrial applications. This unexpected advantage of the ROM is new to our knowledge in this context and could be surprising at first sight. But in fact, this becomes more natural when realizing that POD modes are only used to (i) obtain a good POD manifold; the exact equations themselves are imposed again when (ii) the residual is minimized. Thus, errors in the snaps only affect step (i), while the actual approximation is calculated in step (ii). One can also think that Fourier modes do not bear any information about the solution that is being approximated (they only bear information about the fact that the solution is periodic); the equations themselves are imposed when calculating the actual Fourier expansion, which is the counterpart of our step (ii).

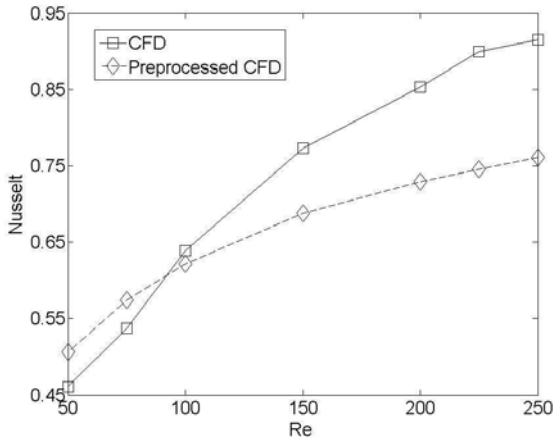


Figure 4: Nusselt number vs. Reynolds number resulting from CFD calculations, both with the code used in the remaining of the paper (solid) and with post-processing on the energy equation (dashed).

5 Conclusions

We have presented a new method for constructing ROMs to calculate steady states of complex fluid flows in a multi-parametric setting. The main idea behind the method was given in section 2, and developed in section 4 using a test problem. The following overall conclusions/remarks are in order:

1. Alternative methods can be constructed that are more appropriate than standard Galerkin to calculate steady states (and, we believe, to unsteady problems as well). The main weaknesses of the latter can be overcome, revising the ideas behind the concept of POD-based ROMs. In particular, localized CFD errors, which are frequently present in industrial calculations, can be dealt with in an efficient way. And artificial stabilizing terms, required by CFD computations, have been completely ignored above. Instead, the exact equations and boundary conditions have been considered to calculate the ROM. Thus, our ROM is robust, namely it is independent of the way snapshots have been calculated. More flexibility is convenient in the definition of the POD modes themselves, treating the various flow variables independently.

2. It is feasible to compute the modes using only information from a limited flow field region, with only weak limitations. We can expect that placing projection windows in that part of the fluid domain that show more structure (which can be usually decided a priori, with only a qualitative knowledge of the expected solution) and selecting equispaced mesh points should be enough to obtain good results. Also, this opens the possibility of deriving the whole flow field using

information from a small part of it. This means, for instance, that databases of practical engineering interest could be reconstructed out of gappy experimental data.

3. Selection of the snapshots is a critical issue since CFD calculation of these is the most expensive part of the process. We believe that it should be possible to design a method to select the snapshots in such a way that only a few of them are enough, if properly selected. Its number should be just somewhat larger than the number of POD modes. The method would provide a dramatic reduction in computational time, since this is essentially associated with CFD; the remaining calculations in our method are quite inexpensive after the improvements introduced above. Such method is the object of our current research.

4. Finally, and somewhat unexpectedly, our analysis (in section 4.3) of the anomalous behavior of point PT3 has shown that our ROM is able to improve the precision of CFD results, when these exhibit large numerical errors. Such unexpected advantage of our ROM is of great interest in industrial applications, since industrial CFD codes frequently exhibit large errors due to time and cost constrains. But this is ahead of the scope of the paper, and again the object of our current research.

Acknowledgements

This research was supported by the Spanish Ministry of Education and Science (MEC) under Projects DPI2005-05572 and TRA2007-65699

References

[1] D. Alonso, A. Velazquez, J.M. Vega, *Robust reduced order modeling of heat transfer in a back step flow*, Int. J. Heat Mass Transfer **52** (2009) 1149–1157.

[2] D. Alonso, A. Velazquez, J.M. Vega, *A method to generate computationally efficient reduced order models*, Comput. Methods Appl. Mech. Engrg. **In Press**.

[3] B. Mendez and A. Velazquez, *Finite point solver for the simulation of a 2-D laminar incompressible unsteady flows*, Comput. Methods Appl. Mech. Engrg. **193** (2004) 825–848.

[4] A. Velazquez, J.R. Arias, B. Mendez, *Laminar heat transfer enhancement downstream of a backward facing step using a pulsating flow*, Int. J. Heat Mass Transfer **51** (2008), 2075–2089.

[5] F.P. Incropera, D.P. Dewitt, *Introduction to Heat Transfer*, John Wiley & Sons (1996), Table A-6, pp 764.

[6] L. Kaiktsis, G.E. Karniadakis, S.A. Orzag, *Onset of the three dimensionality, equilibria and early transition in flow over a backward facing step*, J. Fluid Mech. **231** ((1991) 501–558.

[7] D. Barkley, M.G.M. Gomes, R. Henderson, *Three dimensional instability in flow over a backward facing step*, J. Fluid Mech. **473** (2002) 167–190.

[8] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge Univ. Press (1967)

[9] L.S. Lorente, J. M. Vega, A. Velazquez, *Generation of aerodynamics databases using Singular Value Decomposition*, J. Aircraft **45** (2008), 1779–1788.

[10] L. de Oathauwer, B. de Moor, J. Vandewalle, *On the best rank-one and rank-(R1,R2, ,RN) approximation of higher order tensors*, SIAM J. Matrix Anal. Appl. **21-4** (2000) 1324–1342.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

On symmetry groups of quasicrystals

Viacheslav A. Artamonov[1] **and Sergio Sànchez**[2]

[1] *Department of Algebra, Faculty of Mechanics and Mathematics, Moscow State University*

[2] *Dept. of Applied Mathematics, University Rey Juan Carlos, Madrid, Spain*

emails: artamon@mech.math.msu.su, sergio.sanchez@urjc.es

Abstract

The paper contains a survey of various approaches to a definition of symmetry groups of quasicrystals. There is given a classification of finite groups of symmetries in small dimensions.

Key words: groups, symmetries, mathematical models of quasicrystals
MSC 2000: AMS 20G75

1 Introduction

Symmetry groups of crystals play an important role in the geometric theory of crystals. They help to classify all possible combinations of positions occupied by atoms in materials. A mathematical theory of symmetry groups of crystals was completed in thirties in the last century.

A new alloy $Al_{0,86}Mn_{0,14}$ which found in 1984 admits an icosahedral symmetry which is forbidden in the symmetry theory of crystals. The new metallic alloys are now called quasicrystals.

In the paper using *cut and project* model Q for quasicrystals we introduce a group of symmetries $\operatorname{Sym} Q$ of Q and its distinguished subgroup $\operatorname{Sym}_W Q$ depending on a choice of a window W.

2 Symmetries of crystals

In this section we recall mathematical model of Delaunay of crystalline sets and recall basic results their concerning symmetries.

Let K be a set of all positions in an ideal crystal occupied by atoms. It is supposed that K is a discrete subset in an Euclidean space E. Since an ideal crystal is a solid material located in E we shall consider in E an Euclidean metric $\|x-y\| = \sqrt{(x-y, x-y)}$.

It is assumed that E has a finite dimension. An *isometry* of E is a map $\Phi : E \to E$ preserving a distance between any two vectors, that is $\|\Phi(x) - \Phi(y)\| = \|x - y\|$ for all $x, y \in E$. All isometries of E form a group $\operatorname{Iso} E$ under the operation of multiplications of transformations. The following well known theorem reduces isometries to orthogonal linear operators and transfers, see [10][Chapter 4, §3, Theorem 3]

Theorem 1. *A transformation Φ of an Euclidean space E is an isometry if and only if there exists an orthogonal linear operator ϕ called the* differential *$d\Phi$ of Φ and a vector $a \in E$ such that $\Phi(x) = \phi(x) + a$ for all $x \in E$. In particular Φ is bijective.*

Since an ideal crystal K is solid body the group $\operatorname{Sym} K$ of its symmetries consists of all isometries Φ of the space E such that $\Phi(K) = K$. The following definition of a crystalline set K due to B. Delaunay involves the group $\operatorname{Sym} K$.

Definition 1 ([6])**.** A subset K of an Euclidean space E is *crystalline* or *Delaunay*, if the group of its symmetries $\operatorname{Sym} K$ satisfies the following conditions

1) given a point $A \in E$ then there exists a real number $d(A) > 0$ such that $\|\Phi(A) - A\| < d(A)$ for some $\Phi \in \operatorname{Sym} K$ implies $\Phi(A) = A$;

2) there exists a fixed real number $D > 0$ such that for any two points $A \in K$ and $B \in E$ one can found a transformation $\Psi \in \operatorname{Sym} K$ for which $\|\Psi(A) - B\| < D$.

Take an additive discrete subgroup L of E as an additive abelian group. It is known that E is a finitely generated torsion-free group and therefore L is a free abelian group. Any base of L as a free abelian group consists of linearly independent vectors in E. A discrete subgroup L in additive group of the Euclidean space E of dimension n is a *lattice* if L has a basis consisting n vectors of n elements. Actually it means that $E = \mathbb{R} \otimes_{\mathbb{Z}} L$.

Theorem 2 (Schoenflies–Bieberbach)**.** *Let $\Gamma = \operatorname{Sym} K$ be a symmetry group of a crystalline set $K \subset E$ and $N = N(\Gamma)$ a subset of all transfers in $\operatorname{Sym} K$. Then $N \lhd \Gamma$ and the factorgroup $\Delta = \operatorname{Sym} K/N = d(\operatorname{Sym} K)$ is finite. A subgroup $N(0)$ consisting of all vectors $\{f(0) \in E \mid f \in N\}$ is a lattice in E. Here 0 is the origin.*

The finite group Δ is called a *point* group. Since the group $N(0)$ is a lattice we can conclude that there exists a matrix $C \in \mathrm{GL}(n, \mathbb{R})$ such that $C\Delta C^{-1} \subset \mathrm{GL}(n, \mathbb{Z})$.

Theorem 2 means that in order to classify the groups Γ from group-theoretical point of view we need to pass through three steps:

(i) classify point groups Δ which are finite subgroup,

(ii) find the lattice $N(0) \subset E$ which is invariant under the action of Δ,

(iii) construct a group extension Γ from $N(0)$ and Δ,

$$1 \longrightarrow N \longrightarrow \Gamma \longrightarrow \Delta \longrightarrow 1.$$

 ISBN: 978-84-612-9727-6

The list of point group in the case when $\dim E = 2, 3$ can be found in [2][Chapter2]. For example we have

Theorem 3. *Let a point group Δ be a subgroup in $\mathrm{O}(2,\mathbb{R})$. Then Δ is either a cyclic group $\langle a\rangle_n$ or a dihedral group $\mathbf{D}_n$, where $n = 1, 2, 3, 4, 6$.*

3 Quasicrystals

There are various approaches to a construction of mathematical models of quasicrystals and a definition of their symmetries [11], [9], [1], [5], [12], [13].

We shall adopt the model which is usually call *cut and project scheme*. Let V be an additive locally compact topological Abelian group, U an additive group of a real *physical* vector space of dimension d and M a discrete subgroup in $E = U \oplus V$ such that E/M is compact and $M \cap V = 0$. The group E is often called a *hyperspace* and U a *physical* space. Consider the diagram of group projections

$$\begin{array}{ccccc} U & \xleftarrow{\ \pi\ } & E & \xrightarrow{\ \rho\ } & V \\ & & \cup & & \\ & & M & & \end{array}$$

Note that π is injective on M.

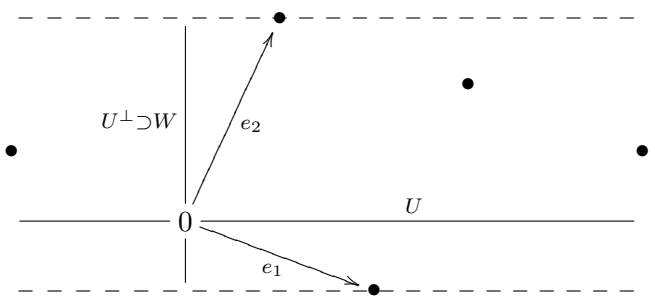
A nonempty compact subset $W \subset V$ is called a *window*, [12], if W is a completion of its interior. Since $\rho(M)$ is dense in V we can conclude that $\rho\,(M) \cap W$ is dense in W. Hence there exists a point $A \in M$ and a base $e_1, \dots, e_n$ of the lattice M, such that the image of the cube

$$K = \{A + \mu_1 e_1 + \cdots + \mu_n e_n \mid 0 \leqslant \mu_i \leqslant 1\} \tag{1}$$

under the projection ρ belongs to W. Then V is a span of elements $\rho(e_1), \dots, \rho(e_n)$.

Put $\mathcal{Q} = \rho^{-1}(W) \cap M$. $Q = \pi(\mathcal{Q})$ is called a *cut and project set* , or a *quasicrystal* in the physical space U. Note that π maps $\mathcal{Q}$ injectively to U. Thus π induces a bijection $\mathcal{Q} \to Q$.

In particular if W is a convex hull of $\rho(e_1), \rho(e_2)$ and $\dim U = \dim V = 1$ then we have the following picture



 ISBN: 978-84-612-9727-6

The bullets in this picture show the points from $\mathcal{Q} = (U \oplus K) \cap M$. Their projections form a quasicrystal Q.

Recall some basic results concerning quasicrystals from [2][Chapter 6, §6.3].

Theorem 4 (Local metric reiteration [2]). *Let $\mathcal{Q}$, Q W, U, E be as above and S a finite subset in $\mathcal{Q}$ having a neighborhood belonging to $U \oplus W$. For any $T > 0$ there exists a vector $x \in M$ such that $\|x\| > T$ and $S + x \in \mathcal{Q}$. In particular there exists a vector $x \in M$ such that $S + x \subseteq \mathcal{Q}$ and $\|\pi(x)\|$ is arbitrary large.*

4 Symmetries of quasicrystals

In contrast with symmetries of crystals in the theory of symmetries of quasicrystals we shall consider not only isometries but larger groups of affine transformations of vector spaces.

Definition. A bijective map Φ of a real vector space E onto itself is affine if Φ maps lines to lines. All affine transformations of E form a group under composition which is denoted by $\mathrm{Aff}\, E$.

Theorem 5 ([10], Chapter 4, §4, Theorem 9). *A map $\Phi : E \to E$ is affine if and only if there exists an invertible linear operator ϕ and a vector $b \in E$ such that*

$$\Phi(B + x) = \phi(x) + b \tag{2}$$

for some $b \in E$, where B is a fixed point in E.

Note that the linear operator ϕ is the differential $d(\Phi)$.

There are several approaches to a definition of a symmetry group of a quasicrystal Q, see [8], [11], [2], [4]. The first idea is inspired by Theorem 4 is to consider the group Sym_W of affine transformation of E mapping $\mathcal{Q}$ bijectively onto itself. Sym_W is called a *proper symmetry group* of $\mathcal{Q}$ and of Q.

Theorem 6. *Let $\Psi \in \mathrm{Sym}_W Q$ and $\Psi(A + x) = \psi(x) + b$ where $A \in \mathcal{Q}$. Then $b = \Psi(A) \in Q$. The physical space U is invariant under the differential $d\Psi$. The lattice M is invariant under Ψ.*

The *general symmetry group* $\mathrm{Sym}\, Q$ of a quasicrystal Q is a subgroup of the group $\mathrm{Aff}\, E$ of all affine transformation of E consisting of all transformations of the hyperspace E such that the physical space U and the lattice M are mapping onto themselves. By Theorem 6 proper subgroup $\mathrm{Sym}_W Q$ is a subgroups in the group Sym.

Theorem 7. *The window W is invariant under the restriction $\rho\Psi \mid_V$ to the space V of a product $\rho\Psi$, provided $\Psi \in \mathrm{Sym}_W Q$. The map ρ^*, sending $\Psi \in \mathrm{Sym}_W Q$ to $\rho\Psi \mid_V$ is a group homomorphism. The group $\rho^*(\mathrm{Sym}_W Q)$ is relatively compact and it is isomorphic to the group of its differentials $d(\mathrm{Sym}_W)$. In particular if W is a convex polygon then the group $\rho^*(\mathrm{Sym}_W Q)$ is finite.*

 ISBN: 978-84-612-9727-6

Corollary 1. *There exists an inner point $F \in W$ such that $\Psi(F) - F \in U$ for all $\Psi \in \mathrm{Sym}_W Q$.*

Choose $F \in W$ as the origin. Then $\Psi(F) \in U$. Hence if $F + x \in Q$ then $\Psi(F + x) = \phi(x) + \Psi(F) \in Q$ or $\rho(\Psi(F + x)) = \rho(\phi(x))$. So W is stable under the action of $d(\mathrm{Sym}_W)$.

Corollary 2. *Let G be a finite subgroup in $\mathrm{Sym}\, Q$, Then G is isomorphic to a subgroup of $d(\mathrm{Sym}\, Q)$.*

Theorem 8. *Let $\mathrm{Sym}\, Q$ contain an element g of order m and*

$$m = p_1^{l_1} \cdots p_r^{l_r}, \quad p_1 < p_2 < \cdots < p_r \tag{3}$$

is a prime decomposition of m. If $p_1 = 2$, $l_1 = 1$ then $\dim E \geqslant \sum_{i=2}^{r} p_i^{l_i - 1}(p_i - 1)$. Otherwise $\dim E \geqslant \sum_{i=1}^{r} p_i^{l_i - 1}(p_i - 1)$.

Proof. According to Corollary 2 we can assume that g is a linear operator. In a complex space $\mathbb{C} \otimes E$ there exists a basis in which g has a diagonal matrix

$$\begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots^{0} & \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}$$

Since the order of g is equal to m each λ_j is a primitive root of degree m_j of 1 and m is the least common multiple of $m_1, \dots, m_n$.

Let $m_{j_1}, \dots, m_{j_t}$ is the maximal subset of different integers from $m_1, \dots, m_n$.

The matrix of g is conjugate to a matrix in $\mathrm{GL}(n, \mathbb{Z})$ because the lattice M is g-invariant. Hence its characteristic polynomial $\chi(T)$ is an integer polynomial. By the assumption each λ_j is a root of $\chi(T)$. Hence all primitive roots of 1 of degree m_j are root of $\chi(T)$. Thus

$$\dim E = \deg \chi(T) = \phi(m_{j_1}) + \cdots + \phi(m_{j_t}) \tag{4}$$

where ϕ is the Euler function. Let

$$m_{j_s} = p_1^{l_{s1}} \cdots p_r^{l_{sr}}, \qquad l_i = \max_s l_{si}, \quad i = 1, \dots, r. \tag{5}$$

Then

$$\phi(m_{j_s}) = \phi\left(p_1^{l_{s1}}\right) \cdots \phi\left(p_r^{l_{sr}}\right).$$

Note that if $A, B \geqslant 2$ then $A + B \leqslant AB$. Thus if either m_{j_s} is odd or divisible by 4 then

$$\phi(m_{j_s}) \geqslant \phi\left(p_1^{l_{s1}}\right) + \cdots + \phi\left(p_r^{l_{sr}}\right).$$

Suppose that $p_1 = 2$ and $l_{s1} = 1$. Then

$$\phi(m_{j_s}) = \phi\left(p_2^{l_{s2}}\right) \cdots \phi\left(p_r^{l_{sr}}\right) \geqslant \phi\left(p_2^{l_{s2}}\right) + \cdots + \phi\left(p_r^{l_{sr}}\right).$$

Using (4), (5) we complete the proof. □

 ISBN: 978-84-612-9727-6

Corollary 3 ([4]). *Let* $\dim U = 2$, $\dim E = 4$ *and* $g \in \mathrm{Sym}_W$ *has order* m. *Then* $m = 1, 2, 3, 4, 5, 6, 8, 10$.

Proof. Suppose that m has a prime decomposition (3). Then $4 \geqslant \sum_{i=2}^{r} p_i^{l_i-1}(p_i - 1)$ for any i. If $l_i \geqslant 2$, then $p_i = 2$ and $l_i = 2, 3$. Thus either $m = 2^l p_2 \cdots p_r$ where $p_2 < \cdots < p_r$ are odd primes and $l = 1, 2, 3$, or $m = p_1 \cdots p_r$ where $2 < p_1 < \cdots < p_r$ are odd primes.

By Theorem 8 in the first case $4 \geqslant 1+p_2-1+\cdots+p_r-1$ that is $3 \geqslant p_2-1+\cdots+p_r-1$ and $r \leqslant 1$. Thus $m = 2,\ 4,\ 8,\ 6,\ 10,\ 12$.

Suppose that m is odd. Then $4 \geqslant p_1 - 1 + \cdots + p_r - 1$ and again $r = 1$. Hence $m = 3, 5$. □

Corollary 4. *Let* $\dim U = 3$, $\dim E \leqslant 6$ *and* $g \in \mathrm{Sym}_W$ *has order* m. *Then* $m = 7, 15, 5, 10, 20, 2, 4, 8, 3, 9, 6, 12, 24$.

Proof. Note that

$$\phi(2^4) = 8, \quad \phi(3^3) = 9 \cdot 2 = 18, \quad \phi(5^2) = 5 \cdot 4 = 20, \quad \phi(7^2) = 7 \cdot 6 = 42, \quad \phi(11) = 10.$$

Hence by Theorem 8 we have

$$m = 2^{l_2} \cdot 3^{l_3} \cdot 5^{l_5} \cdot 7^{l_7}, \quad l_2 \leqslant 3,\ l_3 \leqslant 2,\ l_5, l_7 \leqslant 1.$$

If $l_7 = 1$, then $l_2 = l_3 = l_5 = 0$ and $m = 7$.

Suppose that $l_7 = 0$ and $l_5 = 1$ that is

$$m = 2^{l_2} \cdot 3^{l_3} \cdot 5, \qquad l_2 \leqslant 3,\ l_3 \leqslant 2,$$

If $l_2, l_3 > 0$ then $6 \geqslant 2^{l_2-1} + 2 \cdot 3^{l_3-1} + 4 \geqslant 1 + 2 + 4$ which is impossible.

Let $l_5 = 1, l_3 > 0, l_2 = 0$. Then $6 \geqslant 2 \cdot 3^{l_3-1} + 4$ which means that $l_3 = 1$. In this case $m = 15$.

Let $l_5 = 1, l_3 = 0, l_2 > 0$. Then $6 \geqslant 2^{l_2-1} + 4$ and $l_2 = 0,\ 1,\ 2$. So $m = 5,\ 10,\ 20$.

Suppose finally that $l_5 = l_7 = 0$ and $m = 2^{l_2}3^{l_3}$. If $l_2 > 0$ then $6 \geqslant 2^{l_2-1}$ and $l_2 = 1\, 2,\ 3$, that is $m = 2,\ 4,\ 8$.

If $l_2 = 0,\ l_3 > 0$ then $6 \geqslant 2 \cdot 3^{l_3-1}$ and $l_2 = 1,\ 2$ that is $m = 3,\ 9$.

Finally let $l_2, l_3 > 0$. Then $6 \geqslant 2^{l_2-1} + 2 \cdot 3^{l_2-1}$ and $l_2 = 1,\ 2,\ 3,\ l_3 = 1$. Hence $m = 6,\ 12,\ 24$. □

We shall consider the problem of a classification of finite subgroup in $\mathrm{Sym}\, Q$. It follows from Corollary 2 that we can consider the case when a finite group is a subgroup of $d(\mathrm{Sym}\, Q)$ and therefore it consists of linear operators.

The main result of the paper [4] is a classification of finite subgroups G in the subgroup of linear operators in $\mathrm{Sym}\, Q$ in the case when $\dim U = 2$, $\dim E = 4$. By Corollary 3 an order m of each element $g \in G$ is equal one of the numbers 1, 2, 3, 4, 5, 6, 8, 10, 12. We assume that G is not a point group of a crystalline set from Theorem 3. Under this assumption it is shown that G is a subgroup of direct product of two dihedral groups $\mathbf{D}_{k_1} \times \mathbf{D}_{k_2}$ where one of cases takes place

 ISBN: 978-84-612-9727-6

(1) $k_1 = k_2 = 10$;

(2) $k_1 = 5$, $k_2 = 10$ or $k_1 = 10$, $k_2 = 5$;

(3) $k_1 = k_2 = 8$;

(4) $k_1 = k_2 = 12$.

In each of these cases we show that G belongs to one of types:

a) a cyclic group $\langle B\rangle$;

b) direct product of two cyclic groups $\langle B\rangle \times \langle a\rangle$;

c) a semidirect product of a normal subgroup from the previous case and a cyclic group of order 2.

d) a dihedral group $\mathbf{D}_{10}$ in the case **(2)**.

There is given a representation of each of these groups as rotations and mirror mappings in 4-dimensional space E.

5 The proper symmetry group of a quasicrystal

In this section we shall look at subgroups of the general symmetry group Sym which are proper for some window W.

As in Theorem 7 we have

Theorem 9. *The map $\rho^* : \Psi \to \rho\Psi\mid_V$ is a homomorphism of the group* Sym *into the group of affine transformations of V.*

Proposition 1. *Let Ψ be an affine transformation of E such that M, U are stable under Ψ. Then $\Psi \in$* Sym.

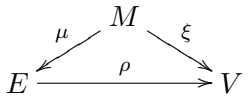
Corollary 5. *Let G be a subgroup of* Sym *such that for any $g \in G$ there is a vector $x \in E$ with a bounded g-orbit. In particular this is the case if either G is periodic , or G is a compact group. Then G is isomorphic to a subgroup of a point group d*(Sym).

Theorem 10. *Let G be a subgroup in* Sym *such that $\rho^*(G)\mid_V$ is relatively compact. Then there exists a window $W \subset V$ such that $G \subseteq \mathrm{Sym}_W Q$.*

6 Other models

Another model of a quasicrystal was proposed in [11].

A *quasilattice* is an Euclidean space V is an additive finitely generated subgroups M in V whose span is equal to V. It is assumed that the rank of M is greater than $\dim V$. Then we get a surjective linear map $\rho : E = \mathbb{R}\otimes_{\mathbb{Z}} M \to V$ for which the diagram



 ISBN: 978-84-612-9727-6

is commutative where μ, ξ are embeddings of M in E and in V.

According to [11], [14] symmetry groups are subgroups G in the isometry group of V under which the quasilattice M is invariant.

Theorem 11. *Let G be a symmetry groups of a quasilattice as it is mentioned above. Then there exists a group H of affine transformations of E such that:*

1) $U = \ker \rho$ is H-invariant H;

2) the lattice $\mu(M) \simeq M$ is also H-invariant;

3) the map π induces a group isomorphism $\rho^* : H \to G$.

Under these settings V is the phase space and $U = \ker \rho$ is the physical space.

If L is a quasilattice in an Euclidean phase space V then a *quasicrystal* in the sense of [14] is a complex function $\hat{\rho} : L \to \mathbb{C}$ such that L as an Abelian groups is generated by the support of $\hat{\rho}$, that is by all elements $x \in L$ such that $\hat{\rho}(x) \neq 0$. A *gauge* function in L is an element from $\hat{L} = \hom(L, \mathbb{R}/\mathbb{Z})$. Two quasicrystals $\hat{\rho}_1,\ \hat{\rho}_1$ are *indistinguishable*, if there exists am gauge function χ on $\hat{L}$ such that $\hat{\rho}_2(x) = \exp(2\pi i \chi(x))\, \hat{\rho}_1(x)$ for all $x \in L$.

A *symmetry* of a quasicrystal $\hat{\rho}$ is an orthogonal linear operator g such that

1) L is g-invariant,

2) $\hat{\rho} \circ g$ and $\hat{\rho}$ are indistinguishable.

By Theorem 11 the symmetry group in this sense is a subgroup of a general symmetry group Sym.

Another model is considered in [3]. A subset Λ in an Euclidean space E is a Meyer set if $\Lambda - \Lambda$ is a Delaunay set. A Pisot-Vijayaraghavan number, or *PV-number* is an algebraic integer $\beta > 1$ such that its Galois conjugates have absolute values small er than 1. A*cyclotomic PV-number* with a symmetry of order n is PV-numver β such that $\mathbb{Z}[\beta] = \mathbb{Z}\left[2\cos\frac{2pi}{N}\right]$. In quadratic case the Galois conjugate $\beta' = \pm\frac{1}{\beta}$. A *beta-expansion* of a positive real number x is an infinite sequence $x_i,\ i \leqslant k = \left[\log_\beta x\right]$, the integer part, such that $x = \sum_{i \leqslant k} x_i \beta_i$. Here $x_k = [\frac{x}{\beta^k}]$, $r_k = \{\frac{x}{\beta^k}\}$ and $x_i = [\beta r_{i+1}]\,,\ r_i = \{\beta r_{r+1}\}$ for $i < r$. Put $\mathbb{Z}_\beta = \{x \in \mathbb{R} \mid |x| = x_k\beta^k + \cdots + x_0\}$. Then $\beta\mathbb{Z}_\beta \subseteq \mathbb{Z}_\beta$ and $\mathbb{Z}_{\underline{e}ta} = \mathbb{Z}_\beta$. It is shown in [3] that $\mathbb{Z}_\beta,\ \mathbb{Z}_\beta - \mathbb{Z}_\beta$ are Delaunay sets. Put $\Gamma_1(\beta) = \mathbb{Z}_\beta + \mathbb{Z}_\beta\zeta,\ \zeta = \exp\frac{2\pi}{N}$. It is shown in [3, Theorem 1 and 2] a symmetry group of $\Gamma_1(\beta)$ consisting of all affine planar transformations leaving $\Gamma_1(\beta)$ invariant is found for some specific β. As in Theorem 11 it can be proved that these groups are subgroups of a 2Dsymmetry group Sym.

In the book [2, Chapter 6] it was suggested that inverse semigroups should play a substantial role in symmetries of quasicrystals. Indeed a semigroup is *inverse* if for any element x there exists a unique element x^* such that $xx^*x = x,\ x^*xx^* = x^*$. The basic example of an inverse semigroup is the following. Let X be a set and $S(X)$ is the set of all bijections between subsets in X which are considered as partial transformations of X. Then $S(X)$ is a semigroup with respect to multiplication of partial maps. If

 ISBN: 978-84-612-9727-6

$f : B \to C$ is a bijection between subsets B, C in X, then $f^* : C \to B$ is the inverse map. Any inverse semigroup can be embedded into an inverse semigroup of the form $S(X)$ for some set X. It means that inverse semigroups are related to partial bijections. This is precisely the case of symmetries of quasicrystals. This ideas was developed in the paper [15].

Acknowledgements

This work has been partially supported by Russian foundation for basic research, grant 09-01-00058.

References

[1] G. Aragón, J.Ĺ. Aragón, F. Davila, A. Gomez, M.Á. Rodriguez, *Geometric Algebra with Applications in Science and Engineering,* Eds. E. Bayro-Corrochano and G. Sobczyk Birkhḧuser (Boston, Febrero 2001) pp.371-386.

[2] V. A. Artamonov and Yu. L. Slovokhotov *Groups and their applications in physics, chemistry and crystalography*, Moscow: Pulishing center "Academia", 2005, P.512.

[3] Avi Elkharrat, Christiane Frougny, Jean-Pierre Gazeau, Jean-Louis Verger-Gaugry , *Symmetry groups for beta-lattices*,Theoretical Computer Science, **319** (2004), 281-315.

[4] V. A. Artamonov, S. Sanchez, *Remarks on symmetries of 2Dquaiscrystals*, Proc. of the Conference on computational and Mathematical Methods in Science and Engineering, (CMMSE-2006), University Rey Juan Carlos, Madrid, Spain, September 21-25, 2006, 59-70.

[5] J. Hermisson, Ch. Richard, M. Baake, *A Guide to the symmetry structure of quasiperiodic tiling classes,* J. Phys. I France, **7** (1997), 1003-1018.

[6] B. Delonais, N. Padurov, A. Alexandrov, *Mathematical foundations of a structure analysis of crystals.* Moscow: ONTI. GTTI, 1934.

[7] D. Fried, W. D. Goldman, *Three-dimensional affine crystallographic groups,* Adv. in Math. **47** (1983), 1-49.

[8] A. Janner, *Crystallographic symmetries of quasicrystals*, Phase Transitions, **43** (1993), 35-47.

[9] C. Janot, *Quasicrystals* Oxford:Carleron Press, 1994.

[10] A. I. Kostrikin, *Introduction to Algebra, II, Linear algebra,* Moscow: Fizmarlint, 2001.

[11] Le Ty Kyok Thang, S. A. Puinikhin, V. A. Sadov, *Geometry of quasicrystalls.* Russian math. surveys, **48** (1993), N 1, 41-102.

[12] R. V. Moody *Model sets: a survey.* In the book: From quasicrystals to more Complex Systems. Springer, Berlin, pp. 145-166, arXiv:math.MG/0002020 v1

[13] P. A. B. Pleasants, *Designer quasicrystals: cut-and-project sets with preassigned properties.* In the book: Directions in Mathematical Quasicrystals (M. Baake and R. V. Moody, eds.), CRM Monograph Series, AMS, Providence, Rhode Island, pp 93-138.

[14] B. N. Fisher, D. A. Rabson, *Applications of group cohomology to the classification of quasicrystal symmetries,* J. Phys. A:Math. Gen., **36** (2003), 10195-10214.

[15] J. Kellendonk, M. V. Lawson, *Universal groups for point-sets and tillings,* J. Algebra, **276** (2004), no. 2, 462–492.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Control of the particle number in particle simulations

Franck Assous[1]

[1] *Department of Mathematics & Computer Sciences, Ariel University Center*

emails: franckassous@netscape.net

Abstract

Particle method is a well-known approach that has been used for a long time in charged particle beams or plasma physics modeling. In recent years, particle based methods have become widespread tools for approximating solutions of ordinary/partial differential equations in a variety of fields. However an intrinsec constraint on the use of particle methods is the need to control the number and the distribution of the particles in particle simulations. The aim of this paper is to propose a way to control it, while preserving the physics of the considered problem.

Key words: particle method, coalescence

1 Introduction

Particle methods have been used for a long time to numerically solve problems in charged particle beams or plasma physics modeling [1], [2]. In recent years, particle methods have become tools for approximating solutions in a variety of fields. One can find examples in diffusion [3] or convection-diffusion [4] problems, medical application [5], chemical engineering [6] among other. In these methods, a solution of a given equation is represented by a collection of particles, located in points $\mathbf{x}_k$ and carrying masses ω_k. Equations of evolution in time are then written to describe the dynamics of the location of the particles and their weights.

Nevertheless, an intrinsec constraint on the use of particle methods is the need to control the number and the distribution of the particles in the simulations. The first reason is to limit to a "reasonable" level the total number of particles when dealing with source terms. Depending on the applications, they can result from ionization process, collisions terms, re-emission from the boundaries, etc. Another reason can be to enforce a quasi constant number of particles per cell, for accuracy purpose, when using such simulations. Finally, it is sometimes worthwhile to reorganize the particle distribution in order to reduce the numerical noise.

In order to preserve the physics of the problem, the control of the particle number has to fulfill some constraints, such as the mass, momentum or energy conservation, or the positivity of the mass and of the energy. The aim of this paper is to propose a method to control the particle number which preserves the physics of the problem. We hope this could provide a useful tool of simulation in several fields. Let us take an example of application in chemistry modeling.

Following [7], consider the simulation of the dynamics of a ball milling process. Basically, this is a trajectography/collision problem: one has to find the trajectory $\mathbf{x}(t)$ of grains of powder and milling balls, which are represented by particles (see Figure 1), submitted to a given force $\mathbf{F}$ (that represents collision, loss of energy, etc.). The ball

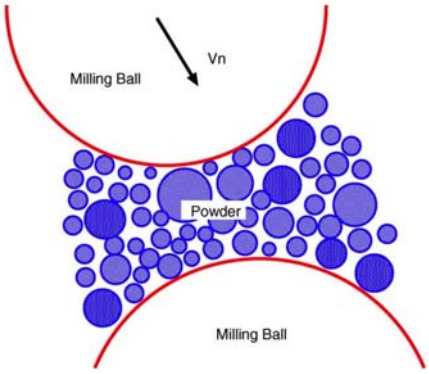


Figure 1: A particle model to investigate the influence of powder on ball impact. ©*Feng et al.[7]*

milling process, namely the collision of the balls on the grains of powder, generates finer powder particles, but leads to increase the particle number. For this reason, it can be useful to control it, especially in such cases to limit it to a reasonable level. For this reason, this is refers as a *coalescence* process.

In this article, we propose a method for coalescing particles in such simulations. We will expose it in the framework of a Particle-In Cell (*PIC*) approach, which is widely used for plasma physics simulation. From this process, we deduce an algorithm of coalescence that conserves the particle and cell charge and current densities, and the particle energy. The positivity of the mass and of the energy is also preserved. Section 2 is devoted to the presentation of the general framework of *PIC* codes. The principle of the coalescence process is then introduced in Section 3. Finally, numerical results are given and a conclusion is drawn.

2 Particle Method

Consider several populations of charged particles (ions, electrons, etc.) moving in a 2D or 3D bounded domain Ω. The motion of these charged particles is described in terms of particle distribution functions $f_\alpha(\mathbf{x}, \mathbf{v}, t)$ by the Vlasov equation

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_\alpha + \frac{q_\alpha}{m_\alpha}(\boldsymbol{\mathcal{E}}(\mathbf{x}, t) + \mathbf{v} \times \boldsymbol{\mathcal{B}}(\mathbf{x}, t)) \cdot \nabla_{\mathbf{v}} f_\alpha = S\,,$$

where $\mathbf{x}$ and $\mathbf{v}$ are respectively the position and the velocity of the particles, m_α and q_α denote the mass and the electric charge for each species of particle α, and S is a source term (ionization, collisions, etc.). For the sake of simplicity, we shall consider in the following only one species of charged particles (with an electric charge q, a mass m and a distribution function f). However, this work can easily be applied to several species of particles.

The electromagnetic fields $\boldsymbol{\mathcal{E}}(\mathbf{x}, t)$ and $\boldsymbol{\mathcal{B}}(\mathbf{x}, t)$ are solution of the Maxwell system

$$\frac{\partial \boldsymbol{\mathcal{E}}}{\partial t} - c^2\, \mathbf{curl}\, \boldsymbol{\mathcal{B}} = -\frac{1}{\varepsilon_0}\boldsymbol{\mathcal{J}}, \tag{1}$$

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial t} + \mathbf{curl}\, \boldsymbol{\mathcal{E}} = 0, \tag{2}$$

$$\mathrm{div}\, \boldsymbol{\mathcal{E}} = \frac{1}{\varepsilon_0}\rho, \tag{3}$$

$$\mathrm{div}\, \boldsymbol{\mathcal{B}} = 0, \tag{4}$$

or of any kind of equations approximating the Maxwell system ((quasi-)electro or magnetostatics, Darwin, etc.). The right-hand sides ρ and $\boldsymbol{\mathcal{J}}$ are obtained from the solution f of the Vlasov equation according to:

$$\rho = q \int_{\mathbb{R}^3} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}\ , \quad \boldsymbol{\mathcal{J}} = q \int_{\mathbb{R}^3} \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}. \tag{5}$$

The Vlasov equation is numerically solved by means of a particle method: the distribution function $f(\cdot, \cdot, t)$ is approximated at any time t, by a linear combination of delta functions

$$f(\mathbf{x}, \mathbf{v}, t) \simeq \tilde{f}(\mathbf{x}, \mathbf{v}, t) = \sum_{k=1}^{N} w_k\, \delta(\mathbf{x} - \mathbf{x}_k(t))\delta(\mathbf{v} - \mathbf{v}_k(t))\ , \tag{6}$$

where each term of the sum can be identified with a macro-particle, characterized by its weight w_k, its position $\mathbf{x}_k$ and its velocity $\mathbf{v}_k$, and N is the number of macro-particles. By introducing this particle approximation of f in the expression (5), we obtain a particle approximation of the charge and current densities as follows :

$$\tilde{\rho} = q \int_{\mathbb{R}^3} \tilde{f}(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}\ = q \sum_{k=1}^{N} w_k\, \delta(\mathbf{x} - \mathbf{x}_k(t))\ , \tag{7}$$

$$\tilde{\boldsymbol{\mathcal{J}}} = q \int_{\mathbb{R}^3} \mathbf{v} \tilde{f}(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} = q \sum_{k=1}^{N} w_k \mathbf{v}_k\, \delta(\mathbf{x} - \mathbf{x}_k(t))\ . \tag{8}$$

 ISBN: 978-84-612-9727-6

This is a representation of the right-hand sides ρ and $\boldsymbol{\mathcal{J}}$ in terms of delta functions, that are called *particle variables*.

Since Maxwell's equations are solved by a grid method (Finite Difference, Finite Volume or Finite Element methods), there is a need to define the right-hand sides ρ, $\boldsymbol{\mathcal{J}}$ at the vertices of the mesh used for compute $\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{B}}$. Let us denote by $\mathcal{T}$ the mesh of the computational domain Ω, by $\{a_i\}$ the vertices of $\mathcal{T}$ and by $\lambda_i, 1 \leq i \leq n$ the shape functions related to the field approximation. We define the values of the charge and current densities at the nodes of the mesh by

$$\rho_i = \frac{\displaystyle\int_\Omega \tilde{\rho}(\mathbf{x})\,\lambda_i(\mathbf{x})\,d\mathbf{x}}{\displaystyle\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}} = \frac{q\displaystyle\sum_{k=1}^{N} w_k \lambda_i(\mathbf{x}_k)}{\displaystyle\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}},$$

$$\boldsymbol{\mathcal{J}}_i = \frac{\displaystyle\int_\Omega \tilde{\boldsymbol{\mathcal{J}}}(\mathbf{x})\,\lambda_i(\mathbf{x})\,d\mathbf{x}}{\displaystyle\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}} = \frac{q\displaystyle\sum_{k=1}^{N} w_k \mathbf{v}_k \lambda_i(\mathbf{x}_k)}{\displaystyle\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}}.$$

Since $\displaystyle\sum_i \lambda_i = 1$, this projection procedure preserves the total charge and the total current of particles.

3 A coalescence algorithm

As any other method allowing to control the number of particles, a coalescence process can be global or local. A global approach consists in replacing the set of all the particles in a domain by another set containing a different number of particles, without changing the grid quantities one wants to conserve. In a local approach, the previous procedure is applied cell by cell. One can choose in which element of the mesh the particles are replaced, while preserving the local mesh and/or particle quantities. In this paper, we are interested in a local coalescence process. In each element T of the mesh $\mathcal{T}$, the set of N_T particles $(w_k, \mathbf{x}_k, \mathbf{v}_k)_{k=1,N_T}$ are replaced by a set of N'_T particles $(w'_l, \mathbf{x}'_l, \mathbf{v}'_l)_{l=1,N'_T}$. To preserve the main features of the distribution function, the coalescence process must meet the particle conditions:

- the mass conservation $\displaystyle\sum_{l=1}^{N'_T} w'_l = \sum_{k=1}^{N_T} w_k,$
- the momentum conservation $\displaystyle\sum_{l=1}^{N'_T} w'_l \mathbf{v}'_l = \sum_{k=1}^{N_T} w_k \mathbf{v}_k,$

 ISBN: 978-84-612-9727-6

- and if it is possible, the energy conservation $\sum_{l=1}^{N'_T} w'_l |\mathbf{v}'_l|^2 = \sum_{k=1}^{N_T} w_k |\mathbf{v}_k|^2$,

with the positivity constraint $w'_l > 0, \;\; l = 1, \ldots, N'_T$.

The control of particle number that we present hereafter is general in the sense that it can be applied to increase or decrease the total number of particle, eventually locally in the mesh. In this article, we will expose a coalescence method: when the number of particles in a cell is too large, they are replaced by a lower number of particles. It is also a quite general method, since it can be applied to two dimensional triangle or quadrilateral meshes as well as three dimensional tetrahedral or hexahedral meshes.

The method derived here is devoted to the definition of a coalescence algorithm that conserves the charge and current density in a two-dimensional triangular mesh. For the energy conservation, see [8]. Let us denote by T a chosen fixed triangle of the mesh, and by $\lambda_i, i = 1, 2, 3$ the linear basis functions associated to the nodes of the triangle T. We are interested to conserve the mesh variables and the particle variables as well. For doing that, one basically uses the property that if the local coalescence method conserves the mesh charge and current densities, it also conserves the corresponding particle variables. Then, it is sufficient to consider the mesh quantities.

Now, the control of the particle number that we have developed is based on the the following remark: Following ([9]), a particle approximation of a function can be related to a numerical quadrature formula. Indeed, let $g(\mathbf{x})$ be a given function, and consider a N-points quadrature formula on T

$$\int_T g(\mathbf{x}) \, d\mathbf{x} \simeq \sum_{k=1}^{N} \alpha_k g(\boldsymbol{\xi}_k) \, , \tag{9}$$

where $\boldsymbol{\xi}_k$ and α_k denote the quadrature points and weights. Using (9), we have

$$\int_T f(\mathbf{x}) \psi(\mathbf{x}) \, d\mathbf{x} \simeq \sum_{k=1}^{N} \alpha_k f(\boldsymbol{\xi}_k) \psi(\boldsymbol{\xi}_k) \, , \quad \forall \psi \;\; \text{a test function},$$

which defines a particle approximation of f as $f(x) \simeq \sum_{k=1}^{N} w_k \delta(\mathbf{x} - \boldsymbol{\xi}_k)$, where the weights w_k are given by $w_k = \alpha_k f(\boldsymbol{\xi}_k)$.

The construction of the position and weight of the new particles can now be described in three steps :

1. Find a function g defined on T that satisfies :

$$\int_T g(\mathbf{x}) \lambda_i(\mathbf{x}) \, d\mathbf{x} = \sum_{k=1}^{N} w_k \lambda_i(\mathbf{x}_k) \stackrel{\text{def}}{=} Q_i, \quad i = 1, 2, 3 \, , \tag{10}$$

 ISBN: 978-84-612-9727-6

where N, w_k and $\mathbf{x}_k$ are the number, the weights and the positions of the particles to be replaced.

2. choose on the triangle an accurate N'-points quadrature formula characterized by $(x'_l, \alpha_l), l = 1, \ldots, N'$, its quadrature points and weights. N' and x'_l will be the number and the positions of the new particles, defined in the triangle T.

3. Define the weights of the new particles by applying the quadrature formula, namely
$$\int_T g(\mathbf{x})\lambda_i(\mathbf{x})\, dx \simeq \sum_{l=1}^{N'} \alpha_l g(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l) \quad i = 1, 2, 3 \ ,$$
so we have $w'_l = \alpha_l g(\mathbf{x}'_l)$

With this choice, the mesh charge density conservation property is ensured, and consequently, the particle charge density is also conserved. It can be also proved that this approximation is of the same order as the quadrature formula approximation applied to $\int_T g(\mathbf{x})\lambda_i(\mathbf{x})\, d\mathbf{x}$.

It remains now to choose a well adapted function $g(\mathbf{x})$. Obviously, there is several different possible choices. As $g(\mathbf{x})$ must satisfy three conditions defined by (10), one introduces three unknows in its definition, namely

$$g(\mathbf{x}) = \sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}) \ ,$$

where the three coefficients g_j are the solutions to the 3×3 linear system

$$\sum_{j=1}^{3} g_j \int_T \lambda_j(\mathbf{x})\lambda_i(\mathbf{x})\, dx = Q_i, \quad i = 1, 2, 3$$

and the new weights are defined by $w'_l = \alpha_l \sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}'_l)$. Note that this solution does not necessarily ensure the positivity of the particle weights w'_l, which is a condition that *must* be fulfilled. To ensure this positivity, a sufficient condition consists in choosing a quadrature formula with positive weights ($\alpha_l > 0, \forall l$) and to make sure that $g(\mathbf{x}) > 0, \forall \mathbf{x} \in T$. It can exist some cases where the positivity conditions are not fulfilled, depending on the initial repartition of the particles in the triangle before coalescence. A typical case is when most of the particles have clustered together near a node or an edge. In these cases, the coalescence is not performed. The velocities of the new particles are constructed component by component in the same way as the positions and the weights.

The method we present here was developed to be implemented in a two dimensional *PIC* code where the field solver uses a P^1 finite element method on an unstructured mesh of triangles. As an illustration, we present some results obtained by using the

 ISBN: 978-84-612-9727-6

algorithm constructed from the 7 points Gauss-Hammer formula. We consider a mesh made up with 512 triangles. On every triangle, 30 particles are created. Their positions are chosen at random uniformly on the triangle and their velocities according to a maxwellian distribution.

The results after coalescence show a relative error on the mesh and on the particle charge and current densities conservation of 10^{-8} for a single precision computation. The number of particles before coalescence is 15360, the number after is 4389, and there are 35 triangles where the coalescence is forbidden.

4 Conclusion

We have proposed a way of controlling the particle number in particle simulations. A first application of this method was exposed for coalescing particle in *PIC* codes, which gives interesting results. Particular attention was paid on the conservation of the mesh and particle charge and current densities. Moreover, the number of particles after coalescence can be controlled by choosing an appropriate quadrature formula. We think that this approach can be useful in a variety of fields, as for instance in chemical processing. This certainly requires to extend the method to others type of meshes: quadrilaterals in the 2D case, tetrahedra or hexahedra in the 3D one. Finally it would also be interesting to try other types of interpolation functions, other quadrature formulae, such as monte-carlo methods for other types of application.

References

[1] C.K. Birdsall and A.B. Langdon, *Plasmas Physics via Computer Simulation* New York: Mac.Graw-Hill, 1985.

[2] R.W. Hockney and J.W. Eastwood, *Computer simulation using particles*, Adam Hilger imprint by IOP Publishing Ltd, 1988.

[3] P. Degond and F.-J.Mustieles, *A deterministic approximation of diffusion equations using particles* SIAM J. Sci. Stat. Comput **11** (1990) 293–310.

[4] P. Degond and S. Mas-Gallic, *The weighted particle method for convection-diffusion equations*, Math. Comput. **53** (1989) 509–525.

[5] G. Coppola, S.J. Sherwin and J.Peiro, *Nonlinear particle tracking for high-order elements* J Comput Phys. **172** (2001) 356–386.

[6] N.G. Deen, M. van Sint Annaland, M.A. van der Hoef and J.A.M. Kuipers, *Review of discrete particle modeling of fluidized beds*, Chem. Eng. Sc. **62** (2007) 28–44.

 ISBN: 978-84-612-9727-6

[7] Y.T. Feng, K. Han and D.R.J. Owen, *Discrete element simulation of the dynamics of high energy planetary ball milling processes*, Mater. Sc. Eng. **A 375-377** (2004) 815–819.

[8] F. Assous, T. Pougeard-Dulimbert and J.Segré, *A New Method for Coalescing Particles in PIC Codes*, J. Comput. Phys. **187** (2003) 550–571.

[9] P.A. Raviart, *An Analysis of Particle Methods* Springer Verlag, Berlin,1985.

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009 30 June, 1–3 July 2009.

Fast simulation of one-layer shallow water systems using CUDA architectures

Marc de la Asunción[1], José M. Mantas[1] and Manuel J. Castro[2]

[1] *Depto. Lenguajes y Sistemas Informáticos, Universidad de Granada*

[2] *Depto. Análisis Matemático, Universidad de Málaga*

emails: marc@correo.ugr.es, jmmantas@ugr.es, castro@anamat.cie.uma.es

Abstract

The numerical solution of shallow water systems is useful for several applications related to geophysical flows but the dimensions of the domains impose great demands of computing power. This fact suggests the use of powerful accelerators to obtain numerical results in reasonable times. This paper addresses how to speed up considerably the numerical solution of a first order well-balanced finite volume scheme for 2D one-layer shallow water systems by using modern Graphics Processing Units (GPUs) supporting the NVIDIA CUDA programming model. An algorithm which describes how the potential data parallelism of this method can be exploited by using the CUDA model is presented and implemented on GPUs. Numerical experiments show the high efficiency of this CUDA solver in comparison with an efficient CPU implementation of the solver and with respect to a previously existing GPU implementation of the solver based on a shading language.

1 Introduction

The shallow water equations, formulated under the form of a conservation law with source terms, are widely used to model the flow of a layer of fluid under the influence of gravity forces. The numerical solution of these models is useful for several applications related to geophysical flows, such as the simulation of rivers, channels or dambreak problems. These simulations impose great demands on computing power due to the dimensions of the domains (space and time) and extremely efficient high performance solvers are required to solve and analyze these problems in reasonable execution times.

Since the numerical solution of shallow water systems exhibits a lot of exploitable parallelism, several works have dealt with the acceleration of these simulations by using parallel hardware. An interesting numerical scheme to simulate shallow water systems and an efficient parallel implementation of this scheme for a PC cluster are presented in [1]. This parallel implementation has been improved in [2] by using SSE-optimized

software modules. Although these improvements have made it possible to obtain results in faster computational times, the simulations still require too much runtime despite the efficient use of all the resources of a powerful PC cluster.

Modern Graphics Processing Units (GPUs) offer hundreds of processing units optimized for massively performing floating point operations in parallel and can be a cost effective way to speed up the numerical solution of several mathematical models in science and engineering (see [9] for a review of the topic). Moreover, in computationally intensive tasks like the one considered in this paper, these architectures are able to obtain a substantially higher performance than can a general purpose CPU.

There are previous proposals to port shallow water numerical solvers to GPU platforms In [7], a explicit central-upwind scheme is implemented on a NVIDIA GeForce 7800 GTX card to simulate the one-layer shallow water system and a speedup from 15 to 30 is achieved with respect to a CPU implementation. An efficient implementation of the numerical scheme presented in [1] on GPUs is described in [8], obtaining two orders of magnitude speedup on a NVIDIA Geforce 8800 Ultra card with respect to a monoprocessor implementation. These previous proposals are based on the OpenGL graphics application programming interface [10] and the Cg shading language [3].

The use of graphics-specific programming languages and interfaces complicates the programming of GPUs in scientific applications for those who were unfamiliar with computer graphics and the code obtained is difficult to understand and maintain.

Recently, NVIDIA has developed the CUDA programming toolkit [5] consisting in an extension of the C language which facilitates the programming of GPUs for general purpose applications by preventing the programmer to deal with the graphics details of the GPU. CUDA supports multiple memory address spaces and its programming model maps well to the GPU architecture enabling the efficient exploitation of the hardware.

Our goal is to accelerate the numerical solution of shallow water systems by using GPUs supporting CUDA. In particular, the one layer shallow water numerical solver which is parallelized in [1] and [8] has been adapted to the CUDA architecture to obtain much better response times with a lower programming effort.

The next section describes the underlying numerical scheme and its main sources of parallelism. A brief introduction to the CUDA architecture and programming model is presented in Section 3. The design and implementation of the CUDA version of the numerical solver is described in Section 4. Section 5 presents and analyzes the results obtained when the solver is applied to several meshes using several GPUs. Finally, Section 6 summarizes the main conclusions and presents the lines for further work.

2 Numerical Scheme and parallelism sources

2.1 The one-layer shallow water system

The one-layer shallow water system is a system of conservation laws with source terms which models the flow of a homogeneous fluid shallow layer that occupies a bounded subdomain $D \subset \mathbb{R}^2$ under the influence of a gravitational acceleration g. The system has the following form:

 ISBN: 978-84-612-9727-6

$$\frac{\partial W}{\partial t} + \frac{\partial F_1}{\partial x}(W) + \frac{\partial F_2}{\partial y}(W) = \begin{bmatrix} 0 \\ gh \\ 0 \end{bmatrix} \frac{\partial H}{\partial x} + \begin{bmatrix} 0 \\ 0 \\ gh \end{bmatrix} \frac{\partial H}{\partial y} \tag{1}$$

being

$$W = \begin{pmatrix} h \\ q_x \\ q_y \end{pmatrix}, \qquad F_1(W) = \begin{bmatrix} q_x \\ \frac{q_x^2}{h} + \frac{1}{2}gh^2 \\ \frac{q_x q_y}{h} \end{bmatrix}, \qquad F_2(W) = \begin{bmatrix} q_y \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{1}{2}gh^2 \end{bmatrix}.$$

where $h(x, y, t) \in \mathbb{R}$ denotes the thickness of the water layer at point (x, y) at time t, $H(x, y)$ is the depth function measured from a fixed level of reference and $q(x, y, t) = (q_x(x, y, t), q_y(x, y, t)) \in \mathbb{R}^2$ is the mass-flow of the water layer at point (x, y) at time t.

2.2 The numerical scheme

Our problem consists in studying the time evolution of $W(x, y, t)$ satisfying System (1).

In accordance with the description given in [1], the discretization of System (1) by means of a Finite Volume scheme is presented. First, the computational domain D is divided into L discretization cells or finite volumes, $V_i \subset \mathbb{R}^2$, which are assumed to be quadrangles. Given a finite volume V_i, $N_i \in \mathbb{R}^2$ is the centre of V_i, $\aleph_i$ is the set of indexes j such that V_j is a neighbour of V_i; Γ_{ij} is the common edge of two neighbour cells V_i and V_j, and $|\Gamma_{ij}|$ is its length; $\boldsymbol{\eta}_{ij} = (\eta_{ij,x}, \eta_{ij,y})$ is the unit vector which is normal to the edge Γ_{ij} and points towards the cell V_j (see Figure 1).

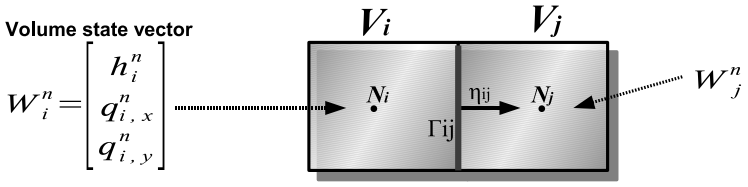


Figure 1: Finite volumes

Assume that the approximations at time t^n, W_i^n, have already been calculated. To advance in time, with Δt being the time step, the following numerical scheme is applied (see [1] for more details about the numerical scheme):

$$W_i^{n+1} = W_i^n - \frac{\Delta t}{|\, V_i \,|} \sum_{j \in \aleph_i} |\, \Gamma_{ij} \,|\, F_{ij}^-, \tag{2}$$

being

$$\begin{aligned} F_{ij}^- &= P_{ij}^-(A_{ij}(W_j^n - W_i^n) - S_{ij}(H_j - H_i)), \\ P_{ij}^- &= \frac{1}{2}\mathcal{K}_{ij} \cdot (I - \mathrm{sgn}(\mathcal{D}_{ij})) \cdot \mathcal{K}_{ij}^{-1}, \end{aligned}$$

 ISBN: 978-84-612-9727-6

where $|V_i|$ is the area of V_i , $H_l = H(N_l)$ with $l = 1, \ldots, L$, $A_{ij}^n \in \mathbb{R}^{3\times3}$ and $S_{ij}^n \in \mathbb{R}^3$ depend on W_i^n and W_j^n, D_{ij}^n is a diagonal matrix whose coefficients are the eigenvalues of A_{ij}^n and the columns of $K_{ij}^n \in \mathbb{R}^{3\times3}$ are the associated eigenvectors (see [1] for more details).

In order to compute the n-th time step, the following condition can be used [1]):

$$\Delta t^n = \min_{i=1,\ldots,L} \left\{ \left[\frac{\sum_{j\in\aleph_i} \mid \Gamma_{ij} \mid\mid\mid D_{ij}^n \mid\mid_\infty}{2\gamma \mid V_i \mid} \right]^{-1} \right\} \tag{3}$$

where γ, $0 < \gamma \leq 1$, is the CFL (Courant-Friedrichs-Lewy) parameter.

2.3 Parallelism sources

Figure 2 shows a graphical description of the main sources of parallelism obtained from the mathematical description of the numerical scheme. In this figure, the main calculation phases are identified with circled numbers and the main sources of data parallelism are indicated. A `Parfor each` <*data_item*> block denotes that the calculation affected by it can be performed simultaneously for each data item of a set (the data items can represent the volumes or the edges of the mesh). The arrows connecting two computing phases represent data dependences between the two phases.

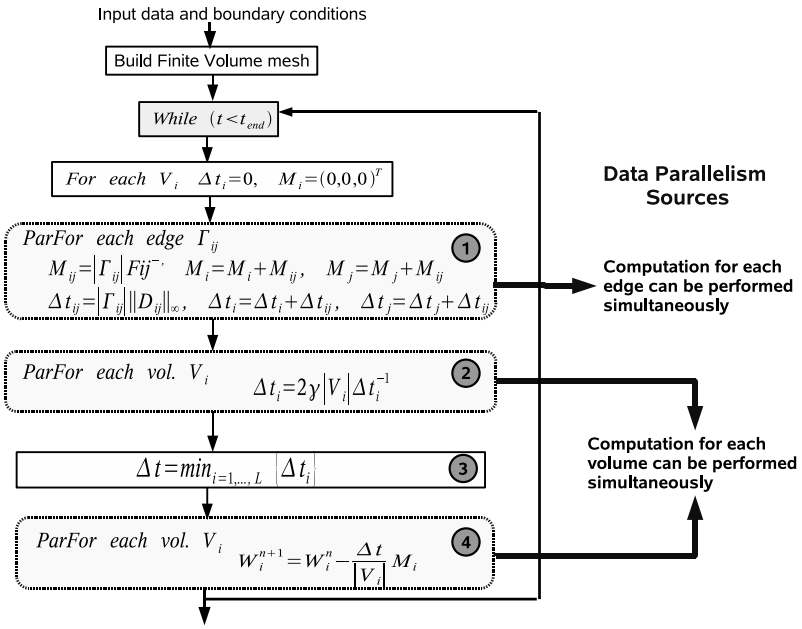


Figure 2: Main calculation phases in the parallel algorithm

Initially, the finite volume mesh must be constructed from the input data. Then the time stepping process is repeated until the final simulation time is reached. At the $(n + 1)$-th time step, Equation (2) must be evaluated to update the state of each

cell. Each of the main calculation phases of the evaluation present a high degree of parallelism because the computation at each edge or volume is independent with respect to that performed or associated to the other edges or volumes:

1. **Edge-based calculations**: Two calculations must be performed for each edge Γ_{ij} communicating two cells V_i and V_j ($i, j \in \{1, \ldots, L\}$):
   a) Vector $M_{ij} =\mid \Gamma_{ij} \mid F_{ij}^{-} \in \mathbb{R}^3$ must be computed as the contribution of each edge to the sum associated to the adjacent cells V_i and V_j (see Equation (2)). This contribution can be computed independently for each edge and must be added to the partial sums associated to each cell (M_i and M_j).
   b) The value $\Delta t_{ij} =\mid \Gamma_{ij} \mid\mid\mid D_{ij}^{n} \mid\mid_{\infty}$ can be computed independently for each edge and added to the partial sums associated to each cell (Δt_i and Δt_j) as an intermediate step to compute the n-th time step Δt^n (see Equation (3)).
2. **Computation of the local Δt for each volume**: For each volume V_i, the value of Δt_i is modified to compute the local Δt per volume according to Equation (3). In the same way, the computation for each volume can be performed in parallel.
3. **Computation of Δt^n**: The minimum of all the local Δt values previously obtained for each volume must be computed.
4. **Computation of W_i^{n+1}**: The $(n+1)$-th state of each volume (W_i^{n+1}) must be approximated from the n-th state using the data computed in the previous phases. This phase can also be completed in parallel (see Figure 2).

Since the numerical scheme exhibits a high degree of potential data parallelism, it is good candidate to be implemented on CUDA architectures.

3 CUDA Architecture and Programming Model

According to the CUDA framework, both the CPU and the GPU maintain their own memory. It is possible to copy data from CPU memory to GPU memory and vice versa.

The GPU is formed by a set of Single Instruction Multiple Data (SIMD) multiprocessors, each one having 8 processors (see Figure 3). At any clock cycle, each processor of the multiprocessor executes the same instruction, but operates on different data.

A function executed on the GPU is called a *kernel*. A kernel is executed forming a grid of thread blocks, which run logically in parallel. All blocks and threads have spatial indices, so that the spatial position of each thread could be identified programmatically. Each thread block runs in a single multiprocessor. A *warp* is the number of threads that can run concurrently in a multiprocessor. Warp size is 32 threads. Each block is split into warps, and periodically a scheduler switches from one warp to another. This allows to hide the high latency when accessing the GPU memory, since some threads can continue their execution while other threads are waiting.

A thread that executes on the GPU has access to the following memory spaces:

 ISBN: 978-84-612-9727-6

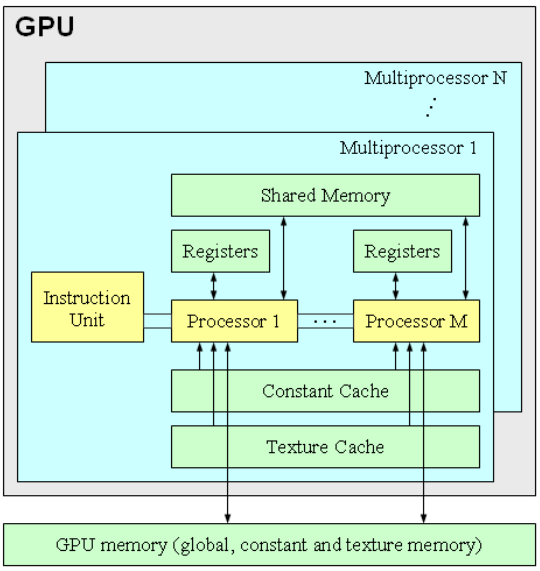


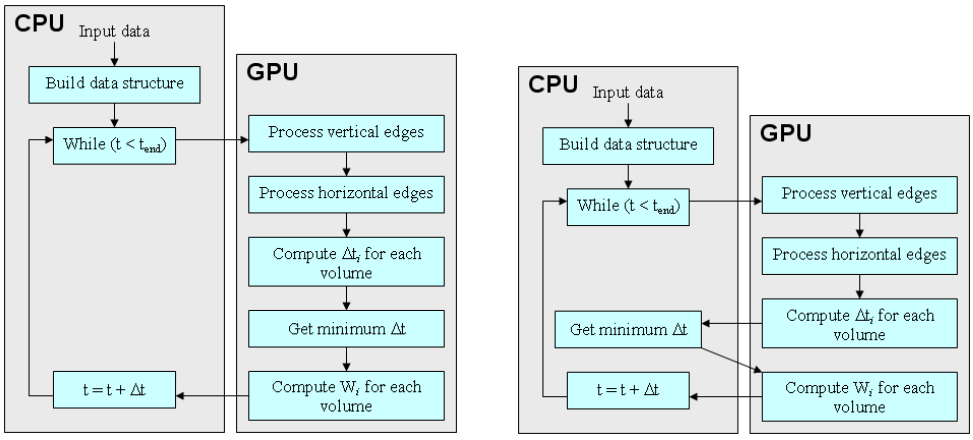
Figure 3: GPU architecture.

- Registers: Each thread has its own readable and writable registers.
- Shared memory: Shared by all threads of a block. Its size is only 16 KB. Readable and writable only from the GPU. It is faster than global memory.
- Global memory: Shared by all blocks of a grid. Readable and writable from CPU and GPU. It is slow due to its high latency.
- Constant memory: Shared by all blocks of a grid. Readable from GPU and writable from CPU. Its size is 64 KB and it is cached, making it faster than global memory if the data is in cache. Cache size is 8 KB per multiprocessor.
- Texture memory: Shared by all blocks of a grid. Readable from GPU and writable from CPU. It is cached and optimized from 2D spatial locality, i.e. it is especially suited for each thread to access its closer neighborhood. Cache size varies between 6 and 8 KB per multiprocessor.

In GT200 architecture, each multiprocessor has 16384 registers, which are split and assigned to the threads that execute concurrently on that multiprocessor. Therefore, the number of registers used by each thread is an important factor to achieve the maximum usage of the GPU. For example, if a kernel (i.e. a thread) uses 32 registers, only $16384/32 = 512$ threads can run concurrently on a multiprocessor. Above 16 registers, the more registers used, the less threads can run concurrently.

Some other tips are suggested in the CUDA documentation [4] to achieve maximum performance, for example, the number of threads per block should be a multiple of 64, and the recommended is 192 or 256 if a thread does not require many registers.

 ISBN: 978-84-612-9727-6



(a) (b)

Figure 4: General steps of the parallel algorithm implemented in CUDA. (a) When there are more than a particular number of volumes, (b) Otherwise.

4 Algorithmic details of the CUDA version

In this section we describe the parallel algorithm we have developed and its implementation in CUDA. We consider problems consisting in a bidimensional regular finite volume mesh. The general steps of the parallel algorithm are depicted in Figure 4. Each processing step executed on the GPU is assigned to a CUDA kernel. Depending on the total number of volumes, the process of obtaining the minimum of the local Δt_i of the volumes ($1 \leq i \leq L$) is performed on the CPU or on the GPU.

Next, we describe in detail each step:

- *Build data structure*: In this step, the data structure that will be used on the GPU is built. For each volume, we store its initial state (h, q_x and q_y) and its depth H. We define an array of `float4` elements, where each element represents a volume and contains the former parameters. This array is stored as a texture. Each edge only needs the data of its two adjacent volumes, that is, each thread only needs two accesses to global memory for getting the needed data. With such a few memory reads per thread, texture memory is a better option than shared memory, which has more sense when each thread has to use many global memory elements, and each thread of a block loads a small part of all these elements.

  The area of the volumes and the length of the vertical and horizontal edges are precalculated and passed directly to the CUDA kernels that need them.

  We can know at runtime if an edge or volume is frontier or not and the value of the normal $\boldsymbol{\eta}_{ij}$ of an edge by checking the position of the thread in the grid.

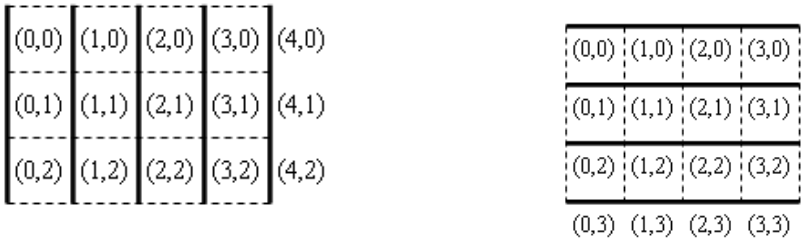
- *Process vertical edges* and *process horizontal edges*: The key of this algorithm is the division of the edge processing into vertical and horizontal edge processing. For vertical edges, $\eta_{ij,y} = 0$ and therefore all the operations where this term

 ISBN: 978-84-612-9727-6

takes part can be discarded, increasing efficiency. Similarly, for horizontal edges, $\eta_{ij,x} = 0$ and all the operations where this term takes part can be avoided.

In the vertical edge processing, each thread represents a vertical edge, while in the horizontal edge processing, each thread represents a horizontal edge. Figure 5 shows how the threads are spatially distributed in a block of the grid. In both vertical and horizontal edge processing, each thread computes the contribution of the edge to their adjacent volumes, that is, it performs the following computations:

- $|\Gamma_{ij}|F_{ij}^{-}$. This is the edge contribution for obtaining the next state of its adjacent volumes (see Equation 2).
- $|\Gamma_{ij}|||D_{ij}||_{\infty}$. This is the edge contribution for obtaining the local value of Δt_i of its adjacent volumes (see Equation 3).



(a) Vertical edge processing (b) Horizontal edge processing

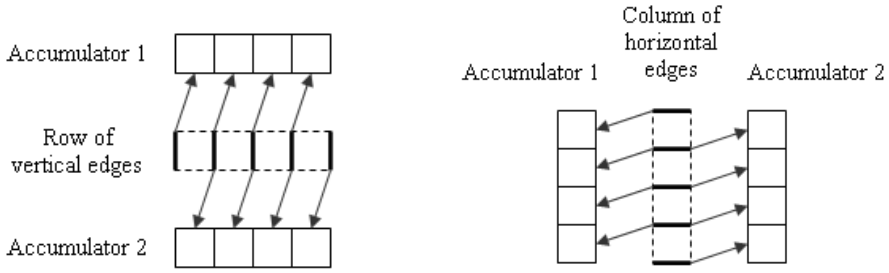
Figure 5: 4×3 finite volume mesh showing the spatial distribution of the threads in a block of the grid. For each thread, its (x, y) position is indicated.

A crucial aspect that must be defined is the way the edges (threads) synchronize each other when contributing to a particular volume, that is, how these contributions are summed for each volume. An internal edge contributes to its two adjacent volumes, while a frontier edge only contributes to its adjacent volume Instead of using a global data structure and a critical section in the thread for summing all the contributions for each volume, which would reduce performance, we have solved this issue by creating two accumulators, each one being an array of `float4` elements. The size of each accumulator is the total number of volumes. Each element of the accumulators contains four `float` values corresponding to the former contributions (a 3×1 vector and a `float` value). Then, in the processing of vertical edges:

- Each vertical edge, except the right frontier, writes in the first accumulator the contribution to its right volume.
- Each vertical edge, except the left frontier, writes in the second accumulator the contribution to its left volume.

Next, the processing of horizontal edges is performed in an analogous way. Figure 6 shows this process graphically. At this point, we only have to sum the corres-

ponding positions in the first and second accumulator to get the final contribution term for each volume. This is done in steps described next.



(a) Vertical edge processing (b) Horizontal edge processing

Figure 6: Computing the sum of the contributions of the edges of each volume.

- *Compute Δt_i for each volume*: In this step, each thread represents a volume and computes the local Δt_i of the volume i, $1 \leq i \leq L$. Each thread sums the contribution value (a `float` value) stored in the position of the first accumulator corresponding to the volume, and the contribution value stored in the same position of the second accumulator, thus getting the final contribution of the edges to the local Δt_i of the volume. Let S_i be this value. Then, Δt_i is obtained as: $\Delta t_i = 2\gamma \,|\, V_i \,|\, S_i^{-1}$ (see Equation 3).

- *Get minimum Δt*: This step finds the minimum of the local Δt_i of the volumes. Depending on the number of volumes, it is performed on the CPU or the GPU. If the number of volumes is greater than a threshold value (which is chosen experimentally), we apply a reduction algorithm on the GPU. Otherwise, the array of the Δt_i values is copied from GPU to CPU and the computation is carried out on the CPU, since it is more efficient. The reduction algorithm applied to perform this step on GPU is the kernel 7 (the most optimized one) of the reduction sample included in the CUDA Software Development Kit [5].

- *Compute W_i for each volume*: In this step, each thread represents a volume and updates the state W_i of the volume i, $1 \leq i \leq L$. Each thread sums the contribution value (a 3×1 vector) stored in the position of the first accumulator corresponding to the volume, and the contribution value stored in the same position of the second accumulator, thus getting the final contribution of the edges to the state of the volume. Let M_i be this value. Then, the new W_i is obtained as: $W_i^{n+1} = W_i^n - \frac{\Delta t}{|V_i|} M_i$ (see Equation 2):

5 Experimental Results

A problem consisting in an unsteady flow in a 1 m $\times$ 10 m rectangular channel is considered in order to compare the performance of our CUDA implementation and a

 ISBN: 978-84-612-9727-6

GPU solver based on Cg (which is described in [8]). For this test problem, the depth function is given by $H(x,y) = 1 - \frac{\cos(2\pi x)}{2}$ and the initial condition is:

$$W_i^0(x,y) = \begin{pmatrix} h^0(x,y) \\ 0 \\ 0 \end{pmatrix}, \quad \text{where} \quad h^0(x,y) = \begin{cases} H(x,y) + 2 & \text{if } x < 5 \\ H(x,y) & \text{otherwise} \end{cases}$$

Six uniform meshes of the domain, Q_k, $k = 0, \ldots, 5$, are constructed such that the number of volumes of the mesh Q_k is $(2^k \cdot 100) \times (2^k \cdot 10)$.

The numerical scheme is run in the time interval [0, 5] except for the mesh Q_5, which is solved for the time interval [0, 0.1]. CFL parameter is $\gamma = 0.9$ and wall boundary conditions ($\mathbf{q} \cdot \eta = 0$) are considered.

We also have implemented a serial CPU version of the CUDA algorithm, that is, vertical and horizontal edges are also processed separately. The CPU program is implemented in C++ and the Eigen library [6] is used for operating with matrices. We have used the `double` and `float` data types in CPU and GPU, respectively.

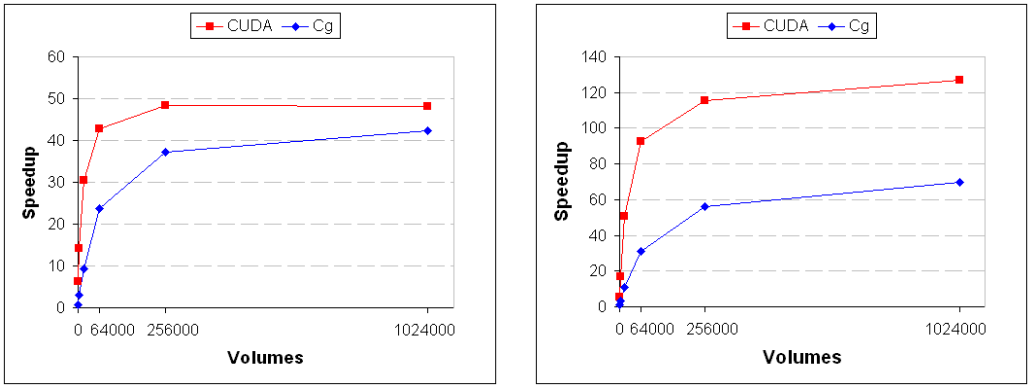
All the programs were executed on a Core2 Quad Q9550 with 2 GB RAM. Graphics cards used were a GeForce 8800 Ultra and a GeForce GTX 280. The threshold of 4096 volumes has been chosen to compute the minimum Δt on the GPU. Figure 8 shows the evolution of the fluid for our test problem. Table 1 shows the execution times in seconds for all the meshes and programs and Figure 7 shows graphically the speedup obtained in the Cg and CUDA implementations with respect to the CPU implementation. We can see that the execution times of the CUDA implementation outperform that of Cg in all cases with both graphic cards. For big problems, using a 8800 Ultra, the CUDA and Cg programs achieve a speedup with respect to the CPU version of approximately 48 and 40, respectively. Using a GTX 280, for big problems, the performance gain reached by the CUDA program almost doubles that of Cg, achieving a speedup in the 3200×320 example of more than 120 with respect to the CPU version.

				8800 Ultra		GTX 280	
Mesh	Volumes	t_{end}	CPU	Cg	CUDA	Cg	CUDA
Q_0	100×10	5.0	0.29	0.37	0.046	0.34	0.052
Q_1	200×20	5.0	2.28	0.77	0.16	0.74	0.13
Q_2	400×40	5.0	18.24	1.95	0.6	1.70	0.36
Q_3	800×80	5.0	162.3	6.82	3.8	5.20	1.75
Q_4	1600×160	5.0	1354.0	36.38	28.0	24.03	11.75
Q_5	3200×320	0.1	218.3	5.15	4.53	3.12	1.72

Table 1: Execution times in seconds for all the meshes and programs.

We have also compared the numerical solutions obtained in the CPU program with double precision and the CUDA program with simple precision. Table 2 shows the L1 norm of the difference between the solutions obtained with both implementations at time $t = 5.0$ for all meshes except Q_5. As it can be seen, the order of magnitude of the L1 norm is acceptable and is maintained constant when the mesh size is increased, which confirms the accuracy of the numerical solutions computed on the GPU.

 ISBN: 978-84-612-9727-6



(a) Using a GeForce 8800 Ultra (b) Using a GeForce GTX 280

Figure 7: Speedup obtained with the Cg and CUDA implementations with respect to the CPU implementation in all meshes.

Mesh	Volumes	t_{end}	diff h	diff q_x	diff q_y
Q_0	100×10	5.0	$5.79 \cdot 10^{-6}$	$2.09 \cdot 10^{-5}$	$2.31 \cdot 10^{-6}$
Q_1	200×20	5.0	$8.01 \cdot 10^{-6}$	$2.56 \cdot 10^{-5}$	$2.16 \cdot 10^{-6}$
Q_2	400×40	5.0	$7.77 \cdot 10^{-6}$	$2.35 \cdot 10^{-5}$	$2.32 \cdot 10^{-6}$
Q_3	800×80	5.0	$5.80 \cdot 10^{-6}$	$2.12 \cdot 10^{-5}$	$2.24 \cdot 10^{-6}$
Q_4	1600×160	5.0	$7.48 \cdot 10^{-6}$	$2.50 \cdot 10^{-5}$	$2.33 \cdot 10^{-6}$

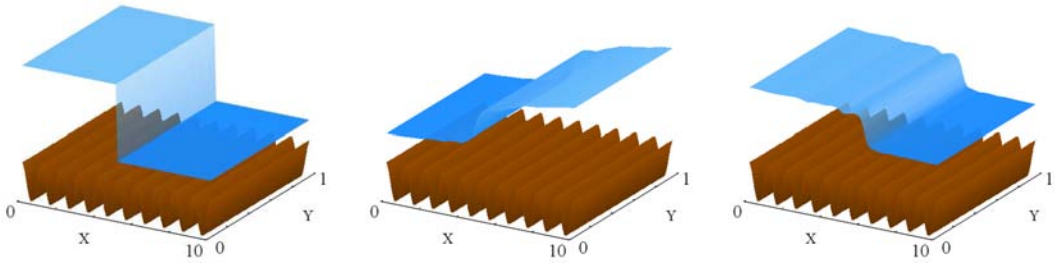
Table 2: L1 norm of the difference between the numerical results obtained with the CPU and CUDA implementations.

6 Conclusions and further work

An efficient first order well-balanced finite volume solver for one-layer shallow water systems, capable of efficiently exploiting the parallel processing power of GPUs, has been derived using the CUDA framework. This solver implements optimization techniques to parallelize efficiently the numerical scheme on the CUDA architecture. Simulations carried out on a GeForce GTX 280 card were found to be up to two orders of magnitude faster than an efficient CPU version of the solver for big-size uniform problems and twice faster than a GPU version based on a graphics-specific language. These simulations also show that the numerical solutions obtained with the solver are accurate enough for practical applications. As further work, we propose to extend the strategy to enable efficient simulations on irregular and nonstructured meshes and to address the simulation of two-layer shallow water systems.

Acknowledgements

J. Mantas acknowledges partial support from DGI-MEC project MTM2008-06349-C03-03. M. Castro acknowledges partial support from DGI-MEC project MTM2006-08075.



(a) t = 0.0 (b) t = 2.5 (c) t = 5.0

Figure 8: Graphical representation of the fluid evolution at different time instants.

References

[1] M. J. Castro, J. A. García-Rodríguez, J. M. González-Vida, C. Parés, *A parallel 2d finite volume scheme for solving systems of balance laws with non-conservative products: Application to shallow flows*, Comput. Meth. Appl. Mech. Eng., 195 (2006) 2788-2815.

[2] M. J. Castro, J. A. García-Rodríguez, J. M. González-Vida, C. Parés, *Solving shallow-water systems in 2D domains using Finite Volume methods and multimedia SSE instructions*, J. Comput. Appl. Math., 2007.

[3] R. Fernando, M. J. Kilgard, *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*, Addison-Wesley (2003).

[4] NVIDIA, *NVIDIA CUDA Programming Guide Version 2.1.* (2008). http://www.nvidia.com/object/cuda_develop.html.

[5] NVIDIA, *CUDA home page*, http://www.nvidia.com/object/cuda_home.html.

[6] *Eigen 2.0.1.*, http://eigen.tuxfamily.org.

[7] T. R. Hagen, J. M. Hjelmervik, K.-A. Lie, J. R. Natvig, M. Ofstad Henriksen, *Visual simulation of shallow-water waves*, Simul. Model. Pract. Theory, 13 (2005) 716-726.

[8] M. Lastra, J. M. Mantas, C. Ureña, M. J. Castro and J. A. García, *Simulation of Shallow-Water systems using Graphics Processing Units*, Submitted to Mathematics in Industry (2009).

[9] M. Rumpf, R. Strzodka, *Graphics Processor Units: New Prospects for Parallel Computing*, Lecture Notes in Computational Science and Engineering, 51 (2006) 89-121 .

[10] D. Shreiner, M. Woo, J. Neider, T.Davis, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2.1*, Addison-Wesley Professional (2007).

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

A residual-based a posteriori error estimator for an augmented mixed method in elasticity

Tomás P. Barrios[1], Edwin M. Behrens[2] and María González[3]

[1] *Departamento de Matemática y Física Aplicadas, Universidad Católica de la Santísima Concepción*

[2] *Dpto. de Ingeniería Civil, Universidad Católica de la Santísima Concepción*

[3] *Departamento de Matemáticas, Universidad de A Coruña*

emails: tomas@ucsc.cl, ebehrens@ucsc.cl, mgtaboad@udc.es

Abstract

We present an a posteriori error estimator of residual type for the augmented mixed finite element method introduced in [12] for the linear elasticity system in the plane with mixed boundary conditions. We proved in [3] that this a posteriori error estimator is reliable and efficient. We provide numerical experiments that illustrate the performance of the corresponding adaptive algorithm and confirm the theoretical properties of the a posteriori error estimator.

Key words: Mixed finite element, augmented formulation, a posteriori error estimator, linear elasticity
MSC 2000: AMS 65N15, 65N30, 65N50, 74B05

1 Introduction

Recently, a new stabilized mixed finite element method for the linear elasticity problem in the plane was presented and analyzed in [12]. The approach there is based on the introduction of suitable Galerkin least-squares terms arising from the constitutive and equilibrium equations, and from the relation defining the rotation in terms of the displacement. The resulting augmented variational formulations and the associated Galerkin schemes are well posed for appropriate values of the stabilization parameters, and the latter becomes locking-free for the pure displacement problem and asymptotically locking-free in case of mixed boundary conditions. In particular, the discrete scheme allows the use of Raviart-Thomas spaces of the lowest order to approximate the stress tensor, piecewise linear elements for the displacement, and piecewise constants for the rotation. When mixed boundary conditions are considered, the essential one (Neumann) is imposed weakly, which yields the introduction of the trace of the

displacement as a Lagrange multiplier. This unknown is approximated by piecewise linear elements on an independent partition of the Neumann boundary, whose mesh size needs to satisfy a compatibility condition with the mesh size of the triangulation of the domain.

Motivated by the competitive character of the augmented mixed finite element scheme introduced in [12], we derived in [5] an a posteriori error estimator of residual type for the case of pure Dirichlet boundary conditions, and proved its reliability and efficiency. Recently, we extended the analysis to the case of mixed boundary conditions (see [3]). Here, we present the a posteriori error estimator derived in [3] and provide some numerical experiments that confirm the theoretical properties of the estimator and illustrate the ability of the corresponding adaptive algorithm to recognize the singularities and large stress regions of the solution. First, we recall the continuous and discrete augmented formulations proposed in [12] for the elasticity problem with mixed boundary conditions.

2 The augmented mixed finite element method

Let $\Omega \subset \mathbb{R}^2$ be a bounded and simply connected domain with Lipschitz-continuous boundary Γ, and let Γ_D and Γ_N be two disjoint subsets of Γ such that Γ_D has positive measure and $\Gamma = \bar{\Gamma}_D \cup \bar{\Gamma}_N$. Given a volume force $\mathbf{f} \in [L^2(\Omega)]^2$ and a traction $\mathbf{g} \in [H^{-1/2}(\Gamma_N)]^2$, we consider the following problem: determine the displacement vector field $\mathbf{u}$ and the symmetric stress tensor field $\boldsymbol{\sigma}$ of a linear elastic material occupying the region Ω and satisfying

$$\left\{ \begin{array}{rcll} \boldsymbol{\sigma} & = & \mathcal{C}\,\mathbf{e}(\mathbf{u}) & \text{in } \Omega \\ -\mathbf{div}(\boldsymbol{\sigma}) & = & \mathbf{f} & \text{in } \Omega \\ \mathbf{u} & = & \mathbf{0} & \text{on } \Gamma_D \\ \boldsymbol{\sigma}\mathbf{n} & = & \mathbf{g} & \text{on } \Gamma_N \end{array} \right.$$

where $\mathbf{e}(\mathbf{u}) := \frac{1}{2}\,(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathtt{t}})$ is the strain tensor of small deformations, and $\mathcal{C}$ is the elasticity tensor determined by Hooke's law, that is,

$$\mathcal{C}\,\boldsymbol{\zeta} \;:=\; \lambda\,\mathrm{tr}(\boldsymbol{\zeta})\,\mathbf{I} \;+\; 2\,\mu\,\boldsymbol{\zeta} \qquad \forall\,\boldsymbol{\zeta} \,\in\, [L^2(\Omega)]^{2\times 2}\,,$$

where $\mathbf{I}$ is the identity matrix of $\mathbb{R}^{2\times 2}$ and $\lambda,\, \mu \,>\, 0$ are the Lamé parameters. We denote by $\mathbf{n}$ the unit outward normal to Γ.

The augmented variational formulation proposed in [12] relies on the mixed method of Hellinger and Reissner, which provides simultaneous approximations of the displacement $\mathbf{u}$ and the stress tensor $\boldsymbol{\sigma}$. The symmetry of $\boldsymbol{\sigma}$ is imposed weakly, introducing the rotation $\boldsymbol{\gamma} := \frac{1}{2}(\nabla \mathbf{u} - (\nabla \mathbf{u})^{\mathtt{t}})$ as an additional unknown (see [1]). The Neumann boundary condition is also imposed in a weak sense and the Lagrange multiplier $\boldsymbol{\xi} := -\mathbf{u}|_{\Gamma_N}$ is introduced as the associated unknown (see [2]). The corresponding dual-mixed variational formulation satisfies the hypotheses of the Babuška-Brezzi theory. However, it is difficult to derive explicit finite element subspaces yielding a stable discrete scheme. In particular, when mixed boundary conditions with non-homogeneous Neumann data

 ISBN: 978-84-612-9727-6

are imposed, the PEERS elements can be applied but, since the Neumann boundary condition becomes essential, they yield a non-conforming Galerkin scheme. This was one of the main motivations for the augmented formulation from [12].

There, the usual dual-mixed variational formulation is enriched with residuals arising from the constitutive and equilibrium equations, and from the relation defining the rotation as a function of the displacement. The resulting augmented variational formulation reads: find $((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),\boldsymbol{\xi}) \in \mathbf{H}\times\mathbf{Q}$ such that

$$\begin{array}{rcll} A((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),(\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta})) + B((\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta}),\boldsymbol{\xi}) & = & F(\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta}) & \forall\,(\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta}) \in \mathbf{H}\,, \\ B((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),\boldsymbol{\chi}) & = & G(\boldsymbol{\chi}) & \forall\,\boldsymbol{\chi} \in \mathbf{Q}\,, \end{array} \tag{1}$$

where $\mathbf{H} := H(\mathbf{div};\Omega)\times[H^1_{\Gamma_D}(\Omega)]^2\times[L^2(\Omega)]^{2\times 2}_{\mathtt{skew}}$ and $\mathbf{Q} := [H^{1/2}_{00}(\Gamma_N)]^2$. The bilinear forms $A:\mathbf{H}\times\mathbf{H}\to\mathbb{R}$ and $B:\mathbf{H}\times\mathbf{Q}\to\mathbb{R}$ are given by

$$\begin{aligned} A((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),(\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta})) &:= a(\boldsymbol{\sigma},\boldsymbol{\tau}) + b(\boldsymbol{\tau},(\mathbf{u},\boldsymbol{\gamma})) - b(\boldsymbol{\sigma},(\mathbf{v},\boldsymbol{\eta})) \\ &+\kappa_1\int_\Omega \big(\,\mathbf{e}(\mathbf{u}) - \mathcal{C}^{-1}\,\boldsymbol{\sigma}\,\big) : \big(\,\mathbf{e}(\mathbf{v}) + \mathcal{C}^{-1}\,\boldsymbol{\tau}\,\big) \;+\; \kappa_2\int_\Omega \mathbf{div}(\boldsymbol{\sigma})\cdot\mathbf{div}(\boldsymbol{\tau}) \\ &+\kappa_3\int_\Omega \left(\boldsymbol{\gamma} - \frac{1}{2}(\nabla\mathbf{u} - (\nabla\mathbf{u})^{\mathtt{t}})\right) : \left(\boldsymbol{\eta} + \frac{1}{2}(\nabla\mathbf{v} - (\nabla\mathbf{v})^{\mathtt{t}})\right), \end{aligned}$$

and $B((\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta}),\boldsymbol{\chi}) := \langle\boldsymbol{\tau}\mathbf{n},\boldsymbol{\chi}\rangle_{\Gamma_N}$, where the bilinear forms $a: H(\mathbf{div};\Omega)\times H(\mathbf{div};\Omega)\to\mathbb{R}$ and $b: H(\mathbf{div};\Omega)\times Q\to\mathbb{R}$ are defined by

$$a(\boldsymbol{\zeta},\boldsymbol{\tau}) := \int_\Omega \mathcal{C}^{-1}\boldsymbol{\zeta} : \boldsymbol{\tau} = \frac{1}{2\,\mu}\int_\Omega \boldsymbol{\zeta}:\boldsymbol{\tau} - \frac{\lambda}{4\,\mu\,(\lambda+\mu)}\int_\Omega \mathrm{tr}(\boldsymbol{\zeta})\,\mathrm{tr}(\boldsymbol{\tau})$$

and

$$b(\boldsymbol{\tau},(\mathbf{v},\boldsymbol{\eta},\boldsymbol{\chi})) := \int_\Omega \mathbf{v}\cdot\mathbf{div}(\boldsymbol{\tau}) + \int_\Omega \boldsymbol{\tau}:\boldsymbol{\eta} + \langle\boldsymbol{\tau}\mathbf{n},\boldsymbol{\chi}\rangle_{\Gamma_N}$$

The linear functionals $F:\mathbf{H}\to\mathbb{R}$ and $G:\mathbf{Q}\to\mathbb{R}$ are given by

$$F(\boldsymbol{\tau},\mathbf{v},\boldsymbol{\eta}) := \int_\Omega \mathbf{f}\cdot(\,\mathbf{v} - \kappa_2\,\mathbf{div}(\boldsymbol{\tau})\,) \qquad G(\boldsymbol{\chi}) := \langle\mathbf{g},\boldsymbol{\chi}\rangle_{\Gamma_N}$$

The idea is to choose the positive parameters κ_1, κ_2 and κ_3 independent of λ and such that (1) satisfies the hypotheses of the Babuška-Brezzi theory. The following result is proved in [12, Theorem 3.4].

Theorem 2.1 *Assume that $(\kappa_1,\kappa_2,\kappa_3)$ is independent of λ and such that $0<\kappa_1<2\,\mu$, $0<\kappa_2$ and $0<\kappa_3<\dfrac{\kappa_1\,k_D}{1-k_D}$, where $k_D\in(0,1)$ is the constant of Korn's first inequality. Then, the augmented variational formulation (1) has a unique solution $((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),\boldsymbol{\xi})\in\mathbf{H}\times\mathbf{Q}$, and there exists a positive constant C, independent of λ, such that*

$$\|((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),\boldsymbol{\xi})\|_{\mathbf{H}\times\mathbf{Q}} \le C\left(\|\mathbf{f}\|_{[L^2(\Omega)]^2} + \|\mathbf{g}\|_{[H^{-1/2}(\Gamma_N)]^2}\right).$$

 ISBN: 978-84-612-9727-6

From now on, we assume that the vector of parameters $(\kappa_1, \kappa_2, \kappa_3)$ satisfies the assumptions of Theorem 2.1, and that Ω is a polygonal region. Let $\mathbf{H}_h := H_h^{\boldsymbol{\sigma}} \times H_{D,h}^{\mathbf{u}} \times H_h^{\boldsymbol{\gamma}}$ and $\mathbf{Q}_{\tilde{h}}$ be any finite element subspaces of $\mathbf{H}$ and $\mathbf{Q}$, respectively, where h and $\tilde{h}$ are two positive parameters. Then, a Galerkin scheme associated to the augmented variational formulation (1) reads: find $((\boldsymbol{\sigma}_h, \mathbf{u}_h, \boldsymbol{\gamma}_h), \boldsymbol{\xi}_{\tilde{h}}) \in \mathbf{H}_h \times \mathbf{Q}_{\tilde{h}}$ such that for all $(\boldsymbol{\tau}_h, \mathbf{v}_h, \boldsymbol{\eta}_h) \in \mathbf{H}_h$ and $\boldsymbol{\chi}_{\tilde{h}} \in \mathbf{Q}_{\tilde{h}}$,

$$\begin{array}{rcl} A((\boldsymbol{\sigma}_h, \mathbf{u}_h, \boldsymbol{\gamma}_h), (\boldsymbol{\tau}_h, \mathbf{v}_h, \boldsymbol{\eta}_h)) + B((\boldsymbol{\tau}_h, \mathbf{v}_h, \boldsymbol{\eta}_h), \boldsymbol{\xi}_{\tilde{h}}) & = & F(\boldsymbol{\tau}_h, \mathbf{v}_h, \boldsymbol{\eta}_h) \\ B((\boldsymbol{\sigma}_h, \mathbf{u}_h, \boldsymbol{\gamma}_h), \boldsymbol{\chi}_{\tilde{h}}) & = & G(\boldsymbol{\chi}_{\tilde{h}}) \end{array} \tag{2}$$

We remark that the properties of the bilinear form B are not directly transferred to the discrete level and need to be proved for each particular choice of the corresponding finite element subspaces.

Let $\{\mathcal{T}_h\}_{h>0}$ be a regular family of triangulations of $\bar{\Omega}$. We assume that $\bar{\Omega} = \cup\{\, T \,:\, T \in \mathcal{T}_h \,\}$. Given a triangle $T \in \mathcal{T}_h$, we denote by h_T its diameter and define the mesh size $h := \max\{\, h_T \,:\, T \in \mathcal{T}_h \,\}$. We also assume that each point in $\bar{\Gamma}_D \cap \bar{\Gamma}_N$ is a vertex of $\mathcal{T}_h$, for all $h > 0$. In addition, given an integer $\ell \geq 0$ and $S \subset \mathbb{R}^2$, we denote by $\mathcal{P}_\ell(S)$ the space of polynomials in two variables defined in S of total degree at most ℓ, and for each $T \in \mathcal{T}_h$, we denote by $\mathcal{RT}_0(T)$ the local Raviart-Thomas space of the lowest order. Finally, let $\gamma_{\tilde{h}} = \{\tilde{e}_1, \tilde{e}_2, ..., \tilde{e}_m\}$ be an independent partition of the Neumann boundary Γ_N, where $\tilde{h} := \max\{|\tilde{e}_j| :\ j = 1, ..., m\}$. Then, one possibility is to choose

$$H_h^{\boldsymbol{\sigma}} := \left\{ \boldsymbol{\tau}_h \in H(\mathbf{div}; \Omega) \,:\, \boldsymbol{\tau}_h|_T \in [\mathcal{RT}_0(T)^{\mathtt{t}}]^2, \quad \forall\, T \in \mathcal{T}_h \right\},$$

$$H_{D,h}^{\mathbf{u}} := \left\{ \mathbf{v}_h \in [C(\bar{\Omega})]^2 \,:\, \mathbf{v}_h|_T \in [\mathcal{P}_1(T)]^2, \quad \forall\, T \in \mathcal{T}_h, \quad \mathbf{v}_h = \mathbf{0} \quad \text{on } \Gamma_D \right\},$$

$$H_h^{\boldsymbol{\gamma}} := \left\{ \boldsymbol{\eta}_h \in [L^2(\Omega)]^{2\times 2}_{\mathtt{skew}} \,:\, \boldsymbol{\eta}_h|_T \in [\mathcal{P}_0(T)]^{2\times 2} \quad \forall\, T \in \mathcal{T}_h \right\},$$

and

$$\mathbf{Q}_{\tilde{h}} := \left\{ \boldsymbol{\chi}_{\tilde{h}} \in [C(\Gamma_N)]^2 \cap [H_{00}^{1/2}(\Gamma_N)]^2 \,:\, \boldsymbol{\chi}_{\tilde{h}}|_{\tilde{e}_j} \in [\mathcal{P}_1(\tilde{e}_j)]^2 \quad \forall\, j \in \{1, ..., m\} \right\}.$$

Let $\{e_1, e_2, ..., e_n\}$ be the partition of Γ_N defined by $\mathcal{T}_h$, and assume that the family of triangulations $\{\mathcal{T}_h\}_{h>0}$ is uniformly regular near Γ_N, that is, there exists $C > 0$, independent of h, such that $|e_j| \geq C\, h$ for all $j \in \{1, ..., n\}$ and for all $h > 0$. Further, let us assume that the independent partition $\{\tilde{e}_1, ..., \tilde{e}_m\}$ of Γ_N is uniformly regular, that is, there exists $C > 0$, independent of $\tilde{h}$, such that $|\tilde{e}_j| \geq C\, \tilde{h}$ for all $j \in \{1, ..., m\}$ and for all $\tilde{h} > 0$. Under these conditions, the unique solvability and stability of the augmented Galerkin scheme (2) and the corresponding Cea's estimate are established in Theorem 4.9 in [12] for the previous finite element subspaces, for all $\tilde{h} \leq h_0$ and for all $h \leq C_0\, \tilde{h}$. Moreover, using the approximation properties of the finite element subspaces (see [6, 10]), the corresponding rate of convergence of the Galerkin scheme (2) is provided (see Theorem 4.10 in [12]).

 ISBN: 978-84-612-9727-6

3 The a posteriori error estimator of residual type

In this section we provide a residual based a posteriori error estimator for the discrete scheme (2). First we introduce several notations. Given $T \in \mathcal{T}_h$, we denote by $E(T)$ the set of its edges, and by E_h the set of all edges of the triangulation $\mathcal{T}_h$. Then we write $E_h = E_h(\Omega) \cup E_h(\Gamma_D) \cup E_h(\Gamma_N)$, where $E_h(S) := \{e \in E_h : e \subseteq S\}$ for $S \subset \mathbb{R}^2$. In what follows, h_e stands for the length of edge $e \in E_h$. We also assume that $h \leq C_0\, \tilde{h}$. Then, we can assume, without loss of generality, that each side $e_i \in E_h(\Gamma_N)$, $i = 1, \ldots, n$, is contained in a side $\tilde{e}_j$, for some $j \in \{1, \ldots, m\}$; in this case, we denote by $\tilde{h}_{e_i} = |\tilde{e}_j|$. Further, given $\boldsymbol{\tau} \in [L^2(\Omega)]^{2\times 2}$ such that $\boldsymbol{\tau}|_T \in [C(T)]^{2\times 2}$ on each $T \in \mathcal{T}_h$, an edge $e \in E(T) \cap E_h(\Omega)$, for some $T \in \mathcal{T}_h$, and the unit tangential vector $\mathbf{t}_T$ along e, we denote by $J[\boldsymbol{\tau}\mathbf{t}_T]$ the jump of $\boldsymbol{\tau}$ across e, that is, $J[\boldsymbol{\tau}\mathbf{t}_T] := (\boldsymbol{\tau}|_T - \boldsymbol{\tau}|_{T'})|_e \mathbf{t}_T$, where $T' \in \mathcal{T}_h$ is such that $T \cap T' = e$. Abusing notation, when $e \in E_h(\Gamma)$, we write $J[\boldsymbol{\tau}\mathbf{t}_T] := \boldsymbol{\tau}|_e \mathbf{t}_T$. We recall here that, if $\mathbf{n}_T := (n_1, n_2)^{\mathtt{t}}$ is the unit outward normal to ∂T, then $\mathbf{t}_T := (-n_2, n_1)^{\mathtt{t}}$. The normal jumps $J[\boldsymbol{\tau}\mathbf{n}_T]$ can be defined analogously.

For each $T \in \mathcal{T}_h$, we define the local error indicator

$$
\begin{aligned}
\eta_T^2 :=\ & \|\mathbf{f} + \mathbf{div}(\boldsymbol{\sigma}_h)\|^2_{[L^2(T)]^2} \ + \ \tfrac{1}{4}\, \|\boldsymbol{\sigma}_h - \boldsymbol{\sigma}_h^{\mathtt{t}}\|^2_{[L^2(T)]^{2\times 2}} \\
& + \|\boldsymbol{\gamma}_h - \tfrac{1}{2}(\nabla \mathbf{u}_h - (\nabla \mathbf{u}_h)^{\mathtt{t}})\|^2_{[L^2(T)]^{2\times 2}} + h_T^2\, \|\, \mathbf{curl}(\mathcal{C}^{-1}\boldsymbol{\sigma}_h - \nabla \mathbf{u}_h + \boldsymbol{\gamma}_h)\|^2_{[L^2(T)]^2} \\
& + h_T^2\, \|\, \mathbf{curl}(\mathcal{C}^{-1}(\mathbf{e}(\mathbf{u}_h) - \mathcal{C}^{-1}\boldsymbol{\sigma}_h))\|^2_{[L^2(T)]^2} \\
& + h_T^2\, \|\mathbf{div}(\mathbf{e}(\mathbf{u}_h) - \tfrac{1}{2}\,\mathcal{C}^{-1}(\boldsymbol{\sigma}_h + \boldsymbol{\sigma}_h^{\mathtt{t}}))\|^2_{[L^2(T)]^2} \\
& + h_T^2\, \|\mathbf{div}(\boldsymbol{\gamma}_h - \tfrac{1}{2}(\nabla \mathbf{u}_h - (\nabla \mathbf{u}_h)^{\mathtt{t}}))\|^2_{[L^2(T)]^2} \\
& + \sum_{e \in E(T)} h_e\, \|J[(\mathcal{C}^{-1}\boldsymbol{\sigma}_h - \nabla \mathbf{u}_h + \boldsymbol{\gamma}_h)\mathbf{t}_T]\|^2_{[L^2(e)]^2} \\
& + \sum_{e \in E(T)} h_e\, \|J[(\mathcal{C}^{-1}(\mathbf{e}(\mathbf{u}_h) - \mathcal{C}^{-1}\boldsymbol{\sigma}_h))\mathbf{t}_T]\|^2_{[L^2(e)]^2} \\
& + \sum_{e \in E(T) \cap E_h(\Omega \cup \Gamma_N)} h_e\ \|J[(\mathbf{e}(\mathbf{u}_h) - \frac{1}{2}\,\mathcal{C}^{-1}(\boldsymbol{\sigma}_h + \boldsymbol{\sigma}_h)^{\mathtt{t}})\mathbf{n}_T]\|^2_{[L^2(e)]^2} \\
& + \sum_{e \in E(T) \cap E_h(\Omega \cup \Gamma_N)} h_e\ \|J[(\boldsymbol{\gamma}_h - \frac{1}{2}(\nabla \mathbf{u}_h - (\nabla \mathbf{u}_h)^{\mathtt{t}}))\mathbf{n}_T]\|^2_{[L^2(e)]^2} \\
& + \log(1+\kappa) \sum_{e \in E(T) \cap E_h(\Gamma_N)} h_e ||\mathbf{g} - \boldsymbol{\sigma}_h \mathbf{n}_T||^2_{[L^2(e)]^2} \\
& + \sum_{e \in E(T) \cap E_h(\Gamma_N)} h_e \left|\left| \frac{d}{d\mathbf{t}_T}(\mathbf{u}_h + \boldsymbol{\xi}_{\tilde{h}}) \right|\right|^2_{[L^2(e)]^2}
\end{aligned}
$$

where $\kappa = \max\{\frac{\tilde{h}_j}{\tilde{h}_j} : \tilde{e}_j \text{ and } \tilde{e}_j \text{ neighbours}\}$. The residual character of each term involved in the definition of η_T is quite clear. We define the global residual error estimator $\eta := \left(\sum_{T \in \mathcal{T}_h} \eta_T^2 \right)^{1/2}$. We proved in [3] that η is both realiable and efficient. This result is stated properly below.

Theorem 3.1 *Let* $((\boldsymbol{\sigma},\mathbf{u},\boldsymbol{\gamma}),\boldsymbol{\xi}) \in \mathbf{H}\times\mathbf{Q}$ *and* $((\boldsymbol{\sigma}_h,\mathbf{u}_h,\boldsymbol{\gamma}_h),\boldsymbol{\xi}_{\tilde{h}}) \in \mathbf{H}_h\times\mathbf{Q}_{\tilde{h}}$ *be the unique solutions of problems* (1) *and* (2)*, respectively, and let us assume that* $\mathbf{g} \in [L^2(\Gamma_N)]^2$*. Then there exist positive constants* $C_{\mathtt{eff}}$ *and* $C_{\mathtt{rel}}$*, independent of* λ*,* h *and* $\tilde{h}$*, such that*

$$C_{\mathtt{eff}}\,\eta \;\leq\; \|((\boldsymbol{\sigma}-\boldsymbol{\sigma}_h,\mathbf{u}-\mathbf{u}_h,\boldsymbol{\gamma}-\boldsymbol{\gamma}_h),\boldsymbol{\xi}-\boldsymbol{\xi}_{\tilde{h}})\|_{\mathbf{H}\times\mathbf{Q}} \;\leq\; C_{\mathtt{rel}}\,\eta\,. \tag{3}$$

The reliability of the error estimator η (upper bound in (3)) is derived combining a technique used in mixed finite element schemes (see, e.g. [8, 9]) with the usual procedure applied to primal finite element methods (see [14]). This technique was used for the first time in [5]. Further, to derive upper bounds for the terms on the Newmann boundary, we followed [4] and used some results from [7]. To prove the so-called efficiency (lower bound in (3)), we proceed as in [8] and [9], and use inverse inequalities (see [10]) and the localization technique introduced in [14], which is based on triangle-bubble and edge-bubble functions.

4 Numerical results

In this section we present some numerical results illustrating the performance of the augmented mixed finite element scheme (2) and the adaptive algorithm based on the a posteriori error estimator η. In what follows, N stands for the total number of degrees of freedom. According to the stability condition required to $\{\tilde{e}_j\}_{j=1}^m$, we set a vertex of $\{\tilde{e}_j\}_{j=1}^m$ every two vertices of $\{e_j\}_{j=1}^n$. The individual errors are denoted by $e(\boldsymbol{\sigma}) := \|\boldsymbol{\sigma}-\boldsymbol{\sigma}_h\|_{H(\mathbf{div};\Omega)}$, $e(\mathbf{u}) := |\mathbf{u}-\mathbf{u}_h|_{[H^1_{\Gamma_D}(\Omega)]^2}$, $e(\boldsymbol{\gamma}) := \|\boldsymbol{\gamma}-\boldsymbol{\gamma}_h\|_{[L^2(\Omega)]^{2\times 2}}$ and $e(\boldsymbol{\xi}) := \|\boldsymbol{\xi}-\boldsymbol{\xi}_h\|_{[H^{1/2}_{00}(\Gamma_N)]^2}$, and the total error is

$$e_{total} := \left\{ [e(\boldsymbol{\sigma})]^2 \,+\, [e(\mathbf{u})]^2 \,+\, [e(\boldsymbol{\gamma})]^2 \,+\, [e(\boldsymbol{\xi})]^2 \right\}^{1/2}.$$

In practice, the individual errors are computed using a Gaussian quadrature rule in each triangle. The effectivity index with respect to η is defined by e_{total}/η. The adaptive algorithm based on η refines, in each step, all triangles $T \in \mathcal{T}_h$ such that $\eta_T \geq \frac{1}{2}\max\{\eta_{T'} : T' \in \mathcal{T}_h\}$; triangles are refined using the *bisection procedure*. Given two consecutive meshes with associated degrees of freedom N and N', and total errors e_{total} and e'_{total}, we defined the experimental rate of convergence as

$$r(e_{total}) := -2\,\frac{\log(e_{total}/e'_{total})}{\log(N/N')}\,.$$

We recall that given the Young modulus E and the Poisson ratio ν of a linear elastic material, the corresponding Lamé constants are defined by $\mu := \frac{E}{2(1+\nu)}$ and $\lambda := \frac{E\,\nu}{(1+\nu)\,(1-2\,\nu)}$. In [3], we tested the robustness of the a posteriori error estimator η with respect to the Poisson ratio (and hence with respect to the Lamé constant λ). In the example below we take $E = 1$ and $\nu = 0.4900$, which yield the following values of μ and λ :

$$\mu = 0.3356 \qquad \lambda = 16.4430$$

 ISBN: 978-84-612-9727-6

The robustness with respect to the parameters $(\kappa_1, \kappa_2, \kappa_3)$ was also tested in [3]. Here, we take $(\kappa_1, \kappa_2, \kappa_3) = \left(\mu, \frac{1}{2\mu}, \frac{\mu}{8}\right)$, which corresponds to a feasible choice, as described in Theorem 2.1. We take Ω as the L-shaped domain $(-1,1)^2 \setminus [0,1]^2$, and $\Gamma_D := \big(\{0\} \times [0,1]\big) \cup \big([0,1] \times \{0\}\big)$. The data $\mathbf{f}$ and $\mathbf{g}$ are chosen so that the exact solution is $u_1(r,\theta) = u_2(r,\theta) = r^{5/3}\sin((2\theta - \pi)/3)$. We remark that the solution is singular at the boundary point $(0,0)$. In fact, the behaviour of $\mathbf{u}$ in a neighborhood of the origin implies that $\mathbf{div}(\boldsymbol{\sigma}) \in [H^{2/3}(\Omega)]^2$ only, which according to Theorem 4.10 in [12], yields $2/3$ as the expected rate of convergence for the uniform refinement. In addition, the solution show a large stress region in a neighborhood of the boundary point $(0,0)$.

In Tables 5.1 and 5.2 we provide the individual and total errors, the experimental rates of convergence, the a posteriori error estimators and the effectivity indexes for the uniform and adaptive refinements, respectively. We observe from these tables that the errors of the adaptive procedure decrease much faster than those obtained by the uniform one, which is confirmed by the experimental rates of convergence. This fact can also be seen in Figure 5.1, where we display the total error e_{total} versus the degrees of freedom N for both refinements.

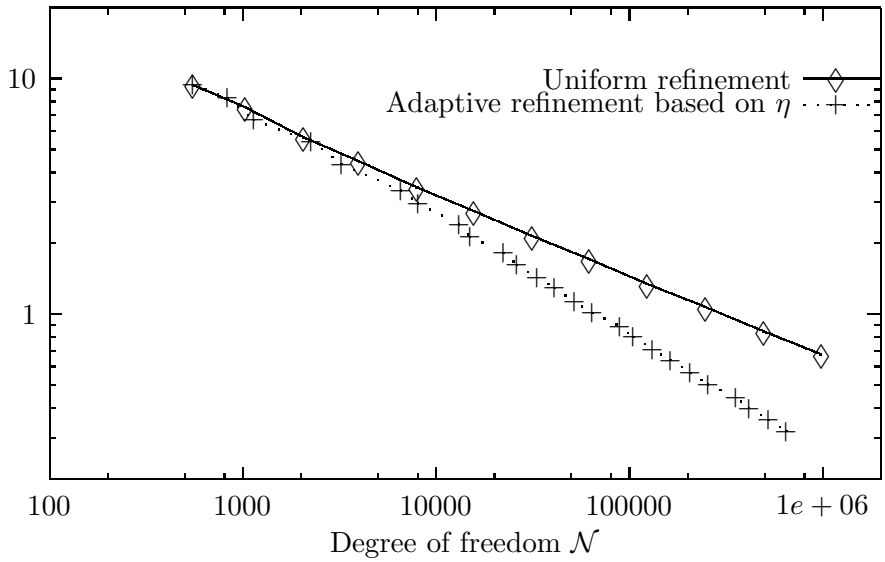


Figure 5.1: Total errors e_{total} vs. dof for the uniform and adaptive refinements.

 ISBN: 978-84-612-9727-6

Table 5.1: Total errors, experimental rates of convergence, a posteriori error estimators and effectivity indexes for the uniform refinement.

N	e_{total}	$r(e_{total})$	η	e_{total}/η
550	0.9362E+1	—	0.5205E+2	0.1799
1030	0.7567E+1	0.6784	0.3932E+2	0.1925
2062	0.5609E+1	0.8630	0.2570E+2	0.2182
3982	0.4470E+1	0.6898	0.1868E+2	0.2393
7966	0.3449E+1	0.7475	0.1283E+2	0.2688
15646	0.2748E+1	0.6738	0.9226E+1	0.2978
31294	0.2149E+1	0.7094	0.6501E+1	0.3306
62014	0.1714E+1	0.6616	0.4676E+1	0.3665
124030	0.1346E+1	0.6970	0.3341E+1	0.4028
246910	0.1074E+1	0.6545	0.2417E+1	0.4445
493822	0.8452E+0	0.6924	0.1746E+1	0.4842
985342	0.6752E+0	0.6505	0.1274E+1	0.5299

Table 5.4: Total errors, experimental rates of convergence, a posteriori error estimators and effectivity indexes for the adaptive refinement.

N	e_{total}	$r(e_{total})$	η	e_{total}/η
550	0.9362E+1	—	0.5205E+2	0.1799
830	0.8277E+1	0.5985	0.4128E+2	0.2005
1140	0.6643E+1	1.3862	0.3284E+2	0.2023
2254	0.5413E+1	0.6009	0.2345E+2	0.2308
3246	0.4332E+1	1.2207	0.1879E+2	0.2306
6560	0.3348E+1	0.7324	0.1332E+2	0.2513
8104	0.2949E+1	1.2004	0.1175E+2	0.2510
13176	0.2399E+1	0.8497	0.9219E+1	0.2603
15015	0.2132E+1	1.8088	0.8646E+1	0.2466
22300	0.1827E+1	0.7814	0.7121E+1	0.2565
26210	0.1627E+1	1.4291	0.6483E+1	0.2510
33400	0.1430E+1	1.0695	0.5724E+1	0.2498
40896	0.1296E+1	0.9707	0.5156E+1	0.2513
51830	0.1131E+1	1.1470	0.4584E+1	0.2468
64244	0.1013E+1	1.0304	0.4150E+1	0.2440
89170	0.8910E+0	0.7809	0.3512E+1	0.2537
104520	0.8001E+0	1.3554	0.3209E+1	0.2493
131396	0.7121E+0	1.0181	0.2863E+1	0.2487
162732	0.6406E+0	0.9893	0.2568E+1	0.2495
205668	0.5648E+0	1.0756	0.2289E+1	0.2467
255476	0.5029E+0	1.0695	0.2068E+1	0.2432
354138	0.4444E+0	0.7581	0.1749E+1	0.2540
415684	0.4010E+0	1.2808	0.1601E+1	0.2506
521490	0.3574E+0	1.0165	0.1431E+1	0.2498
649138	0.3199E+0	1.0109	0.1280E+1	0.2500

As shown by the values of $r(e_{total})$, that approach 2/3 for the uniform refinement, the adaptive method is able to recover, at least approximately, the quasi-optimal rate

 ISBN: 978-84-612-9727-6

of convergence $\mathcal{O}(h)$ for the total error. Furthermore, the effectivity indexes remain bounded from above and below, which confirms the reliability and efficiency of η for the adaptive algorithm. Finally, some intermediate meshes obtained with the adaptive refinement are displayed in Figure 5.2. We observe that the method is able to recognize the singularity and the large stress regions of the solutions since the adapted meshes are highly refined around the boundary point $(0, 0)$.

In summary, the numerical experiments underline the reliability and efficiency of the a posteriori error estimator η and strongly demonstrate that the associated adaptive algorithm is much more suitable than a uniform discretization procedure when solving problems with non-smooth solutions.

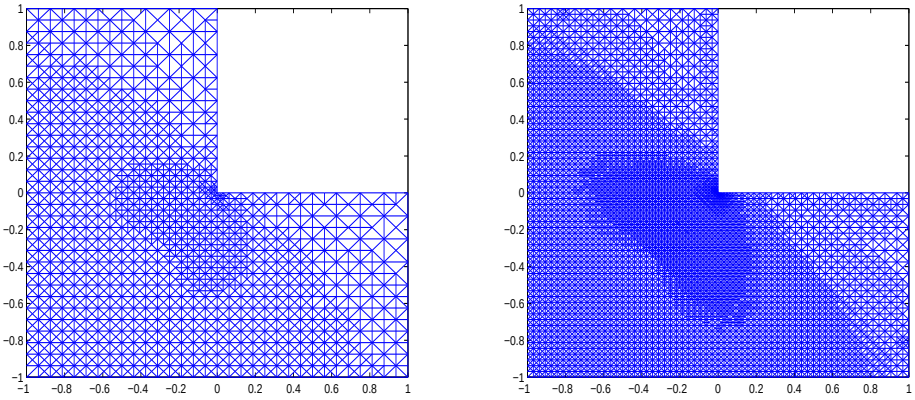


Figure 5.2: Adapted intermediate meshes with 15015 and 64244 d.o.f.

Acknowledgements

This work has been partially supported by MEC (project MTM2004-05796-C02-01), by Xunta de Galicia (project PGIDIT05PXIC30302PN), by CONICYT-Chile through the FONDECYT Grants 11060014 and 11070085, and by the Dirección de Investigación of the Universidad de Católica de la Santísima Concepción.

References

[1] D.N. Arnold, F. Brezzi, and J. Douglas, *PEERS: A new mixed finite element method for plane elasticity*. Japan Journal of Applied Mathematics, vol. 1, pp. 347-367, (1984).

[2] I. Babuška and G.N. Gatica, *On the mixed finite element method with Lagrange multipliers*. Numerical Methods for Partial Differential Equations, vol. 19, 2, pp.192-210, (2003).

[3] T.P. Barrios, E.M. Behrens and M. González. *Residual based a posteriori error estimators for an augmented mixed finite element method in linear elasticity with mixed nonhomogeneous boundary conditions*. Pre-print (2009).

[4] T.P. Barrios and G.N. Gatica, *An augmented mixed finite element method with Lagrange multipliers: A priori and a posteriori error analyses*, J. Comp. Appl. Math., vol. 200, 2, pp. 653-676, (2007).

[5] T.P. Barrios, G.N. Gatica, M. González and N. Heuer, *A residual based a posteriori error estimator for an augmented mixed finite element method in linear elasticity*, ESAIM: Mathematical Modelling and Numerical Analysis, vol. 40, 5, pp. 843-869, (2006).

[6] F. Brezzi and M. Fortin, Mixed and Hybrid Finite Element Methods. Springer Verlag, 1991.

[7] C. Carstensen, *An a posteriori error estimate for a first-kind integral equation*. Mathematics of Computation, vol. 66, 217, pp. 139-155, (1997).

[8] C. Carstensen, *A posteriori error estimate for the mixed finite element method*. Mathematics of Computation, vol. 66, 218, pp. 465-476, (1997).

[9] C. Carstensen and G. Dolzmann, *A posteriori error estimates for mixed FEM in elasticity*, Numer. Math., vol. 81, pp. 187-209, (1998).

[10] P.G. Ciarlet, The Finite Element Method for Elliptic Problems. North-Holland, 1978.

[11] P. Clément, *Approximation by finite element functions using local regularisation*. RAIRO Modélisation Mathématique et Analyse Numérique, vol. 9, pp. 77-84, (1975).

[12] G.N. Gatica, *Analysis of a new augmented mixed finite element method for linear elasticity allowing RT_0 - P_1 - P_0 approximations*, ESAIM: Mathematical Modelling and Numerical Analysis, vol. 40, 1, pp. 1-28, (2006).

[13] M. Lonsing and R. Verfürth, *A posteriori error estimators for mixed finite element methods in linear elasticity*, Numer. Math. 97 (4): 757-778, 2004.

[14] R. Verfürth, A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. Wiley-Teubner (Chichester), 1996.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

An algorithm for Bang-Bang control of fixed-head hydroplants

L. Bayón[1], J.M. Grau[1], M.M. Ruiz[1] and P.M. Suárez[1]

[1] *Department of Mathematics, University of Oviedo. Spain*

emails: bayon@uniovi.es, grau@uniovi.es, mruiz@uniovi.es, pedrosr@uniovi.es

Abstract

This paper deals with the optimal control problem that arise when a hydraulic system with fixed-head hydroplants is considered. In the frame of a deregulated electricity market the resulting Hamiltonian for such systems is linear in the control variable and results in an optimal singular/bang-bang control policy. To avoid difficulties associated with the computation of optimal singular/bang-bang controls, an efficient and simple optimization algorithm is proposed. The computational technique is illustrated on one example.

Key words: Optimal Control, Singular/Bang-Bang Problems, Hydroplants
MSC 2000: 49J30

1 Introduction

The computation of optimal singular/bang-bang controls is of particular interest to researchers because of the difficulty in obtaining the optimal solution. Several engineering control problems, such as chemical reactor start-up or hydrothermal optimization problems, are known to have optimal singular/bang-bang problems. This paper deals with the optimal control (OC) problem that arises when addressing the new short-term problems that are faced by a generation company in a deregulated electricity market. Our model of the spot market explicitly represents the price of electricity as a known exogenous variable and we consider a system with fixed-head hydroplants. These plants, with a large capacity reservoir, are the most important in the electricity market. The resulting Hamiltonian for such systems, H, is linear in the control variable, u, and results in an optimal singular/bang-bang control policy.

In general, the application of Pontryagin's Maximum Principle (PMP) is not well suited for computing singular control problems as it fails to yield a unique value for the control. Different methods for determining optimal controls with a possibly singular part have already been developed. In [1], the switching function is used as a constraint

and the resulting problem is solved as a differential algebraic equation (DAE) problem. Other popular approaches are the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm and other decay methods taken from nonlinear optimization [2], Maurer's Method [3], which converts the problem into a two point boundary value problem (TPBVP) that can be solved by the multi shooting method, and the Method by Fraser-Andrews [4], which determines the structure using orthogonal functions.

Another method that has been used by a number of researcher is the ε-method by Bell and Jacobson. This method [5] involves solving the singular/bang-bang optimal control problem as the limit of a series of nonsingular problems. The problem then becomes well defined so that methods based on PMP can be used. However, some existing numerical methods for handling such problems behave poorly. One alternative, Iterative Dynamic Programming (IDP) [6], has been used and applications to different types of problems have been reported. Recently [7] Maurer et al. presented a numerical scheme for computing optimal bang-bang controls on problems with a larger number of switchings. They assume that every component of the optimal control is bang-bang and that there are only finitely many switching times. Such a bang-bang control can be computed by solving an induced optimization problem, using the durations of the bang-bang arcs as optimization variables instead of the switching times.

In this paper we propose a simple and efficient optimization algorithm that avoids all the difficulties that the above methods present. The algorithm has been specifically developed for a hydraulic problem and we remark that no approach has yet been developed to find the bang-bang solution to our hydro-problem. The paper is organized as follows. In Section 2, we present the mathematical environment of our work: the singular optimal control problem with control appearing linearly. In Section 3, we present the mathematical models of our fixed-head hydroplant. In Section 4 we formulate our optimization problem: *profit maximization of fixed-head hydroplants in a deregulated electricity market* and prove that singular controls can be excluded. In Section 5 we describe the algorithm that provides the structure of bang-bang arcs. The results of the application of the method to a numerical example are presented in Section 6. Finally, the main conclusions of our research are summarized in Section 7.

2 General statement of the singular OC problem

Let us assume a system given by: a state $x(t) \in \mathbb{R}^n$ at time $t \in [0,T]$, a control $u(t) \in U(t) \subset \mathbb{R}^m$, where u is piecewise continuous and $U(t)$ is compact for every $t \in [0,T]$, a state equation $x'(t) = f(t,x(t),u(t))$ almost everywhere, an initial condition $x(0) = x_0$ and final condition $x(T) \in Z \neq \varnothing$, where $[0,T]$ is fixed, and the scalar functions g and L with a suitable domain. The following problem is called the Bolza problem **(P)**:

Find an admissible pair (x,u) *on* $[0,T]$ *such that the functional*

$$J(u) = g(x(T)) + \int_0^T L(t,x(t),u(t))dt$$

 ISBN: 978-84-612-9727-6

becomes maximal. If $g \equiv 0$, we call **(P)** a Lagrange problem, while **(P)** is called a Mayer problem if $L \equiv 0$. We define the Hamiltonian:

$$H(u, x, \lambda, t) := L(t, x, u) + \lambda^T f(t, x, u)$$

where $\lambda^T \in \mathbb{R}^n$ holds. We assume that every f_i $(i = 1, ..., n)$ is continuous in (t, x, u), that the derivatives $\frac{\partial}{\partial t} f_i$ and $\nabla_x f_i$ exist and are continuous in (t, x, u) for every i. Furthermore we assume that $g \in C^1$ and that **(P)** has a solution (x^*, u^*) with $Z = \mathbb{R}^n$. The following theorem is often very useful in solving Bolza problems:

Theorem 1 (PMP). *Under the above hypothesis, there thus exists an absolutely continuous function* $\lambda : [0, T] \to \mathbb{R}^n$ *with the following properties*:

a) $x' = H_\lambda$ *and* $\lambda' = -H_x$ *along* (x^*, u^*)
b) $H(u^*(t), x^*(t), \lambda(t), t) = \max\{H(u, x^*(t), \lambda(t), t) \mid u \subset U(t)\}$ *for every* $t \in [0, T]$
c) $\lambda \neq 0$ *on* $[0, T]$
d) $\lambda(T)dx(T) - dg = 0$ (*transversality condition*)

In the usual case, the optimality condition

$$H(u^*(t), x^*(t), \lambda(t), t) = \max\{H(u, x^*(t), \lambda(t), t) \mid u \subset U(t)\} \tag{1}$$

is used to solve for the extremal control in terms of the state and adjoint (x, λ). Normally, the optimality condition is imposed as $H_u = 0$ and this system of equations is solved for the control vector $u(t)$. Additionally, since u^* is to maximize H, the Hessian must be positive definite: $H_{uu} < 0$ (Legendre-Clebsch (LC) condition).

We now consider the case of scalar control appearing linearly (H_{uu} is singular):

$$\begin{aligned} &\max \int_0^T [f_1(t, x) + u f_2(t, x)]dt \\ &x' = g_1(t, x) + u g_2(t, x); \quad x(0) = x_0 \\ &u_{\min} \leq u(t) \leq u_{\max} \end{aligned} \tag{2}$$

The variational Hamiltonian is linear in u and can be written as

$$H(u, x, \lambda, t) := f_1(t, x) + \lambda g_1(t, x) + [f_2(t, x) + \lambda g_2(t, x)]u$$

The optimality condition (maximize H w.r.t. u) leads to:

$$u^*(t) = \begin{cases} u_{\max} & \text{if} \quad f_2(t, x) + \lambda g_2(t, x) > 0 \\ u_{\text{sing}} & \text{if} \quad f_2(t, x) + \lambda g_2(t, x) = 0 \\ u_{\min} & \text{if} \quad f_2(t, x) + \lambda g_2(t, x) < 0 \end{cases}$$

and u^* is undetermined if $\Phi(x, \lambda) \equiv H_u = f_2(t, x) + \lambda g_2(t, x) = 0$. The function Φ is called the switching function. If $\Phi(x^*(t), \lambda(t)) = 0$ only at isolated time points, then the optimal control switches between its upper and lower bounds, which is said to be a bang-bang type control (i.e. the problem is not singular). The times when the OC switches from $u_{\max}$ to $u_{\min}$ or vice-versa are called switch times.

 ISBN: 978-84-612-9727-6

If $\Phi(x^*(t), \lambda(t)) = 0$ for every t in some subinterval $[t', t'']$ of $[0, T]$, then the original problem is called a singular control problem and the corresponding trajectory for $[t', t'']$, a singular arc. The case when Φ vanishes over an interval is more troublesome, because the optimality condition is vacuous, since $H(u, x^*(t), \lambda(t), t)$ is independent of u. In the singular case, PMP yields no information on the extremal (or stationary) control.

In order to find the control on a singular arc, we use the fact that H_u remains zero along the whole arc. Hence, all the time derivatives are zero along such an arc. By successive differentiation of the switching function, one of the time derivatives may contain the control u, in which case u can be obtained as a function of x and λ. The next result (see [8]) is important.

Proposition 1. *If H_u is successively differentiated with respect to time, then u cannot first appear in an odd-order derivative.*

As u first appears in an even-order derivative, we denote this by $\frac{d^{2q}(H_u)}{dt^{2q}}$ and q is the order of the singular arc. An important theorem (see [8]) is the necessary condition for a singular arc to be optimal: the Generalized Legendre-Clebsch (GLC) condition.

Theorem 2 (GLC Condition). *If $x^*(t), u^*(t)$ are optimal on a singular arc, then, for scalar u,*

$$(-1)^q \frac{\partial}{\partial u}\left[\frac{d^{2q}(H_u)}{dt^{2q}}\right] \leq 0$$

3 Hydroplant performance models

Conventional hydroplants are classified as run-of-river plants and storage plants. Run-of-river plants have little storage capacity and use water as it becomes available. The water not utilized is spilled. Storage plants are associated with reservoirs that have significant storage capacity. During periods of low power requirements, water can be stored and then released when demand is high.

A basic physically-based relationship between the active power generated P (in MW) by a hydroplant and the rate of water discharge, q (in m^3/s), and the effective head, h (in m), is given by

$$P = 0.0085\ q.h.\eta(q, h)$$

where η is a function of q and h. A variety of models have been proposed in the literature [9], [10] due to the diversity of plant types and their characteristics (see Table I). The appropriate choice of mathematical models for representing the physical system is a crucial aspect when addressing any optimization problem. In this paper we consider the approximation presented by El-Hawary [9] to be the most appropriate on account of its precision and flexibility.

 ISBN: 978-84-612-9727-6

Table I. Hydroplant models.

Glimn-Kirchmayer	$q = K\psi(h)\phi(P)$	$\psi(h) = \alpha h^2 + \beta h + \gamma$ $\phi(P) = xP^2 + yP + z$
Hildebrand	$q = \sum_{i=0}^{L}\sum_{j=0}^{K} C_{ij}P^i h^j$	(L and K are usually taken to be 2)
Hamilton-Lamont	$q = \psi(h)\frac{\phi(P)}{h}$	$\psi(h) = \alpha h^2 + \beta h + \gamma$ $\phi(P) = xP^3 + yP + z$
Arvanitidis-Rosing	$P = qh[\beta - e^{-\alpha S}]$	(S is reservoir storage)

El-Hawary's Model. In this model the output power P (MW) is given by

$$P = \frac{qh}{G}$$

where q is the rate of water discharge (m^3/h), h is the effective water head (m), and G is the efficiency (m^4/h.MW). For the sake of simplicity, we assume the rate of water spillage and the penstock head losses to be negligible. Thus, we have $h = y - y_T$, where y is the forebay elevation and y_T the tailrace elevation. In most cases, a typical linear variation between y_T and the discharge, q, and a typical linear elevation-storage curve may be assumed:

$$y(t) = [y_0 + B_y s(t)] - [y_{T_0} + B_T q(t)]$$

where $s(t)$ is the reservoir storage. The reservoir's dynamics may be suitably described by the equation

$$\frac{ds(t)}{dt} = i(t) - q(t) \rightarrow s(t) = S_0 + i \cdot t - Q(t)$$

being i the natural inflow (that is, in general, assumed constant), $Q(t)$ being the volume discharged up to the instant t by the plant and S_0 the initial volume. So, we have that

$$\begin{aligned} &P(t, Q(t), q(t)) := A(t) \cdot q(t) - B \cdot Q(t) \cdot q(t) - C \cdot q^2(t) \\ &A(t) = \frac{(y_0 - y_{T_0}) + B_y(S_0 + i \cdot t)}{G};\ B = \frac{B_y}{G};\ C = \frac{B_T}{G} \end{aligned} \tag{3}$$

This is a *variable-head* model and the hydroplant's hydraulic generation, P, is a function of $Q(t)$ and $q(t)$. According to El-Hawary's model, power output is a function of discharge and the head. For a large capacity reservoir, it is practical to assume that the effective head is constant over the optimization interval. Here the *fixed-head* hydroplant model is defined and P is represented by the linear equation:

$$P(t) = \frac{(y_0 - y_{T_0}) + B_y(S_0)}{G} q(t) = Aq(t) \tag{4}$$

 ISBN: 978-84-612-9727-6

4 Structure of the solution of the optimization problem

For convenience of formulation, in this section we introduce this new notation: $q(t) \equiv z'(t); Q(t) \equiv z(t)$. Let $P(t, z(t), z'(t))$ be the function of the hydroplant's hydraulic generation, where $z(t)$ is the volume that is discharged up to the instant t by the plant, and $z'(t)$ the rate of water discharge of the plant at the instant t. If we assume that b is the volume of water that must be discharged during the entire optimization interval $[0, T]$, the following boundary conditions will have to be fulfilled:

$$z(0) = 0, z(T) = b$$

Throughout the paper we assume that $P(t, z, z') : [0, T] \times \mathbb{R}^+ \times \mathbb{R}^+ \longrightarrow \mathbb{R}^+$; that is, we shall only admit non-negative volumes, $z(t)$, and rates of water discharge, $z'(t)$ (pumped-storage plants will be not considered). Besides the previous statement, we consider $z'(t)$ to be bounded by technical constraints

$$q_{\min} \leq z'(t) \leq q_{\max}, \quad \forall t \in [0, T]$$

No transmission losses will be considered in our study; this is a crucial aspect when addressing the optimization problem from a centralized viewpoint. From the perspective of a generation company and within the framework of the new electricity market, said losses are not relevant, as power generators currently receive payment for all the energy they generate in power plant bars.

This study constitutes a modification of previous papers by the authors [11], [12], where a variable-head model (3) was considered. When the term $-C \cdot q^2(t)$ is considered, the Hamiltonian is not linear in u and the control is not singular/bang-bang. The Hamiltonian is also not linear in u when transmission losses are considered using the classic Kirchmayer model: $P_L = BP(t)^2$; P_L being the losses.

In our problem, the objective function is given by revenue during the optimization interval $[0, T]$

$$F(z) = \int_0^T L(t, z(t), z'(t))dt = \int_0^T \pi(t) P(t, z(t), z'(t))dt$$

Revenue is obtained by multiplying the hydraulic production of the hydroplant by the clearing price $\pi(t)$ at each hour t. Our model of the spot market explicitly represents the price of electricity as a known exogenous variable. Here the fixed-head hydroplant model (4) for P is used. In keeping with the previous statement, our objective functional in continuous time form is

$$\max_z F(z) = \max_z \int_0^T \pi(t)\ A\ z'(t)dt$$

$$\text{on} \quad \Omega = \left\{ z \in \widehat{C}^1[0, T] \mid z(0) = 0, z(T) = b;\ q_{\min} \leq z'(t) \leq q_{\max}, \forall t \in [0, T] \right\}$$

 ISBN: 978-84-612-9727-6

where $\widehat{C}^1$ is the set of piecewise C^1 functions. A standard Lagrange type OC problem of type (2) can be mathematically formulated as follows:

$$\begin{aligned}
&\max \int_0^T A\pi(t)udt = \max \int_0^T f(t)udt \\
&z' = u; \quad z(0) = 0, z(T) = b \\
&u_{\min} \leq u(t) \leq u_{\max}
\end{aligned}$$

With the aim of obtaining a solution numerically, we first attempt to determine the structure of the solution; that is, the sequence of the bang-bang and the singular parts. We define the Hamiltonian:

$$H(u, x, \lambda, t) := f(t)u + \lambda u = [f(t) + \lambda]u$$

The switching function is $\Phi(x, \lambda) \equiv H_u = f(t) + \lambda$. The optimality condition (1) leads to:

$$u^*(t) = \begin{cases} u_{\max} & \text{if} \quad f(t) + \lambda > 0 \\ u_{\text{sing}} & \text{if} \quad f(t) + \lambda = 0 \\ u_{\min} & \text{if} \quad f(t) + \lambda < 0 \end{cases} \tag{5}$$

On the other hand, the co-state equation of PMP allows us to obtain:

$$\lambda' = -H_z = 0 \rightarrow \lambda = \lambda_0 \text{ (cte)} \tag{6}$$

To find the control on a singular arc, we use the fact that H_u remains zero along the whole arc. By differentiation of the switching function, we obtain

$$\begin{aligned}
&\frac{d}{dt}H_u = \frac{d}{dt}[f(t) + \lambda] = f'(t) = A\pi'(t) = 0 \\
&\qquad \ldots \\
&\frac{d^n}{dt^n}H_u = A\pi^{(n)}(t) = 0
\end{aligned}$$

We can see that in the successive derivatives of H_u w.r.t. t, doesn't appear the control u. We have only derivatives of the clearing price $\pi(t)$. The presence of singular arcs in the solution are thus ruled out.

5 Algorithm for the Bang-Bang solution

Having ruled out the presence of singular arcs, we now determine the bang-bang segments and the boundary on which the solution is situated. To obtain the optimal solution, we apply (5) and (6), obtaining

$$u^*(t) = \begin{cases} u_{\max} & \text{if} \quad f(t) > -\lambda_0 \\ u_{\min} & \text{if} \quad f(t) < -\lambda_0 \end{cases} \tag{7}$$

The algorithm that leads to the optimal solution (7) comprises the following steps:

 ISBN: 978-84-612-9727-6

(i) First, $f(t)$ must be interpolated to obtain a continuous function. Note that in real electricity markets, the clearing price $\pi(t)$ is only known at each hour ($t = 1, 2, ..., 24$). In this paper we have used linear interpolation with good results.

(ii) Second, we have to determine the switch times: $t_1, t_2, ...$ These instants are calculated solving the equation

$$f(t) = -\lambda$$

(iii) Third, the optimal value λ_0 must be determined in order for:

$$z_\lambda(T) = \sum_{i=1}^{N_s} \delta_i \cdot q_{\max} + (T - \sum_{i=1}^{N_s} \delta_i) \cdot q_{\min} = b$$

δ_i being the duration of the i-th bang-bang segment in the upper bound $u_{\max}$, N_s the number of such segments, and $z_\lambda(T)$ the final volume obtained for each λ. Figure 1 illustrates the proposed method.

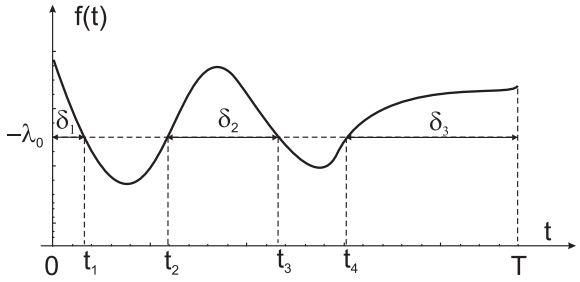


Figure 1. Illustration of the method.

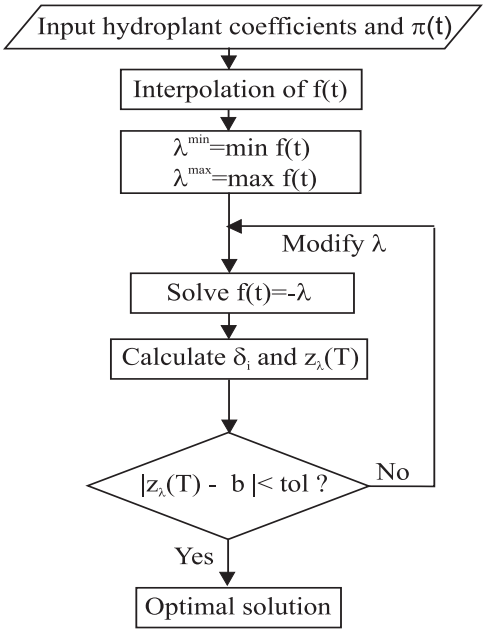


Figure 2. Computational flow of the proposed algorithm.

 ISBN: 978-84-612-9727-6

(iv) To calculate an approximate value of λ_0, we propose an iterative method (like, for example, bisection or the secant method) using this condition

$$Error = |z_\lambda(T) - b| < tol$$

to finalize the algorithm. As we shall see in the next section, the secant method has provided satisfactory results using these initial values:

$$\lambda^{\min} = \min f(t); \quad \lambda^{\max} = \max f(t)$$

6 Example

A program was written using the Mathematica package to apply the results obtained in this paper to an example of a hydraulic system made up of one fixed-head hydroplant. The hydroplant data are summarized in Table II.

Table II: Hydroplant coefficients.

$G(m^4/\,\mathrm{h}\,.\mathrm{MW})$	$b(m^3)$	$S_0(m^3)$	$y_0(m)$	$y_{T_0}(m)$	$B_y(m^{-2})$
319840	$45\ 10^6$	$2.395\ 10^8$	6.18166	5	$2.89386\ 10^{-8}$

We shall also consider the technical constraints: $q_{\min} = 0$; $q_{\max} = 3.94258\ 10^6\ (m^3/\,\mathrm{h})$, which correspond, respectively, to $P_{\min} = 0; P_{\max} = 100$ (MW). With these coefficients, the hydraulic model is:

$$P(t) = 0.0000253641\ q(t)$$

In this paper, we focus on the problem that a generation company faces when preparing its offers for the day-ahead market. Thus, the classic optimization interval of $T = 24$ h . was considered. The clearing price $\pi(t)$ $(euros/h \cdot \mathrm{MW})$ corresponding to one day was taken from the Spanish electricity market [13]. The known values of $\pi(t) : t = 1, 2, ..., 24$ were linearly interpolated (see Figure 3).

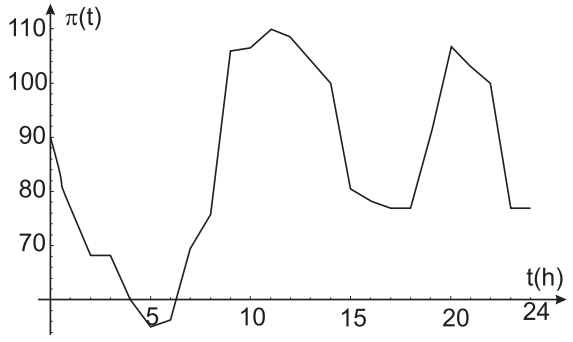


Figure 3. Clearing price $\pi(t)$.

The solution may be constructed in a simple way by taking into account the above algorithm. In this example we have:

$$f(t) = 0.0000253641\ \pi(t)$$
$$\lambda^{\min} = \min f(t) = 0.00139528$$
$$\lambda^{\max} = \max f(t) = 0.00279005$$

 ISBN: 978-84-612-9727-6

The secant method was used to calculate the approximate value of λ for which

$$Error = |z_\lambda(T) - b| < tol$$

with $tol = 50$ (m^3). The optimal value obtained is $\lambda_0 = 0.002107617885177008$ and the switch times are:

$$t_1 = 0.528346,\ t_2 = 8.24259,\ t_3 = 14.8669,\ t_4 = 18.4717,\ t_5 = 22.7328$$

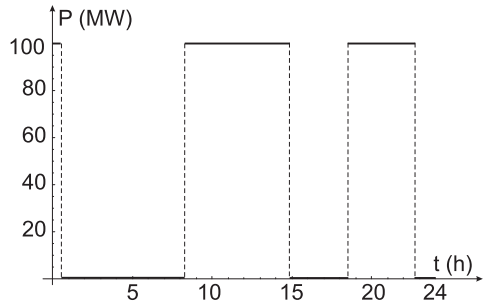


Figure 4. Optimal hydro-power $P(t)$.

Figure 4 presents the optimal hydro-power, P. The profits from the optimal solution are 130908 *euros*.

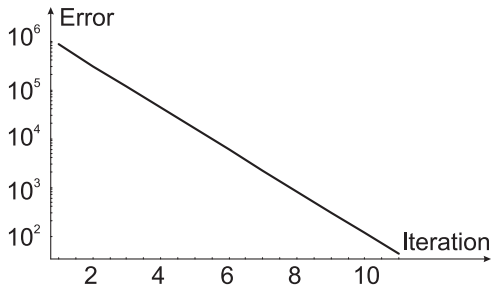


Figure 5. Convergence of the algorithm.

The algorithm runs very quickly (see Figure 5). In the example, 11 iterations were needed and the CPU time required by the program was 0.188 sec on a personal computer (Pentium IV/2GHz).

7 Conclusions and future perspectives

This paper presents a novel method for developing the optimal control problem faced by a fixed-head hydroplant in a deregulated electricity market (no transmission losses). We have proved that singular controls do not exist and, for the first time, a simple and very efficient algorithm has been specifically developed for the resulting bang-bang problem. In spite of its hydraulic origin, it should be noted that our method

may be applied to other problems with the same characteristics. As far as future perspectives are concerned, it would be very interesting to apply this method when the system is made up of variable-head hydroplants of the type: $P(t) = f(Q(t)) \cdot q(t)$ or $P(t) = f(t, Q(t)) \cdot q(t)$.

References

[1] R.M. Lewis, *Defintions of Order and Junction Conditions in Singular Optimal Control Problems*, SIAM Journal Control and Optimization **18** (1980) 21-32

[2] H.T. Jongen, K. Meer and E. Triesch, *Optimization Theory*, Springer, 2004.

[3] H. Maurer, *Numerical Solution of Singular Control Problems Using Multiple Shooting Techniques*, JOTA **18** (1976) 235-257.

[4] G. Fraser-Andrews, *Numerical Methods for Singular Optimal Control*, JCTA **61** (1989) 377-401.

[5] D.J. Bell and D.H. Jacobson, *Singular Optimal Control Problems*, Mathematics in Science and Engineering **117**, Academic Press, London 1975.

[6] R. Luus, *On the Application of Iterative Dynamic Programming to Singular Optimal Control Problems*, IEEE Trans. on Automatic control **37** (1992) 1802-1806.

[7] H. Maurer, C. Buskens, J.H.R. Kim and C. Y. Kaya, *Optimization methods for the verification of second order sufficient conditions for bang–bang controls*, Optim. Control Appl. Meth. **26** (2005) 129-156.

[8] H.J. Kelley, R.E. Kopp and H.G. Moyer, *Singular Extremals, in Topics in Optimization*, Academic Press, 1967, 63-101.

[9] M.E. El-Hawary and G.S. Christensen, *Optimal Economic Operation of Electric Power Systems*, Academic Press, New York, 1979.

[10] D.P. Kothari and J. S. Dhillon, *Power System Optimization*, PHI Learning Pvt. Ltd., 2004.

[11] L. Bayon, J. M. Grau, M. M. Ruiz and P. M. Suarez, *An environmentally constrained economic dispatch: CFBC boilers in the day-ahead market*, International Journal of Computer Mathematics **85(3)** (2008) 345-358.

[12] L. Bayon, J. M. Grau, M. M. Ruiz and P. M. Suarez, *An optimization problem in deregulated electricity markets solved with the nonsmooth maximum principle*, International Journal of Computer Mathematics **86(2)** (2009) 237-249.

[13] Compañía Operadora del Mercado Español de Electricidad, S.A. http://www.omel.es.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Seeking the identities of a ternary quaternion algebra

P. D. Beites[1], A. P. Nicolás[2] and A. P. Pozhidaev[3]

[1] *Departamento de Matemática and Centro de Matemática, Universidade da Beira Interior*

[2] *Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria*

[3] *Sobolev Institute of Mathematics, Novosibirsk State University*

emails: pbeites@mat.ubi.pt, alejandro.p.nicolas@unican.es, app@math.nsc.ru

Abstract

Through computational linear algebra, in the environment GAP, we study the degrees 1 and 2 identities of a simple 4−dimensional ternary quaternion algebra.

Key words: Computational linear algebra, Identities, Filippov algebra, Triple system
MSC 2000: 17-04, 17-08, 17A30

1 Introduction

A few years ago, the theory of n-Lie algebras (now known as Filippov algebras) attracted a lot of attention because of its close relation with the Nambu Mechanics, [5]. This connection was revealed in [7], where those algebras appear under the name of Nambu-Lie algebras. Specifically, the notion of n-Lie algebra is the implicit algebraic concept underlying the Nambu Mechanics. The definition of n-Lie algebra ($n \geq 2$) was introduced by the Russian mathematician Filippov, in 1985, as a natural generalization of the Lie algebra notion, [4].

We consider the ternary Filippov algebra A_1 (following [6], we denote the Filippov algebra A_4 by A_1) equipped with a bilinear, symmetric and non-degenerate form, and the canonical basis. We define, in Section 3, a new multiplication on the underlying vector space of A_1. This new algebra A is said to be a ternary quaternion algebra because it appears analogously to the construction of the quaternions from the Lie algebra sl_2. In this work, our main goal is the study of the degrees 1 and 2 identities of A.

Speaking finite dimensionally, it is possible to determine by hand identities of some degrees valid in an algebra that has a small dimension. But when we deal with an

algebra with a considerable dimension or we are looking for identities of high degree, it's imperative to substitute the hand calculations by computational algebraic methods. The articles of Bremner, Hentzel and Peresi illustrate the applications of computational linear algebra to the study of identities for nonassociative algebras: expansion matrix; representation theory of $\mathcal{S}_n$ — the symmetric group of degree n; pseudorandomvectors.

For the previously mentioned purpose, we decided to use the free computational algebra software `GAP` and Bremner's method — the expansion matrix. Concretely, the information about the structure of the space of identities is given by the nullspace of that matrix and this linear-algebraic data can be translated back into the identities we seek (as in [2] and [3]). We highlight (6) — an associativity identity of A, [6], that played a crucial role in the simplification of the problem.

In what follows, the symbol $:=$ denotes an equality by definition, Φ is a ground field and $\mathrm{ch}(\Phi)$ is the characteristic of Φ.

2 Preliminaries

Given a vector space U over Φ, U is an Ω-*algebra* over Φ if Ω is a system of multilinear algebraic operations defined on U:

$$\Omega = \{w_i : |w_i| = n_i \in \mathbb{N}, i \in I\},$$

where $|w_i|$ denotes the arity of w_i. In particular, we say that U is a *triple system* (or a *ternary algebra*) over Φ if U is equipped with a trilinear map $p : U^3 \to U$. We omit the arity whenever it is clear from the context.

Let L be an Ω-algebra over Φ equipped with a single n-ary operation $[.,\dots,.]$, *i.e.*, an *n-ary algebra*; L is a *Filippov algebra* (or an *n-Lie algebra*, $n \geq 2$) over Φ if, for all $x_1,\dots,x_n,y_2,\dots,y_n \in L$ and $\sigma \in \mathcal{S}_n$:

$$[x_1,\dots,x_n] = sgn(\sigma)[x_{\sigma(1)},\dots,x_{\sigma(n)}], \tag{1}$$

$$[[x_1,\dots,x_n],y_2,\dots,y_n] = \sum_{i=1}^{n}[x_1,\dots,[x_i,y_2,\dots,y_n],\dots,x_n]. \tag{2}$$

If (1) holds, then $[.,\dots,.]$ is said to be *anticommutative*; (2) is called the *generalized Jacobi identity*.

The following example of an $(n+1)$-dimensional n-Lie algebra, an analogue of the 3-dimensional Lie algebra with the cross product as multiplication, appears among the first ones given by V. T. Filippov, [4].

Example 2.1 *Let L be an $(n+1)$-dimensional Euclidean vector space over Φ equipped with the multiplication $[.,\dots,.]$, which is the vector product of $x_1,\dots,x_n \in L$, $n \geq 2$. If $\mathcal{E} = \{e_1, \dots, e_{n+1}\}$ is an orthonormal basis of L then we have:*

$$[x_1,\dots,x_n] = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} & e_1 \\ x_{21} & x_{22} & \dots & x_{2n} & e_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{(n+1)1} & x_{(n+1)2} & \dots & x_{(n+1)n} & e_{n+1} \end{vmatrix},$$

 ISBN: 978-84-612-9727-6

where $x_{1i}, \ldots, x_{(n+1)i}$ *are the coordinates of* x_i*. The vector product is completely determined by the rule:*

$$[e_1, \ldots, e_{i-1}, \widehat{e_i}, e_{i+1}, \ldots, e_{n+1}] = (-1)^{n+1+i} e_i, \tag{3}$$

$i \in \{1, \ldots, n+1\}$*, where the symbol* $\widehat{e_i}$ *means that* e_i *is omitted. The remaining products of the basis vectors are either zero or obtained from* (3) *and* (1)*. The basis* $\mathcal{E}$ *for which the product is written in the form* (3) *is said to be canonical. V. T. Filippov showed that this* Ω*-algebra is an* n*-Lie algebra and denoted it by* A_{n+1}*,* [4].

Let L be a Filippov algebra. A subspace I of L is an *ideal* of L provided that $[I, L, \ldots, L] \subseteq I$. If $[L, \ldots, L] \neq \{0\}$ and L lacks of ideals other than $\{0\}$ and L then we say that L is *simple*.

In [9], W. Ling proved that A_{n+1} is the only $(n+1)$-dimensional simple Filippov algebra over an algebraically closed field of characteristic zero, up to isomorphism.

3 On the identities of the ternary quaternion algebra A

Consider the ternary Filippov algebra A_1 over Φ equipped with a bilinear, symmetric and non-degenerate form $(.,.)$, and the canonical basis $\mathcal{E}$. We define a new multiplication on the underlying vector space of A_1 in the following way:

$$\{x, y, z\} := -(y, z)x + (x, z)y - (x, y)z + [x, y, z]. \tag{4}$$

We denote the obtained algebra by A. The product in A differs from that which appears in [6] by a scalar. We seek the degrees 1 and 2 identities of A (1-identities and 2-identities, respectively), which will be found using the method of the expansion matrix. More details on this process can be found, for instance, in [2] and [3]. From now on, unless stated otherwise, $\mathrm{ch}(\Phi) = 0$.

Theorem 3.1 *The* 1*-identities of* A *follow from:*

$$\{b, a, a\} = \{a, a, b\}. \tag{5}$$

Sketch of the proof using `GAP`

The $\{\mathcal{S}_3\}$*-monomials* (the terms arising from the action of the six permutations in $\mathcal{S}_3$ over the arguments of $\{a, b, c\}$) can be seen as elements of the $\mathcal{S}_3$-module generated by the *basic monomials* $(b, c)\, a$, $(a, c)\, b$, $(a, b)\, c$, and $[a, b, c]$. We create the expansion matrix of $\{.,.,.\}$ in degree 1, which has size 4×6 and that we denote by $\mathcal{X}$. Its rows are labeled by the basic monomials, its columns are labeled by the nonassociative monomials and the entry x_{ij} is given by the coefficient of the i-th basic monomial in the expansion of the j-th $\{\mathcal{S}_3\}$-monomial. Then we compute the nullspace of $\mathcal{X}$, which is spanned by the 1-identities of A:

$$\begin{aligned} \{a, b, c\} + \{a, c, b\} - \{b, c, a\} - \{c, b, a\} &= 0, \\ -\{a, c, b\} + \{b, a, c\} + \{b, c, a\} - \{c, a, b\} &= 0. \end{aligned}$$

 ISBN: 978-84-612-9727-6

It is clear that the second identity can be obtained by the action of the transposition $(a\ b) \in \mathcal{S}_3$ over the first one. So, the first identity generates the whole space of 1-identities of A under the action of $\mathcal{S}_3$. ■

It is always possible to obtain identities of a certain degree d ($d > 1$) from identities of degree $d-1$, procedure which is known as *lift*. The two possible ways to do it are: replacing one variable by a triple, or embedding the identity in a triple. But the interesting part related to the problem of searching 2-identities of A is to find the new ones, that is, those that can not be obtained, in any manner, from the ones of the previous degree. Next result gives one of them.

Theorem 3.2 (Pozhidaev, [6]) *In A the following identity holds:*

$$\{\{a,b,c\},d,e\} = \{a,b,\{c,d,e\}\}. \tag{6}$$

Theorem 3.3 *The identities* (5), (6) *and*

$$\{\{a,b,b\},c,d\} = \{a,\{b,b,c\},d\} \tag{7}$$

$$\{\{a,d,c\},a,b\} = \{a,\{a,c,d\},b\} \tag{8}$$

imply all 2*-identities of A.*

Sketch of the proof using `GAP`

By Theorem 3.2, we only have to consider the following association types to construct the expansion matrix for $\{.,.,.\}$ in degree two: $\{\{a,b,c\},d,e\}$ and $\{a,\{b,c,d\},e\}$. This matrix $\mathcal{X}$ has size 40×240, since in degree 2 there are 240 $\{\mathcal{S}_5\}$-monomials and 40 basic monomials. So, the columns of $\mathcal{X}$ are labeled by the $\{\mathcal{S}_5\}$-monomials: the first 120 columns correspond to the monomials of the first association type ordered lexicographically, while the last 120 columns correspond to the second association type in the same order.

Taking into account that the `GAP`-command `NullspaceMat` returns the nullspace of the transposed of the considered matrix, we create $\mathcal{X}^T$. After making the expansions of the two mentioned association types as linear combinations of the basic monomials, we:

- store the basic monomials, organized by four types, in an assigned list;
- create a matrix with 240 rows and 40 columns, where we store the expansion of $\{\{a,b,c\},d,e\}$ in line 1 and the expansion of $\{a,\{b,c,d\},e\}$ in line 121;
- create the other lines through the action of $\mathcal{S}_5$ over the previous two, using the `GAP`-commands `OnTuplesSets` and `OnTuplesTuples` according to the different types of basic monomials;
- use the `GAP`-command `NullspaceMat`, which returns a basis of the nullspace of $\mathcal{X}$.

 ISBN: 978-84-612-9727-6

As $\mathcal{X}$ has rank 40, the nullspace $\mathcal{N}$ of $\mathcal{X}$ has dimension 200. We select the three 2-identities of A:

$$-\{\{a,e,c\},d,b\}-\{\{a,c,e\},d,b\}+\{a,\{e,c,d\},b\}+\{a,\{c,e,d\},b\}=0, \tag{9}$$

$$\begin{aligned}-\{\{c,a,e\},d,b\}-\{\{a,c,e\},d,b\}+\{\{a,d,c\},e,b\}+\{\{e,d,c\},a,b\}\\ +\{e,\{c,a,d\},b\}-\{a,\{e,c,d\},b\}=0,\end{aligned} \tag{10}$$

$$\begin{aligned}\{\{a,d,c\},e,b\}-\{\{e,b,a\},d,c\}-\{\{b,e,a\},d,c\}-\{\{a,c,d\},b,e\}\\ +\{a,\{b,c,d\},e\}+\{a,\{b,d,c\},e\}=0.\end{aligned} \tag{11}$$

We now apply all the permutations of a,b,c,d,e to these three identities and store the results in a matrix of size 360×240. Then, we use `GAP`-command `RankMat` to compute the rank of this matrix, which is 200. So, these identities generate, under the action of $\mathcal{S}_5$, a subspace of dimension 200, that is, the whole space $\mathcal{N}$. So, (6), (9), (10) and (11) imply all 2-identities of A. ■

Although Theorem 3.1 is stated for $\mathrm{ch}(\Phi)=0$, it can be proved, by a manual way, for $\mathrm{ch}(\Phi)\neq 2$. Using the same approach, we can establish identities (5), (7) and (8) assuming the mentioned characteristic.

Acknowledgements

P. D. Beites was supported by an individual doctoral grant (SFRH/BD/37907/2007) of the FCT (Foundation for Science and Technology of the Portuguese Ministry of Science, Technology and Higher Education). The third author was supported by State Aid of Leading Scientific Schools (project NSh-344.2008.1) and by ADTP "Development of the Scientific Potential of Higher School" of the Russian Federal Agency for Education (Grant 2.1.1.419).

References

[1] P. D. Beites, A. P. Nicolás and A. P. Pozhidaev, *On a ternary quaternion algebra*, in preparation.

[2] M. Bremner and I. Hentzel, *Identities for generalized Lie and Jordan products on totally associative triple systems*, J. Algebra **231** (1) (2000) 387–405.

[3] M. R. Bremner and L. A. Peresi, *Classification of trilinear operations*, Comm. Algebra **35** (9) (2007) 2932–2959.

[4] V. T. Filippov, *n-Lie Algebras*, Siberian Math. J. **26** (6) (1985) 879–891; (translation of *Sib. Mat. Zh.* **26** (6) (1985) 126–140 (russian)).

[5] Y. Nambu, *Generalized Hamiltonian Mechanics*, Phys. Rev. D (3) **7** (8) (1973) 2405–2412.

[6] A. P. Pojidaev, *Enveloping algebras of Filippov algebras*, Comm. Algebra **31** (2) (2003) 883–900.

[7] L. Takhtajan, *On foundation of the generalized Nambu mechanics*, Comm. Math. Phys., **160** (2) (1994) 295–315.

[8] The GAP Team, `GAP` - *Groups, Algorithms and Programming*, U. St. Andrews, available at http://www.gap-system.org/.

[9] W. Ling, *On the structure of n-Lie algebras*, Thesis, Siegen University-GHS-Siegen, iv., 1–61, 1993.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Computer-aided clinker analysis

Bilbao-Castro, J.R.[1], **Martínez, J.A.**[1], **Márquez, A.L.**[1], **García, I.** [1] **and Fernández, J.J.**[1]

[1] *Department of Computer Architecture and Electronics, University of Almería*

emails: jrbcast@gmail.com, jmartine@ual.es, amarquez@ace.ual.es, igarcia@ual.es, jose@ace.ual.es

Abstract

Modern approaches to evaluate the quality of cement are based on microscopy. This evaluation is performed on samples of the cement, clinker nodules, that are prepared with different etches and imaged with, typically, optical microscopy. The abundance of the different minerals, i.e. clinker phases, determine the performance of the cement. As a consequence, the goal is to identify and quantify the amount of these phases in order to characterize the cement quality. Traditionally, microscopy images from cement clinker are analyzed by visual inspection. This is a subjective approach, apart from being a tedious and time-consuming process. The current computer-aided image analysis systems of cement clinker are still made up of relatively simple image processing operations, and are used as mere supporting tools for the users. This work presents a more sophisticated approach for interpretation and quantitative assessment of cement clinker images. It is based on a combination of complex image processing operators that allow a more objective, reproducible analysis of the cement performance. The key of the approach is the segmentation using the Watershed transform, which is based on topology, rather than density, and hence allows processing of images taken under difficult experimental situations. After segmentation, labelling and quantification is straightforward. Our approach thus succeeds in facilitating quantitative analysis of cement clinker so that the users can characterize the performance of the cement.

Key words: clinker, cement, image processing, watershed
MSC 2000: AMS codes (optional)

1 Introduction

Cement industry requires procedures to assess the quality of final and intermediate products. The performance of the cement depends on its composition, the morphology of its microstructure and the manufacturing process. As a consequence, the evaluation of the cement quality is mainly focused on the analysis of the phase composition of the

cement and its texture (crystal size, morphology, abundance and distribution). This is also the procedure to follow when different manufacturing processes are to be compared in terms of the performance of the resulting cement. Early approaches for cement performance evaluation were based on chemistry [1]. Modern approaches are rather based upon microscopy, as it provides a direct means to quantify microstructure features. Moreover, there already are guides, compiled by specialists in the field, to facilitate the interpretation of the microscopical observations [2, 3]. However, automated image analysis would be of invaluable help as it could allow fast, reproducible and objective quantification of the properties.

Cement clinker nodules are manufactured from a mixture of, mainly, limestone and clay, which is sintered in a kiln at high temperatures. These clinker nodules are then crushed to yield a fine powder to produce cement. The procedure to analyze the cement [3] starts with the extraction of a sample of clinker, either uncrushed or partially crushed, that is then embedded in epoxy resin. A section is then cut and polished to reveal a cross section. Different chemical etches are then used to ease analysis of the various phases of the clinker and reveal different features. The prepared cross section is then observed with an optical microscope using reflecting light, with magnifications typically in the range 100-500x [3]. Scanning electron microscopes are also used in the field [3, 4], though optical microscopes are widespread and much less costly. Microscopy images acquired with a CCD camera are then subjected to visual inspection and quantitative analysis in terms of phase composition and texture.

The raw materials used for the cement and the manufacturing process produce the various clinker phases. The abundance of the different phases will determine the performance of the cement at early and later stages. Three main phases are typically found in the clinker [3]. The two most abundant are termed 'alite' and 'belite', whereas the third one, which is in fact a mixture of many other phases much less abundant, is termed 'liquid phase' or 'matrix'. Alite appears as angular crystals whose size is indicative of the early strength of the cement. Belite appears as striated rounded crystals whose size is indicative of the later-age strength. Therefore, quantification of the abundance of these phases is essential to characterize the performance of the underlying cement. Apart from the abundance, other features of the phases (e.g. morphology, texture) are also useful to determine the performance more precisely (for details, see [3]).

Automated image analysis has achieved an increasing interest in the field since the end of nineties. The reasons are its abilities to obtain objective, quantitative measures of the clinker phases and other morphological features from the images in an automated or semi-automated fashion. Traditionally, microscopy images from cement clinker have been, however, analyzed by visual inspection, which is a time-consuming process that requires profound knowledge and skill obtained after a long, accumulated experience and practice. In that sense, guides prepared by specialists have been crucial to the training of users [2, 3].

Computer-aided image analysis systems for clinker images are intended to provide users with qualitative and quantitative information of the cement in an automated or semi-automated way. These systems have traditionally relied upon relatively simple image processing techniques [5, 6, 4]. Though simple, so far they have turned out to

 ISBN: 978-84-612-9727-6

be of some help for users by supplying with estimates of the percentage of the minerals (alite, belite, liquid phase) present in the sample. These measures have helped the users characterize the performance of the cement from relatively good images. However, time-consuming user intervention or complicated heuristic rules have to be used to refine the results [6].

In this work, we propose the use of more sophisticated image processing methods for the analysis of clinker images. These methods are more suitable to work under difficult situations where recognition of the clinker phases by simple thresholds are precluded by noise or the presence of numerous residual small objects with high density. In this work, we propose the use of the Watershed transform to identify the phases. This technique is based on topology, not density levels, and is thus more appropriate to deal with these difficult cases. Afterwards, a labelling procedure is used to obtain quantitative estimates of the phases. Therefore, this approach is more suitable for computer-assisted clinker analysis in an automated or semi-automated way, reducing the overhead of the user intervention and increasing the productivity.

2 Methods

Computer-aided image analysis of clinker images have been based on relatively simple techniques thus far. These techniques mainly rely on segmentation based on thresholds [5, 6, 4]. In some of these works, simple morphological operators were also used to refine the segmentation results and remove small objects [6], or the own user could refine the result thanks to a graphical user interface [5]. Once the images are segmented into classes (alite, belite, liquid phase), a simple sum of the number of pixels for each phase is computed to yield the area fraction of each cement component, which is indicative of the ratio of the volume of a mineral to that of all minerals in the cement sample. This, in turn, help to quantitatively characterize the performance of the cement.

In this work, we propose to use the Watershed transform to segment the clinker images into the phases and a subsequent labelling procedure to provide quantitative measures of the phases of the sample. The Watershed technique carries out the segmentation based on the topology of the image. Therefore, it is better suited for segmentation of the clinker images because it is not based on density thresholds and thus avoids the need to devise complicated heuristic approaches to find out a proper global threshold and to get rid of residual small objects [6].

Prior to the segmentation, a filtering step is needed to reduce the noise of the image in order to maximize the chances for an automated segmentation procedure. Classical Gaussian or kernel-based filtering methods succeed in reducing the noise, but at the expense of blurring the edges. Therefore, they should be avoided; otherwise, the resulting filtering image would have the limits of the regions blurred, thus make segmentation more challenging. Instead, the median filtering is a classical nonlinear filtering method that manages to clean noise while preserving the edges and the features present in the image [10].

Therefore, our approach for computer-assisted image analysis of clinker images

 ISBN: 978-84-612-9727-6

consists in (1) preliminary filtering of the image with the median filtering, (2) segmentation of the image into regions by means of the Watershed transform, (3) some possible user interaction for refinement purposes, and (4) region labelling. Afterwards, (5) the statistical analysis of the clinker phases encompassed by the region limits is straightforward, thereby providing users with quantitative information immediately. As the filtering step is basically a preprocessing step for the segmentation, hereafter it will be considered as part of the segmentation itself.

2.1 Image preprocessing

An image normalization step is usually desirable before applying certain image processing algorithms. Thus, undesired features present in the images are removed or attenuated so subsequent image processing algorithms yield more robust results. A typical normalization step consists of improving images signal-to-noise ratio (SNR) through noise reduction. Apart from noise reduction, such techniques can also be applied to eliminate very small details that are not of importance for the final result and would disturb the normal functioning of the applied algorithms. In our case, when segmenting clinker images we do not want very small details appear as single regions but to be a part of other, bigger neighbour regions. There exist many different noise reduction algorithms that could help to pre-process clinker images. Nevertheless, more advanced image filtering algorithms are usually computationally very expensive.

2.1.1 Median filtering

The median filtering algorithm is a noise reduction algorithm based on assigning each pixel the median value of it and its neighbours. It removes noise as well as small details while preserving global features of the image. The user will determine how many pixels are to be considered as neighbours by selecting a maximum distance for them (in pixels) in relation to the central pixel. Figure 1 shows an example of the median filter with a neighbourhood distance=1. In this case, 180 (the value of the centre's pixel) is a value that diverges a lot from the neighbours. Therefore, it is likely to be a result of noise and, hence, it should be ignored and changed it by the median filtering. This makes that the final value for the pixel should be similar to the neighbour pixels. It the pixel's value was similar to the surrounding values, it would conserve its original value (or quite a similar one).

As shown, filtering a single channel (i.e. grey-scale) image using the median filtering is quite simple (and fast) [10]. Nevertheless, clinker images are usually coloured and not grey-scale. Therefore, we should consider how to filter images having three different channels (Red, Green, Blue, RGB) and using the median filter. The so-called vector median filtering [11] is a good response. It considers RGB components as members of a vector and filtering is done by means of vectors instead of single values. The theory sounds good but its implementation is more complex and the algorithms takes much longer time than the single-channel implementation. We propose to consider the RGB image as three independent images and process each of them separately. Our experience

 ISBN: 978-84-612-9727-6

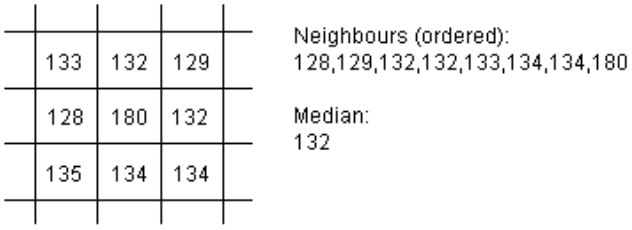
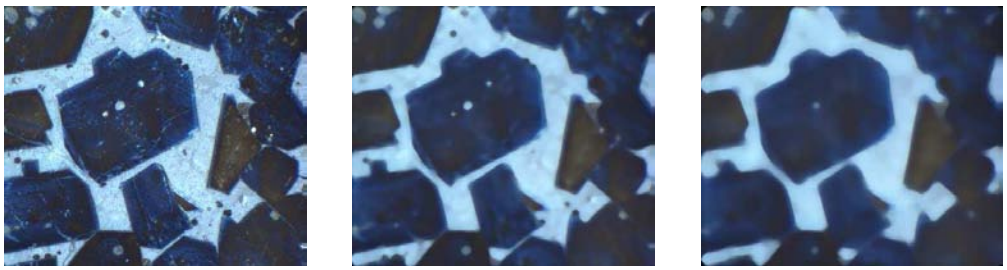


Figure 1: Pixel update based on neighbourhood in median filtering algorithm



(a) (b) (c)

Figure 2: (a) the original clinker image. Note all the small elements that we do not really need to perform a segmentation. (b), the same filtered image using a median filter as proposed in text, with a neighbourhood distance N=4. (c), in this case, a neighbourhood distance of N=8 is used. Note that as the neighbourhood distance is increased, more information is lost on small details.

(see Figure 2) has shown that results do not differ significantly between those obtained using the vector implementation and those obtained through the three independent channels one. Therefore we suggest that a simple median filter is used to pre-process clinker images.

2.2 Image segmentation

Image segmentation is an image processing technique consisting of partitioning the original image into regions of interest (sets of pixels). Such regions are differentiated depending on pixels similarities regarding features like colour, texture, brightness, etc.. Image segmentation is an essential tool for many computer vision applications where meaningful areas must be identified in the image prior to its analysis. Therefore, image segmentation converts a raw image (data) into useful information that can be interpreted. Different algorithms fit better for different purposes and a satisfactory general purpose segmentation algorithm does not exist.

Image segmentation based on histograms has been broadly used for clinker images analysis [6]. Its main advantage is its ease of implementation and its computational

simplicity. Nevertheless, tuning this kind of algorithms for a proper image segmentation is mainly based on human expertise and does not always yield good results. Other image segmentation algorithms families rely on more sophisticated approaches to obtain better results with a reduced human expertise need. In this regard, we propose a watershed transform-based segmentation algorithm as a good candidate for clinker images segmentation.

2.2.1 The watershed transform

The watershed transform allows general-purpose image segmentation based on "topological" characteristics of the image. Pixels are assigned a "height" depending, for example, on the image gradient. This way, lower pixels will be those in zones where there is a small variation between neighbour pixels, On the other hand, higher pixels correspond to zones of the image where gradient is bigger. The final result of the transformation can be seen as a geographical map with watersheds and catchment basins (see Figure 3). From this point the idea is to flood the map with simulated water until only the watersheds remain in form of lines delimiting different segments of the image.

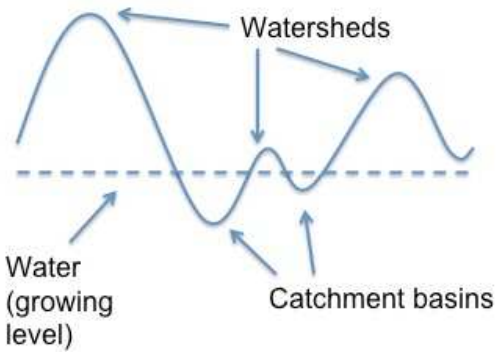


Figure 3: Conceptual explanation of the watershed algorithm.

There exist different implementations of the watershed algorithm. All of them divide the image into many small pieces, generally leading to an over-segmentation. Some variants of the algorithm allow to reduce this over-segmentation, leading to more reasonable results. Nevertheless, we consider that such approaches (like Multi-resolutional [8, 9] or marker-driven) do not fit well to our problem, where very small details of the clinker phases would pass unnoticed. From the over-segmented image a new process starts that will join small, neighbour areas into bigger ones based on their similarity. This step is crucial to obtain a satisfactory segmentation of the image.

Computationally speaking, the watershed transform is costly in terms of required CPU time. It is an iterative process with two clearly differentiated parts. First, the gradient image is divided into a myriad of small regions based on similarity between pixels. Second, the resulting set of small regions is iteratively reduced by generating bigger regions. A new region is created by merging two neighbour regions whose similarity is maximum. This last step is, by far, the most computationally intensive

stage and involves to compare all possible combinations of neighbour regions pairs and update such values each time two regions are merged.

2.3 Image regions labelling

Generally, after a segmentation process comes the labelling process. It consists of assigning a meaning to the resulting segmented regions. That is assigning each region a label which describes it depending on the nature of the problem being treated. For example, in clinker analysis, once the image is segmented, it is interesting to assign materials (alite, belite, etc..) or features (cracks, pores, etc..) to segmented regions. Once labels are applied it is possible to extract interesting statistics about mineral clinker composition.

3 Workflow proposal

We propose the next protocol for segmenting clinker images (see Figure 4):

a) Filter image to reduce noise. This step prevents over-segmentation of the image. A median filter is advised due to its simplicity and good results without a big lost in image information.

b) The filtered image is transformed into a topological representation; for example, the gradient operator is performed. The resulting image is grey-scale, with darker zones being those of less change between neighbour pixels.

c) The watershed algorithm is applied, over the resulting image from the previous point, with an objective of N regions. This means that the image is over-segmented and then regions are iteratively joined based on their similarity until N regions are remaining. N is user-selected and will vary between images. What we should have here is a black/white image, with i.e. white representing the lines delimiting each region.

d) The user fine-tunes the result by joining by hand some remaining regions (if needed). This would give a very close approximation to the ideal segmentation the user is seeking. For a better user experience, we propose that regions are somewhat depicted on top of the original, coloured image. This way, the user gets the original image with lines delimiting the different regions.

e) Very small details that passed unnoticed can be segmented by re-running from 'a' in subregions within the image.

4 Results

Some graphical results are shown in Figure 5. From top to bottom; on the left we have an original microscopic image of clinker grains. Next, and after a previous median

 ISBN: 978-84-612-9727-6

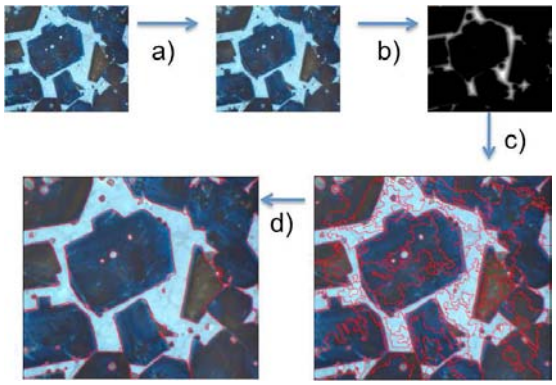
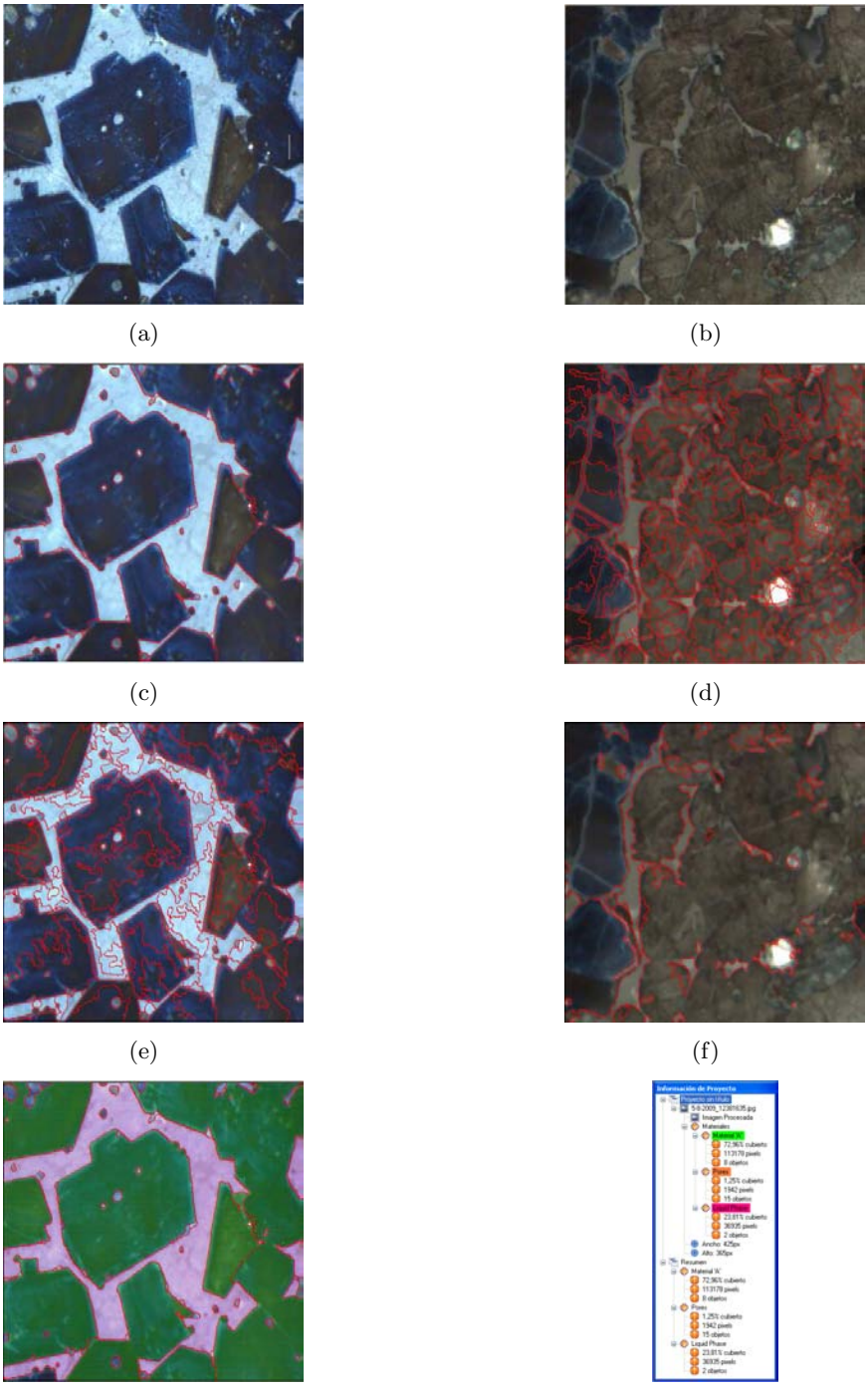


Figure 4: This image shows the proposed workflow for segmenting clinker images. It starts with a microscopic image of a clinker preparation. The filtering stage follows with the application of a median filter. Later, the gradient image is calculated as the input for the watershed algorithm. The last two images represent the oversegmented image and the fine-segmented result.

filtering of the image with a neighbourhood distance of 2, the watershed algorithm is run with an objective number of regions N=200. It is clear that some oversegmentation (despite filtering) has happened. To reduce oversegmentation, the user just needs to select those regions wanted to be joined and create a single bigger region. The third image shows the result after user's intervention, with regions clearly separating different elements of the image. Finally, different regions of the segmented image are labelled with different meanings (alite, belite, etc..). The last image on the right column shows statistical results related to this image, indicating the percentage of each component present in the preparation. The three first images on the right column show the same procedure performed over a different, more complex clinker image. As shown, and in spite of the more complex nature of the image, the process results in a satisfactory segmentation of the image.

5 Conclusions

Quantitative and qualitative analysis of clinker grains is an important step to asses the quality and characteristics of cements. Clinker image analysis gives a clear response about the influence of the different materials and techniques used to prepare it. Therefore, a broad knowledge of the process of clinker formation is of vital importance for cement manufacturers. Currently, clinker images analysis is performed by hand by experienced workers who determine the composition of the clinker and its features (holes, pores, cracks, etc..). The work proposed here suggests that computer aided analysis of clinker grains should be possible through different image processing techniques. More specifically, the application of a watershed algorithm to perform clinker image segmentation has proven to be very efficient when complementing user interaction. The most important part of the process is the image segmentation as labelling is trivial once



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5: Graphical results obtained when analyzing different clinker images.

 ISBN: 978-84-612-9727-6

different regions of the image are defined. From a computational point of view, image filtering as well as image labelling are cheaper in terms of computing demands. Nevertheless, the segmentation process is quite time-consuming. We propose parallelizing the algorithm taking advantage of new multi-core processors and/or GPUs.

Acknowledgements

The authors wish to thank S. Martínez, A.B. Ibernón and F. Martínez (Holcim España S.A. - Fábrica de Gádor) for helpful disscussions and support throughout the work. Project supported by Holcim España S.A. through the contract 400629 (ClinkerView) with the University of Almería. Additional support from grants MCI-TIN2008-01117 and JA-P06-TIC01426 is also acknowledged. J.R. Bilbao-Castro is a fellow of the Spanish Juan de la Cierva postdoctoral contract program, co-financed by the European Social Fund.

References

[1] H. F. W. Taylor, *Cement industry, 2nd ed.*, Thomas Telford publishing, London, 1997.

[2] Y. Ono, *Fundamental Microscopy of Portland Cement Clinker*, Chichibu Onoda Cement, Chichibu Onoda Cement Co. (1995) 192–196.

[3] D. H. Campbell, *Microscopical Examination and Interpretation of Portland Cement and Clinker*, Natalie C. Holz (Editor). Portland Cement Association. Stokie (USA), 1999.

[4] P. Stutzman, *Scanning electron microscopy imaging of hydraulic cement microstructure.*, Cement and Concrete Composites **26** (2004) 957–966.

[5] K. Theisen, *Quantitative Determination of Clinker Phases and Pore Structure Using Image Analysis*, Proceedings of the Nineteenth International Conference On Cement Microscopy (1997) 30–44, Cincinnati, Ohio.

[6] M. Jourlin, B. Roux and R. M. Faure, *Recognition of clinker phases by automatic image analysis.*, Cement and Concrete Composites **23** (2001) 207–214.

[7] H. Sun, J. Y. Yang and M. Ren, *A fast watershed algorithm based on chain code and its application in image segmentation*, PRL **9** (2005) 1266–1274.

[8] J. B. Kim and H. J. Kim, *Multiresolution-based watersheds for efficient image segmentation*, PRL **24** (2003) 473–488.

[9] Y. Hu and T. Nagao, *A matching method based on marker-controlled watershed segmentation*, In proc. of ICIP04 **1** (2004) 283–286.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 3rd edition*, Prentice Hall, 2008.

[11] J. T. Astola, P. Haavisto and Y. Neuvo, *Vector Median Filters*, Proc. IEEE **78** (1990) 678–689.

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009 30 June, 1–3 July 2009.

Fractional calculus and Levy flights: modelling spatial epidemic spreading

João Pedro Boto[1] **and Nico Stollenwerk**[1]

[1] *Centro de Matemática e Aplicações Fundamentais, Universidade de Lisboa, Avenida Prof. Gama Pinto 2, 1649-003 Lisboa, Portugal*

emails: jboto@ptmat.fc.ul.pt, nico@ptmat.fc.ul.pt

Abstract

We investigate fractional derivatives, especially fractional Laplacian operators, leading to Lévy flights. These notions will be applied to epidemic processes, like the stochastic spatially extended SIS process and models with reinfection, as super-diffusion is a more realistic mechanism of spreading epidemics than ordinary diffusion.

Key words: fractional calculus, fractional Laplace operator, Lévy flight, spatial stochastic epidemics, Kolmogorov-Fisher equation

1 Introduction

Classical derivatives of integer order have been generalized historically in various ways to derivatives of fractional order [2]. On important application of such fractional derivatives is the notion of the fractional Laplacian operator in the theory of Lévy flights. This leads to the notion of sub- and super-diffusion, well applicable in reaction-diffusion systems [3]. In epidemiological systems especially the super-diffusion case is of interest as description of more realistic spreading than normal diffusion on regular lattices.

To understand even basic epidemiological processes it is often necessary to investigate well the spatial spreading since all epidemic processes happen on spatially restricted networks [4]. We have previously studied epidemic processes with reinfection on regular lattices [10] as they also appear in the physics literature [7]. A crucial question in such systems is in how far basic notions like finite spreading and phase diagrams hold not only for ordinary diffusion but also in the super-diffusion case [4, 5]. Wider processes with multi strain interaction [9, 1] could be treated similarly. As our prime example here we will investigate the susceptible-infected- susceptible SIS epidemic, which leads in the framework of reaction diffusion processes to the well know Kolmogorov-Fisher equation [8, 6].

2 Historic ways of generalizing derivatives

There are many definitions for the derivative of arbitrary real order μ [2]. The one that we use bellow is based on the fact that the Fourier transform $\mathcal{F}$ of a function satisfies the relation

$$\mathcal{F}\left(\frac{\partial f}{\partial x_\nu}\right) = ik_\nu \mathcal{F}(f).$$

For any constant coefficient partial differential operator

$$P\left(\frac{\partial}{\partial x_1}, ..., \frac{\partial}{\partial x_n}\right),$$

where P is a polynomial in n variables, we thus have

$$\mathcal{F}\left(P\left(\frac{\partial}{\partial x_1}, ..., \frac{\partial}{\partial x_n}\right) f\right) = P(ik_1, ..., ik_n).$$

In particular, when $P(x_1, ..., x_n) = -|x|^2 = -x_1^2 - ... - x_n^2$then

$$P\left(\frac{\partial}{\partial x_1}, ..., \frac{\partial}{\partial x_n}\right) = \Delta$$

is the n dimensional Laplace operator. The Weyl derivative is then defined by

$$\mathbf{D}^\mu f = \mathcal{F}^{-1}\left[|k|^\mu \mathcal{F}[f]\right]$$

with $k = (k_1, ..., k_n)$. Symbolically $\mathbf{D}^\mu = (-\Delta)^{\mu/2}$ is called the fractional power of the Laplacian of exponent $\mu/2$. We will use the definition of the Laplacian via the Fourier representation below. In the following we will investigate ordinary diffusion in more detail and show how to generalize to super-diffusion. As an example for an application we will investigate the so called susceptible-infected-susceptible epidemic process, which leads in approximation neglecting higher correlation to the well known Kolmogorov-Fisher equation. This can easily be extended to other epidemic processes, like the SIR system or such with reinfection [10].

3 Ordinary diffusion

The simple stochastic differential equation

$$\frac{d}{dt}\ x = \varepsilon(t) \tag{1}$$

with a random variable $\varepsilon(t)$ describes a one-dimensional random walk in space. For probability distributions with finite variance and independent random draws at each time step the distribution of the process converges to a Wiener process with Gaussian distribution.

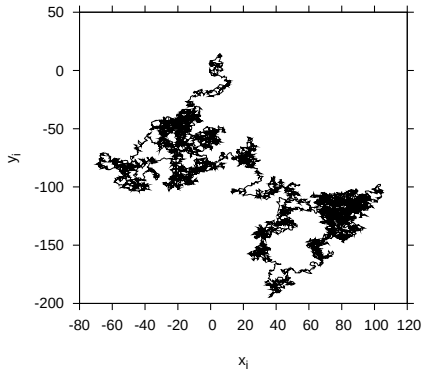


Figure 1: *Random walk with Gaussian distributed steps.*

Hence, for simplicity we can start the process immediately with Gauss normally distributed and stochastically independent random kicks

$$p(\varepsilon) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{\varepsilon^2}{2}} \quad . \tag{2}$$

For the Langevin-type equation (1) and independent Gaussian noise, Eq. (2), we obtain for the distribution of the process $p(x,t)$ the Fokker-Planck equation as a simple diffusion equation

$$\frac{\partial}{\partial t}\ p(x,t|x_0,t_0) = \frac{1}{2}\frac{\partial^2}{\partial x^2}\ p(x,t|x_0,t_0) \tag{3}$$

with the solution

$$p(x,t|x_0,t_0) = \frac{1}{\sqrt{2\pi(t-t_0)}}\ e^{-\frac{(x-x_0)^2}{2(t-t_0)}} \tag{4}$$

which is a Gaussian distribution with mean value $\mu = x_0$ and time dependent variance $\sigma^2 = (t-t_0)$. The mean displacement $\sqrt{\langle (x-x_0)^2 \rangle}$ has the famous $\sqrt{t}$ behaviour.

3.1 Simulation of an ordinary diffusion process

To give an impression how processes under ordinary diffusion and under super-diffusion work, we will first simulate a random walker for graphical clarity in two dimensions. Hence the Langevin equations for the x and y components of the vector $\underline{x} = (x,y)^{tr}$ are given by

$$\begin{aligned} \frac{d}{dt}x &= \varepsilon(t) \\ \frac{d}{dt}y &= \eta(t) \end{aligned} \tag{5}$$

with independent noise sources $\varepsilon(t)$ and $\eta(t)$. Or simply given in time discrete form we have

$$\begin{aligned} x_{n+1} &= x_n + \varepsilon_n \\ y_{n+1} &= y_n + \eta_n \end{aligned} \tag{6}$$

 ISBN: 978-84-612-9727-6

The Gaussian distributed random numbers ε_n and η_n can be generated from uniformly distributed random numbers on the unit interval, as given e.g. by the Marsaglia random generator, by the Box-Muller algorithm. Fig. 1 shows a simulation of a Gaussian random walker in two dimensions, starting at the origin, for 10 000 iteration steps. A random walker going with equal probability to one of its four von Neumann neighbouring sites on a regular two dimensional lattice would look on a large scale similar to the Gaussian random walker. The distribution $p(x,t|x_0,t_0)$ of the lattice random walker converges for long times and distances to the distribution of the Gaussian random walker.

3.2 Fourier representation of the ordinary diffusion process

The Fourier transform of the probability $p(x,t) := p(x,t|x_0 = 0, t_0 = 0)$ of the Wiener process, Eq. (4), is simply

$$\tilde{p}(k,t) = e^{-k^2 t} \tag{7}$$

and the Fokker-Planck equation is in Fourier space given by

$$\frac{\partial}{\partial t}\ \tilde{p}(k,t) = -k^2 \cdot \tilde{p}(k,t) \tag{8}$$

which now can be easily generalized to other powers of k than the power of 2 for normal diffusion.

4 Super-diffusion

To describe super-diffusion we generalize the Fourier representation of the diffusion process to $\mu \in (0,2]$ in the solution

$$\tilde{p}(k,t) = e^{-|k|^\mu t} \tag{9}$$

which corresponds to

$$\frac{\partial}{\partial t}\ \tilde{p}(k,t) = -|k|^\mu \cdot \tilde{p}(k,t) \tag{10}$$

in the Fokker-Planck equation. By inverse Fourier transformation we obtain in real space representation

$$\frac{\partial}{\partial t}\ p(x,t) = -\left(-\Delta_x\right)^{\mu/2} p(x,t) \quad . \tag{11}$$

For $\mu \in (0,1)$ the fractional Laplacian operator $(-\Delta_x)^{\mu/2}$ is given by

$$\left(-\Delta_x\right)^{\mu/2} p(x,t) = C_\mu \int_{-\infty}^{\infty} \frac{p(x,t) - p(y,t)}{|x-y|^{1+\mu}}\, dy \tag{12}$$

with constant

$$C_\mu = \frac{2^{-\mu}\pi^{3/2}}{\Gamma(1+\frac{\mu}{2})\Gamma(\frac{1+\mu}{2})\sin\left(\frac{\mu\pi}{2}\right)} \quad .$$

 ISBN: 978-84-612-9727-6

For $\mu \in (1,2)$ the fractional Laplacian operator $(-\Delta_x)^{\mu/2}$ is given by

$$(-\Delta_x)^{\mu/2} p(x,t) = C_\mu \int_{-\infty}^{\infty} \frac{\triangle_{x-y}^{-} [p(x,t) - p(y,t)]}{|x-y|^{1+\mu}} \, dy \tag{13}$$

where $\triangle_{x-y}^{-}$ is the retarded-difference operator,

$$\begin{aligned} \triangle_{x-y}^{-} [p(x,t) - p(y,t)] &= p(x,t) - p(y,t) - (p(x-(x-y),t) - p(y-(x-y),t) & (14) \\ &= p(x,t) - 2p(y,t) + p(2y-x,t) \quad , & (15) \end{aligned}$$

with constant

$$C_\mu = \frac{\left(2^{1-\mu} - 1\right) \pi^{3/2}}{\Gamma(1+\frac{\mu}{2}) \Gamma(\frac{1+\mu}{2}) \sin\left(\frac{\mu\pi}{2}\right)} \quad .$$

Or as master equation it can be written

$$\frac{\partial}{\partial t} \; p(x,t) = \int w_{x|y} p(y,t) - w_{y|x} p(x,t) \, dy \tag{16}$$

with transition rate

$$w_{x|y} = \frac{C_\mu}{|x-y|^{1+\mu}} \tag{17}$$

for $\mu \in (0,1)$ and

$$w_{x|y} = \frac{C_\mu}{|x-y|^{1+\mu}} \triangle_{x-y}^{-} \tag{18}$$

for $\mu \in (1,2)$. The solution in real space representation for $t > t_0$ is given by

$$p(x,t|x_0,t_0) = \frac{1}{2\pi} \int e^{-ik(x-x_0) - |k|^\mu (t-t_0)} \, dk \tag{19}$$

or with the function

$$G_\mu(z) = \frac{1}{2\pi} \int e^{-ikz - |k|^\mu} \, dk \tag{20}$$

the solution is

$$p(x,t|x_0,t_0) = \frac{1}{(t-t_0)^{1/\mu}} \, G\left(\frac{x-x_0}{(t-t_0)^{1/\mu}} \right) \quad . \tag{21}$$

The function $G_\mu(z)$ has for large argument $|z| >> 1$ a power law tail

$$G_\mu(z) \sim \frac{1}{|z|^{1+\mu}} \tag{22}$$

which however shows up rather slowly, since the series expansion of $G_\mu(z)$ is given by

$$G_\mu(z) = \frac{1}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k!} \, \Gamma(1+k\mu) \sin\left(\frac{\pi}{2} k\mu\right) \cdot z^{-(k\mu+1)} \tag{23}$$

hence decreases for $k=1$ as $|z|^{-(1+\mu)}$, but higher order terms die off only very slowly.

 ISBN: 978-84-612-9727-6

4.1 Cauchy process

For $\mu = 1$ the integral in the function $G_\mu(z)$ can be solved and gives a Cauchy distribution

$$G_{\mu=1}(z) = \frac{1}{\pi(1+z^2)} \tag{24}$$

leading to the Cauchy process as a special case of super-diffusion.

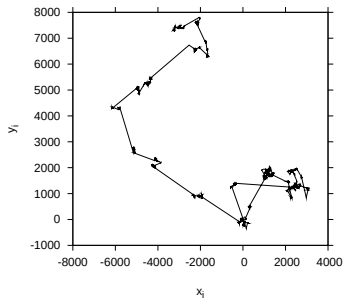


Figure 2: *Lévy flight with Cauchy distributed steps, i.e. Lévy exponent* $\mu = 1$.

4.2 Simulation of super-diffusive Lévy flights

A Cauchy distribution can simply be obtained numerically by dividing two Gauss normally distributed random variables. The simulation of a Cauchy flight in two dimensions is shown in Fig. 2.

For any other Lévy flight exponent μ the generation of random numbers is a bit more involved. We simulate in two dimensions the map

$$\begin{aligned} x_{n+1} &= x_n + r_n \cos(2\pi\varphi_n) \\ y_{n+1} &= y_n + r_n \sin(2\pi\varphi_n) \end{aligned} \tag{25}$$

in random polar coordinates, with φ_n uniformly distributed in the unit interval $[0, 1]$ and r_n random numbers with infinite variance and power law tail $|r|^{-(1+\mu)}$ with Lévy exponent μ.

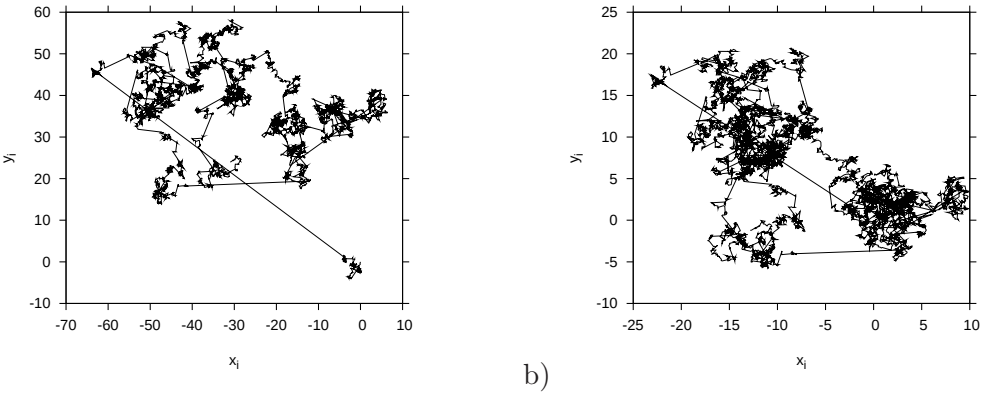
Since the distribution $G_\mu(z)$ cannot be evaluated analytically, and also no closed invertible cumulative distribution function can be given, we cannot simply draw a random number from this Lévy flight distribution. But we can obtain random numbers r with a power law tail, and have to use an upper cut-off r_0. Such a distribution is

$$p(r) = \frac{\mu}{2} r_0^\mu \quad \cdot \quad \begin{cases} |r|^{-(1+\mu)} & \text{for } r_0 \le |r| \le \infty \\ 0 & \text{else} \end{cases} \tag{26}$$

and obtained by using a shot noise distributed random variable s, hence s takes the values -1 and +1 with equal probability, and a uniformly distributed random variable c in the unit interval via

$$r := \frac{r_0 \cdot s}{(1-c)^{1/\mu}} \quad . \tag{27}$$

 ISBN: 978-84-612-9727-6



a) b)

Figure 3: *a) Lévy flight with exponent* $\mu = 1.5$ *and b) with* $\mu = 1.8$.

The distributions of s and c are given by

$$p(s) = \frac{1}{2}\Big(\delta(s+1) + \delta(s-1)\Big) \tag{28}$$

and

$$p(c) = \begin{cases} 1 & \text{for } c \in [0,1] \\ 0 & \text{else} \end{cases} \tag{29}$$

Fig. 3 shows a Lévy flight with exponent $\mu = 1.5$ in two dimensions. As cut-off we use $r_0 = 0.1$. The qualitative differences between ordinary diffusion and super-diffusion with various exponents becomes well visible in the simulations. The Lévy flight type moving pattern of individuals, many local steps but the occasional long distance journey, has consequences for epidemic models of disease spreading in physical space.

4.3 Generalization of fractional Laplacians to higher dimensions

The generalization of the Laplace operator to higher dimensions is straight forward when considering the Fourier representation, hence

$$\mathcal{F}[(-\Delta)^{\mu/2} f](\underline{k}) := |\underline{k}|^{\mu} \cdot \tilde{f}(\underline{k}) \tag{30}$$

for $\underline{k} \in \mathbb{R}^n$. Then in real space via inverse Fourier transform we have the representation, for $\mu \in (0,1)$,

$$\left(-\Delta_{\underline{x}}\right)^{\mu/2} f(\underline{x}) = C_{\mu,n} \int \frac{f(\underline{y}) - f(\underline{x})}{|\underline{x} - \underline{y}|^{n+\mu}}\, d^n y \tag{31}$$

with constant

$$C_{\mu,n} = \frac{2^{-\mu}\pi^{1+n/2}}{\Gamma\left(1+\frac{\mu}{2}\right)\Gamma\left(\frac{n+\mu}{2}\right)\sin\left(\frac{\mu\pi}{2}\right)} \quad .$$

via inverse Fourier transform.

5 From stochastic epidemic models to reaction-diffusion processes

The SIS epidemics is an autocatalytic process given by the reaction scheme

$$\begin{aligned} S+I &\xrightarrow{\beta} I+I \\ I &\xrightarrow{\alpha} S \end{aligned}$$

and can be described via a master equation to capture the population noise of the epidemiological model (see [11] for a more detailed description of the SIS process).

The stochastic spatially extended SIS epidemic process on general lattice or network topologies is given by the following dynamics for the probability p of the state of a network

$$\begin{aligned} \frac{d}{dt}p(I_1,I_2,...,I_N,t) &= \sum_{i=1}^{N}\beta\left(\sum_{j=1}^{N}J_{ij}I_j\right)I_i\ \ p(I_1,...,1-I_i,...,I_N,t) \\ &+\sum_{i=1}^{N}\alpha(1-I_i)\ \ p(I_1,...,1-I_i,...,I_N,t) \\ &-\sum_{i=1}^{N}\left[\beta\left(\sum_{j=1}^{N}J_{ij}I_j\right)(1-I_i)+\alpha I_i\right]\ \ p(I_1,...,I_i,...,I_N,t) \end{aligned} \tag{32}$$

for variables $I_i \in \{0,1\}$ and adjacency matrix (J_{ij}). Local quantities like the expectation value of infected at a single lattice point, which in reaction diffusion systems corresponds to the local density $u(x,t)$ are given by

$$\langle I_i\rangle(t) := \sum_{I_1=0}^{1}\sum_{I_2=0}^{1}...\sum_{I_N=0}^{1} I_i\ p(I_1,I_2,...,I_N,t) \quad . \tag{33}$$

For such quantities dynamics can be derived using the original dynamics of the stochastic process description for $p(I_1,I_2,...,I_N,t)$. In such dynamics for local quantities there appears the discretized diffusion operator in the case of lattice models

$$\Delta\langle I_i\rangle := \sum_{j=1}^{N} J_{ij}(\langle I_j\rangle - \langle I_i\rangle) \tag{34}$$

and defines a generalized Laplace-operators for other network topologies, coded in the adjacency matrix (J_{ij}). Considering the local quantity $\langle I_i\rangle(t)$, which in a countinuous space model corresponds to the local density $u(x,t)$ with spatial variable x corresponding to i and lattice spacing a from our lattice model going to zero, we obtain

$$\frac{d}{dt}\langle I_i\rangle = \beta\sum_{j=1}^{N} J_{ij}\langle(1-I_i)I_j\rangle - \alpha\langle I_i\rangle \quad . \tag{35}$$

 ISBN: 978-84-612-9727-6

Hence

$$\frac{d}{dt}\langle I_i\rangle = \beta\sum_{j=1}^{N} J_{ij}(\langle I_j\rangle - \langle I_i\rangle) + \beta\sum_{j=1}^{N} J_{ij}\langle I_i\rangle - \beta\sum_{j=1}^{N} J_{ij}\langle I_i I_j\rangle - \alpha\langle I_i\rangle \quad (36)$$

where we now use the discrete version of the diffusion operator $\Delta\langle I_i\rangle = \sum_{j=1}^{N} J_{ij}(\langle I_j\rangle - \langle I_i\rangle)$ for the first term of the sum on the right hand side of the equation. Further, in the term $-\beta\sum_{j=1}^{N} J_{ij}\langle I_i I_j\rangle$ we apply a local mean field assumption in the sense that local correlations can be neglected and coarse grained hence $\langle I_i I_j\rangle - \langle I_i\rangle\langle I_j\rangle \approx 0$ and $\langle I_i\rangle\langle I_j\rangle \approx \langle I_i\rangle\langle I_i\rangle$. Furthermore, we use $Q_i = \sum_{j=1}^{N} J_{ij}$ for the total number of neighbours of lattice site i, and in regular lattices $Q_i = Q$ as a single constant for the number of neighbours of any lattice site. Hence we finally obtain

$$\frac{d}{dt}\langle I_i\rangle = \beta Q\langle I_i\rangle\Big(1 - \langle I_i\rangle\Big) - \alpha\langle I_i\rangle + \beta\Delta\langle I_i\rangle \quad . \quad (37)$$

This is for lattice spacing going to zero, hence $u(x,t) = \langle I_i\rangle$ nothing but the Kolmogorov-Fisher equation in the form

$$\frac{\partial}{\partial t}\, u = r\, u\left(1 - \frac{u}{k}\right) + \chi\Delta u \quad (38)$$

where we identify the growth rate $r = \beta Q - \alpha$, the carrying capacity $k = \left(1 - \frac{\alpha}{\beta Q}\right)$ and diffusion constant $\chi = \beta$. Often the carrying capacity is simply set to unity, as well as the diffusion constant.

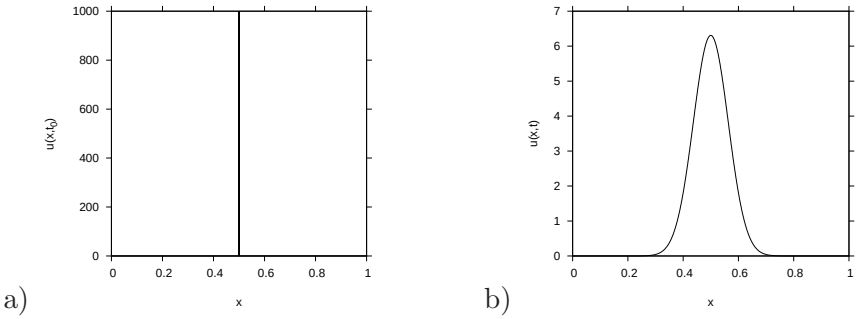


Figure 4: *Integration over the unit interval of ordinary diffusion. a) Initial state is a delta function. b) The final state after some integration time is a Gaussian.*

5.1 First numeric results for reaction diffusion type spatial epidemics

For ordinary diffusion we use the usual discretization for the second derivative in space, hence

$$u_i(t + \Delta t) = u_i(t) + \Delta t \cdot \left(\chi \cdot \frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2}\right) \quad (39)$$

with diffusion constant $\chi = 0.1$. For Fig. 4 we use $\Delta x = 1000$, integration time $t_{max} = 0.02$ and resolution $r_t = 10000$, hence $\Delta t = 0.00002$. The initial δ-peak, Fig.

4 a), is given by zero on the whole unit interval and $1/\Delta x$ at the middle. Then the resulting Gaussian curve, Fig. 4 b), for the final state is invariant under changes of resolution in time and space. The Kolmogorov-Fisher equation shows for slow diffusion, $\chi = 0.0001$, a rapid convergence to the stationary state $u^* = 1 - \frac{\alpha}{\beta Q} = 0.5$, here for $\alpha = 1$ and $\beta Q = 2$ around the center, and a slowly moving diffusion front towards the boundaries. Integration time is $t_{max} = 20$. We then test the program numerically with the representation using the adjacency matrix J_{ij} as discribed in the epidemiological models, hence the Laplacian becomes

$$\frac{\partial^2}{\partial x^2} u_j = \sum_{\ell=1}^{N} J_{j\ell} \frac{1}{\left| \frac{\ell}{N} - \frac{j}{N} \right|^2} (u_\ell - u_j) \tag{40}$$

which gives the same analytics and numerics as the previously used form Eq. (39).

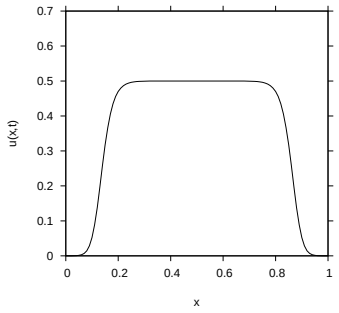


Figure 5: *Kolmogorov-Fisher equation.*

We now use the Fourier representation of a function $u(x,t)$ on the unit interval in discretized form with N discretization points, hence for $u_j(t)$

$$u_j = \sum_{f=1}^{N} \hat{u}_k \cdot e^{2\pi i \frac{j}{N} k} \tag{41}$$

with the Fourier transform

$$\hat{u}_k = \frac{1}{N} \sum_{j=1}^{N} u_j \cdot e^{-2\pi i \frac{j}{N} k} \tag{42}$$

where we have as relations between continuous and discretized version $x = \frac{j}{N}$ and $\Delta x = \frac{1}{N}$. Here Δx just is the difference in x, not to confuse with the Laplace operator Δu which is $\Delta u = \partial^2 u / \partial x^2$ in one dimension. As exact result we obtain using Fourier transformation and back transformation

$$\begin{aligned} \frac{\partial^2}{\partial x^2} u_j &:= \frac{1}{\Delta x^2} (u_{j-1} - 2u_j + u_{j+1}) \\ &= \sum_{\ell=1}^{N} u_\ell \, \frac{1}{N} \sum_{k=1}^{N} e^{2\pi i \frac{j-\ell}{N} k} \, \frac{1}{\Delta x^2} \cdot 2 \left(cos \left(2\pi \frac{k}{N} \right) - 1 \right) \end{aligned} \tag{43}$$

 ISBN: 978-84-612-9727-6

hence the form

$$\frac{\partial^2}{\partial x^2}u_j = \sum_{\ell=1}^{N} w_{j\ell} \cdot u_\ell \tag{44}$$

with $w_{j\ell}$ as specified above through the Fourier transform. We observe that due to 2π periodicity we have

$$2\left(cos\left(2\pi\frac{k}{N}\right)-1\right) = 2\left(cos\left(2\pi\frac{N-k}{N}\right)-1\right) \tag{45}$$

and for the left hand side of Eq. (45) the approximation

$$2\left(cos\left(2\pi\frac{k}{N}\right)-1\right) \approx -\left(2\pi\frac{k}{N}\right)^2 \tag{46}$$

for small values of k and

$$2\left(cos\left(2\pi\frac{N-k}{N}\right)-1\right) \approx -\left(2\pi\frac{N-k}{N}\right)^2 \tag{47}$$

for small $N-k$, hence large k. The power of 2 in Eqs. (43) to (47) gives the handel to generalize to other powers $\mu < 2$ for the superdiffusive case.

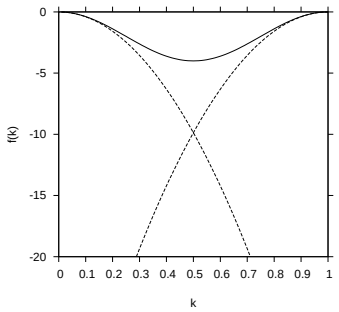


Figure 6: *Expression* $2\left(cos\left(2\pi\frac{k}{N}\right)-1\right)$ *and its quadratic approximations Eqs. (46) and (47).*

The exact result Eq. (43) gives good numerical results compared with Eq. (39), whereas the approximation using Eqs. (46) and (47) shows for moderate step size numerical instabilities due to the not well approximated intermediate part of the k spectrum, see Fig. 6. However, first numerical tests have shown that the exact result, Eq. (43), can be perturbed away from the quadratic power without large numerical errors. For larger perturbations $u(x,t)$ becomes slightly negative at the tails. Further analysis has to be performed on this topic. The description of the Laplacian in form Eq. (44) is in complete analogy to Eq. (40) and can be used to describe the epidemic process we investigate for extensions away from the ordinary diffusion case towards super-diffusion.

Acknowledgements

We thank Luis Sanchez and Jose Francisco Rodrigues, Lisbon, Dirk Brockmann, Göttingen and Chicago, and Frank Hilker, Lisbon and Bath, for stimulating discussions on the present topic. We further thank FCT, Portugal, and the EU Framework Program 7 for financial support.

References

[1] M. Aguiar, B.W. Kooi and N. Stollenwerk, *Epidemiology of dengue fever: A model with temporary cross-immunity and possible secondary infection shows bifurcations and chaotic behaviour in wide parameter regions*, Math. Model. Nat. Phenom. **3** (2008) 48–70.

[2] J.P. Boto *Review on fractional derivatives*, CMAF manuscript (2009).

[3] D. Brockmann and L. Hufnagel *Front propagation in reaction-superdiffusion dynamics: Taming Lévy flights with fluctuations*, Phys. Review Lett. **98** (2007) 178301.

[4] D. Brockmann, L. Hufnagel and T. Geisel (2006) *The scaling laws of human traveling*, *Nature* **418** (2006) 462–465.

[5] D. Brockmann *Superdiffusion in scale-free inhomogeneous environments*, Dissertation at Georg-August-University Göttingen, 2003.

[6] R.A. Fisher (1937) *The wave of advance of advantageous genes*, Annals of Eugenics **7** (1937) 353–369.

[7] P. Grassberger, H. Chaté and G. Rousseau (1997) *Spreading in media with long-time memeory*, Phys. Rev. **E 55** (1997) 2488–2495.

[8] A. Kolmogorov, I. Petrovski and N. Piscounov *A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem*, Moscow Univ. Bull. Ser. Internat. Sec. **A 1** (1937) 1–25.

[9] N. Stollenwerk, M.C.J. Maiden and V.A.A. Jansen *Diversity in pathogenicity can cause outbreaks of menigococcal disease*, Proc. Natl. Acad. Sci. USA **101** (2004) 10229–10234.

[10] N. Stollenwerk, J. Martins and A Pinto *The phase transition lines in pair approximation for the basic reinfection model SIRI*, Physics Letters A **371** (2007) 379–388.

[11] S. van Noort and N. Stollenwerk, *From dynamical processes to likelihood functions: an epidemiological application to influenza*, Proceedings of 8th Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2008, ISBN 978-84-612-1982-7 (2008).

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Detection of faults and gradient faults from scattered data with noise

Mira Bozzini[1] **and Milvia Rossini**[1]

[1] *Dipartimento di Matematica e Applicazioni, Università di Milano Bicocca*

emails: mira.bozzini@unimib.it, milvia.rossini@unimib.it

Abstract

In this paper we discuss the problem of detecting the faults and or the gradient faults of a function when scattered and noisy data are given.

Key words: Discontinuity detection, fault, gradient fault

1 Introduction

Detection of discontinuities is very important in many scientific applications including signal and image processing, geology, geophysics, economics, medicine.

Here we want to consider the case of discontinuities in the function and also in the first derivatives. In two dimensions, the problem is to detect the curves across which the function or its partial derivatives are discontinuous. We call such curve *fault* in the first case and *gradient fault* in the second. Their accurate detection is of crucial importance to analyze and recover a certain phenomenon correctly. In fact the most important information is carried by irregular structures. We can think for instance to depth or subsoil faults which represent discontinuities caused by severe movements of the earth crust. Their localization provides useful information on the occurrence of oil reservoirs. Another important example is the analysis of medical images as the magnetic resonance (MRI) where the fault lines may indicate the presence of some pathology. Moreover in many problems of geophysical interest, one has to deal with data that exhibit also gradient faults. This occurs when describing the topography of seafloor surfaces, mountains with drainage patterns and in general the shape of geological entities.

In any case, discretely defined surfaces that exhibit such features can not correctly recovered without the knowledge of the position of the discontinuity curves and the type of discontinuity. The typical problem that occurs is over-smoothing near gradient faults and Gibb's phenomenum near the faults.

The importance of detecting the discontinuities curves of a function, is also evident from the literature where we find several methods related to this subject. In particular there is a wide literature about the fault detection, often referred as edge detection (see for instance [1], [2], [3], [7], [9],[11], [12], [13], [14]). Many of these methods are based on wavelets and multiresolution techniques [2], [12], [13], [14] that are very popular in image processing where we have pixel (gridded) data with very large samples of size at least 2^{16}. On the other hand, we find only some strategy for detecting gradient discontinuities when gridded data are given [6], [10].

The aim of the paper is to give a method to detect the discontinuity curves on unknown functions $f(\mathbf{x})$, $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ by a sample of scattered and noisy data with large size N but not extra large, i.e. $N < 2^{16}$. Let S be the sample

$$S = \{(\mathbf{x}_i, \tilde{f}_i),\ i = 1, \ldots, N\}. \tag{1}$$

The point locations $X = \{\mathbf{x}_i \in \Omega \subset \mathbb{R}^2\}$ are scattered with a uniform distribution in $\Omega = [0, 1]^2$, and the assigned values are such that

$$\tilde{f}_i = f(\mathbf{x}_i) + e_i, \quad i = 1, \ldots, N \tag{2}$$

where

1. e_i are i.i.d random variables with expected value $E(e_i) = 0$ and unknown covariance matrix $C = \sigma^2 I$, being I the identity matrix of order N. We assume that the noise to signal ratio, $\sigma/\|f\|_2$, is small.

2. The function $f(\mathbf{x})$ or its gradient $\nabla f(\mathbf{x})$ are discontinuous across an unknown curve Γ of Ω and smooth in any neighborhood of Ω which does not intersect Γ.

Our aim is to provide a method to detect the position of Γ and to say if it is a fault or gradient fault.

We also want to discuss the problem of approximating this curve. As we shall see in the next paragraphs, our method identifies a set of points in Ω near the curve. These points allow us to find a strip domain in which the curve lies with high probability. The next problem is to provide an approximation $\hat{\Gamma}$ of Γ. This is an open and difficult question. In fact it is not sufficient to accurately reconstruct Γ. It is also necessary that the approximation $\hat{\Gamma}$ respects the partition of the sample given by Γ. Otherwise, we would have incorrect information on the phenomenon we are studying. The techniques that we find in the literature (see for instance [7] and [9]), do not take into account this aspect which is of crucial importance. In fact, in the case of non-regular surface approximation, the recovering will be poor near the discontinuity curve especially in the case of faults (see Fig. 3).

In §3 we propose a first attempt to recover a linear piecewise continuous fault. For instance, this situation happens in geophysical surfaces with faults generated by tectonic movements that cause fractures in the ground following piecewise linear paths.

Finally we want to stress that the proposed method allows to distinguish between discontinuities and sharp variations (see §2.3).

2 Detection and Classification of the discontinuity curve

2.1 A simple case

We start by considering the simplest case of exact and gridded data. Let G_n denote a grid of points in Ω with step-size $h_n = 1/n$

$$G_n = \{\mathbf{z}_{i_n,j_n},\, \mathbf{z}_{i_n,j_n} = (i_n h_n, j_n h_n), \quad i_n, j_n = 0, \ldots, n\}, \tag{3}$$

and let F_n be the values of a function $f(\mathbf{x})$ evaluated at the points of G_n

$$F_n = \{f(\mathbf{z}_{i_n,j_n}), \quad i_n, j_n = 0, \ldots, n\}. \tag{4}$$

For $i_n, j_n = 1, \ldots, n-1$, we consider the centered differences $\Delta_n f(\mathbf{z}_{i_n,j_n})$ applied to the data (4)

$$\begin{aligned} \Delta_n f(\mathbf{z}_{i_n,j_n}) &= [\Delta_{x,n} f(\mathbf{z}_{i_n,j_n}), \Delta_{y,n} f(\mathbf{z}_{i_n,j_n})] & (5)\\ &= [f(\mathbf{z}_{i_n+1,j_n}) - f(\mathbf{z}_{i_n-1,j_n}), f(\mathbf{z}_{i_n,j_n+1}) - f(\mathbf{z}_{i_n,j_n-1})], & (6) \end{aligned}$$

and we indicate with $\boldsymbol{\Delta}_{x,n}$ and $\boldsymbol{\Delta}_{y,n}$ the sets

$$\boldsymbol{\Delta}_{x,n} = \{\Delta_{x,n} f(\mathbf{z}_{i_n,j_n}), \quad i_n, j_n = 1, \ldots, n-1\}, \tag{7}$$

$$\boldsymbol{\Delta}_{y,n} = \{\Delta_{y,n} f(\mathbf{z}_{i_n,j_n}), \quad i_n, j_n = 1, \ldots, n-1\}. \tag{8}$$

With simple arguments, it is easy to see that the elements of $\boldsymbol{\Delta}_{x,n}$ and $\boldsymbol{\Delta}_{y,n}$ have different behaviors depending on the grid point $\mathbf{z}_{i_n,j_n}$ are close to the curve Γ.

The idea of using difference operators to characterize function discontinuities is not new, in fact in [2] and [8] they are used to detects jumps in univariate and bivariate functions. Here we want to study the asymptotic behaviour of these operators and also to consider difference operators coming from the discretizazion of $\Delta^2 := \partial^2/\partial x^2 + \partial^2/\partial y^2$. As you will see, they allow us to identify not only the faults but also the gradient faults. We indicate with Q_{i_n,j_n} the square

$$Q_{i_n,j_n} = [(i_n - 1)h_n, (i_n + 1)h_n] \times [(j_n - 1)h_n, (j_n + 1)h_n] \tag{9}$$

centered at $\mathbf{z}_{i_n,j_n}$. Let us consider a difference operator

$$\Delta^2_{1,n} f(\mathbf{z}_{i_n,j_n}) = \sum_{l,r=-1}^{1} \gamma_{l,r} f(\mathbf{z}_{i_n+l,j_n+r})$$

associated to a discretization $(h_n^{-2}\Delta^2_{1,n} f(\mathbf{z}_{i_n,j_n}))$ of the Laplace operator Δ^2. The classical one considers only the vertical and horizontal directions with coefficients given by

$$\gamma = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

 ISBN: 978-84-612-9727-6

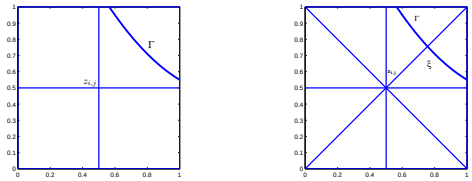


Figure 1:

Here we prefer to use different coefficients leading to an isotropic discretization of Δ^2 (see [4])

$$\gamma = \frac{1}{6}\begin{pmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{pmatrix} \tag{10}$$

which considers the diagonal directions and gives information also when Γ does not intersect the horizontal and the vertical directions of Q_{i_n,j_n} but only the diagonal ones (see Fig. 1). As before we indicate with $\mathbf{\Delta}^2_{1,n}$ the set

$$\mathbf{\Delta}^2_{1,n} = \{\Delta^2_{1,n} f(\mathbf{z}_{i_n,j_n}), \quad i_n, j_n = 1, \ldots, n-1\}. \tag{11}$$

We call A_{i_n,j_n} the set of points where the curve intersects the horizontal, vertical and diagonal directions of Q_{i_n,j_n}. For simplicity, we assume that in any Q_{i_n,j_n}, Γ intersects the four directions only one time at the most. The possible intersection points are denoted by

$$\begin{aligned} \boldsymbol{\xi}_o = (\xi_{ox}, \xi_{oy}) &= \Gamma \bigcap \overline{\mathbf{z}_{i_n-1,j_n}\mathbf{z}_{i_n+1,j_n}}, \\ \boldsymbol{\xi}_v = (\xi_{vx}, \xi_{vy}) &= \Gamma \bigcap \overline{\mathbf{z}_{i_n,j_n-1}\mathbf{z}_{i_n,j_n+1}}, \\ \boldsymbol{\xi}_{d_1} = (\xi_{d_1x}, \xi_{d_1y}) &= \Gamma \bigcap \overline{\mathbf{z}_{i_n-1,j_n-1}\mathbf{z}_{i_n+1,j_n+1}}, \\ \boldsymbol{\xi}_{d_2} = (\xi_{d_2x}, \xi_{d_2y}) &= \Gamma \bigcap \overline{\mathbf{z}_{i_n-1,j_n+1}\mathbf{z}_{i_n+1,j_n-1}}. \end{aligned}$$

We set

$$\text{jump}_x f|_{\boldsymbol{\xi}_o} = f(\xi^+_{ox}, j_n h_n) - f(\xi^-_{ox}, j_n h_n), \quad \text{jump}_y f|_{\boldsymbol{\xi}_v} = f(i_n h_n, \xi^+_{oy}) - f(i_n h_n, \xi^-_{oy}),$$

and we indicate respectively with $\text{jump} f|_{Q_{i_n,j_n}}$ and $\text{jump} \nabla f|_{Q_{i_n,j_n}}$ an average of the possible jumps of f and ∇f along the four directions in Q_{i_n,j_n}. With simple arguments, we prove that:

Proposition 1 *Let h_n be a given step-size. For any $i_n, j_n = 1, \ldots, n-1$, we have that*

1. *when Γ intersects $(\mathbf{z}_{i_n-1,j_n}, \mathbf{z}_{i_n+1,j_n})$ in $\boldsymbol{\xi}_o$*

$$\Delta_{x,n} f(\mathbf{z}_{i_n,j_n}) = \begin{cases} \text{jump}_x f|_{\boldsymbol{\xi}_o} + O(h_n), & \text{if } \Gamma \text{ is a fault} \\ O(h_n) & \text{if } \Gamma \text{ is a gradient fault}, \end{cases} \tag{12}$$

while

$$\Delta_{x,n} f(\mathbf{z}_{i_n,j_n}) = O(h_n) \quad \text{when} \quad \Gamma \bigcap [\mathbf{z}_{i_n-1,j_n}, \mathbf{z}_{i_n+1,j_n}] = \emptyset; \tag{13}$$

 ISBN: 978-84-612-9727-6

2. *when* Γ *intersects* $(\mathbf{z}_{i_n,j_n-1}, \mathbf{z}_{i_n,j_n+1})$ *in* $\boldsymbol{\xi}_v$

$$\Delta_{y,n} f(\mathbf{z}_{i_n,j_n}) = \begin{cases} \mathrm{jump}_y f|_{\boldsymbol{\xi}_v} + O(h_n), & \text{if } \Gamma \text{ is a fault} \\ O(h_n), & \text{if } \Gamma \text{ is a gradient fault} \end{cases} \tag{14}$$

while

$$\Delta_{y,n} f(\mathbf{z}_{i_n,j_n}) = O(h_n) \quad \text{when} \quad \Gamma \bigcap [\mathbf{z}_{i_n,j_n-1}, \mathbf{z}_{i_n,j_n+1}] = \emptyset; \tag{15}$$

3. *when* $A_{i_n,j_n} \neq \emptyset$ *and at least one of its point is an interior point of* Q_{i_n,j_n}

$$\Delta^2_{1,n} f(\mathbf{z}_{i_n,j_n}) = \begin{cases} \mathrm{jump} f|_{Q_{i_n,j_n}} + O(h_n), & \text{if } \Gamma \text{ is a fault} \\ \mathrm{jump} \nabla f|_{Q_{i_n,j_n}} h_n + O(h_n^2), & \text{if } \Gamma \text{ is a gradient fault,} \end{cases} \tag{16}$$

otherwise

$$\Delta^2_{1,n} f(\mathbf{z}_{i_n,j_n}) = O(h_n^2). \tag{17}$$

We now discuss what happens when h_n decreases to zero. This study allows us to characterize the points of Γ.

We consider a sequence of nested grids $G_{\bar{n}} \subset \cdots \subset G_n \subset G_{n+1} \ldots$, $n = 2^i \bar{n}$, $h_n = h_{\bar{n}}/2^i$ with $i = 0, 1, 2, \ldots$, and the associated sets $\boldsymbol{\Delta}_{x,n}$, $\boldsymbol{\Delta}_{y,n}$, $\boldsymbol{\Delta}^2_{1,n}$.

Fixed a grid $G_{\hat{n}}$, we take the $\hat{j}_{\hat{n}}$th row (and respectively the $\hat{i}_{\hat{n}}$th column). Let $\boldsymbol{\eta}_{\hat{y}} = (x, \hat{y})$ be a point on the line $y = \hat{y}$ (and correspondingly let $\boldsymbol{\eta}_{\hat{x}} = (\bar{x}, y)$ be a point on the line $x = \hat{x}$). For $n > \hat{n}$, the sequence of nested grids G_n determines on $y = \hat{y}$ a sequence of nested intervals $I_{\boldsymbol{\eta}_{\hat{y}},n} = [\mathbf{z}_{i_n-1,j_n}, \mathbf{z}_{i_n+1,j_n}]$ containing $\boldsymbol{\eta}_{\hat{y}}$ (and correspondingly on $x = \hat{x}$ a sequence $I_{\boldsymbol{\eta}_{\hat{x}},n} = [\mathbf{z}_{i_n,j_n-1}, \mathbf{z}_{i_n,j_n+1}]$ containing $\boldsymbol{\eta}_{\hat{x}}$). We indicate with $\boldsymbol{\Delta}_{x,n}(\boldsymbol{\eta}_{\hat{y}})$ and $\boldsymbol{\Delta}_{y,n}(\boldsymbol{\eta}_{\hat{x}})$ the sequences of the centered differences of $f(\mathbf{z}_{i_n,j_n})$ associated with $I_{\boldsymbol{\eta}_{\hat{y}},n}$ and $I_{\boldsymbol{\eta}_{\hat{x}},n}$. Analogously, fixed a point $\boldsymbol{\eta}$ of Ω, we denote with $Q_{\boldsymbol{\eta},n} = [(i_n - 1)h_n, (i_n + 1)h_n] \times [(j_n - 1)h_n, (j_n + 1)h_n]$ a sequence of nested squares containing $\boldsymbol{\eta}$ and we indicate with $\boldsymbol{\Delta}^2_{1,n}(\boldsymbol{\eta})$ the sequence of the isotropic difference of $f(\mathbf{z}_{i_n,j_n})$ associated with $Q_{\boldsymbol{\eta},n}$. From Proposition 1, we get:

Corollary 2 *When* $n \to \infty$ *we have that*

$$\boldsymbol{\Delta}_{x,n}(\boldsymbol{\eta}_{\hat{y}}) \to \mathrm{jump}_x f|_{\boldsymbol{\eta}_{\hat{y}}}, \ \left(\boldsymbol{\Delta}_{y,n}(\boldsymbol{\eta}_{\hat{x}}) \to \mathrm{jump}_y f|_{\boldsymbol{\eta}_{\hat{x}}}\right) \tag{18}$$

if $\boldsymbol{\eta}_{\hat{y}}(\boldsymbol{\eta}_{\hat{x}}) \in \Gamma$ *and* Γ *is a fault curve while if* $\boldsymbol{\eta}_{\hat{y}}(\boldsymbol{\eta}_{\hat{x}}) \in \Gamma$ *and* Γ *is a gradient curve or* $\boldsymbol{\eta}_{\hat{y}}(\boldsymbol{\eta}_{\hat{x}}) \notin \Gamma$

$$\boldsymbol{\Delta}_{x,n}(\boldsymbol{\eta}_{\hat{y}}) \to 0 \quad (\boldsymbol{\Delta}_{y,n}(\boldsymbol{\eta}_{\hat{x}}) \to 0)\,; \tag{19}$$

Now we consider the asymptotic behavior of the isotropic differences (11).

Corollary 3 *Let* $\boldsymbol{\eta}$ *be a point of* Ω. *When* $n \to \infty$, *we have that*

1.

$$\frac{\boldsymbol{\Delta}^2_{1,n}(\boldsymbol{\eta})}{h_n} \to \infty \tag{20}$$

if $\boldsymbol{\eta} \in \Gamma$ *and* Γ *is a fault curve;*

 ISBN: 978-84-612-9727-6

2.

$$\frac{\boldsymbol{\Delta}^2_{1,n}(\boldsymbol{\eta})}{h_n} \to C_{\boldsymbol{\eta}} \tag{21}$$

if $\boldsymbol{\eta} \in \Gamma$ *and* Γ *is a gradient curve, being* $C_{\boldsymbol{\eta}}$ *a constant depending on the gradient jump at* $\boldsymbol{\eta}$*;*

3.

$$\frac{\boldsymbol{\Delta}^2_{1,n}(\boldsymbol{\eta})}{h_n} \to 0 \tag{22}$$

if $\boldsymbol{\eta} \notin \Gamma$.

2.2 The general case: scattered and noisy data

From the given scattered sample S, we construct a gridded pseudodata set

$$S_{G_n} = \{(\mathbf{z}_{i_n,j_n}, \tilde{u}_{i_n,j_n}), \quad i_n, j_n = 0, \ldots, n\}. \tag{23}$$

Namely, we consider a suitable step-size $h_n = 1/n$, the associated grid (3) and an integer $n_0 << N$. For each grid point, we indicate with $\mathcal{U}_{i_n,j_n}$ the circular neighborhood centered at $\mathbf{z}_{i_n,j_n}$ containing n_0 points $\mathbf{x}_k^{i_n,j_n} \in X$, with $\tilde{f}_k^{i_n,j_n}$ the corresponding sample values, and with μ_{i_n,j_n} its radius. We define $\tilde{u}_{i_n,j_n}$ to be the average

$$\tilde{u}_{i_n,j_n} = \frac{1}{n_0} \sum_{k=1}^{n_0} \tilde{f}_k^{i_n,j_n}. \tag{24}$$

It is worthwhile to remark that in this way we smooth the noise corrupting the data; in fact the random variables $\tilde{u}_{i_n,j_n}$ have the following expected values and variances

$$E(\tilde{u}_{i_n,j_n}) = \frac{1}{n_0} \sum_{k=1}^{n_0} f(\mathbf{x}_k^{i_n,j_n}), \quad Var(\tilde{u}_{i_n,j_n}) = \frac{\sigma^2}{n_0}.$$

By applying the discrete operators Δ_n, $\Delta^2_{1,n}$ to (23), we obtain the estimators

$$\Delta_{x,n}\tilde{u}_{i_n,j_n}, \quad \Delta_{y,n}\tilde{u}_{i_n,j_n}, \quad \Delta^2_{1,n}\tilde{u}_{i_n,j_n}, \; i_n, j_n = 1, \ldots, n-1, \tag{25}$$

and the sets

$$\widetilde{\boldsymbol{\Delta}}_{x,n} = \{\Delta_{x,n}\tilde{u}_{i_n,j_n}, \quad i_n, j_n = 1, \ldots, n-1\}, \tag{26}$$

$$\widetilde{\boldsymbol{\Delta}}_{y,n} = \{\Delta_{y,n}\tilde{u}_{i_n,j_n}, \quad i_n, j_n = 1, \ldots, n-1\}, \tag{27}$$

$$\widetilde{\boldsymbol{\Delta}^2}_{1,n} = \{\Delta^2_{1,n}\tilde{u}_{i_n,j_n}, \quad i_n, j_n = 1, \ldots, n-1\}. \tag{28}$$

Using the results of Proposition 1, we obtain immediately

Proposition 4 *For any* $i_n, j_n = 1, \ldots, n-1$, *we have that*

 ISBN: 978-84-612-9727-6

1. *when* Γ *intersects* $\overline{\mathbf{z}_{i_n-1,j_n}\mathbf{z}_{i_n+1,j_n}}$ *in* $\boldsymbol{\xi}_o$

$$E(\Delta_{x,n}\tilde{u}_{i_n,j_n}) = \begin{cases} C^x_{i_n,j_n} + O(h_n), & \text{if } \Gamma \text{ is a fault} \\ O(h_n) & \text{if } \Gamma \text{ is a gradient fault,} \end{cases} \tag{29}$$

while

$$E(\Delta_{x,n}\tilde{u}_{i_n,j_n}) = O(h_n) \quad \text{if} \quad \Gamma \bigcap \overline{\mathbf{z}_{i_n-1,j_n}\mathbf{z}_{i_n+1,j_n}} = \emptyset; \tag{30}$$

2. *when* Γ *intersects* $\overline{\mathbf{z}_{i_n,j_n-1}\mathbf{z}_{i_n,j_n+1}}$ *in* $\boldsymbol{\xi}_v$

$$E(\Delta_{y,n}\tilde{u}_{i_n,j_n}) = \begin{cases} C^y_{i_n,j_n} + O(h_n), & \text{if } \Gamma \text{ is a fault} \\ O(h_n), & \text{if } \Gamma \text{ is a gradient fault} \end{cases} \tag{31}$$

while

$$E(\Delta_{y,n}\tilde{u}_{i_n,j_n}) = O(h_n) \quad \text{if} \quad \Gamma \bigcap \overline{\mathbf{z}_{i_n,j_n-1}\mathbf{z}_{i_n,j_n+1}} = \emptyset; \tag{32}$$

3. *when* $A_{i_n,j_n} \neq \emptyset$

$$E(\Delta^2_{1,n}\tilde{u}_{i_n,j_n}) = \begin{cases} C^1_{i_n,j_n} + O(h_n), & \text{if } \Gamma \text{ is a fault} \\ D^1_{i_n,j_n}h_n + O(h_n^2), & \text{if } \Gamma \text{ is a gradient fault,} \end{cases} \tag{33}$$

$$E(\Delta^2_{1,n}\tilde{u}_{i_n,j_n}) = O(h_n^2) \quad \text{if} \quad A_{i_n,j_n} = \emptyset, \tag{34}$$

$C^x_{i_n,j_n}$, $C^y_{i_n,j_n}$, *and* $C^1_{i_n,j_n}$, *are constants depending on the jumps of continuity of* f *at some points of* Q_{i_n,j_n} *and* $D^1_{i_n,j_n}$ *is a constant depending on the jumps of continuity of* ∇f *at some points of* Q_{i_n,j_n}.

Proposition 5 *The variances of (25) are*

$$Var(\Delta_{x,n}\tilde{u}_{i_n,j_n}) \leq \frac{2\sigma^2}{n_0}, \;\; Var(\Delta_{y,n}\tilde{u}_{i_n,j_n}) \leq \frac{2\sigma^2}{n_0}, \;\; \text{and } Var(\Delta^2_{1,n}\tilde{u}_{i_n,j_n}) \leq \frac{13\sigma^2}{n_0}. \tag{35}$$

We now study the asymptotic behavior of (25). For $N \to \infty$ and $n \to \infty$, we consider again a sequence of nested grids $G_{\bar{n}} \subset \cdots \subset G_n \subset G_{n+1} \ldots$, $n = 2^i\bar{n}$, $h_n = h_{\bar{n}}/2^i$ with $i = 0, 1, 2, \ldots$. Proceedings as in §2.1 and using the results of Propositions 4 and 5, we get:

Proposition 6 *When* $N \to \infty$, $h_n \to 0$, $n_0 \to \infty$ *so that* $\sqrt{N}h_n \to \infty$, $\mu_{i_n,j_n} \to 0$, $\mu_{i_n,j_n}n_0 \to \infty$, *and* $h_n n_0 \to \infty$, *we have that in probability*

1. *if* $\boldsymbol{\eta}_{\hat{y}}(\boldsymbol{\eta}_{\hat{x}}) \in \Gamma$ *and* Γ *is a fault curve*

$$\frac{\widetilde{\boldsymbol{\Delta}}_{x,n}(\boldsymbol{\eta}_{\hat{y}})}{h_n} \to \infty \quad \left(\frac{\widetilde{\boldsymbol{\Delta}}_{y,n}(\boldsymbol{\eta}_{\hat{x}})}{h_n} \to \infty \right), \tag{36}$$

while if $\boldsymbol{\eta}_{\hat{y}}(\boldsymbol{\eta}_{\hat{x}}) \in \Gamma$ *and* Γ *is a gradient curve or* $\boldsymbol{\eta}_{\hat{y}}(\boldsymbol{\eta}_{\hat{x}}) \notin \Gamma$

$$\frac{\widetilde{\boldsymbol{\Delta}}_{x,n}(\boldsymbol{\eta}_{\hat{y}})}{h_n} \to K^x_{\boldsymbol{\eta}_{\hat{y}}}, \quad \left(\frac{\widetilde{\boldsymbol{\Delta}}_{y,n}(\boldsymbol{\eta}_{\hat{x}})}{h_n} \to K^y_{\boldsymbol{\eta}_{\hat{x}}} \right); \tag{37}$$

 ISBN: 978-84-612-9727-6

2. if $\boldsymbol{\eta} \in \Gamma$ *and* Γ *is a fault curve*

$$\frac{\widetilde{\boldsymbol{\Delta}^2}_{1,n}(\boldsymbol{\eta})}{h_n} \to \infty; \tag{38}$$

3. if $\boldsymbol{\eta} \in \Gamma$ *and* Γ *is a gradient curve*

$$\frac{\widetilde{\boldsymbol{\Delta}^2}_{1,n}(\boldsymbol{\eta})}{h_n} \to C_{\boldsymbol{\eta}}; \tag{39}$$

4. if $\boldsymbol{\eta} \notin \Gamma$

$$\frac{\widetilde{\boldsymbol{\Delta}^2}_{1,n}(\boldsymbol{\eta})}{h_n} \to 0. \tag{40}$$

2.3 Working with a fixed h_n

In practice, we work with a given sample and consequently a fixed h_n. The previous results suggest how to perform the detection. First, we look for the position of the possible fault curve by considering and analyzing the quantities $\Delta_{x,n}\tilde{u}_{i_n,j_n}/h_n$ and $\Delta_{y,n}\tilde{u}_{i_n,j_n}/h_n$, $i_n, j_n = 1, \ldots, n-1$.

Proposition 6 allows us to say that they assume "big" values at the points $\mathbf{z}_{i_n,j_n}$ near the fault, otherwise they have "small" values with respect to the previous ones.

The set of possible fault points ("big values") can be detected by a classification method which separates the two different class of points ("big values", "small values"). For instance, this can be done by fixing threshold values depending on the range of $\Delta_{x,n}\tilde{u}_{i_n,j_n}/h_n$ and $\Delta_{y,n}\tilde{u}_{i_n,j_n}/h_n$.

We have to remark that, in the selected class, we can have also set of points for which $\frac{\Delta_{x,n}\tilde{u}_{i_n,j_n}}{h_n}$ and/or $\frac{\Delta_{y,n}\tilde{u}_{i_n,j_n}\tilde{u}_{i_n,j_n}}{h_n}$ are "big" because the constants $K^x_{\boldsymbol{\eta}_{\hat{x}}}$ and/or $K^y_{\boldsymbol{\eta}_{\hat{y}}}$ are large. These points correspond to high gradients.

Typically, high gradient points lie in a bidimensional region R of Ω and they do not follow the behavior of a curve of Ω. This allows us to discriminate them.

In Fig. 2, it is shown a function with a fault and a zone of high gradients (left). On the right, we can see the points that follows a curve and the region corresponding the the sharp variation of f. Here we have considered a sample of $N = 4900$ scattered points. We indicate by $D_f = \{(\mathbf{z}^f_l, \tilde{u}^f_l),\, l = 1, \ldots, n_f\}$ the points of S_{G_n} corresponding to the locations $\mathbf{z}^f_l$ detected as fault points. If $D_f \neq \emptyset$, the unknown curve Γ is classified as fault. In a second step, we look if there is a gradient fault. We consider the set $S_{G_n} \setminus D_f$. We select from it the points $\mathbf{z}^g_l$ at which $\frac{\Delta^2_{1,n}\tilde{u}_{i_n,j_n}}{h_n}$ takes "big values": these locations are detected as gradient fault points. We indicate by $D_g = \{(\mathbf{z}^g_l, \tilde{u}^g_l),\, l = 1, \ldots, n_g\}$. the elements of S_{G_n} corresponding to the locations $\mathbf{z}^g_l$. The unknown curve Γ is classified as a gradient fault. In both cases the distances between the detected points and the curve is less or equal to h_n.

 ISBN: 978-84-612-9727-6

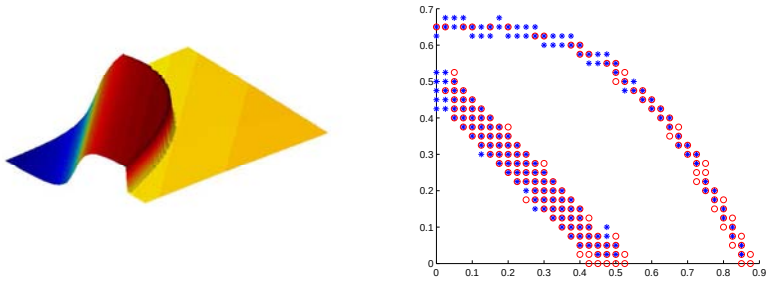


Figure 2: $N = 4900$, $h_n = 1/40, \frac{\sigma}{\|f\|^2} \sim 0.015$

3 Some remarks on the approximation of Γ

When approximating a fault or a gradient fault, one must take into account some fundamental aspects. In the following we illustrate these issues and show how we can proceed in the case of faults.

We observe that, having used the centered differences and $\Delta^2_{1,n}$, the sets D_f and D_g detect a stripe $\mathfrak{S}$ of Ω where most likely Γ lies.

Let us assume that the curve runs from west to east and that it divides Ω into the disjoint subsets Ω_N and Ω_S. Γ separates the given sample S in two disjoint sub-samples S_N and S_S.

The recovering $\hat{\Gamma}$ is effective if it observes the same sample parting provided by the unknown curve. This means that it is not sufficient to require that $\hat{\Gamma}$ is a good approximation of Γ. In the following we give an outline of how you can recover a linear piecewise and continuous fault. The construction of $\hat{\Gamma}$ of Γ, starts with a first rough approximation that will be used to find the slope changes. We construct a new set of points $\mathbf{z}^\star_j \in \mathfrak{S}$ by associating to each $\mathbf{z}^f_l$ the average with the points $\mathbf{z}^f_i$ whose distance from $\mathbf{z}^f_l$ is less or equal to $\sqrt{2}h_n$ and, in order to obtain a behavior more close to the piecewise linear one, we apply again the average procedure to the new set with a distance less than h_n. Let $Z^\star = \{\mathbf{z}^\star_j(x^\star_j, y^\star_j)\ j = 1, \ldots, n^\star\}$ be the so obtained set ordered with respect to $x^\star_j$ and purified by eventual coincident points. The set $Z^\star$ has a global piecewise linear behavior and we detect those points $\bar{\mathbf{z}}_i \in Z^\star$, $i = 1, \ldots, \bar{m}$, corresponding to slope changes by considering the locations where the second differences of the $Z^\star$ elements move away from zero. A first approximation of Γ is given by the linear spline $\hat{\Gamma}^1$ interpolating the points of the set

$$\bar{Z} = \{\mathbf{z}^\star_1, \bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2 \ldots, \bar{\mathbf{z}}_{\bar{m}}, \mathbf{z}^\star_{n^\star}\}.$$

As already said, it is of crucial importance that the final approximation $\hat{\Gamma}$ respects the sample classification given by Γ. This is particularly true when we want to recover the unknown surface f. To this end we show the example discussed in [5]. We approximate the test function (Fig. 5 on the left) by using the approximation $\hat{\Gamma}^1$ of Γ (on the right of Fig. 3)which does not separate the sample points correctly. The recovering (on the left of Fig. 3) presents undue oscillations.

 ISBN: 978-84-612-9727-6

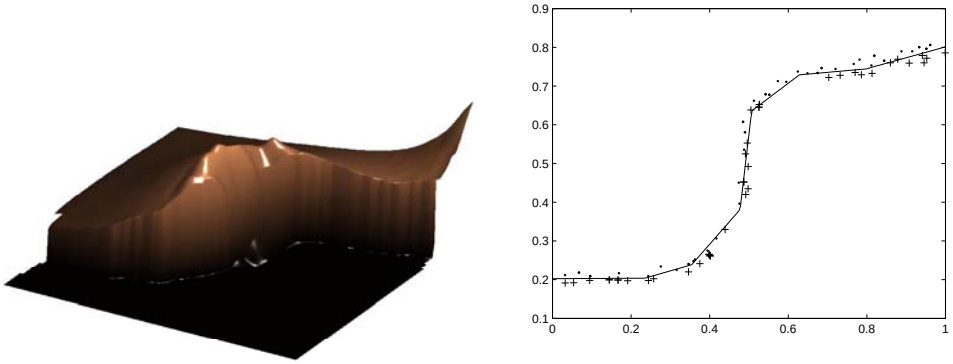


Figure 3: Left: Approximation of the test function). Right: the approximation $\hat{\Gamma}^1$ of Γ,+ sample locations belonging to $\Omega_S \bigcap \mathfrak{S}$, $\cdot$ sample locations belonging to $\Omega_N \bigcap \mathfrak{S}$.

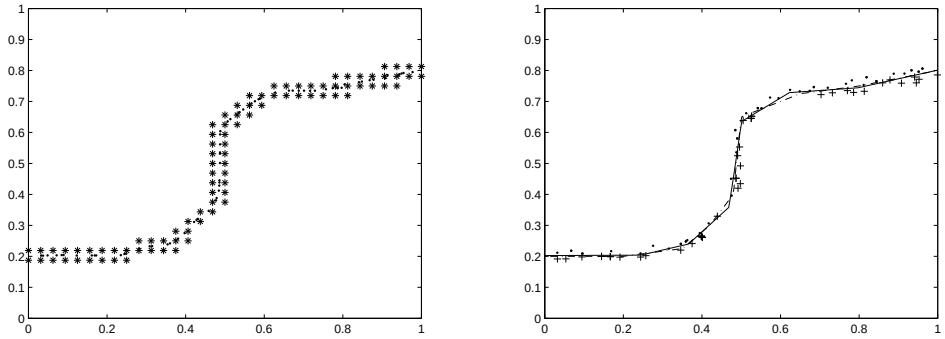


Figure 4: Example 1 (Section 5). Left. $\star$: the detected points $\mathbf{z}_l^f$, diamond: the points of $Z^\star$. Right. – the final approximation $\hat{\Gamma}$, $\cdot -$ the true fault Γ.

We indicate with $\hat{\Omega}_N$ and $\hat{\Omega}_S$ the two parts in which Ω is divided by $\hat{\Gamma}$, and with $\hat{S}_N$ and $\hat{S}_S$ the sample data having $\mathbf{x}_i \in \hat{\Omega}_N$ and $\mathbf{x}_j \in \hat{\Omega}_S$. We need that $S_N \equiv \hat{S}_N$ and $S_S \equiv \hat{S}_S$. Obviously, a wrong classification can happen in $\mathfrak{S}$. Then it is necessary to establish whether a sample point with $\mathbf{x}_i \in \mathfrak{S}$ belongs to S_S or to S_N. For this purpose, we take the set S^1 of the points in S such that the distance $d(\mathbf{x}_i, \mathbf{z}_l^f) \leq h_n$ for some $\mathbf{z}_l^f \in D_f$. By considering all the possible distances between the elements of S_1, and using standard algorithms of cluster analysis, it is possible to divide S^1 in two classes S_N^1 and S_S^1 whose locations $\mathbf{x}_i$ are in Ω_N and Ω_S respectively. Now, by using S_N^1 and S_S^1, we verify whether $\hat{\Gamma}^1$ respects the classification, otherwise we recursively modify it segment by segment maintaining the continuity. In Fig. 4, it is shown the final approximation $\hat{\Gamma}$ of Γ. The maximum error and the root mean least squares error computed on a grid of 200 point are $e_\infty = 0.046$ and $e_2 = 0.007$ respectively.

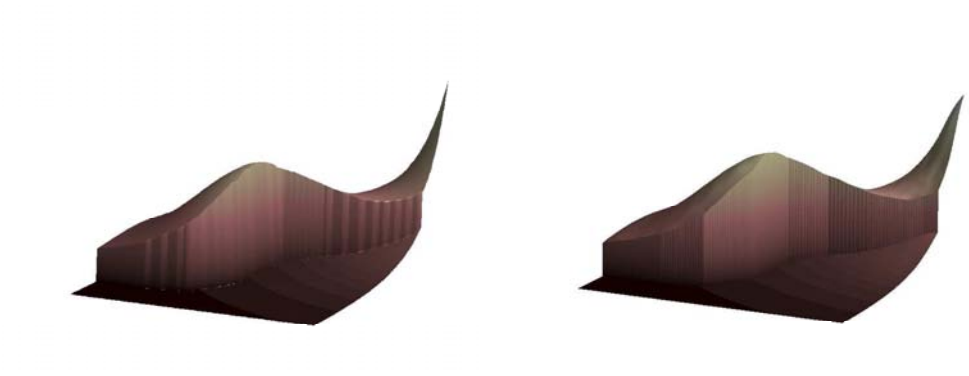


Figure 5: Left: the true surface. Right: The approximation.

To complete the discussion, we show in Fig. 5 (right) the recovering obtained in [5] by using $\hat{\Gamma}$.

References

[1] G. Allasia, R. Besenghi, A. De Rossi, *A scattered data approximation scheme for the detection of fault lines,* Mathematical methods for curves and surfaces (Oslo, 2000), 25–34, Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 2001.

[2] F. Arandiga, F. Cohen, A. Donat, R. Dyn, N., Matei, B. *Approximation of piecewise smooth functions and images by edge-adapted (ENO-EA) nonlinear multiresolution techniques,* Appl. Comput. Harmon. Anal. **24**, no. 2, (2008) 225–250.

[3] R. Archibald, A. Gelb, J. Yoon, *Polynomial fitting for edge detection in irregularly sampled signals and images,* SIAM J. Numer. Anal. **43**, no. 1, (2005) 259–279.

[4] M. Bozzini, C. Rabut, M. Rossini, *A multiresolution analysis with a new family of polyharmonic B-splines.* In Curve and Surface Fitting: Avignon 2006, A. Cohen, J. L. Merrien and L. Schumaker (eds.), Nashboro Press (2007) 51-60.

[5] M. Bozzini, L. Lenarduzzi, M. Rossini, *Non-regular surface approximation.* In press on Lecture Notes in Computer Science, Springer.

[6] D. Cates, A. Gelb, *Detecting derivative discontinuity locations in piecewise continuous functions from Fourier spectral data,* Numer. Algor. **46**, no. 1, (2007) 59–84.

[7] A. Crampton, J. C. Mason, *Detecting and approximating fault lines from randomly scattered data,* Numer. Algor. **39**, no. 1-3, (2005) 115–130.

[8] A. Gelb, E. Tadmor, *Adaptive edge detectors for piecewise smooth data based on the minmod limiter,* J. Sci. Comput. **28** (2006) 279–306.

[9] T. Gutzmer, A. Iske, *Detection of Discontinuities in Scattered Data Approximation,* Numer. Algor. **16** (1997) 155-170.

[10] M. C. Lòpez de Silanes, M. C. Parra, J. J. Torrens, *Vertical and oblique fault detection in explicit surfaces,* J. Comput. Appl. Math. **140** (2002) 559-585.

[11] M. C. Lòpez de Silanes, M. C. Parra, J. J. Torrens, *On a new characterization of finite jump discontinuities and its application to vertical fault detection,* Math. Comput. Simulation **77**, no. 2-3 (2008) 247–256.

[12] M. Rossini, *Irregularity Detection from Noisy Data in One and Two Dimension,* Numer. Algor. **16** (1997) 283-301.

[13] M. Rossini, *2D- Discontinuity Detection from Scattered Data,* Computing **61** (1998) 215-234.

[14] M. Rossini, *Detecting discontinuities in two dimensional signal sampled on a grid.* In press on J. Numer. Anal. Ind. Appl. Math. (1st issue-Volume 4)(2009).

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Growth of Individuals in Randomly Fluctuating Environments

Carlos A. Braumann[1], Patrícia A. Filipe[1], Clara Carlos[2] and Carlos J. Roquete[3]

[1] *Centro de Investigação em Matemática e Aplicações, Universidade de Évora*

[2] *Escola Superior de Tecnologia do Barreiro, Instituto Politécnico de Setúbal*

[3] *Instituto de Ciências Agrárias e Ambientais Mediterrânicas, Universidade de Évora*

emails: braumann@uevora.pt, pasf@uevora.pt, Clara.Carlos@estbarreiro.ips.pt, croquete@uevora.pt

Abstract

Many of the deterministic models proposed in the literature for the growth of an individual animal (or plant) from birth to maturity can be written in the form $dY(t) = \beta(A - Y(t))dt$, where $Y(t) = h(X(t))$, with h a strictly increasing continuously differentiable function and $X(t)$ the size of the individual at age t. Of course, different models have specific functional forms for h (e.g., the Gompertz model has $h(x) = \ln x$, and the Bertalanffy-Richards model has $h(x) = x^c$). Note that $A = h(S)$, where S is the asymptotic (maturity) size. In a randomly fluctuating environment, we propose stochastic differential equation (SDE) models of the type $dY(t) = \beta(A - Y(t))dt + \sigma dW(t)$, where σ is an environmental noise intensity parameter and $W(t)$ is a standard Wiener process. Properties of the model are deduced, including studying the time required for an animal to reach a given size. Statistical issues concerning parameter estimation and prediction will also be tackled and a real life application to cattle data will be shown. The results are useful for optimization issues in livestock management (applications in forestry are also possible). Of course, when a livestock producer has several animals, it is likely that the average asymptotic size S varies from animal to animal according to some distribution. This generalization is also considered here.

Key words: stochastic differential equations, individual growth models, estimation, bovine weight

MSC 2000: 60H10; 60J60; 92D99

1 Introduction

Usually, random variations in individual growth data have been treated by classical regression models. The traditional assumption of regression models that observed deviations from the regression curve are independent at different ages would be realistic if the deviations were due to measurement errors. It is totally unrealistic when the deviations are due to random changes on growth rates induced by environmental random fluctuations. For instance, in such regression models, an animal having a size much below the growth curve at a given age has equal probability of being above or below the curve a day later. Stochastic differential equation (SDE) models do not have such shortcomings. They are built precisely to incorporate the dynamics of the growth process. Furthermore, unlike regression models, in which prediction of future sizes is based on the growth curve, in SDE models the predictions are based on the current size of the animal taking into account the growth dynamics and the effect environmental random fluctuations have on such dynamics.

In section 2 we introduce a class of SDE models for the individual growth of animals (or plants) from birth to maturity and study its properties. The models are based on deterministic models with an added term to account for the environmental fluctuations. The class we are going to consider includes the stochastic versions of a vast range of deterministic growth models that have been traditionally used in the literature.

In section 3, we study the statistical issues of parameter estimation for the case of one animal which size is observed at different ages. Estimation methods for the case where we have observations from several animals are also developed. For illustration we use bovine data of the Mertolengo breed provided by C. J. Roquete. This summarizes results obtained in [4], [6] and [7]. We can also predict future animal sizes (we refer the reader to [4]).

In section 4, we characterize the time for an animal to reach a given size, which could be the size at which the animal is sold to the meat market. The result is similar to a result obtained in [3] and in section 4 of [2] for the extinction time of population size models in a random environment. Now, however, we do not look for the time to reach a low size threshold but rather a high size threshold.

In section 5, we generalize our results to the case when the average asymptotic size S varies randomly from animal to animal due to genetic differences or differences in their life conditions.

Section 6 contains the conclusions.

2 SDE models

Many of the deterministic models proposed in the literature for the growth of an individual animal (or plant) from birth to maturity can be written in the form

$$dY(t) = \beta(A - Y(t))dt, \tag{1}$$

where $Y(t) = h(X(t))$, with h a strictly increasing continuously differentiable function and $X(t)$ the size (weight, length, height, volume, etc.) of the individual at age t.

 ISBN: 978-84-612-9727-6

We may think of $Y(t)$ as a modified size, i.e. the size of the animal in a modified (usually nonlinear) scale. Of course, $A = h(S)$, where S is the asymptotic (maturity) size, and the parameter β describes how fast is the approach to this asymptotic value. We will assume to be known the initial size $X(t_0) = x_0$ measured at an initial age of measurement t_0 and denote $y_0 = h(x_0)$ the initial modified size. Of course, to use transcendental functions h, we have to use an adimensional $X(t)$, that is to consider $X(t)$ as a pure number with no physical dimensions, for which we have to define a fixed unit of size and take $X(t)$ as the number of such units the size of the animal at age t comprises.

Of course, different models have specific functional forms for h. The most commonly used models (and there are many) can be written in the form (1) for an adequate choice of h. For example, two of the most commonly used models are the Gompertz model, which corresponds to $h(x) = \ln x$, and the Bertalanffy-Richards model, which corresponds to $h(x) = x^c$ with $c > 0$. For example, if $X(t)$ is the weight or the volume, we may use in the Bertalanffy-Richards model $c = 1/3$ so that $Y(t)$ becomes a kind of typical length of the animal.

In a randomly fluctuating environment, we propose autonomous stochastic differential equation (SDE) models of the type

$$dY(t) = \beta\,(A - Y(t))\,dt + \sigma dW(t), \;\; Y(t_0) = y_0, \tag{2}$$

where σ is an environmental noise intensity parameter and $W(t)$ is a standard Wiener process.

This type of stochastic models have been applied in the particular case of specific forms of h. For instance, in [8] one can see an application to tree growth and, more recently, in [11], an application to fish gowth. But most authors studying growth use inappropriate regression methods, adjusting data by minimum squares to a growth curve, usually the solution of an ordinary differential equation of type (1), thus assuming that the differences between the curve and the observations are or behave like measurement errors. This is totally inadequate if the deviations are, as we assume here, the result of environmental fluctuations of a random nature that affect growth.

The solution of (2) is a homogeneous diffusion process with drift coefficient $a(y) = \beta\,(A - y)$ and diffusion coefficient $b(y) = \sigma^2$. The drift coefficient is the mean speed of growth described by $Y(t)$ and the diffusion coefficient gives a measure of the local magnitude of the fluctuations.

To solve the equation is convenient to use the change of variable $Z(t) = Y(t)e^{\beta t}$ to obtain the SDE $dZ(t) = A\beta e^{\beta t}dt + \sigma e^{\beta t}dW(t)$ and solve it by direct integration between t_0 and t. We get $Z(t) = Z(t_0) + A\left(e^{\beta t} - e^{\beta t_0}\right)dt + \sigma\int_{t_0}^{t} e^{\beta u}dW(u)$ and finally the explicit solution

$$Y(t) = A - (A - y_0)e^{-\beta(t-t_0)} + \sigma e^{-\beta t}\int_{t_0}^{t} e^{\beta u}dW(u). \tag{3}$$

Since we have a deterministic integrand, the distribution of the stochastic integral in

 ISBN: 978-84-612-9727-6

(3) is Gaussian with zero mean and variance $\int_{t_0}^{t} \left(e^{\beta u}\right)^2 du$. Consequently,

$$Y(t) \curvearrowright \mathcal{N}\left(A - (A - y_0)e^{-\beta(t-t_0)}, \frac{\sigma^2}{2\beta}\left(1 - e^{-2\beta(t-t_0)}\right)\right), \tag{4}$$

meaning that $Y(t)$ has a Gaussian distribution with the indicated mean and variance. Leting $t \to +\infty$, one sees that there is an asymptotic distribution of Y, which is $\mathcal{N}\left(A, \frac{\sigma^2}{2\beta}\right)$. It is also possible to show that the process is ergodic (see, for instance, [1]). From (3), one obtains, for $s < t$, that $Y(t) = A-(A-Y(s))e^{-\beta(t-s)}+\sigma e^{-\beta t}\int_s^t e^{\beta u}dW(u)$. Therefore, the transition distribution between s and t, i.e., the conditional distribution of $Y(t)$ given $Y(s) = y$, is

$$Y(t)|_{Y(s)=y} \curvearrowright \mathcal{N}\left(A - (A - y)e^{-\beta(t-s)}, \frac{\sigma^2}{2\beta}\left(1 - e^{-2\beta(t-s)}\right)\right) \tag{5}$$

and the transition density is

$$p(t, z|s, y) = \frac{1}{\sqrt{2\pi\frac{\sigma^2}{2\beta}\left(1 - e^{-2\beta(t-s)}\right)}} \exp\left(-\frac{(z - A + (A - y)e^{-\beta(t-s)})^2}{2\frac{\sigma^2}{2\beta}\left(1 - e^{-2\beta(t-s)}\right)}\right). \tag{6}$$

From those transient, asymptotic and transition distributions of the modified size, one can easily obtained the corresponding distributions of the real size X.

Notice that, even though the animal size keeps varying driven by the environmental fluctuations, its distribution settles down to an asymptotic regimen.

3 Parameter estimation and prediction

Assume we have an animal (or plant) growing according to model (2). The parameter vector to be estimated is $\mathbf{p} = (A, \beta, \sigma)$.

Assume we measure the animal size at ages $t_0 < t_1 < t_2 < \ldots < t_n$. Let $X_k = X(t_k)$ be the size at age t_k $(k = 0, 1, 2, \ldots, n)$ and let the corresponding modified size be $Y_k = h(X_k)$. Assuming the initial value $Y_0 = y_0$ to be known. Since the process is Markov, the likelihood function of the observations is (in terms of the modified size) just the product of the transition densities (given by (6)) between consecutive ages of observation, and so the log-likelihood is given by

$$\begin{aligned} L(\mathbf{p}) &= -\frac{n}{2}\ln(\pi) - \frac{n}{2}\ln\left(\sigma^2\right) + \frac{n}{2}\ln\left(\beta\right) - \frac{1}{2}\sum_{k=1}^{n}\ln\left(1 - e^{-2\beta\delta_k}\right) \\ &\quad -\frac{\beta}{\sigma^2}\sum_{k=1}^{n}\frac{\left(y_k - A + (A - y_{k-1})\,e^{-\beta\delta_k}\right)^2}{1 - e^{-2\beta\delta_k}}, \end{aligned} \tag{7}$$

where $\delta_k = t_k - t_{k-1}$.

The maximum likelihood (ML) estimator $\hat{\mathbf{p}} = (\hat{A}, \hat{\beta}, \hat{\sigma})$ is just obtained by maximization of $L(\mathbf{p})$. As $n \to +\infty$ and $t \to +\infty$, the ML estimators are consistent and

asymptotically Gaussian with mean $\mathbf{p}$ and variance-covariance matrix $\mathbf{V} = \mathbf{F}^{-1}$, where $\mathbf{F}$ is the Fisher information matrix with elements given by $\mathbf{F}_{i,j} = -E\left[\partial^2(\mathbf{p})/\partial p_i \partial p_j\right]$. The expressions for $\mathbf{F}_{i,j}$ can be explicitly computed using the properties of the $Y(t)$ process. They do depend on the true (unknown) parameter values $\mathbf{p}$ and can be approximated by replacing these true values by their ML estimators $\hat{\mathbf{p}}$. This leads, by inverting the resulting approximation $\hat{\mathbf{F}}$ of the matrix $\mathbf{F}$, to an approximation $\hat{\mathbf{V}}$ of $\mathbf{V}$, which allows the construction of asymptotic confidence intervals for the parameters.

The data we have used for illustration was taken from cattle, namely the Mertolengo cattle breed and was collected by C. J. Roquete. This cattle breed is, at the moment, considered by many as the Portuguese breed with higher progression in terms of population increment and market potential. The data we work with comes from animals raised in "Herdade da Abóboda" in the Serpa region at the left margin of the Guadiana river. The animals were raised in pasture, together with their mothers during nursing and latter supplemented with silage when pasture is in shortage (from August till January).

In [6] we have applied, using software R, these techniques to the evolution of the weight (using 1 Kg as the weight unit) of a Mertolengo cow, where we have 79 observations since birth ($t_0 = 0$) to over 5 years of age. We have used several stochastic models, i.e. the model class (2) with several choices of the function h. Here we just present the results for the Gompertz model ($h(x) = \ln x$) and the Bertalanffy-Richards model with $c = 1/3$ ($h(x) = x^{1/3}$). Table 1 presents the ML parameter estimates and their asymptotic 95% approximated confidence intervals. instead of showing the parameter A (asymptotic average modified weight), we chose to present the equivalent parameter $S = h^{-1}(A)$, which is the asymptotic average weight (notice, however, that this is not an arithmetic average). The value of β is per year and the parameter σ^2 is also per year. Figure 1 shows the trajectory of the cow's observed weight and the curves corresponding to the "average" behavior of the models using the ML parameter values of S and β (curves corresponding to the case of no environmental fluctuations, i.e. $\sigma = 0$).

Table 1: Maximum likelihood estimates and asymptotic 95% approximated confidence intervals for the parameters S, β and σ using data on one Mertolengo cow.

	S	β	σ
Gompertz	407.1 ± 60.5	1.472 ± 0.354	0.226 ± 0.036
Bertalanffy-Richards with $c = 1/3$	422.4 ± 81.6	1.096 ± 0.419	0.525 ± 0.083

Let us consider the case where, instead of one animal, we have observations from m animals, all having the same parameters and being independent realizations of the stochastic process. Let, for animal number j ($j = 1, 2, ..., m$), $t_{j,k}$ ($k = 0, 1, 2, ..., n_j$) be the ages at which we observe its size $X_{j,k} = X(t_{j,k})$ and let $Y_{j,k} = h(X_{j,k})$, $\delta_{j,k} =$

 ISBN: 978-84-612-9727-6

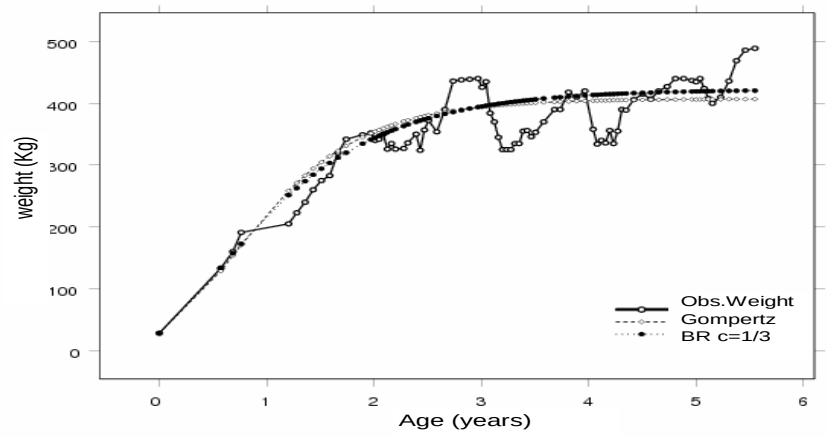


Figure 1: Observed growth curves (weight in Kg) of a Mertolengo cow. Growth curves of the Gompertz and Bertalanffy-Ricards ($c = 1/3$) models using ML estimators (see Table 1) of the parameters and puting $\sigma = 0$ (i.e., assuming absence of environmental fluctuations).

$t_{j,k} - t_{j,k-1}$. Let $L_j(\mathbf{p})$ be the likelihood based on the observations of animal number j:

$$\begin{aligned} L_j(\mathbf{p}) &= -\tfrac{n_j}{2}\ln(\pi) - \tfrac{n_j}{2}\ln\left(\sigma^2\right) + \tfrac{n_j}{2}\ln\left(\beta\right) - \tfrac{1}{2}\textstyle\sum_{k=1}^{n_j}\ln\left(1-e^{-2\beta\delta_{j,k}}\right) \\ &\quad -\tfrac{\beta}{\sigma^2}\textstyle\sum_{k=1}^{n_j}\dfrac{\left(y_{j,k}-A+\left(A-y_{j,k-1}\right)e^{-\beta\delta_{j,k}}\right)^2}{1-e^{-2\beta\delta_{j,k}}}. \end{aligned} \tag{8}$$

The log-likelihood for all observations is just the sum of these log-likelihoods:

$$L_{1,2,\ldots,m}(\mathbf{p}) = \sum_{j=1}^{m} L_j(\mathbf{p}). \tag{9}$$

We now work to optimize $L_{1,2,\ldots,m}(\mathbf{p})$ and the process of obtaining the ML estimators and their asymptotic confidence intervals is similar.

We have applied these procedure to observations on $m = 97$ Mertolengo cows for which we had a total of 2129 observations. The results are in Table 2.

Table 2: Maximum likelihood estimates and asymptotic 95% approximated confidence intervals for the parameters S, β and σ using data on 97 Mertolengo cows.

	S	β	σ
Gompertz	411.2 ± 8.0	1.676 ± 0.057	0.302 ± 0.009
Bertalanffy-Richards with $c = 1/3$	425.7 ± 9.4	1.181 ± 0.057	0.597 ± 0.019

 ISBN: 978-84-612-9727-6

Either for one or several trajectories (one or several animals), if the sample size and/or the age range are not sufficiently long for the asymptotically based approximations to be reasonably precise, one can use bootstrap methods (see [5] and [7]).

4 Time to reach a given size

Let us consider thresholds q^* and Q^*, one low and one high, for the animal size $X(t)$.

We are interested in the time required for an animal to reach size Q^* for the first time, which can be a maturity size for selling the animal to the meat market. Since $Y(t)$ and $X(t)$ are related through the strictly increasing function h, this is also the first passage time of $Y(t)$ (modified size) by $Q = h(Q^*)$. Let us denote it by T_Q.

Let $q = h(q^*)$ and assume that $-\infty < q < y_0 < Q < +\infty$ (q and Q both in the interior of the state space of Y). Let T_q be the first passage time of $Y(t)$ by q and let $T_{qQ} = \min(T_q, T_Q)$ be the first passage time of $Y(t)$ through either of the thresholds q and Q. Denote the k-th order moment of T_{qQ} by

$$U_k(y_0) = E[(T_{qQ})^k | Y(0) = y_0].$$

Let us define, in the interior of the state space, the scale and speed measures of $Y(t)$. The scale density is

$$s(y) = \exp\left(-\int_{y^*}^{y} \frac{2a(\theta)}{b(\theta)} d\theta\right) \tag{10}$$

and the speed density is

$$m(y) = \frac{1}{s(y)b(y)}, \tag{11}$$

where y^* is an arbitrary (but fixed) point in the interior of the state space and where a and b are the drift and diffusion coefficients of $Y(t)$. In our case, $a(y) = \beta(A - y)$ and $b(y) = \sigma^2$, and so,

$$\begin{aligned} s(y) &= C \exp\left(-\tfrac{2\beta A}{\sigma^2} y + \tfrac{\beta}{\sigma^2} y^2\right) \\ m(y) &= \tfrac{1}{\sigma^2 s(y)}, \end{aligned} \tag{12}$$

where C is a constant. The "distribution" functions of these measures are the scale function and speed function defined by $S(z) = \int_{x^*}^{z} s(u)du$ and $M(z) = \int_{x^*}^{z} m(u)du$, where x^* is an arbitrary (but fixed) point in the interior of the state space. The associated scale measure and speed measures are defined, for Borel sets B, by $S(B) = \int_B s(u)du$ and $M(B) = \int_B m(u)du$.

One can see, for instance in [9] or [10], that

$$u(y_0) := P[T_Q < T_q | Y(0) = y_0] = \frac{S(y_0) - S(q)}{S(Q) - S(q)}. \tag{13}$$

and that $U_k(y_0)$ satisfies the differential equation

$$\frac{1}{2} b(y_0) \frac{d^2 U_k(y_0)}{dy_0^2} + a(y_0) \frac{dU_k(y_0)}{dy_0} + kU_{k-1}(y_0) = 0,$$

 ISBN: 978-84-612-9727-6

which is easily seen to be equivalent to

$$\frac{1}{2}\frac{d}{dM(y_0)}\left(\frac{dU_k(y_0)}{dS(y_0)}\right)+kU_{k-1}(y_0)=0. \tag{14}$$

Integrating with respect to $M(y_0)$ and with respect to $S(y_0)$, using the conditions $U_k(q)=U_k(Q)=0$ $(k=1,2,...)$ and (13), one obtains, for $k=1,2,...$, the solution

$$\begin{array}{ll} U_k(y_0) & =2u(y_0)\int_{y_0}^{Q}(S(Q)-S(\xi))kU_{k-1}(\xi)m(\xi)d\xi \\ & +2(1-u(y_0))\int_{q}^{y_0}(S(\xi)-S(q))kU_{k-1}(\xi)m(\xi)d\xi. \end{array} \tag{15}$$

Since $U_0(y_0)\equiv 1$, one can iteratively obtain the moments of any arbitrary order of T_{qQ}.

One can also obtain a differential equation for the Laplace transform $\Lambda(y_0):=E[\exp(-\lambda T_{qQ})|Y(0)=y_0]$, namely $\frac{1}{2}b(y_0)\frac{d^2\Lambda(y_0)}{dy_0^2}+a(y_0)\frac{d\Lambda(y_0)}{dy_0}-\lambda\Lambda(y_0)=0$, with condition $\Lambda(q)=\Lambda(Q)=1$. If one can solve this equation, the p.d.f. of T_{qQ} can be obtained by inverting the Laplace transform.

We can apply (15) to our model (2). Since the process $Y(t)$ is ergodic, we can obtain the distribution (and moments) of T_Q as the limiting case of the distribution (moments) of T_{qQ} when $q\downarrow-\infty$.

Let us denote by $V_k(y_0):=E[(T_Q)^k|Y(0)=y_0]$ the k-th order moment of T_Q. Taking the limit as $q\downarrow-\infty$ in (15), one obtains

$$V_k(y_0)=2\int_{y_0}^{Q}s(\xi)\left(\int_{-\infty}^{\xi}kV_{k-1}(\theta)m(\theta)d\theta\right)d\xi. \tag{16}$$

For our model, (12) holds. Replacing in (16) with $k=1$ and using $V_0(y_0)\equiv 1$, one gets, after computing the inner integral and making some changes of variables,

$$E[T_Q]=V_1(y_0)=\frac{1}{\beta}\int_{\sqrt{2\beta}(y_0-A)/\sigma}^{\sqrt{2\beta}(Q-A)/\sigma}\frac{\Phi(y)}{\phi(y)}dy, \tag{17}$$

where Φ and ϕ are the distribution function and the probability density function of a standard normal random variable. Replacing this result in (16) with $k=2$, one obtains, after some manipulations

$$E\left[(T_Q)^2\right]=V_2(y_0)=\frac{2}{\beta^2}\int_{\sqrt{2\beta}(y_0-A)/\sigma}^{\sqrt{2\beta}(Q-A)/\sigma}\frac{1}{\phi(z)}\int_{-\infty}^{z}\frac{\Phi^2(y)}{\phi(y)}dydz+\left(V_1(y_0)\right)^2. \tag{18}$$

Using (17) and (18), one obtains

$$VAR[T_Q]=\frac{2}{\beta^2}\int_{\sqrt{2\beta}(y_0-A)/\sigma}^{\sqrt{2\beta}(Q-A)/\sigma}\frac{1}{\phi(z)}\int_{-\infty}^{z}\frac{\Phi^2(y)}{\phi(y)}dydz. \tag{19}$$

To obtain the mean and variance of T_Q (which, by the way, are typically of the same order of magnitude), one needs to numerically integrate in (17) and (19).

To give an example, consider the unit of weight to be 1 Kg and the case of a Mertolengo cow born with a weight of $X(0)=x_0=30$ and assume we want to determine

 ISBN: 978-84-612-9727-6

the time required for the animal to reach the size of $Q^* = 380$. Assume the cow grows according to a Gompertz model (model (2) with $h(x) = \ln x$) with parameters $S = 407.1$, $\beta = 1.472$ per year and $\sigma^2 = (0.226)^2$ per year (values estimated by ML for the cow considered in Table 1). We have $Q = \ln 380$, $y_0 = \ln 30$ and $A = \ln 407.1$. We have obtained as mean time to reach the desired threshold 1.51 years and as standard deviation of that time 1.01 years.

5 Case of A randomly varying among individuals

So far we have considered either one individual or several independent individuals having the same parameters. However, although it is advisable to work with groups of individuals that are relatively homogeneous (that is the case of our cows that are all from the same Mertolengo breed and all of the Rosilho type), some genetic heterogeneity among individuals is almost inevitable. Sometimes, also the raising conditions of the individuals are subjected to some heterogeneity. We now consider the case where such heterogeneity affects the asymptotic average size S, which is now allowed to vary randomly and independently among individuals according to some probability distribution. We will assume that, in terms of $A = h(S)$, which is now a random variable (with different values for different animals), the distribution is Gaussian $\mathcal{N}\left(\alpha, \theta^2\right)$. Assume further that A is independent of the Wiener process $W(t)$.

Looking at an animal for which $A = a$, we have that $Y(t)$ conditional on $A = a$ follows the SDE (2) with A replaced by a. Therefore, $Y(t)$ conditional on $A = a$ is given by (3) with A replaced by a.

Therefore, the conditional distribution of $Y(t)$ given A is

$$\mathcal{N}\left(A - (A - y_0)e^{-\beta(t-t_0)}, \frac{\sigma^2}{2\beta}\left(1 - e^{-2\beta(t-t_0)}\right)\right).$$

For a randomly chosen animal, we need the unconditional distribution of $Y(t)$, which is therefore Gaussian with mean

$$m(t) = E[Y(t)] = E[E[Y(t)|A]] = E[A - (A - y_0)e^{-\beta(t-t_0)}] = \alpha - (\alpha - y_0)e^{-\beta(t-t_0)}$$

and variance

$$\begin{aligned} VAR\left[Y(t)\right] &= E\left[E\left[\left(Y(t) - m(t)\right)^2 |A\right]\right] \\ &= E\left[E\left[\left((A-\alpha)\left(1 - e^{-\beta(t-t_0)}\right) + \sigma e^{-\beta t}\int_{t_0}^{t} e^{\beta u} dW(u)\right)^2 |A\right]\right] \\ &= E\left[(A-\alpha)^2\left(1 - e^{-\beta(t-t_0)}\right)^2 + \sigma^2 e^{-2\beta t}\int_{t_0}^{t} e^{2\beta u} du\right] \end{aligned}$$

Therefore, the unconditional distribution of $Y(t)$ is

$$Y(t) \curvearrowright \mathcal{N}\left(\alpha - (\alpha - y_0)e^{-\beta(t-t_0)}, \theta^2\left(1 - e^{-\beta(t-t_0)}\right)^2 + \frac{\sigma^2}{2\beta}\left(1 - e^{-2\beta(t-t_0)}\right)\right). \quad (20)$$

 ISBN: 978-84-612-9727-6

Notice that the variance is the sum of two components, one due to random variation of A among the animals and the other due to environmental fluctuations (similar to the case of fixed A). The same happens to the asymptotic distribution

$$Y(t) \curvearrowright \mathcal{N}\left(\alpha, \theta^2 + \frac{\sigma^2}{2\beta}\right).$$

We have also developed estimation methods for this more general situation that are soon to be published.

Should there be some nontrivial variability among animals with respect to A, this generalized model, in which A varies randomly from animal to animal, is quite useful if there is little or no data on the specific animals being raised at the moment, as is commonly the case. Predictions and optimization can still be made if there is prior knowledge of the parameters due to previous studies based on data from many animals of the same breed.

6 Conclusions

We have proposed a general class of stochastic differential equation models for the growth of the size of individual animals in a randomly varying environment. These models might be useful in cattle breeding or forestry in order to optimize the exploitation of such resources. We have argued that regression methods are not adequate for modeling growth under environmental varying conditions (the environment is taken here in the broad sense of factors that affect the growth rate of the animal). The class of models we propose are stochastic versions of a deterministic general model that is both simple in form and flexible enough to comprise most of the commonly used deterministic growth models. The trick is to use an increasing function h that converts the animal size to a modified size that follows a simple dynamical equation.

We have studied the properties of the model, with emphasis on the distribution of the size at an age t and on the transitional distribution between two ages. We have also seen that the models lead to a stochastic equilibrium in which, although the animal keeps changing driven by the environmental fluctuations, the probability distribution of the animal size settles down to an equilibrium distribution, which we also compute.

We have then studied how to estimate the model parameters by maximum likelihood and how to obtain approximate confidence intervals for the estimators when we have data on measurements at several ages of one animal or of several animals. Prediction of future sizes and their confidence intervals is also possible and easily done but not shown here. We refer the reader to other publication for the use of bootstrap methods to improve the approximation in the computation of confidence intervals of the parameter estimators when the data is less abundant or spread over a short age span.

We have determined explicit expressions (in the form of simple integrals that can be numerically computed) for the mean and standard deviation of the time required for an animal to reach a given size for the first time. When that size is the size at which the animal is supposed to be sold for the meat market (or a tree to be cut), one can

 ISBN: 978-84-612-9727-6

see the economic importance of such result. There are many more important economic issues that can be handled, including optimization problems.

Since animals may differ in their asymptotic average sizes due to random genetic or other differences, we have considered also the case where the average asymptotic size varies randomly among animals according to a probability distribution. We have obtained the probability distribution of the animal size. In terms of the modified size, its mean has the same expression as for the case of fix average asymptotic size (except that we use the mean value instead of the fixed value of the modified average asymptotic size). However the variance of the modified size is the sum of two components, one due to the environmental variations during the animal's life (the same as in the fixed average asymptotic size case) and the other due to variations of the average asymptotic size among animals.

Acknowledgements

This work was financed by FCT (Fundação para a Ciência e a Tecnologia) within the research project PTDC/MAT/64297/2006. The first three authors are members of the Centro de Investigação em Matemática e Aplicações and the fourth author is a member of the Instituto de Ciências Agrárias e Ambientais Mediterrânicas, both research centers of the Universidade de Évora financed by FCT.

References

[1] C. A. Braumann, *Introdução às Equações Diferenciais Estocásticas*, Edições SPE, Lisboa, 2005.

[2] C. A. Braumann, *Growth and extinction of populations in randomly varying environments*, Computers and Mathematics with Appl. **56** (2008), 631–644.

[3] C. Carlos and C. A. Braumann, *Tempos de extinção para populações em ambiente aleatório e cálculos de Itô e de Stratonovich*, Ciência Estatística, Actas do XIII Congresso Anual da Sociedade Portuguesa de Estatística (L. Canto e Castro, E. G. Martins, C. Rocha, M. F. Oliveira, M. M. Leal and F. Rosado, eds.), Edições SPE (2006) 229–238.

[4] P. A. Filipe and C. A. Braumann, *Animal growth in random environments: estimation with several paths*, Bull. Int. Statistical Institute **LXII** (2007), in press.

[5] P. A. Filipe and C. A. Braumann, *Modelling individual animal growth in random environments*, Proceedings of the 23rd International Workshop on Statistical Modelling (P. H. C. Eilers, ed.) (2008) 232–237.

[6] P. A. Filipe, C. A. Braumann and C. J. Roquete, *Modelos de crescimento de animais em ambiente aleatório*, Estatística Ciência Interdisciplinar, Actas do XIV

Congresso Anual da Sociedade Portuguesa de Estatística (M. E. Ferrão, C. Nunes and C. A. Braumann, eds.), Edições SPE (2007) 401–410.

[7] P. A. Filipe, C. A. Braumann and C. J. Roquete, *Crescimento individual em ambiente aleatório: várias trajectórias*, Estatística - da Teoria à Prática, Actas do XV Congresso Anual da Sociedade Portuguesa de Estatística (M. M. Hill, M. A. Ferreira, J. G. Dias, M. F. Salgueiro, H. Carvalho, P. Vicente and C. A. Braumann, eds.), Edições SPE (2008) 259–268.

[8] O. Garcia, *A stochastic differential equation model for the height of forest stands*, Biometrics (1983) 1059–1072.

[9] I. I. Ghikman and A. V. Skorohod, *Stochastic Differential Equations*, Springer, Berlin, 1991.

[10] S. Karlin and H. M. Taylor, *A Second Course in Stochastic Processes*, Academic Press, New York, 1981.

[11] Lv. Qiming and J. Pitchford, *Stochastic Von Bertalanffy models, with applications to fish recruitment*, J. Theoretical Biology **244** (2007) 640–655.

 ISBN: 978-84-612-9727-6

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Virtual detectors, transformation and application to pattern recognition problems

Alexander Buslaev[1] and Marina Yashina[2]

[1] *Department of mathematics, Moscow State Automobile and Road Technical University*

[2] *Department of of math. cybernetics, Moscow Technical University of Communication and Informatics*

emails: apal2006@yandex.ru, yash-marina@yandex.ru

Abstract

In artificial intelligence systems there is an actual problem to computerize the process of primary collecting via video stream. In this connection computer vision is developing. In this paper the mathematical approach to processing algorithms of virtual detectors and application to pattern recognition problems are considered. The recovery problem of object with smooth boundary is researched. This method is generalized for object with piece-smooth boundary. The method of an estimation of "image depth" is presented and the measure on a digital matrix is introduced for object detecting problem on a homogeneous background.

Key words: Virtual detectors, video stream, piece-smooth boundary, object recovery

1 Introduction

We consider video stream with fixed panorama and periodicity Δt. There are virtual sensors on the monitor screen. We call them as control domain (CD). CD are a rectangle domain on screen to register and cumulate color perturbations: x_{ij}, where $i = 1, .., m$, m is quantity of sensors, $j = 1, n$, n is quantity of registrations.

It is possible to consider three variants of coordinates x_{ij}:

(a) binary $x_{ij} = 0 \vee 1$;

(b) gray-level intensity $x_{ij} \in [0, 1]$,

(c) and RGB-color coordinates $x_{ij} = (r, g, b)$.

We should take into account that distances up to physical objects are various which images are present in video stream frames. Hence always when a recognition problem is formulating it is necessary to know "depth of the space" forming the image.

2 Classical cinema (depth of the image is const)

We assume, the camera observes the screen located perpendicularly, and events are generated by means of a projector on the screen. It means that visible and generated distortions are minimum. So, the screen is a white background on which the object moves, for example, a black square. *(If the square does not move, we observe, apparently, Malevich's Square).*

It is required the following:

a) to recover the square position at the fixed time moment;

or

b) to restore a trajectory of the square centre for example.

If using "development", i.e. vertical (or horizontal) scan of the screen, so the algorithm looks as follows:

1) Find the first vertical line with black points (a top A);

2) Search minimum and maximum (the lowermost and uppermost point) deviations (tops $C,\ D$);

3) Find last vertical line with black points (a top B).

Thereby, practically in all cases, the square tops are defined by vertical and horizontal boundaries. Thus, if the size of a square $M \times M$ is small in comparison with the size of the screen (for example, $N^2 = N \times N$), the number of operations for search has an order N^2 (search almost all points of the screen, $N(N - M)$).

We consider another strategy. We will choose a uniform network of detectors with distance between neighborhoods $(M - 1)$ on the screen, i.e. $(N/(M - 1))^2$ points only. Then at any position of the square it will be obligatory contain a point from a control network (x^*, y^*) and it will be detected.

To recover a figure completely it will enough check up a square with the side of $2\sqrt{2}M$ with the centre in (x^*, y^*), i.e. make $8M^2$ operations. Thus, total amount of operations:

$$\frac{N^2}{(M-1)^2} + 8M^2 \tag{1}$$

The presented strategy will have advantage over the previous algorithm if $M > 2$.

3 Recovery of the area with a smooth boundary

3.1. We consider one-coherent area with finite smooth boundary. It is required to construct a field of control detectors on the screen for image restoration *constant depth* with the set accuracy.

The problem, obviously, is reduced to restoration the smooth line by discrete amount of points set in the metrics l^2_∞ with a mistake. Methods of this problem solution are widely presented in managements of experiments planning [1], numerical methods [3], interpolation theories [2], splines [3], [4].

The question of control approached points choice can be solved taking into account restrictions on boundary curvature (see Fig. 1).

 ISBN: 978-84-612-9727-6

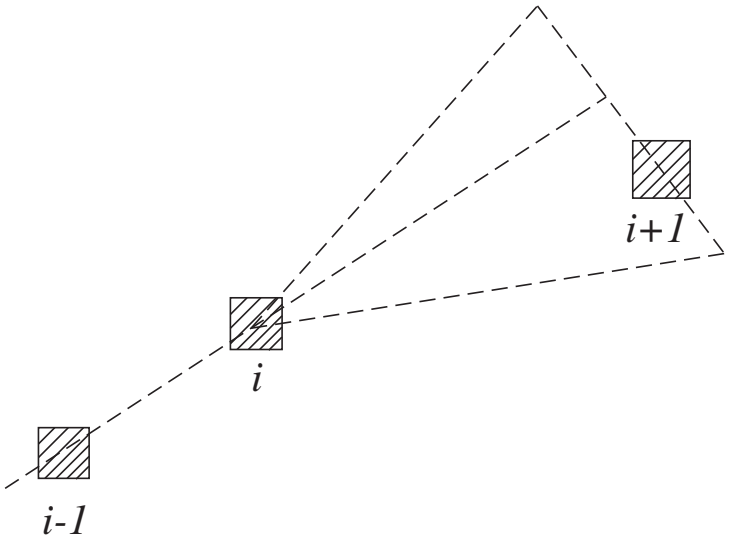


Figure 1: Control domains (CD)

Search area of points with number $i+1$ is essentially narrowed

$$\rho(i,i+2) \sim \rho(i,i-1);$$

$$\frac{\left(\overrightarrow{i-1,i}\right) \times \left(\overrightarrow{i,i+1}\right)}{||\overrightarrow{i-1,i}| \times |\overrightarrow{i,i+1}|} \leq C. \tag{2}$$

3.2. It is possible to consider a problem of the recovery of a point trajectory, which depending on time moves on smooth boundary. I.e. we have two functions $(x_1(t), x_2(t))$, for which, for example,

$$|\ddot{x}_i(t)| \leq 1, \qquad i = 1, 2, ... \tag{3}$$

And we can measure $x_i(t)$, $i = 1, 2...$ in derivative points t with accuracy ε. Theoretically it is an accuracy of trajectory restoration and a parametrical vector function. However it is unreal to make calculations in each point for recovery of $(x_1(t), x_2(t))$. In this case the problem formulation in Tikhonov and Stechkin's sense is correct.

4 Depth of the image and a measure on a digital matrix

At the fixed image, i.e. a condition of a digital matrix of the image generator, the depth of each pixel is measured by distance to an image source.

Means, "all" pixels are defined by distance to a source (prototype) of this pixel and the image size of object inversely proportional to distance to object.
Hence, the metrics on the image with variable depth can be simulated with fractional-rational function. In this case the size of the virtual detector inversely is proportional

 ISBN: 978-84-612-9727-6

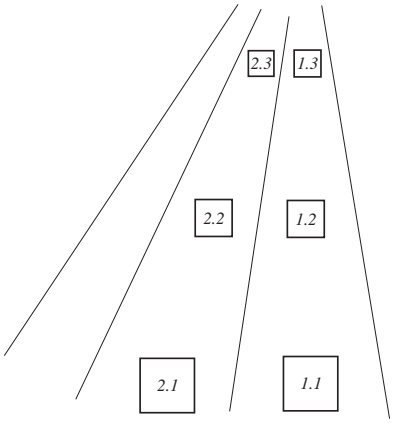


Figure 2: The volume image. Prospect is the image with depth.

to the depth of corresponding area of the image. As an example we consider the traffic image on a multilane highway as shown in Fig. 2, [5].

There are three detectors x_1,x_2,x_3 in fig. 2, they can solve some problems:

(1). Traffic intensity on a lane

$$i = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3$$

where α_i, $i = 1, 2, 3$ weight co-ordinates,

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \qquad \alpha_i \geq 0.$$

The choice of coefficients depends on concomitant factors.

(2). The estimation of movement co-ordinates, speed and acceleration.

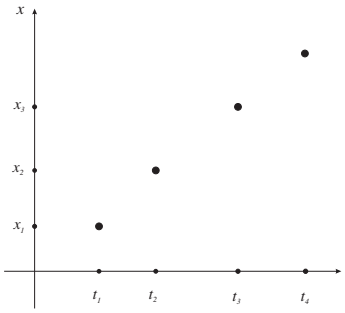


Figure 3: Recovery of movement trajectory

It is necessary to carry out monotonous interpolation of monotonously increasing table in points $t_i, i = 1, 2...$ and with a continuous first derivative. It can be made, for example, by means of parabolic splines with nodes in intermediate points $t_1 < \tau_1 < t_2 < \tau_2 < ...$, [2]. Concerning other approaches see [4].

(3). Division (segregation) of objects, moving on different lanes.

 ISBN: 978-84-612-9727-6

The object is defined as border of moving with identical speed set of pixels of images at processing consecutive shots (method of contours). If in a contour gets two (some) virtual detectors, then it is detected as individual on that strip, which is closer to an axis of the video generator (by default), or, in the presence of obvious properties, for example, a shadow, with a directive choice, [7],[8], [9].

(4). Recovery of a shape of the object which rotates around a fixed points by projections, [6].

It allows the development of computer vision based on network video camera with rotary mechanisms with recognition system.

Acknowledgements

This work has been partially supported by RFBR grant No 08-01-000959-a.

References

[1] Albert J., Nilson E. Uolsh J., *The theory of splines and its application*, . Mir, 1972

[2] Stechkin S.B., Subbotin J.N., *Splines in calculus mathematics*, M. Science, 1976

[3] Tihonov A.N., Samarsky A.A., *Methods of the theory of incorrect problems*, M. Nauka, 1987.

[4] Zavjalov J.S., Kvasov B.I., Miroshnichenko V.L., *Methods of a spline functions*, Moscow, Nauka, 1980.

[5] Buslaev A.P., Dorgan V.V., Prikhodko V.M., Travkin V.Ur., Yashina M.V., *Image recognition and monitoring of road pavement, traffic flows and movement safety factors*, Vestnik MADI(STU), ISBN 5-79620061-5. No 4, (2005) 102–109

[6] Buslaev A.P., Guo J.M., Wang N.J., Yashina M.V., *On recovery of plane object shape by projections.* , To appear in Journal of Mathematical Imaging and Vision, (2009)

[7] Buslaev A.P., Kuzmin D.M., Yashina M.V., *Computer methods of information processing and images recognition in problems of traffic and communication. Part 1–4*, Teacher book. MTICI (2008).

[8] Buslaev A.P., Kuzmin D.M., Yashina M.V., *Computer methods of information processing. Mobile Road Laboratory.* MADI(STU) (2008).

[9] Buslaev A.P., Novikov , V.M. Prikhodko, A.G. Tatashev, Yashina M.V., *Stochastic and simulation approach to traffic*, Mir, Moscow (2003).

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009 30 June, 1–3 July 2009.

A numerical code for fast interpolation and cubature at the Padua points

M. Caliari[1], S. De Marchi[1], A. Sommariva[2] and M. Vianello[2]

[1] *Department of Computer Science, University of Verona*

[2] *Department of Pure and Applied Mathematics, University of Padua*

emails: marco.caliari@univr.it, stefano.demarchi@univr.it, alvise@math.unipd.it, marcov@math.unipd.it

Abstract

We present a numerical code in Matlab/Octave that implements fast versions of the Lagrange interpolation formula, and of the corresponding algebraic cubature formula, at the so-called Padua points in rectangles.

Key words: Padua points, fast algorithms, Lagrange interpolation, Fast Fourier Transform, algebraic cubature

1 Introduction

In this talk we discuss an efficient implementation in Matlab/Octave of bivariate interpolation and cubature at the so-called Padua points. Such points are the first known example of optimal points for total degree polynomial interpolation in two variables, with a Lebesgue constant increasing like log square of the degree; see [1, 2, 4, 5]. Moreover, the associated algebraic cubature formula has shown a very good behavior, comparable to that of the one-dimensional Clenshaw–Curtis rule, cf. [9].

The $N = (n+1)(n+2)/2 = \dim\left(\mathbb{P}_n^2\right)$ Padua points, $n > 0$, are the set

$$\mathrm{Pad}_n = \{\boldsymbol{\xi} = (\xi_1, \xi_2)\} = \left\{\gamma\left(\frac{k\pi}{n(n+1)}\right), \quad k = 0, \ldots, n(n+1)\right\}$$

where $\gamma(t)$ is their "generating curve" (cf. [1])

$$\gamma(t) = (-\cos((n+1)t), -\cos(nt)), \quad t \in [0, \pi] \tag{1}$$

The Padua points (for n even) were introduced for the first time in [4, formula (9)] (in that formula there is a misprint, $n-1$ has to be replaced by $n+1$). Denoting by C_{n+1} the set of the $n+1$ Chebyshev–Gauss–Lobatto points

$$C_{n+1} = \{z_j^n = \cos((j-1)\pi/n),\ j = 1, \ldots, n+1\}$$

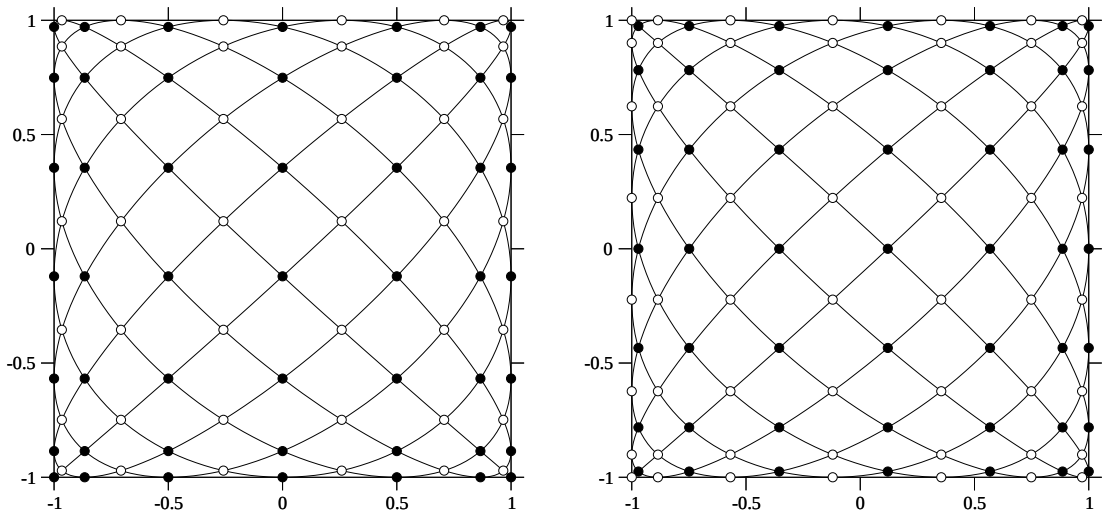


Figure 1: The Padua points with their generating curve for $n = 12$ (left, 91 points) and $n = 13$ (right, 105 points), also as union of two Chebyshev-like grids: filled bullets = $C^{\mathrm{E}}_{n+1} \times C^{\mathrm{O}}_{n+2}$, open bullets = $C^{\mathrm{O}}_{n+1} \times C^{\mathrm{E}}_{n+2}$.

and

$$C^{\mathrm{E}}_{n+1} = \{z^n_j \in C_{n+1},\ j-1 \text{ even}\}$$
$$C^{\mathrm{O}}_{n+1} = \{z^n_j \in C_{n+1},\ j-1 \text{ odd}\}$$

then

$$\mathrm{Pad}_n = (C^{\mathrm{E}}_{n+1} \times C^{\mathrm{O}}_{n+2}) \cup (C^{\mathrm{O}}_{n+1} \times C^{\mathrm{E}}_{n+2}) \subset C_{n+1} \times C_{n+2}$$

which is valid also for n odd (see Fig. 1).

The fundamental Lagrange polynomials of the Padua points are

$$L_{\boldsymbol{\xi}}(\boldsymbol{x}) = w_{\boldsymbol{\xi}} \left(K_n(\boldsymbol{\xi}, \boldsymbol{x}) - \frac{1}{2} \hat{T}_n(\xi_1) \hat{T}_n(x_1) \right) \tag{2}$$

where $K_n(\boldsymbol{x}, \boldsymbol{y})$, with $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{y} = (y_1, y_2)$, is the reproducing kernel of the space $\mathbb{P}^2_n([-1,1]^2)$ equipped with the inner product

$$\langle f, g \rangle = \frac{1}{\pi^2} \int_{[-1,1]^2} f(x_1, x_2) g(x_1, x_2) \frac{\mathrm{d}x_1}{\sqrt{1-x_1^2}} \frac{\mathrm{d}x_2}{\sqrt{1-x_2^2}}$$

that is

$$K_n(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=0}^{n} \sum_{j=0}^{k} \hat{T}_j(x_1) \hat{T}_{k-j}(x_2) \hat{T}_j(y_1) \hat{T}_{k-j}(y_2) \ .$$

Here $\hat{T}_j$ denotes the normalized Chebyshev polynomial of degree j, i.e. $\hat{T}_0 = T_0 \equiv 1$, $\hat{T}_p = \sqrt{2} T_p$, $T_p(\cdot) = \cos(p \arccos(\cdot))$. Moreover, the weights $w_{\boldsymbol{\xi}}$ are

$$w_{\boldsymbol{\xi}} = \frac{1}{n(n+1)} \cdot \begin{cases} 1/2 & \text{if } \boldsymbol{\xi} \text{ is a vertex point} \\ 1 & \text{if } \boldsymbol{\xi} \text{ is an edge point} \\ 2 & \text{if } \boldsymbol{\xi} \text{ is an interior point} \end{cases}$$

 ISBN: 978-84-612-9727-6

We notice that the $\{w_{\boldsymbol{\xi}}\}$ are indeed weights of a cubature formula for the product Chebyshev measure. Such a cubature formula stems from quadrature along the generating curve and is the key to obtaining the Lagrange polynomials (2); cf. [1].

2 Fast interpolation

The polynomial interpolation formula can be written in the bivariate Chebyshev orthonormal basis as

$$\mathcal{L}_n f(\boldsymbol{x}) = \sum_{\boldsymbol{\xi}\in \mathrm{Pad}_n} f(\boldsymbol{\xi}) w_{\boldsymbol{\xi}} \left(K_n(\boldsymbol{\xi},\boldsymbol{x}) - \frac{1}{2}\hat{T}_n(\xi_1)\hat{T}_n(x_1)\right)$$

$$= \sum_{k=0}^{n}\sum_{j=0}^{k} c_{j,k-j}\hat{T}_j(x_1)\hat{T}_{k-j}(x_2) - \frac{1}{2}\sum_{\boldsymbol{\xi}\in \mathrm{Pad}_n} f(\boldsymbol{\xi}) w_{\boldsymbol{\xi}}\hat{T}_n(\xi_1)\hat{T}_0(\xi_2)\hat{T}_n(x_1)\hat{T}_0(x_2)$$

$$= \sum_{k=0}^{n}\sum_{j=0}^{k} c_{j,k-j}\hat{T}_j(x_1)\hat{T}_{k-j}(x_2) - \frac{c_{n,0}}{2}\hat{T}_n(x_1)\hat{T}_0(x_2) \tag{3}$$

where the coefficients are defined as

$$c_{j,k-j} = \sum_{\boldsymbol{\xi}\in \mathrm{Pad}_n} f(\boldsymbol{\xi}) w_{\boldsymbol{\xi}}\hat{T}_j(\xi_1)\hat{T}_{k-j}(\xi_2), \quad 0\le j\le k\le n \tag{4}$$

and can be computed once and for all. First we define the $(n+1)\times(n+2)$ matrix computed corresponding to the Chebyshev-like grid $C_{n+1}\times C_{n+2}$ with entries

$$\mathbb{G}(f) = (g_{r,s}) = \begin{cases} w_{\boldsymbol{\xi}} f(\boldsymbol{\xi}) & \text{if } \boldsymbol{\xi} = (z_r^n, z_s^{n+1}) \in \mathrm{Pad}_n \\ 0 & \text{if } \boldsymbol{\xi} = (z_r^n, z_s^{n+1}) \in (C_{n+1}\times C_{n+2}) \setminus \mathrm{Pad}_n \end{cases}$$

Then, given a vector $S = (s_1,\ldots,s_m)\in[-1,1]^m$, we define the rectangular Chebyshev matrix

$$\mathbb{T}(S) = \begin{pmatrix} \hat{T}_0(s_1) & \cdots & \hat{T}_0(s_m) \\ \vdots & \ldots & \vdots \\ \hat{T}_n(s_1) & \cdots & \hat{T}_n(s_m) \end{pmatrix} \in \mathbb{R}^{(n+1)\times m} \tag{5}$$

Then it is easy to check (see [5]) that the coefficients $c_{j,l}$, $0\le j\le n$, $0\le l\le n-j$ are the entries of the upper-left triangular part of the matrix

$$\mathbb{C}(f) = \mathbb{T}(C_{n+1})\,\mathbb{G}(f)\,(\mathbb{T}(C_{n+2}))^t \tag{6}$$

where now $C_{n+1} = (z_1^n,\ldots,z_{n+1}^n)$ is the *vector* of the Chebyshev–Gauss–Lobatto points. A slightly more refined algorithm can be obtained, by exploiting the fact that the Padua points are union of two Chebyshev subgrids. Indeed, defining the two matrices

$$\mathbb{G}_1(f) = \left(w_{\boldsymbol{\xi}} f(\boldsymbol{\xi})\,,\ \boldsymbol{\xi} = (z_r^n, z_s^{n+1}) \in C_{n+1}^{\mathrm{E}}\times C_{n+2}^{\mathrm{O}}\right)$$

$$\mathbb{G}_2(f) = \left(w_{\boldsymbol{\xi}} f(\boldsymbol{\xi})\,,\ \boldsymbol{\xi} = (z_r^n, z_s^{n+1}) \in C_{n+1}^{\mathrm{O}}\times C_{n+2}^{\mathrm{E}}\right)$$

 ISBN: 978-84-612-9727-6

then we can compute the coefficient matrix as

$$\mathbb{C}(f) = \mathbb{T}(C^{\mathrm{E}}_{n+1})\,\mathbb{G}_1(f)\,(\mathbb{T}(C^{\mathrm{O}}_{n+2}))^t + \mathbb{T}(C^{\mathrm{O}}_{n+1})\,\mathbb{G}_2(f)\,(\mathbb{T}(C^{\mathrm{E}}_{n+2}))^t$$

by multiplying matrices of smaller dimension than those in (6). We term this approach MM (Matrix Multiplication) in the numerical tests.

Here, we pursue an alternative computational strategy, based on the special structure of the Padua points. Indeed, the coefficients $c_{j,l}$ can be rewritten as

$$c_{j,l} = \sum_{\boldsymbol{\xi}\in \mathrm{Pad}_n} f(\boldsymbol{\xi}) w_{\boldsymbol{\xi}} \hat{T}_j(\xi_1)\hat{T}_l(\xi_2) = \sum_{r=0}^{n}\sum_{s=0}^{n+1} g_{r,s}\hat{T}_j(z^n_r)\hat{T}_l(z^{n+1}_s)$$

$$= \beta_{j,l}\sum_{r=0}^{n}\sum_{s=0}^{n+1} g_{r,s}\cos\frac{jr\pi}{n}\cos\frac{ls\pi}{n+1} = \beta_{j,l}\sum_{s=0}^{M-1}\left(\sum_{r=0}^{N-1} g^0_{r,s}\cos\frac{2jr\pi}{N}\right)\cos\frac{2ls\pi}{M}$$

where $N = 2n$, $M = 2(n+1)$ and

$$\beta_{j,l} = \begin{cases} 1 & j = l = 0 \\ 2 & j \neq 0,\ l \neq 0 \\ \sqrt{2} & \text{otherwise} \end{cases} \qquad g^0_{r,s} = \begin{cases} g_{r,s} & 0 \le r \le n \text{ and } 0 \le s \le n+1 \\ 0 & r > n \text{ or } s > n+1 \end{cases}$$

Then, it is possible to recover the coefficients $c_{j,l}$ by the Discrete Fourier Transform

$$\begin{aligned} \hat{g}_{j,s} &= \mathrm{REAL}\left(\sum_{r=0}^{N-1} g^0_{r,s}\mathrm{e}^{-2\pi \mathrm{i} jr/N}\right), \quad 0 \le j \le n,\ 0 \le s \le M-1 \\ \frac{c_{j,l}}{\beta_{j,l}} = \hat{\hat{g}}_{j,l} &= \mathrm{REAL}\left(\sum_{s=0}^{M-1} \hat{g}_{j,s}\mathrm{e}^{-2\pi \mathrm{i} ls/M}\right), \quad 0 \le j \le n,\ 0 \le l \le n-j \end{aligned} \tag{7}$$

According to [5], we call $\mathbb{C}_0(f)$ the interpolation coefficients matrix

$$\mathbb{C}_0(f) = (c'_{j,l}) = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & \cdots & c_{0,n} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} & 0 \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ c_{n-1,0} & c_{n-1,1} & 0 & \cdots & 0 \\ \dfrac{c_{n,0}}{2} & 0 & \cdots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \tag{8}$$

2.1 Evaluation of the interpolant

It is easy to see that the polynomial interpolation formula (3) can be evaluated at any $\boldsymbol{x} = (x_1, x_2) \in [-1,1]^2$ by

$$\mathcal{L}_n f(\boldsymbol{x}) = (\mathbb{T}(x_1))^t\,\mathbb{C}_0(f)\mathbb{T}(x_2)\ .$$

It is also possible to evaluate the polynomial interpolation formula on a set $\boldsymbol{X}$ of target points, at the same time. Given the vector X_1 of the first components of a set of target points and the vector X_2 of the corresponding second components, then

$$\mathcal{L}_n f(\boldsymbol{X}) = \operatorname{diag}\left((\mathbb{T}(X_1))^t \, \mathbb{C}_0(f) \mathbb{T}(X_2) \right) \ . \tag{9}$$

The result $\mathcal{L}_n f(\boldsymbol{X})$ is a (column) vector containing the evaluation of the interpolation polynomial at the corresponding target points.

If the target points are a Cartesian grid $\boldsymbol{X} = X_1 \times X_2$, then it is possible to evaluate the polynomial interpolation in a more compact form

$$\mathcal{L}_n f(\boldsymbol{X}) = \left((\mathbb{T}(X_1))^t \, \mathbb{C}_0(f) \mathbb{T}(X_2) \right)^t \ . \tag{10}$$

3 Fast cubature

In a recent paper [9], the interpolatory cubature formula corresponding to the Padua points has been studied. It has been termed "nontensorial Clenshaw–Curtis cubature" since it is a bivariate analogous of the classical Clenshaw–Curtis quadrature formula (cf. [6]). From the results of the previous section, we can write

$$\int_{[-1,1]^2} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx I_n(f) = \int_{[-1,1]^2} \mathcal{L}_n f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \sum_{k=0}^{n} \sum_{j=0}^{k} c'_{j,k-j} \, m_{j,k-j}$$

$$= \sum_{j=0}^{n} \sum_{l=0}^{n} c'_{j,l} \, m_{j,l} = \sum_{j \text{ even}}^{n} \sum_{l \text{ even}}^{n} c'_{j,l} \, m_{j,l} \tag{11}$$

where the *moments* $m_{j,l}$ are

$$m_{j,l} = \int_{-1}^{1} \hat{T}_j(t) \mathrm{d}t \int_{-1}^{1} \hat{T}_l(t) \mathrm{d}t$$

with

$$\int_{-1}^{1} \hat{T}_j(t) \mathrm{d}t = \begin{cases} 2 & j = 0 \\ 0 & j \text{ odd} \\ \dfrac{2\sqrt{2}}{1-j^2} & j \text{ even} \end{cases}$$

It is often desiderable to have a cubature formula that involves only the function values at the nodes and the corresponding cubature weights. A simple matrix formulation is still available. First, observe that

$$I_n(f) = \sum_{j \text{ even}}^{n} \sum_{l \text{ even}}^{n} c'_{j,l} \, m_{j,l} = \sum_{j \text{ even}}^{n} \sum_{l \text{ even}}^{n} c_{j,l} \, m'_{j,l}$$

 ISBN: 978-84-612-9727-6

with

$$\mathbb{M}_0 = (m'_{j,l}) = \begin{pmatrix} m_{0,0} & m_{0,2} & \cdots & \cdots & m_{0,p_n} \\ m_{2,0} & m_{2,2} & \cdots & m_{2,p_n-2} & 0 \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ m_{p_n-2,0} & m_{p_n-2,2} & 0 & \cdots & 0 \\ m'_{p_n,0} & 0 & \cdots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{([\frac{n}{2}]+1)\times([\frac{n}{2}]+1)}$$

where $p_n = n$ and $m'_{p_n,0} = m_{p_n,0}/2$ for n even, $p_n = n-1$ and $m'_{p_n,0} = m_{p_n,0}$ for n odd.

Now, using the formula for the coefficients (4) we can write

$$I_n(f) = \sum_{\boldsymbol{\xi}\in \mathrm{Pad}_n} \lambda_{\boldsymbol{\xi}}\, f(\boldsymbol{\xi})$$

$$= \sum_{\boldsymbol{\xi}\in C^{\mathrm{E}}_{n+1}\times C^{\mathrm{O}}_{n+2}} \lambda_{\boldsymbol{\xi}}\, f(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi}\in C^{\mathrm{O}}_{n+1}\times C^{\mathrm{E}}_{n+2}} \lambda_{\boldsymbol{\xi}}\, f(\boldsymbol{\xi})$$

where

$$\lambda_{\boldsymbol{\xi}} = w_{\boldsymbol{\xi}} \sum_{j\,\mathrm{even}}^{n} \sum_{l\,\mathrm{even}}^{n} m'_{j,l}\, \hat{T}_j(\xi_1)\hat{T}_l(\xi_2) \tag{12}$$

Defining the Chebyshev matrix corresponding to even degrees

$$\mathbb{T}^{\mathrm{E}}(S) = \begin{pmatrix} \hat{T}_0(s_1) & \cdots & \hat{T}_0(s_m) \\ \hat{T}_2(s_1) & \cdots & \hat{T}_2(s_m) \\ \vdots & \cdots & \vdots \\ \hat{T}_{p_n}(s_1) & \cdots & \hat{T}_{p_n}(s_m) \end{pmatrix} \in \mathbb{R}^{([\frac{n}{2}]+1)\times m}$$

and the matrices of interpolation weights on the subgrids of Padua points, $\mathbb{W}_1 = \left(w_{\boldsymbol{\xi}},\, \boldsymbol{\xi} \in C^{\mathrm{E}}_{n+1}\times C^{\mathrm{O}}_{n+2}\right)^t$, $\mathbb{W}_2 = \left(w_{\boldsymbol{\xi}},\, \boldsymbol{\xi} \in C^{\mathrm{O}}_{n+1}\times C^{\mathrm{E}}_{n+2}\right)^t$ it is then easy to show that the cubature weights $\{\lambda_{\boldsymbol{\xi}}\}$ can be computed in the matrix form

$$\mathbb{L}_1 = \left(\lambda_{\boldsymbol{\xi}},\, \boldsymbol{\xi} \in C^{\mathrm{E}}_{n+1}\times C^{\mathrm{O}}_{n+2}\right)^t = \mathbb{W}_1 . \left(\mathbb{T}^{\mathrm{E}}(C^{\mathrm{E}}_{n+1})\right)^t \mathbb{M}_0\, \mathbb{T}^{\mathrm{E}}(C^{\mathrm{O}}_{n+2})\Big)^t$$

$$\mathbb{L}_2 = \left(\lambda_{\boldsymbol{\xi}},\, \boldsymbol{\xi} \in C^{\mathrm{O}}_{n+1}\times C^{\mathrm{E}}_{n+2}\right)^t = \mathbb{W}_2 . \left(\mathbb{T}^{\mathrm{E}}(C^{\mathrm{O}}_{n+1})\right)^t \mathbb{M}_0\, \mathbb{T}^{\mathrm{E}}(C^{\mathrm{E}}_{n+2})\Big)^t$$

where the dot means that the final product is made componentwise.

An alternative approach is based on the observation that (12) itself is a Discrete Fourier Transform, the roles of the points and of the indexes beeing interchanged. An FFT-based implementation is then feasible, as in the univariate case (cf. [11]).

It is worth recalling that the cubature weights are not all positive, but the negative ones are few and of small size. Indeed, the cubature formula is stable and convergent for every continuous integrand, since, as it has been proved in [9],

$$\lim_{n\to\infty} \sum_{\boldsymbol{\xi}\in \mathrm{Pad}_n} |\lambda_{\boldsymbol{\xi}}| = 4\ .$$

 ISBN: 978-84-612-9727-6

4 Numerical tests

In this section we present some numerical tests on the accuracy and performance of the various implementations of interpolation and cubature at the Padua points. All the experiments have been made by the Matlab/Octave package Padua2DM [3], run in Matlab 7.6.0 on an Intel Core2 Duo 2.20GHz processor.

In Tables 1 and 2 we show the CPU times (seconds) for the computation of the interpolation coefficients and cubature weights at a sequence of degrees, by the MM and the FFT-based algorithms. The results suggest that the FFT approach is preferable for interpolation, whereas the MM approach is better for cubature. Indeed, the MM algorithm is more efficient than the FFT-based one in the cubature instance, since the matrices have lower dimension due to restriction to even indexes, and is competitive with the FFT up to very high degrees.

n	20	40	60	80	100	200	300	400	500
MM	0.003	0.001	0.003	0.004	0.006	0.022	0.065	0.142	0.206
FFT	0.002	0.002	0.002	0.002	0.006	0.029	0.055	0.088	0.137

Table 1: CPU time (in seconds) for the computation of the interpolation coefficients.

n	20	40	60	80	100	200	300	400	500
MM	0.004	0.000	0.001	0.002	0.003	0.010	0.025	0.043	0.071
FFT	0.005	0.001	0.003	0.003	0.005	0.025	0.048	0.090	0.142

Table 2: CPU time (in seconds) for the computation of the cubature weights.

In Figure 2 we report the relative errors of interpolation (top) and cubature (bottom) for the classical Franke test function in $[0,1]^2$ (versus the degree). Here a second advantage of the FFT approach for interpolation appears: it is able to arrive close to machine precision, whereas the MM algorithm stagnates around 10^{-13}. On the contrary, the MM algorithm seems more stable in the cubature than in the interpolation setting. These observations have been confirmed by many other numerical experiments.

In Figures 3 and 4 we show the interpolation and cubature errors versus the number of points (i.e., of function evaluations), for a Gaussian and a C^2 function. Interpolation and cubature at the Padua points are compared with tensorial formulas, and in the case of cubature also with the few known minimal formulas (cf. [8]).

We see two opposite situations. Concerning interpolation, the Padua points perform better than tensor-product Chebyshev–Lobatto points only on regular functions. On the other hand, nontensorial cubature at the Padua points performs always better than tensorial Clenshaw–Curtis cubature (which uses tensor-product Chebyshev–Lobatto points). However, it is less accurate than tensorial Gauss–Legendre–Lobatto and minimal formulas on analytic entire functions, whereas it appears the best one on less regular functions. This phenomenon, confirmed by many other examples (cf. [9])

 ISBN: 978-84-612-9727-6

and present also in 3d with nontensorial cubature at new sets of Chebyshev hyperinterpolation points (cf. [7]), is quite similar to that studied in the univariate case for the classical Clenshaw–Curtis formula (cf. [10]), but is still theoretically unexplained in the multivariate case. Nevertheless, numerical cubature at the Padua points seems to provide one of the best algebraic cubature formulas presently known for the square.

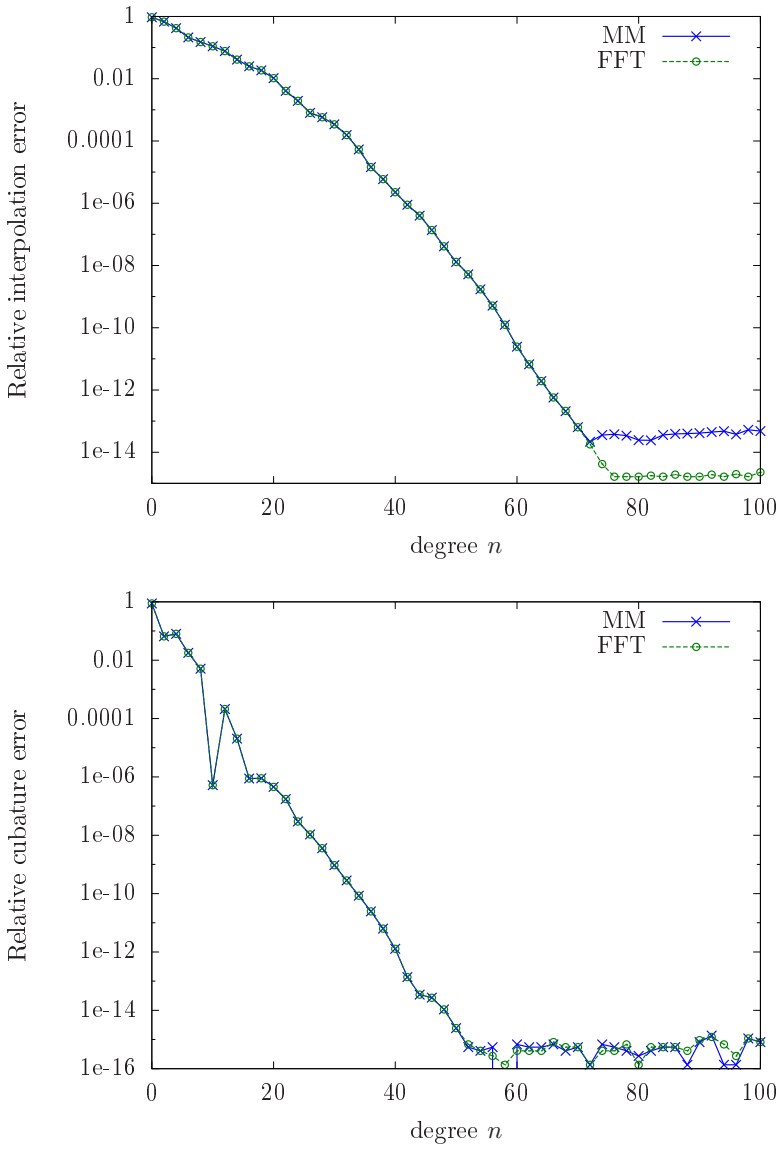


Figure 2: Errors of interpolation (top) and cubature (bottom) versus the interpolation degree for the Franke test function in $[0,1]^2$, by the MM and the FFT-based algorithms.

 ISBN: 978-84-612-9727-6

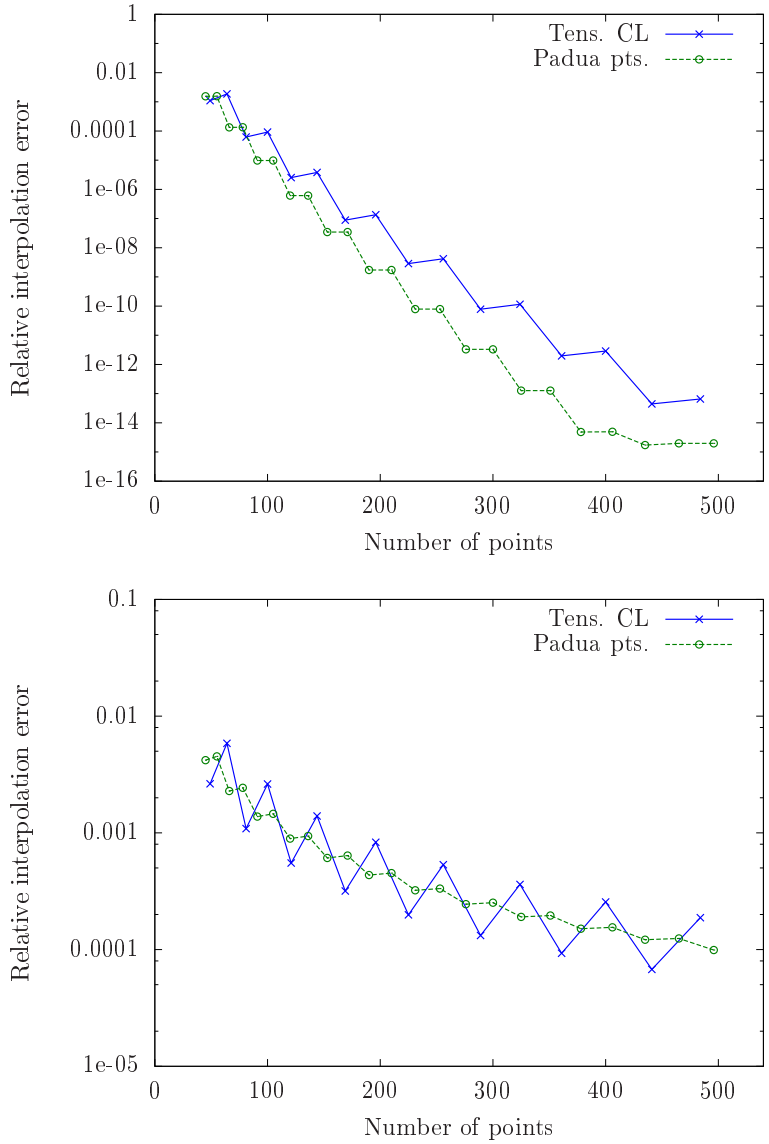


Figure 3: Relative interpolation errors versus the number of interpolation points for the Gaussian $f(\boldsymbol{x}) = \exp(-|\boldsymbol{x}|^2)$ (top) and the C^2 function $f(\boldsymbol{x}) = |\boldsymbol{x}|^3$ (bottom) in $[-1,1]^2$; Tens. CL = Tensorial Chebyshev–Lobatto interpolation.

References

[1] L. Bos, M. Caliari, S. De Marchi, M. Vianello, and Y. Xu: Bivariate Lagrange interpolation at the Padua points: the generating curve approach. J. Approx. Theory **143** (2006) 15–25.

[2] L. Bos, S. De Marchi, M. Vianello, and Y. Xu: Bivariate Lagrange interpolation

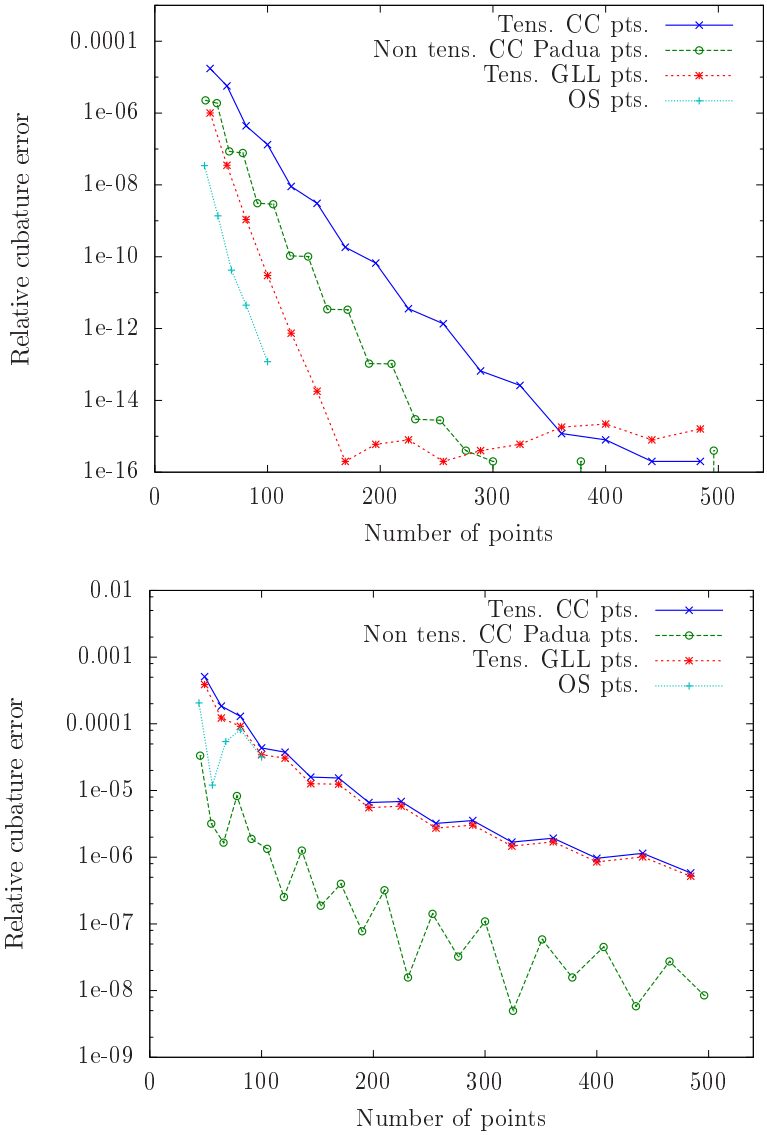


Figure 4: Relative cubature errors versus the number of cubature points (CC = Clenshaw–Curtis, GLL = Gauss–Legendre–Lobatto, OS = Omelyan–Solovyan) for the Gaussian $f(\boldsymbol{x}) = \exp(-|\boldsymbol{x}|^2)$ (top) and the C^2 function $f(\boldsymbol{x}) = |\boldsymbol{x}|^3$ (bottom) in $[-1,1]^2$.

at the Padua points: the ideal theory approach. Numer. Math. **108** (2007) 43–57.

[3] M. Caliari, S. De Marchi, A. Sommariva, and M. Vianello: Padua2DM (a Matlab/Octave code for interpolation and cubature at the Padua points), software available at `http://profs.sci.univr.it/~caliari/software` (2009).

[4] M. Caliari, S. De Marchi, and M. Vianello: Bivariate polynomial interpolation on the square at new nodal sets. Appl. Math. Comput. **165** (2005) 261–274.

[5] M. Caliari, S. De Marchi, and M. Vianello: Algorithm 886: Padua2D: Lagrange Interpolation at Padua Points on Bivariate Domains. ACM Trans. Math. Software **35-3** (2008).

[6] C.W. Clenshaw, A.R. Curtis: A method for numerical integration on an automatic computer. Numer. Math. **2** (1960) 197–205

[7] S. De Marchi, M. Vianello, and Y. Xu: New cubature formulae and hyperinterpolation in three variables. BIT Numerical Mathematics **49(1)** (2009) 55–73.

[8] I.P. Omelyan and V.B. Solovyan: Improved cubature formulae of high degrees of exactness for the square. J. Comput. Appl. Math. **188** (2006) 190–204.

[9] A. Sommariva, M. Vianello, and R. Zanovello, R.: Nontensorial Clenshaw–Curtis cubature. Numer. Algorithms **49** (2008) 409–427.

[10] L.N. Trefethen: Is Gauss quadrature better than Clenshaw–Curtis?. SIAM Rev. **50** (2008) 67–87.

[11] J. Waldvogel: Fast construction of the Fejér and Clenshaw–Curtis quadrature rules. BIT Numerical Mathematics **46** (2006) 195–202.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Improving Ab-initio Protein Structure Prediction by Parallel Multi-Objective Evolutionary Optimization

J.C. Calvo[1], J. Ortega[1] and M. Anguita[1]

[1] *Department of Computer Architecture and Computer Technology, CITIC-UGR*
University of Granada

emails: jccalvo@ugr.es, jortega@atc.ugr.es, manguita@atc.ugr.es

Abstract

Protein structure prediction (PSP) is an open problem with many useful applications in disciplines such as Medicine, Biology and Biochemistry. As this problem presents a vast search space and the analysis of each protein structure requires a significant amount of computing time, is necessary to take advantage of high performance parallel computing platforms as well as to define efficient search procedures in the space of possible protein conformations. In this paper we propose a parallel multi-objective evolutionary procedure that includes techniques to take advantage of the knowledge of the known protein structures through the so called rotamer library, and strategies to simplify the search space and adaptive mutation operator.

1 Introduction

Proteins have important biological functions such as the enzymatic activity of the cell, attacking diseases, transport and biological signal transduction, among others. They are chains of amino acids selected from a set of twenty elements. Whenever an amino acid chain is synthesized, it folds and uniquely determines its 3D structure. Moreover, although the amino acid sequence of a protein provides interesting information, the functionality of a protein is exclusively determined by its 3D structure [1]. Thus, there is a high interest in knowing the 3D structure of any given protein.

It is possible to reach the 3D structure of a protein experimentally by using methods such as X-ray crystallographic and nuclear magnetic resonance (NMR). Nevertheless, these processes are quite complex and costly as they would require months of expert work and laboratory resources. This situation comes clear if considering that less than a 25% of the protein structures included in the PDB (Protein Data Bank) have been solved.

An alternative approach is to use high performance computing. This computer approach is called protein structure prediction (PSP) and implies predicting the 3D structure of a protein given its sequence of amino-acids.Recently, efforts in protein structure prediction such as Rosetta@Home [2] and Predictor@Home [3] have been made using grid or global computing. These proposal tries to augment previous methods and algorithms by orders of magnitude more computing power to improve the prediction quality [2].

Computational approaches to PSP can be divided into two main alternatives: template modeling techniques such as homology and template-free or *ab initio* procedures [4, 5]. Template-modeling methods for protein structure and function prediction are based on the experimental conclusion that homologous proteins have similar folds and functions. This way, once a homology between the query (newly sequenced) protein and some known protein is determined, it is possible to derive some knowledge about the structure and function of the query protein. The homology of two proteins is mainly determined by the similarity of their respective amino acid sequences. In fact, there is a rule of thumb [6] indicating that two proteins of about 100 amino acids and 25% of identities in their sequence are related by evolution with a probability of about 50%. Nevertheless, as there are also proteins with low levels of sequence similarity that present similar structure and function, there are approaches that propose methods for comparing proteins which are not based on determining an alignment of their amino acids sequences but on other techniques such as profile methods, hidden Markov models and sequence signature libraries. Now, the most common way of managing the sequences of a protein family is to build a multiple alignment of the sequences and describe it by a generalized sequence with information about mutations allowed in each position of the family (i.e. a profile). Then, this profile can be used in a dynamic programming algorithm where the scoring is position dependent [6]. The *ab initio* alternative does not require any homology and can be applied when an amino acid sequence does not correspond to any other known one. This paper uses the *ab initio* approach that, nevertheless, could be able to take into account the knowledge extracted by template modeling techniques. In this case, to predict the protein 3D structure, we need to know the relation between primary and tertiary structures. Although this relation is not trivial, and there are many factors affecting the folding process that produces the final 3D conformation, we are looking for the native tertiary conformation with minimum free energy. Therefore, we do not take into account external factors that may influence the protein folding process, such as temperature, neighbor proteins, and other conditions into the cell. We are considering Protein Structure Prediction (PSP) rather than Protein Folding (PF).

Many algorithms have been proposed to solve the PSP by optimizing an objective or energy function, usually by using evolutionary algorithms [7, 8]. Nevertheless, over the last few years, some new approaches have been suggested that model the PSP problem as a multi-objective problem [1,9]. There are several reasons for trying a multi-objective approach. For example, as indicated in [9], there are works that demonstrate that some evolutionary algorithms improve their effectiveness when they are applied to multi-objective algorithms [10]. Indeed, in [1] it is argued that PSP problem can

be naturally modelled as a multi-objective problem because the protein conformations could involve tradeoffs among different objectives as it is experimentally shown by analyzing the conflict between bonded and non-bonded energies.

The procedure proposed in this paper is based on a multi-objective evolutionary algorithm. As evolutionary algorithms are population-based meta-heuristics they allow efficient and easy to implement workload distribution among the processors of the parallel/distributed platform at hand. Although some multiobjective optimization approaches to PSP have been proposed [1, 5], up to our knowledge, their parallelization has not been studied in depth. In [11], it is proposed a parallel hybrid evolutionary algorithm that includes a conjugated gradient-based hill climbing local search method. In our procedure, we include a method to manage torsion angles to reduce the complexity of the search space by using the backbone-rotamer library.

The paper has been structured as follows. In Section II we introduce the concepts related with multi-objective optimization and defines the different methods that our multi-objective optimization process needs in this problem. In the Section III the techniques proposed to improve the performance of our multiobjective protein structure predictor are shown. This multi-objective predictor is based in the I-PAES [12] algorithm, whose parallelization in this PSP context is presented in the next section. Finally, Section V provides the experimental results and Section VI the conclusions of the paper.

2 The proposed multi-objective approach

This section describes the components required to solve the PSP problem by a multi-objective optimization using the *ab initio* approach. These components are: the cost function, initialization, the mutation operators and the set of variables.

A multi-objective optimization problem [13] can be defined as the problem of finding a vector (1) that satisfies a given restriction set (2) and optimizes the function vector in (3). The objectives are usually in conflict between themselves, thus, optimizing one of them is carried out at the expense of the values of the others. This leads to the need of making a compromise, which implies the concept of Pareto optimality. In a multi-objective optimization problem, a decision vector x* is said to be a Pareto optimal solution if there is not any other feasible decision vector, x, that improves one objective without worsening at least one of the other objectives, given P the set of Pareto optimal solutions, (4). Usually, there are many vectors which are Pareto optimal. These solutions are called non-dominated. The set of all non-dominated solutions, in the decision space, determines the Pareto front in the objective space. Finally, a decision maker can select the solutions in its preferred front zone [9].

$$x = [x_1, x_2, ..., x_n] \tag{1}$$

$$g(x) \leq 0, h(x) = 0 \tag{2}$$

 ISBN: 978-84-612-9727-6

$$f(x) = \{f_1(x), f_2(x), , f_m(x)\} \tag{3}$$

$$\forall a, b \in P(\exists i, j \in \{1, 2, \cdots, n\} | (f_i(a) < f_i(b)) \wedge (f_j(a) > f_j(b))) \tag{4}$$

Cost function: Although a realistic measure of protein conformation quality should probably imply considering quantum mechanics principles, it would be too computationally complex to become useful. Thus, as it is usual, we have used the Chemistry at HARvard Macromolecular Mechanics (CHARMM) and AMBER99 energy functions [1, 14]. These are the most popular all-atom force field used for studying macromolecules and proteins respectively. We have considered its implementation at the TINKER library package [15]. As we use a multi-objective evolutionary optimization formulation of the PSP problem, the different terms of the energy function have to be transformed into several objectives. In [1] it is distinguished between bond and non-bond energies. We use this idea, although we have introduced some modifications according to the characteristics of the solution domain. Analyzing these energies we can observe that the Van Der Wall energy term has higher change range than others. Accordingly, this last energy terms can be hidden by the Van Der Wall energy term. Thus to optimize this energy appropriately, we propose a cost function with three objectives as follows: the bond energy and two more objective for the non-bond energies, one for Van Der Wall and other for the rest of non-bond terms. Also, we propose to take in account the difference between the probabilistic 3D protein structure (see next paragraph) and the current protein structure. To do that, we use another cost function configuration: bond energy, non-bond energy and difference with the probabilistic protein 3D structure.

Initialization: We have developed two initialization methods: random and probabilistic. The random method sets each variable with a random value according to the constraints of the variable. The probabilistic method uses the rotamer libraries to set each amino-acid at its most probable conformation.

Mutation operators: In [1] there are two different mutation operators. The first one is used to explore the search space, and the second one performs a local search optimization. We propose a third mutation operator, this method generate a mutation more conservative than the mutation to explore, and bigger than the mutation to perform a local search. This operator is executed with the same probability as the first one, and it mutates the side-chain with a gaussian function defined in the rotamer library. Doing that, we can avoid some local minima.

Set of variables: Any cost function depends on a set of variables that define the search space. These variables provide the required information to build the 3D conformation of the protein. In this paper we use torsion angles to represent the conformation of the protein, because this representation needs less variables than other alternatives. Three torsion angles are required in the backbone per each amino acid and some additional torsion angles depending on the side-chain defined by the different residues of the amino-acids in the 1D structure [1] (Figure 1).

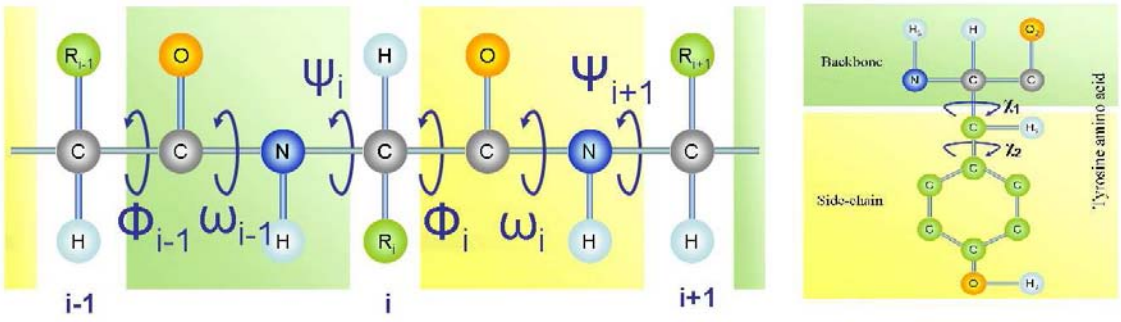


Figure 1: (left) Backbone angles ϕ, ψ and ω. (right) Side-chain angles χ_i in the tyrosine amino acid.

In many optimization problems, once a suitable cost function is found, it would not be difficult to obtain an efficient evolutionary algorithm that generates sufficient good solutions for the problem. This is not the case in the PSP problem. The high dimensionality of the space of conformations makes difficult for an ab initio procedure to find adequate structures for complex proteins. Thus, any population-based procedure needs a set of adequate individuals to to cut the search space and configure the initial population. These individuals should represent promising points in the search space by comprising the most part of the available knowledge on protein structures. Some alternatives should be taken into account to reduce the search space such as secondary and super-secondary structure prediction of rotamers libraries [16], that have statistical information about the conformation of the amino acids in known molecules.

Template-based modeling is the most reliable and accurate approach to the protein structure prediction problem. The procedure proposed in this paper could take advantage of techniques that are common in the template-based approaches to drive the searching of a multi-objective evolutionary algorithm towards the most promising zones of the space.

Thus, although the PSP problem implies to predict the tertiary structure of a given protein from its primary structure, it could be a good idea to use predictions of the secondary and super-secondary structures as they give us information about the amino acids involved in one of these structures, determining some constraints in the torsion angles of each amino acid (as shown in Table 1). In order to get the super-secondary structure given its secondary structure, we have to analyze the conformation of the residues in the short connecting peptide between two secondary structures. They are classified into five types, namely, a, b, e, l or t [17]. Sun *et al.* [17] developed a method to predict the eleven most frequently occurring super-secondary structures: H-b-H, H-t-H, H-bb-H, H-ll-E, E-aa-E, H-lbb-H, H-lba-E, E-aal-E, E-aaal-E and H-l-E where H and E are α helix and β strand, respectively. In this way a reduction in the search space of the PSP problem is obtained.

Moreover, side-chain torsion angles have interesting dependencies. Dumbrack et

 ISBN: 978-84-612-9727-6

Table 1: Search space of each angle ϕ and ψ depending on the position of the super-secondary structure they are.

Super-secondary structure	ϕ	ψ
H (α helix)	[-75, -55]	[-50, -30]
E (β strand)	[-130, -110]	[110, 130]
a	[-150, -30]	[-100, 50]
b	[-230, -30]	[100, 200]
e	[30, 130]	[130, 260]
l	[30, 150]	[-60, 90]
t	[-160, -50]	[50, 100]
undefined	[-180, 0]	[-180, 180]

al. [16] give many rotamers libraries that help us to identify constraints about these torsion angles. An example of these libraries is the backbone-independent rotamer library. Given an amino acid, this library includes constraints for its side-chain torsion angles.

In this work we have applied a new method to manage torsion angles using the backbone-dependent rotamer library that, contrary to backbone-independent rotamer library, includes the dependency between side-chain and backbone torsion angles. In this way, we reduce the set of variables involved in the optimization process by eliminating the side-chain torsion angles. Nevertheless we can not eliminate side-chain torsion angles without adding another mechanism to take them into account. This mechanism selects the most probable conformation of the side-chain in each amino acid depending on the backbone torsion angles.

3 Strategies to improve the evolutionary prediction

In this section we show two new methods to improve the predictive capabilities in an evolutionary algorithm solving the PSP problem. The first method create a simplified search space to be used by the EA in the first steps and the second method define an adaptive probability in each amino-acid to be selected for the mutation operator of the EA.

Simplified search space: In the first part of the EA (for example: first 10% of fitness function evaluations), the search space is a simplification of the real one. The real search space, as we have seen before, consist in two variables per amino-acid, each variable is in real codification and has a range of movement. The simplified search space consist in only one variable per amino-acid, and this variable has only 4 possible values. This way, the EA can observe the diversity of the search space. After this period, the search space becomes the real one.

By using this method, as it is shown in Figure 2, the Evolutionary Algorithm can get a first representation on the protein, not refined, but the structure may be in a

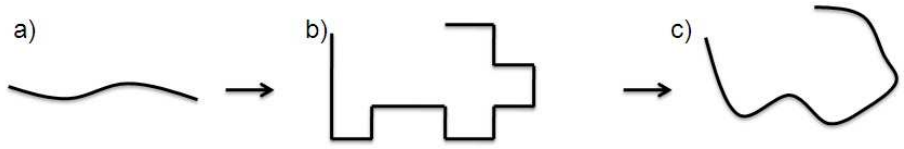


Figure 2: (a) Amino-acid sequence (b) Structure after the simplified search space period. (c) Structure at the end of the algorithm.

correct shape. Then in the rest of the time, the EA can refine the structure to get a good prediction.

Amino-acid mutation probability: In a traditional EA there is an uniform probability in each variable to be mutated. We present a new way to manage mutation probabilities in the PSP problem. In the PSP problem there are bond energies and non-bond energies. The bond energies are independent of the area where the amino-acid is, and of the actual shape of the structure. These energies are only affected by the next and previous amino-acid. But, non-bond energies depends on the actual shape of the structure.

Analyzing these facts, we can observe in Figure 3 that a mutation in the extremes of the protein (b in Figure 3) has less impact than mutations in the middle of the protein (a in Figure 3), because the first one does not involve non-bond energies, but a mutation in the middle of the protein could involve a lot of interactions between atoms in different areas of the protein. Respect to the actual shape of the structure, we can see that an high congested area of the shape (c *right* in Figure 3) is going to have a lot of interactions, this way it is interesting to use more computation in that area than in a zone with less amino-acids (c *left* in Figure 3).

The proposal method to manage these probabilities is an adaptive method. Along the time, the algorithm save the accurate of each mutation depending on the amino-acid. With this information it can put more probability in the amino-acids with better evolutions. It is like a temperature in each amino-acid: a hot amino-acid is more probable to be selected for a mutation, if the mutation gets a better structure in terms of energy, the amino-acid becomes hotter, if the mutation gets a worse structure, the amino-acid lose temperature. In any case, in each step all the amino-acids lose temperature. With this approach the probability can be adapted to the shape, because of the continuously temperature losing.

4 Parallel I-PAES for PSP

The multi-objective evolutionary algorithm we have implemented is based on PAES [12]. In each iteration, the PAES algorithm generates a single child and decides whether to select the child or keep the parent as the current solution. Aiming at a parallel implementation of PAES that preserves the exact algorithm behaviour, a naive parallelization scheme can be described as follows. Given a current solution, PAES will frequently

 ISBN: 978-84-612-9727-6

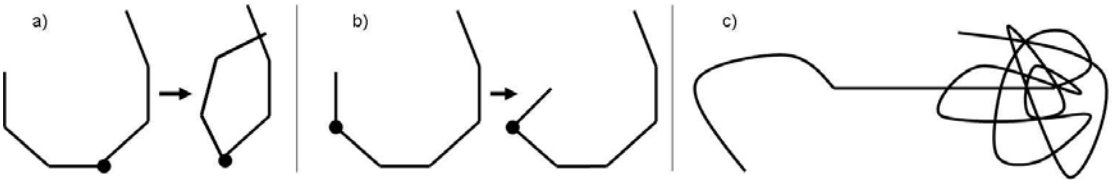


Figure 3: (a) Mutating an amino-acid in the middle of the sequence. (b) Mutating an amino-acid in one extreme of the sequence. (c) A protein shape with an high congested area (right) and a low congested area (left).

need to generate a number of offspring solutions before an acceptable offspring is found, which replaces the current solution (i.e. before we have a change in generation). Hence, if we have a number of n processors available, we can generate n offsprings and use these processors to simultaneously (in a single time step) generate and evaluate an ordered set of n prospective offsprings for the current solution. The master node then scans the fitness values of all n offsprings in order and accepts the first one of these that fullfills PAES acceptance criterion. In this way, the original PAES strategy is maintained. It is evident that the efficiency of this parallelization scheme may vary strongly dependent on the number of children generated (i.e. the number of processors available) and the difficulty of the optimization task. When the search is very easy (e.g. at the beginning of the optimization process) or when a large number of processors are available, the parallel strategy is likely to "waste" a large number of evaluations. Instead, when it is difficult to find a better solution, the strategy works with high efficiency. In Figure 4 a) we show the difference between a sequental algorithm and a parallel algorithm with the same behaviour

In this paper we have developed a more elaborated parallelization scheme, which attempts to minimize the number of "wasted" evaluations by limiting the number of offsprings that are generated simultaneously for a given current solution. The discrepancy between the number of offspring generated and the number of processors available can then be used to generate and evaluate the next offspring generations, in an effort to maximize the number of total iterations covered in a single time step.

In each iteration, the algorithm has to take a decision between the parent node and the new node, hence there is two nodes implied in each decision. Representing a parallel time step, we have many decisions at the same time, and could be that one parent node take part in few decisions. We have to create a new view of the PAES tree to represent each decision on the evolution process separately. In this way, we can allocate the resources in the tree. In Figure 4 c) we show the new prediction tree versus the normal one. In the prediction tree, we move copy the parent node to the left child, the right child is the real child of the parent. Therefore, initially the comparison was between the parent and the child, but now the comparison involves the two children. Anyway, the comparison is the same in both representations.

Assuming a fixed probability $p = 0.5$ of generating a favorable mutation, we can

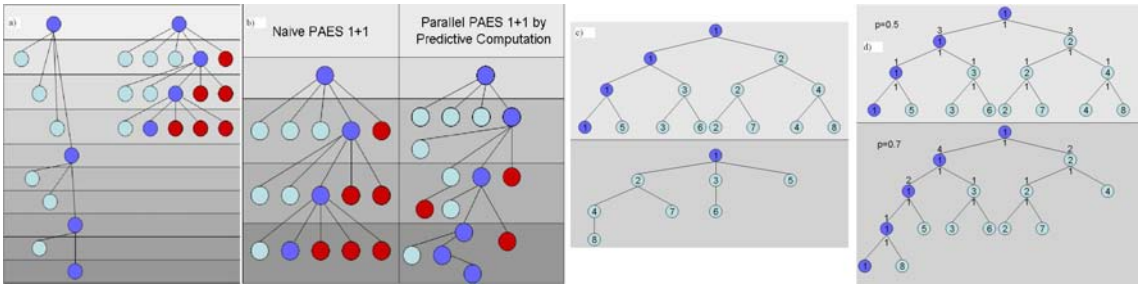


Figure 4: (a) Sequential vs. Parallel PAES. (b) Parallel PAES vs. Parallel PAES by Adaptive Predictive Computation. (c) Different representations of one parallel step: (up) We have to select one final node, (down) we can select any node. (d) Different parallel steps depending on the adaptive parameter p

optimally distribute the processors available based on a static evaluation tree, as illustrated at top in Figure 4 d). However, in a realistic optimization scenario, the probability p will be different to 0.5 and is likely to vary over time, resulting in different shapes of the optimal evaluation tree (bottom in Figure 4 d)).

As we show in Figure 4 b), the prediction tree approach can perform better than the naive scheme. It is going to depend on the quality of the prediction factor.

Analyzing the behaviour of the tree parallelization scheme and the naive one, we can see that the naive is going to work fine if the behaviour of the problem is fixed in keeping in the parent node. In that case, we can use a prediction factor $p = 0$, in this case, the tree parallelization scheme is going to work equal than the naive one. But in other cases, the tree parallelization scheme can take advantage over the naive approach.

5 Results

In this section we provide and analyze the results obtained with our procedure. First of all we use the 1CRN protein [1, 18] as a benchmark. As the cost functions used to model the PSP problem only approximate the conformation energy of the protein, it is not enough to measure the quality of the 3D structure obtained. Thus, we should accomplish a comparison between a given known protein structure and the solution obtained by optimizing the cost function that model the corresponding PSP problem. The most famous measure of similarity between predicted and known native structures is the RMSD [1] (5). RMSD computes the 3D difference between two structures, therefore the lower the molecule RMSD, the better is the 3D structure obtained.

$$RMSD(a,b) = \sqrt{\frac{\sum_{i=1}^{n} |\tau_{ai} - \tau_{bi}|^2}{n}} \tag{5}$$

We have executed an I-PAES with our heuristics and methods for more than 20 times along 250.000 cost function evaluations and, in each execution, we have selected

 ISBN: 978-84-612-9727-6

the solution with better RMSD in the Pareto front. After computing the mean of these RMSDs we have obtained the results of Table 2. As it is shown in this table, the best configuration for our algorithm corresponds to the use of the Van Der Wall term as the third objective along with a probabilistic initialization.

Then, once we studied the different alternatives, we selected the best configuration for the I-PAES algorithm and we added the previously described strategies to improve the search. This way, we were able to improve the mean RSMD up to 7.13 +-0.64 $\mathring{A}$ (by using the amino acid adaptive probabilistic mutation and 20% of the time in the simplified search space).

With the best configuration of the algorithm for 1CRN protein, we tested the algorithm with other proteins, showing the minima RMSD, the mean RMSD and the deviation (Table 3) computing the $RMSD_{C\alpha}$ in the core region.

Table 2: Provided algorithm for 1CRN protein.

# Objectives	Initialization	3rd objective	mean RMSD	deviation
2	Random	-	11.40 $\mathring{A}$	1.78 $\mathring{A}$
2	Probabilistic	-	10.35 $\mathring{A}$	1.50 $\mathring{A}$
3	Random	Probabilistic	9.72 $\mathring{A}$	1.23 $\mathring{A}$
3	Probabilistic	Probabilistic	7.86 $\mathring{A}$	1.03 $\mathring{A}$
3	Random	Van Der Wall	8.49 $\mathring{A}$	1.19 $\mathring{A}$
3	Probabilistic	Van Der Wall	*7.78* $\mathring{A}$	0.94 $\mathring{A}$

Table 3: Best configuration of the proposed algorithm (3 objectives, probabilistic initialization and Van Der Wall term as 3rd objective) and best configuration of technics (amino-acid mutation probability and 20% of the time in the simplified search space) for a set of proteins.

Protein	# amino-acids	core region	min. RMSD	mean RMSD	deviation
1CRN	46	[7-36]	3.48 $\mathring{A}$	5.12 $\mathring{A}$	0.33 $\mathring{A}$
1PLW	5	[1-5]	0.79 $\mathring{A}$	0.79 $\mathring{A}$	0.00 $\mathring{A}$
1UTG	70	[4-64]	4.36 $\mathring{A}$	5.15 $\mathring{A}$	0.51 $\mathring{A}$
1ROP	63	[3-60]	3.84 $\mathring{A}$	4.18 $\mathring{A}$	0.26 $\mathring{A}$
1ZDD	34	[3-32]	4.00 $\mathring{A}$	4.57 $\mathring{A}$	0.41 $\mathring{A}$

The parallel version of this algorithm needs more analysis to get a real behaviour, but the first tests allow us to see a good performance. The sequential algorithm takes around 30 hours, and the parallel version needs only 2 hours with 28 processors. Thus, after our first tests we have obtained an acceptable efficiency according to the sequential behaviour of the algorithm, which has a lot of dependencies between iterations.

 ISBN: 978-84-612-9727-6

6 Conclusion

The PSP problem joins biological and computational concepts. It requires accurate and tractable models of the conformations energy. Thus, there is a long way to go to find useful solutions to the problem for proteins of realistic sizes. Our contribution in this paper deals with a new procedure for PSP based on a parallel multi-objective evolutionary algorithm. It allows a reduction in the number of variables and few heuristics to improve the quality of the solutions such as the three objectives cost function. Also we propose two new strategies to improve the prediction quality in the PSP problem. One of them improves the computation efforts by using an adaptive probability of mutation in each amino-acid, whereas the other uses a simplified search space in the first part of the evolution. The multi-objective evolutionary procedure implemented is based on PAES and it has been parallelized through an adaptive and predictive approach thus providing speedups between 13 and 15 with up to 28 processors.

Acknowledgment

This paper has been supported by the Spanish Ministerio de Educacion y Ciencia under project TIN2007-60587.

References

[1] V. Cutello, G. Narcisi, and G. Nicosia, "A multi-objetive evolutionary approach to the protein structure prediction problem." *J. R. Soc. Interface*, vol. 3, pp. 139–151, 2006.

[2] P. Bradley, K. Misura, and D. Baker, "Toward high-resolution de novo structure prediction for small proteins," *Science*, vol. 309, pp. 1868–1871, 2005.

[3] M. Taufer, C. An, A. Kerstens, and C. Brooks, "Predictor@home: A 'protein structure prediction supercomputer' based on global computing." *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, 2006.

[4] C. Branden and J. Tooze, "Introduction to protein structure." iSBN 0-81-532305-0.

[5] J. Handl, D. Kell, and J. Knowles, "Multiobjective optimization in bioinformatics and computational biology." *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 4, no. 2, pp. 279–292, April 2007.

[6] L. Rychlewski, L. Jaroszewski, W. Li, and A. Godzik, "Comparison of sequence profiles. strategies for structural predictions using sequence information," *Protein Science*, vol. 9, pp. 232–241, 2000.

[7] N. Krasnogor, W. Hart, J. Smith, and D. Pelta, "Protein structure prediction with evolutionary algorithm." in *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.

[8] C. Cotta, "Hybrid evolutionary algorithms for protein structure prediction under the hpnx model," *Advances in Soft Computing*, vol. 2, pp. 525–534, 2005.

[9] R. Day, J. Zydallis, and G. Lamont, "Solving the protein structure prediction problem through a multiobjective genetic algorithm." *Nanotech*, vol. 2, pp. 32–35, 2002.

[10] J. Zydalis, A. V. Veldhuizen, and G. Lamont, "A statistica comparison of moeas including the momga-ii," in *Proc. 1st Int. Conference on Evolutionary Multicriterion Optimization*, 2001, pp. 226–240.

[11] A.-A. Tantar, N. Melab, E.-G. Talbi, B. Parent, and D. Horvath, "A parallel hybrid genetic algorithm for protein structure prediction on the computational grid," *Future Generation Computer Systems*, vol. 23, pp. 398–409, 2007.

[12] J. Knowles and D. Corne, "The pareto archived evolution strategy : A new baseline algorithm for pareto multiobjective optimisation," *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 98–105, 1999.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii." *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[14] B. Wathen, "Hydrophobic residue patterning in β-strands and implication for β-sheet nucleation." [Online]. Available: http://qcse.queensu.ca/conferences/documents/BrentWathen.ppt

[15] TINKER, "Software tools for molecular design." [Online]. Available: http://dasher.wustl.edu/tinker/

[16] R. Dunbrack and F. Cohen, "Bayesian statistical analysis of protein sidechain rotamer preferences," *Protein Sci*, vol. 6, pp. 1661–1681, 1997.

[17] Z. Sun and B. Jiang, "Patterns and conformations of commonly occurring supersecondary structures (basic motifs) in protein data bank." *Journal of Protein Chemistry*, vol. 15, no. 7, 1996.

[18] RCSB, "Protein data bank (pdb)." [Online]. Available: http://www.pdb.org

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Numerical Approximation of Elliptic Control Problems with Finitely Many Pointwise Constraints

Eduardo Casas[1] **and Mariano Mateos**[2]

[1] *Departamento de Matemática Aplicada y Ciencias de la Computación, Universidad de Cantabria*

[2] *Departamento de Matemáticas, Universidad de Oviedo*

emails: eduardo.casas@unican.es, mmateos@uniovi.es

Abstract

We study a control problem governed by a semilinear elliptic partial differential equation. Bound constraints are imposed on the control, as well as finitely many pointwise constraints are imposed on the state. Both equality and inequality constraints are considered. Theoretical results published in [3] are quoted and a complete numerical analysis of the problem is developed. Detailed proofs will be provided in a forthcoming paper.

Key words: optimal control, numerical approximation, error estimates
MSC 2000: 65N30, 65N15, 49M05, 49M25

1 Introduction

Let $\Omega \subset \mathbb{R}^n$, $n = 2$ or $n = 3$ be a $C^{1,1}$ open bounded convex domain and Γ be its boundary. Consider $f : \Omega \times \mathbb{R} \to \mathbb{R}$ and $L : \Omega \times \mathbb{R}^2 \to \mathbb{R}$ Carathèodory functions. Let n_e and n_i be nonnegative integer numbers. Let us take $n_e + n_i$ different points in Ω, $\{x_j\}_{j=1}^{n_e+n_i} \subset \Omega$ and real numbers σ_j for $1 \le j \le n_e + n_i$. Finally we will also consider two Lipschitz functions α, β such that $-\infty < \alpha(x) < \beta(x) < +\infty$ for every $x \in \bar{\Omega}$. We are concerned with the control problem:

$$(P)\begin{cases} \min J(u) = \displaystyle\int_\Omega L(x, y_u(x), u(x))dx \\ \text{subject to } (y_u, u) \in (C(\bar{\Omega}) \cap H_0^1(\Omega)) \times L^\infty(\Omega), \\ \alpha(x) \le u(x) \le \beta(x) \text{ for a.e. } x \in \Omega, \\ y_u(x_j) = \sigma_j \text{ for } 1 \le j \le n_e, \\ y_u(x_j) \le \sigma_j \text{ for } n_e + 1 \le j \le n_e + n_i, \end{cases}$$

 ISBN: 978-84-612-9727-6

where $y_u \in H_0^1(\Omega)$ is the solution of the state equation

$$Ay_u + f(\cdot, y_u) = u \text{ in } \Omega,\ y = 0 \text{ on } \Gamma, \tag{1}$$

and A denotes the elliptic operator given by

$$Ay = -\sum_{i,j=1}^{n} \partial_{x_j}(a_{i,j}\partial_{x_i} y),$$

where $a_{i,j} \in C^{0,1}(\bar{\Omega})$ and there exists $\lambda_A > 0$ such that

$$\lambda_A \|\xi\|^2 \leq \sum_{i,j=1}^{n} a_{i,j}(x)\xi_i\xi_j \ \forall \xi \in \mathbb{R}^n \text{ and } \forall x \in \bar{\Omega}.$$

1.1 Main assumptions

Let $p > n$ be fixed. Through all the paper we will suppose that:

(A1) f is of class C^2 w.r.t. the second variable. For a.e. $x \in \Omega$

$$f(\cdot, 0) \in L^p(\Omega),\ \partial_y f(x, y) \geq 0$$

For every $M > 0$ there exists $C_{f,M} > 0$ such that

$$|\partial_y f(x, y)| + |\partial^2_{yy} f(x, y)| \leq C_{f,M},\ |\partial^2_{yy} f(x, y_1) - \partial^2_{yy} f(x, y_2)| \leq C_{f,M}|y_1 - y_2|$$

for a.e. $x \in \Omega$ and $|y|$, $|y_1|$, $|y_2| \leq M$.

(A2) L is of class C^2 w.r.t. the second and third variables, $L(\cdot, 0, 0) \in L^1(\Omega)$. For every $M > 0$ there exists $C_{L,M} > 0$ and $\psi_M \in L^p(\Omega)$ such that

$$|\partial_y L(x, y, u)| \leq \psi_M(x),\ \|D^2_{(y,u)} L(x, y, u)\| \leq C_{L,M},$$

$$|\partial_u L(x_1, y, u) - \partial_u L(x_2, y, u)| \leq C_{L,M}|x_1 - x_2|$$

$$\|D^2_{(y,u)} L(x, y_1, u_1) - D^2_{(y,u)} L(x, y_2, u_2)\| \leq C_{L,M}(|y_1 - y_2| + |u_1 - u_2|),$$

for a.e. $x \in \Omega$ and all $|y|, |y_1|, |y_2|, |u|, |u_1|, |u_2| \leq M$ and $D^2_{(y,u)} L(x, y_i, u_i)$ is the Hessian matrix

$$D^2_{(y,u)} L(x, y_i, u_i) = \begin{pmatrix} \partial^2_{yy} L(x, y_i, u_i) & \partial^2_{yu} L(x, y_i, u_i) \\ \partial^2_{uy} L(x, y_i, u_i) & \partial^2_{uu} L(x, y_i, u_i) \end{pmatrix}$$

We will also suppose that there exists $\lambda_L > 0$ such that

$$\partial^2_{uu} L(x, y, u) \geq \lambda_L \tag{2}$$

for a.e. $x \in \Omega$ and $(y, u) \in \mathbb{R}^2$.

 ISBN: 978-84-612-9727-6

2 The continuous problem

Through this section we will state some facts about problem (P). For the details see [3] and the references therein.

Definition 2.1 *We will say that $\bar{u}$ is a local solution of problem (P) if there exists $\rho > 0$ such that $J(\bar{u}) < J(u)$ for every $u \in U_{ad}$ such that $\|\bar{u} - u\|_{L^\infty(\Omega)} < \rho$.*

For $2 \le r \le p$, let $G : L^r(\Omega) \to W^{2,r}(\Omega)$ be the solution operator, defined by $G(u) = y_u$ and for $1 \le j \le n_e + n_i$ let $G_j : L^r(\Omega) \to \mathbb{R}$ be the constraint operators defined by $G_j(u) = y_u(x_j) - \sigma_j$. These mappings, as well as the functional $J : L^\infty(\Omega) \to \mathbb{R}$ are of class C^2. For all $u, v \in L^\infty(\Omega)$

$$J'(u)v = \int_\Omega (\partial_u L(x, y_u, u) + \varphi_{0u})v dx$$

and for every $u, v \in L^r(\Omega)$,

$$G_j'(u)v = \int_\Omega \varphi_{ju} v dx$$

where $\varphi_{0u} \in W^{2,p}(\Omega)$ is the solution of

$$A^*\varphi_{0u} + \partial_y f(x, y_u)\varphi_{0u} = \partial_y L(x, y_u, u) \text{ in } \Omega,\ \varphi_{0u} = 0 \text{ on } \Gamma. \tag{3}$$

and $\varphi_{ju} \in W^{1,s}(\Omega) \cap W^{2,p}_{loc}(\Omega \setminus \{x_j\})$ for $1 \le s < n/(n-1)$ is the unique solution of

$$A^*\varphi_{ju} + \partial_y f(x, y_u)\varphi_{ju} = \delta_{x_j} \text{ in } \Omega,\ \varphi_{ju} = 0 \text{ on } \Gamma. \tag{4}$$

Regular controls

For a fixed control $\bar{u}$ and any $\varepsilon > 0$, define

$$\Omega_\varepsilon = \{x \in \Omega :\ \varepsilon < \bar{u}(x) < 1 - \varepsilon\}.$$

Definition 2.2 *Let $\bar{u} \in L^\infty(\Omega)$ be a feasible control for problem (P). Let*

$$I_0 = \{j \le n_e + n_i :\ y_j(x_j) = \sigma_j\}.$$

We will say that the $\bar{u}$ is regular if there exists $\bar{\varepsilon} > 0$ and functions $\{\bar{w}_j\}_{j \in I_0} \subset L^\infty(\Omega)$ such that

1. $\operatorname{supp} \bar{w}_j \subset \Omega_{\bar{\varepsilon}}$.
2. $G_i'(\bar{u})\bar{w}_j = \delta_{i,j}$ *for* $i, j \in I_0$.

Remark 2.3 *Regular controls are quite common. If $\bar{u}$ is a feasible control for problem (P) and there exists $\varepsilon > 0$ such that Ω_ε has a nonempty interior then $\bar{u}$ is regular. Moreover, the functions $\bar{w}_j$ can be chosen to be in $C^\infty(\bar{\Omega})$. and the regularity condition is equivalent to Slater condition.*

 ISBN: 978-84-612-9727-6

Define the Lagrangian function $\mathcal{L} : L^\infty(\Omega) \times \mathbb{R}^{n_e+n_i} \to \mathbb{R}$ as

$$\mathcal{L}(u, \lambda) = J(u) + \sum_{j=1}^{n_e+n_i} \lambda_j G_j(u).$$

Lemma 2.4 *Suppose $\bar{u}$ is a local solution of (P) that is regular in the sense of Definition 2.2. Then there exists real numbers $\{\bar{\lambda}_j\}_{j=1}^{n_e+n_i}$ such that*

$$\bar{\lambda}_j \geq 0 \text{ and } \bar{\lambda}_j G_j(\bar{u}) = 0 \text{ if } n_e + 1 \leq j \leq n_e + n_i, \tag{5}$$

$$\partial_u \mathcal{L}(\bar{u}, \bar{\lambda})(u - \bar{u}) \geq 0 \ \forall u \in \{u \in L^\infty(\Omega) : \alpha(x) \leq u(x) \leq \beta(x) \text{ for a.e. } x \in \Omega\}. \tag{6}$$

For a regular local solution $\bar{u}$, let us also denote $\bar{\varphi}_j = \varphi_{j\bar{u}}$, where $\varphi_{j\bar{u}}$ is defined in (4). We will define the adjoint state of $\bar{u}$ as

$$\bar{\varphi} = \bar{\varphi}_0 + \sum_{j \in I_0} \bar{\lambda}_j \bar{\varphi}_j.$$

The derivative of the Lagrangian function can be expressed in terms of

$$\bar{d}(x) = \partial_u L(x, \bar{y}(x), \bar{u}(x)) + \bar{\varphi}(x).$$

For $\tau \geq 0$ define

$$\Omega^\tau = \{x \in \Omega : \ |\bar{d}(x)| > \tau\}$$

and

$$C^\tau = \{v \in L^2(\Omega) : \text{ satisfying (7)–(10) and } v(x) = 0 \text{ for a.e. } x \in \Omega^\tau\}$$

with

$$G_j'(\bar{u})v = 0 \text{ if } (j \leq n_e) \text{ or } (j > n_e \text{ and } \bar{\lambda}_j > 0) \tag{7}$$

$$G_j'(\bar{u})v \leq 0 \text{ if } (j > n_e, \ G_j(\bar{u}) = 0 \text{ and } \bar{\lambda}_j = 0) \tag{8}$$

$$v(x) \geq 0 \text{ if } \bar{u}(x) = \alpha(x), \tag{9}$$

$$v(x) \leq 0 \text{ if } \bar{u}(x) = \beta(x). \tag{10}$$

Lemma 2.5 *Let $\bar{u}$ be an admissible control for problem (P) that is regular in the sense of Definition 2.2 and satisfies the first order optimality conditions (5) and (6). Suppose further that*

$$\partial_{uu}^2 \mathcal{L}(\bar{u}, \bar{\lambda})v^2 > 0 \ \forall v \in C^0 - \{0\}. \tag{11}$$

Then $\bar{u}$ is a local quadratic solution of (P) in $L^\infty(\Omega)$: there exist $\varepsilon > 0$ and $\delta > 0$ such that

$$J(\bar{u}) + \frac{\delta}{2}\|u - \bar{u}\|_{L^2(\Omega)}^2 \leq J(u)$$

for every admissible control u with $\|u - \bar{u}\|_{L^\infty(\Omega)} \leq \varepsilon$.

 ISBN: 978-84-612-9727-6

3 Numerical Analysis

Here, we define a finite-element based approximation of the optimal control problem (P). To this aim, we consider a family of triangulations $\{\mathcal{T}_h\}_{h>0}$ of $\bar{\Omega}$. Define $\Omega_h = \text{int} \cup_{T\in\mathcal{T}_h} T$. This triangulation is supposed to be regular in the usual sense that we state exactly here. With each element $T \in \mathcal{T}_h$, we associate two parameters $\rho(T)$ and $\sigma(T)$, where $\rho(T)$ denotes the diameter of the set T and $\sigma(T)$ is the diameter of the largest ball contained in T. Let us define the size of the mesh by $h = \max_{T\in\mathcal{T}_h} \rho(T)$. The following regularity assumption is assumed.

(H) - There exist two positive constants ρ and σ such that

$$\frac{\rho(T)}{\sigma(T)} \leq \sigma, \qquad \frac{h}{\rho(T)} \leq \rho$$

hold for all $T \in \mathcal{T}_h$ and all $h > 0$.

The following condition is needed to use a discrete maximum principle.

(A3) The sum of the two angles opposite to any interior edge is $\leq \pi$ if $n = 2$, or the triangulation is nonobtuse for $n \geq 2$

Notice that for h small enough, $x_j \in \Omega_h$ for $1 \leq j \leq n_e + n_i$. Associated with this triangulation we set

$$Y_{h,0} = \{y_h \in C(\bar{\Omega}) \mid y_{h|T} \in \mathcal{P}_1 \text{ for all } T \in \mathcal{T}_h,\ y_h = 0 \text{ in } \Omega \setminus \Omega_h\}$$

and

$$U_h = \{u_h \in L^\infty(\Omega) \mid u_{h|T} \in \mathcal{P}_0 \text{ for all } T \in \mathcal{T}_h,\ u_h = 0 \text{ in } \Omega \setminus \Omega_h\},$$

where $\mathcal{P}_i$ is the space of polynomials of degree less than or equal to i, $i = 0, 1$.

Define the bilinear form

$$a(y, \chi) = \int_{\Omega_h} \left(\sum_{i,j=1}^{n} a_{i,j}(x)\partial_{x_i} y(x)\partial_{x_j}\chi(x) + a_0(x)y(x)\chi(x) \right) dx.$$

To every control $u \in L^\infty(\Omega_h)$, we associate a discrete state $y_h(u)$ as the unique solution in $Y_{h,0}$ of the system of nonlinear equations:

$$a(y_h(u), \chi_h) + \int_{\Omega_h} f(x, y_h(u)(x))\chi_h(x)dx = \int_{\Omega_h} u(x)\chi_h(x)dx \ \forall \chi_h \in Y_{h,0}. \tag{12}$$

For every triangle T let us define $\alpha_T = \inf\{\alpha(x) : x \in T\}$ and $\beta_T = \sup\{\beta(x) : x \in T\}$. Let us also define the functions in U_h, $\alpha_h(x) = \alpha_T$ and $\beta_h(x) = \beta_T$ if $x \in T$. For $1 \leq j \leq n_e + n_i$ let us define $G_{h,j}(u) = y_h(u)(x_j) - \sigma_j$. The set of admissible controls for the discretized problem is

$$\begin{aligned} U_{h,ad} = \{u_h \in U_h :\ & \alpha_h(x) \leq u_h(x) \leq \beta_h(x) \text{ for a.e. } x \in \Omega_h, \\ & G_{h,j}(u_h) = 0 \text{ for } 1 \leq j \leq n_e, \\ & G_{h,j}(u_h) \leq 0 \text{ for } n_e + 1 \leq j \leq n_e + n_i\}. \end{aligned}$$

 ISBN: 978-84-612-9727-6

Problem (P_h) reads like

$$(P_h)\begin{cases}\min J_h(u_h)=\displaystyle\int_{\Omega_h} L(x,y_h(u_h)(x),u_h(x))dx\\ \text{subject to } (y_h(u_h),u_h)\in Y_{h,0}\times U_{h,ad}\end{cases}$$

Remark 3.1 *All the calculations remain valid with the more natural election*

$$\alpha_T=\frac{1}{|T|}\int_T\alpha(x)dx,\ \beta_T=\frac{1}{|T|}\int_T\beta(x)dx$$

except Theorems 4.3 and 5.1.

We will state without proof our main results. For the proofs and some interesting intermediate lemmas, the reader is referred to a forthcoming paper.

Definition 3.2 *Let $\bar{u}_h\in U_{h,ad}$ be a fixed control. For $\varepsilon>0$ define*

$$\Omega_{h,\varepsilon}=\text{int}(\text{clo}\{x\in\Omega_h:\ \alpha_h(x)+\varepsilon<\bar{u}_h(x)<\beta_h(x)-\varepsilon\}).$$

Let $I_{0h}=\{j\le n_e+n_i: y_h(\bar{u}_h)(x_j)=\sigma_j\}$. We will say that $\bar{u}_h$ is regular if there exists $\bar{\varepsilon}_h>0$ and functions $\{\bar{w}_{h,j}\}_{j\in I_{0h}}\subset U_h$ such that

1. $\text{supp}\,\bar{w}_{h,j}\subset\Omega_{h,\varepsilon_h}$.
2. $G'_{h,i}(\bar{u}_h)\bar{w}_{h,j}=\delta_{i,j}$ *for* $i,j\in I_{0h}$.

Define the Lagrangian function for problem (P_h) $\mathcal{L}_h: L^\infty(\Omega)\times\mathbb{R}^{n_e+n_i}\to\mathbb{R}$ as

$$\mathcal{L}_h(u,\lambda)=J_h(u)+\sum_{j=1}^{n_e+n_i}\lambda_j G_{h,j}(u).$$

Lemma 3.3 *Suppose that $\bar{u}_h$ is a local solution of (P_h) that is regular in the sense of Definition 3.2. Then there exist real numbers $\{\bar{\lambda}_{hj}\}_{j=1}^{n_e+n_i}$ such that*

$$\bar{\lambda}_{hj}\ge 0 \text{ and } \bar{\lambda}_{hj}G_{h,j}(\bar{u}_h)=0 \text{ if } n_e+1\le j\le n_e+n_i,$$

$$\partial_u\mathcal{L}_h(\bar{u}_h,\bar{\lambda}_h)(u_h-\bar{u}_h)\ge 0\ \forall u_h\in\{u_h\in U_h:\ \alpha_h(x)\le u_h(x)\le\beta_h(x)\text{ for a.e. } x\in\Omega\}. \tag{13}$$

4 Convergence

We have the following convergence results:

Theorem 4.1 *Let us assume that (P) has at least one regular solution in the sense of definition 2.2. For all $h>0$ let $\bar{u}_h$ be a solution of (P_h) and consider the sequence $\{\bar{u}_h\}_{h>0}$. Then*

1. *there exist weakly*-converging subsequences in $L^\infty(\Omega)$ (still indexed by h),*

2. *if the subsequence* $\{\bar{u}_h\}_{h>0}$ *converges weakly* to* $\bar{u}$*, then* $\bar{u}$ *is a solution of (P),*

3. $\lim_{h\to 0} J_h(\bar{u}_h) = J(\bar{u}) = \inf(P)$ *and*

4. $\lim_{h\to 0} \|\bar{u} - \bar{u}_h\|_{L^2(\Omega)} = 0.$

Furthermore, if $\bar{u}$ *is regular, then*

5. *there exists* $h_0 > 0$ *such that* $\bar{u}_h$ *is regular for* (P_h) *for* $0 < h < h_0$ *and*

6. $\lim_{h\to 0} \bar{\lambda}_h = \bar{\lambda}$

Next we state the behavior of the optimal control near the singularities and we are able to prove uniform convergence. Let $I_{\bar{u}} = \{j \in I_0 : \bar{\lambda}_j \neq 0\}$.

Theorem 4.2 *Under all the assumptions of Theorem 4.1 and supposing further that assumption* ***(A3)*** *is satisfied,*

1. *Let* $j \in I_{\bar{u}}$*. There exists* $R_j > 0$ *such that, for* h *small enough*

$$\bar{u}_h(x) = \begin{cases} \alpha_h(x) & \text{if } \bar{\lambda}_j > 0 \\ \beta_h(x) & \text{if } \bar{\lambda}_j < 0 \end{cases} \quad \forall x \in B_{R_j}(x_j).$$

2. *Finally*

$$\lim_{h\to 0} \|\bar{u} - \bar{u}_h\|_{L^\infty(\Omega_h)} = 0. \tag{14}$$

We finish this section with a kind of reciprocal of the previous result:

Theorem 4.3 *Let* $\bar{u}$ *be a strict local minimum of (P) that is regular in the sense of Definition 2.2. Then there exists* $\rho > 0$ *and* $h_0 > 0$ *such that for every* $0 < h < h_0$*,* (P_h) *has a local minimum* $\bar{u}_h$ *in* $B_\rho(\bar{u})$*. Furthermore,* $\bar{u}_h$ *is regular in the sense of Definition 3.2, the sequence* $\bar{u}_h \to \bar{u}$ *strongly in* $L^\infty(\Omega_h)$ *and* $\bar{\lambda}_{h,j} \to \bar{\lambda}_j$ *for* $j \in I_0$*.*

5 Error estimates

Finally, we are able to get error estimates for both the control and the Lagrange multiplier. Suppose now that $\alpha(x)$ and $\beta(x)$ are constant.

Theorem 5.1 *Let* $\bar{u}$ *be a strict local minimum of (P) that is regular in the sense of Definition 2.2 and satisfies second order optimality condition (11). Then the following estimate holds:*

$$\|\bar{\lambda} - \bar{\lambda}_h\| + \|\bar{u} - \bar{u}_h\|_{L^2(\Omega_h)} \leq Ch$$

 ISBN: 978-84-612-9727-6

References

[1] T. F. D. Andrei Drăgănescu and L. R. Scott, *Failure of the discrete maximum principle for an elliptic finite element problem*, Math. Comp., 74 (2004), pp. 1–23.

[2] E. Casas, *Error estimates for the numerical approximation of semilinear elliptic control problems with finitely many state constraints*, ESAIM Control Optim. Calc. Var., 8 (2002), pp. 345–374 (electronic). A tribute to J. L. Lions.

[3] ——, *Necessary and sufficient optimality conditions for elliptic control problems with finitely many pointwise state constraints*, ESAIM Control Optim. Calc. Var., (2008), pp. – (electronic).

[4] E. Casas and M. Mateos, *Uniform convergence of the FEM. Applications to state constrained control problems*, Comput. Appl. Math., 21 (2002), pp. 67–100. Special issue in memory of Jacques-Louis Lions.

[5] E. Casas, M. Mateos, and J.-P. Raymond, *Error estimates for the numerical approximation of a distributed control problem for the steady-state Navier-Stokes equations*, SIAM J. Control Optim., 46 (2007), pp. 952–982 (electronic).

[6] J. Karátson, S. Korotov, and M. Křížek, *On discrete maximum principles for nonlinear elliptic problems*, Math. Comput. Simulation, 76 (2007), pp. 99–108.

[7] R. Rannacher, *Zur L^∞-Konvergenz linearer finiter Elemente beim Dirichlet-Problem*, Math. Z., 149 (1976), pp. 69–77.

 ISBN: 978-84-612-9727-6

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009 30 June, 1–3 July 2009.

An algorithm for classification of 3-dimensional complex Leibniz algebras

José Manuel Casas[1], Manuel A. Insua[2], Manuel Ladra[2] and Susana Ladra[3]

[1] *Dpto. de Matemática Aplicada I, Universidad de Vigo, 36005 Pontevedra, Spain*

[2] *Departamento de Álgebra, Universidad de Santiago, E-15782 Santiago, Spain*

[3] *Departamento de Computación, Universidad de A Coruña, 15071 A Coruña, Spain*

emails: jmcasas@uvigo.es, avelino.insua@gmail.com, manuel.ladra@usc.es, sladra@udc.es

Abstract

We construct an algorithm running under Mathematica using Gröbner bases which decides in terms of the existence of a non singular matrix P if two Leibniz algebra structures over a finite dimensional $\mathbb{C}$-vector space are representative of the same isomorphism class.

We apply this algorithm in order to obtain a reviewed classification of the 3-dimensional Leibniz algebras given by Ayupov and Omirov.

Key words: Leibniz algebra, Gröbner bases.
MSC 2000: 17A32, 13P10, 68W30.

1 Introduction

A classical problem in Lie algebras theory is to know how many different (up to isomorphisms) finite-dimensional Lie algebras are for each dimension [12, 13].

The classical methods to obtain the classifications essentially solve the system of equations given by the bracket laws, that is for a Lie algebra $\mathfrak{g}$ over a field $\mathbb{K}$ with basis $\{a_1, \ldots, a_n\}$, the bracket is completely determined by the scalars $c_{ij}^k \in \mathbb{K}$ such that

$$[a_i, a_j] = \sum_{k=1}^{n} c_{ij}^k a_k \tag{1}$$

so that the Lie algebra structure is determined by means of the computation of the structure constants c_{ij}^k. The solutions of the system (1) can be computed by different methods, including Gröbner bases techniques [10, 11], nevertheless the classification

must be presented by means of isomorphism classes. So the matter is to know how many different solutions of (1) can be representative of the same isomorphism class.

Leibniz algebras, introduced by Loday in [14] when he studied periodicity phenomenons in algebraic K-theory, are $\mathbb{K}$-vector spaces $\mathfrak{g}$ endowed with a bilinear operation $[-,-]\colon \mathfrak{g}\times\mathfrak{g}\to\mathfrak{g}$ satisfying the Leibniz identity $[x,[y,z]]=[[x,y],z]-[[x,z],y]$, for all $x,y,z\in\mathfrak{g}$. They are a non skew-symmetric version of Lie algebras. The main motivation to study Leibniz algebras is the existence of a (co)homology theory for Leibniz algebras which restricted to Lie algebras provides new invariants.

The classification of Leibniz algebras in low dimensions is obtained for specific classes of Leibniz algebras (solvable, nilpotent, filiform, etc.) [1, 2, 3, 4, 5, 6, 9]. The classification problem is very difficult to handle because the space of solutions of the system (1) becomes very hard to compute, specially for dimensions $n\geq 3$ since it is necessary to solve an $n\times n$ system, and so mistakes are frequent in the literature.

In [7] we have developed an algorithm for testing the Leibniz algebra structure using techniques of Gröbner bases. We have applied this test to the classification of 3-dimensional complex Leibniz algebras showed in [4] and we have detected that the isomorphism class whose representative element is the algebra with basis $\{x,y,z\}$ and bracket given by $[x,y]=\alpha x;[x,z]=\alpha x,[z,y]=x$ and 0 otherwise, doesn't correspond with a Leibiniz algebra structure.

Our goal in the present paper is to obtain a complete classification of the 3-dimensional Leibniz algebras over the field $\mathbb{C}$. To do this, first of all we compute all the solutions of the system of equations which we obtain from (1) having in mind the decomposition $\mathfrak{g}=\mathfrak{g}^{\mathrm{ann}}\oplus\mathfrak{g}_{\mathrm{Lie}}$, the dimension of $\mathfrak{g}^{\mathrm{ann}}$, the Leibniz identity and Gröbner bases computations.

To reach our goal, we construct an algorithm running under Mathematica using Gröbner bases which compares two solutions and it decides if there exists an isomorphism between them or not in terms of the existence of a non singular matrix P, that is, given two different structures $(\mathfrak{g},[-,-]_1)$ and $(\mathfrak{g},[-,-]_2)$ which are solutions of the system (1), we must verify if they are isomorphic. For that, it is necessary to check the existence of a non singular matrix P satisfying the equation

$$P\cdot[a_i,a_j]_1=[P\cdot a_i,P\cdot a_j]_2 \tag{2}$$

for all $i,j\in\{1,2,3\}$. In this step we use computational methods to construct the matrix P, whose computer program is presented in section 4. With the computations carried out with this program, we present in section 3 a reviewed classification of 3-dimensional Leibniz algebras given in [4], obtaining the new isomorphism class 2 (f).

This technique can be extended to any dimension.

2 On Leibniz algebras

Definition 2.1. *A Leibniz algebra $\mathfrak{g}$ is a $\mathbb{K}$-vector space equipped with a bilinear map $[-,-]:\mathfrak{g}\times\mathfrak{g}\to\mathfrak{g}$ satisfying the Leibniz identity*

$$[x,[y,z]]=[[x,y],z]-[[x,z],y],\ \text{for all } x,y,z\in\mathfrak{g}\,. \tag{3}$$

 ISBN: 978-84-612-9727-6

When the bracket satisfies $[x, x] = 0$ for all $x \in \mathfrak{g}$, then the Leibniz identity (3) becomes the Jacobi identity, so a Leibniz algebra is a Lie algebra. Hence, there is a canonical inclusion functor from the category **Lie** of Lie algebras to the category **Leib** of Leibniz algebras. This functor has as left adjoint the Liezation functor which assigns to a Leibniz algebra $\mathfrak{g}$ the Lie algebra $\mathfrak{g}_{\mathrm{Lie}} = \mathfrak{g}/\mathfrak{g}^{\mathrm{ann}}$, where $\mathfrak{g}^{\mathrm{ann}} = \langle\{[x, x], x \in \mathfrak{g}\}\rangle$.

Examples 2.2.

1. Lie algebras
2. Let A be a K-associative algebra equipped with a K-linear map $D : A \to A$ satisfying
$$D(a(Db)) = DaDb = D((Da)b), \text{ for all } a, b \in A \tag{4}$$
Then A with the bracket $[a, b] = aDb - Dba$ is a Leibniz algebra.
If $D = Id$ we obtain the Lie algebra structure associated to an associative algebra. If D is an idempotent algebra endomorphism ($D^2 = D$) or D is a derivation of square zero ($D^2 = 0$), then D satisfies equation (4) and the bracket gives rise to a structure of non-Lie Leibniz algebra.
3. Let D be a dialgebra [15]. Then $(D, [-,-])$ is a Leibniz algebra with respect to the bracket defined by $[x, y] = x \dashv y - y \vdash x, x, y \in D$.
4. Let $\mathfrak{g}$ be a differential Lie algebra, then $(\mathfrak{g}, [-,-]_d)$ with $[x, y]_d := [x, dy]$ is a non-Lie Leibniz algebra.

A homomorphism of Leibniz algebras is a $\mathbb{K}$-linear map $\Phi : \mathfrak{g} \to \mathfrak{h}$ such that $\Phi([x, y]_{\mathfrak{g}}) = [\Phi(x), \Phi(y)]_{\mathfrak{h}}$, for all $x, y \in \mathfrak{g}$. In case of finite dimensional Leibniz algebras $\mathfrak{g}$ and $\mathfrak{h}$, the homomorphism Φ can be represented by means of a matrix P.

Consider two Leibniz algebras $(\mathfrak{g}, [-,-]_1)$ and $(\mathfrak{g}, [-,-]_2)$ with the same underlying $\mathbb{K}$-vector space with basis $\{a_1, \ldots, a_n\}$ and different structure given by the brackets $[-,-]_1$ and $[-,-]_2$. If there exists a non singular matrix P such that the change of variables given by P provides the following diagram commutative

$$\begin{array}{ccc} \mathfrak{g} \times \mathfrak{g} & \xrightarrow{[-,-]_1} & \mathfrak{g} \\ {\scriptstyle P\times P}\uparrow & & \downarrow{\scriptstyle P^{-1}} \\ \mathfrak{g} \times \mathfrak{g} & \xrightarrow{[-,-]_2} & \mathfrak{g} \end{array}$$

that is to say, the following identity holds

$$P \cdot [a_i, a_j]_2 = [P \cdot a_i, P \cdot a_j]_1 ,$$

and then the Leibniz algebras are isomorphic.

Proposition 2.3. *(Consistency algorithm) [8] If we have polynomials $f_1, \ldots, f_s \in \mathbb{C}[x_1, \ldots, x_n]$ then $f_1, \ldots, f_s$ have no common zero in $\mathbb{C}^n$ if and only if, the Gröbner basis of the ideal generated for $f_1, \ldots, f_s$ is $\{1\}$.*

 ISBN: 978-84-612-9727-6

So, if we put together this two facts we have the following algorithm:

Input: Two Leibniz algebras $(L, [-,-]_1)$ and $(L, [-,-]_2)$ with $dim_{\mathbb{C}}(L) = n$ and basis $\{a_1, \ldots, a_n\}$.

Output: True if $(L, [-,-]_1)$ is isomorphic to $(L, [-,-]_2)$ and False in other case.

1. Compute the following system of equations $(P = (p_{ij}))$

$$P \cdot [a_i, a_j]_2 - [P \cdot a_i, P \cdot a_j]_1 = 0; \quad i, j \in \{1, \ldots, n\}\,.$$

2. To ensure that P is going to be non-singular, we add the following relation with a new variable Y:

$$Det[P] \cdot Y - 1 = 0\,.$$

3. Compute a Gröbner basis G of the ideal $< \{P \cdot [a_i, a_j]_2 - [P \cdot a_i, P \cdot a_j]_1\}_{i,j \in \{1,\ldots,n\}} \cup \{Det[P] \cdot Y - 1\} >$.

4. ¿$G = \{1\}$?

   4.1. Yes.

   Return False.

   4.2. No.

   Return True.

3 Application

We devote the present section to apply our technique together with the test developed in [7] in order to obtain the classification of the 3-dimensional complex Leibniz algebras and to compare it with the classification given in [4].

We will use as an invariant of classification $\mathfrak{g}^{\text{ann}}$ and taking into account that:

$$0 \longrightarrow \mathfrak{g}^{\text{ann}} \longrightarrow \mathfrak{g} \longrightarrow \mathfrak{g}_{\text{Lie}} \longrightarrow 0 \tag{5}$$

is a (split) short exact sequence we can conclude that $\mathfrak{g} = \mathfrak{g}^{\text{ann}} \oplus \mathfrak{g}_{\text{Lie}}$ as $\mathbb{K}$-vector spaces.

In the sequel, let $\mathfrak{g}$ be a 3-dimensional Leibniz algebra over $\mathbb{C}$ and $\{a_1, a_2, a_3\}$ a $\mathbb{C}$-basis of $\mathfrak{g}$ such that $a_i \in \mathfrak{g}^{\text{ann}}$ or $a_i \in \mathfrak{g}_{\text{Lie}}$, $i \in \{1, 2, 3\}$.

$\mathfrak{g}_{\text{Lie}}$ is a Lie algebra, so we can suppose that we have chosen a $\mathbb{C}$-basis of $\mathfrak{g}$, which verifies that the restriction of the bracket to $\mathfrak{g}_{\text{Lie}}$ is in a canonical form.

So, if $\mathfrak{g}^{\text{ann}} =< \{a_1\} >$ the bracket will be something like this [1]:

$$\begin{pmatrix} \theta & \alpha_1 \cdot a_1 & \alpha_2 \cdot a_1 \\ \theta & \alpha_3 \cdot a_1 & \alpha_4 \cdot a_1 + x \cdot a_2 \\ \theta & \alpha_5 \cdot a_1 - x \cdot a_2 & \alpha_6 \cdot a_1 \end{pmatrix}, \quad x \in \{0, 1\} \tag{6}$$

[1] $\theta = (0,0,0), a_1 = (1,0,0), a_2 = (0,1,0), a_3 = (0,0,1)$

 ISBN: 978-84-612-9727-6

($[a_i, a_j]$ is the entry of matrix (6) which is placed at the row i and column j)
And if $\mathfrak{g}^{\text{ann}} =< \{a_1, a_2\} >$ we will have:

$$\begin{pmatrix} \theta & \theta & \alpha_1 \cdot a_1 + \alpha_2 \cdot a_2 \\ \theta & \theta & \alpha_3 \cdot a_1 + \alpha_4 \cdot a_2 \\ \theta & \theta & \alpha_5 \cdot a_1 + \alpha_6 \cdot a_2 \end{pmatrix} \tag{7}$$

The case $\dim \mathfrak{g}^{\text{ann}} = 0$ is not considered because it implies that $\mathfrak{g}$ is a Lie algebra and its classification is well-known [13].

The following step is to apply Leibniz identity to each case, and so we will obtain a system of equations.

if $\mathfrak{g}^{\text{ann}} =< \{a_1\} >$ and $x = 0$ the system is:

$$\begin{aligned} \alpha_2 \cdot (\alpha_3 \cdot \alpha_6 - \alpha_4 \cdot \alpha_5) &= 0 \\ -\alpha_2 \cdot \alpha_5 + \alpha_1 \cdot \alpha_6 &= 0 \\ -\alpha_2 \cdot \alpha_3 + \alpha_1 \cdot \alpha_4 &= 0 \end{aligned}$$

if $\mathfrak{g}^{\text{ann}} =< \{a_1\} >$ and $x = 1$ the system is:

$$\begin{aligned} \alpha_1 &= 0 \\ \alpha_3 \cdot (\alpha_4 - \alpha_5) &= 0 \\ -\alpha_4 - \alpha_5 + \alpha_2 \cdot \alpha_5 &= 0 \\ \alpha_3 \cdot (2 - \alpha_2) &= 0 \end{aligned}$$

And finally, if $\mathfrak{g}^{\text{ann}} =< \{a_1, a_2\} >$ then Leibniz identity does not generate any equation except trivial equation $0 = 0$.

From the discussion of each system of equations we will obtain many Leibniz algebras but these algebras are sometimes isomorphic, then we will apply the algorithm after Proposition 2.3 to obtain a classification of Leibniz algebras in isomorphism classes. If we work in this way we reach the following classification of 3-dimensional Leibniz algebras. All the non written brackets are equal to zero.

1. Case 1: $dim_{\mathbb{C}}(\mathfrak{g}^{\text{ann}}) = 0$ (Lie algebras case)
   (a) $[a_2, a_3] = a_1$.
   (b) $[a_1, a_2] = a_1$.
   (c) $[a_1, a_3] = a_1; [a_2, a_3] = \alpha \cdot a_2, (\alpha \neq 0)$.
   (d) $[a_1, a_3] = a_1 + \beta \cdot a_2; [a_2, a_3] = a_2, (\beta \neq 0)$.
   (e) $[a_1, a_2] = a_3; [a_2, a_3] = a_1; [a_3, a_1] = a_2$.
2. Case 2: $dim_{\mathbb{C}}(\mathfrak{g}^{\text{ann}}) = 1$ (non-Lie Leibniz algebras)
   (a) $[a_2, a_2] = \gamma \cdot a_1, \gamma \in \mathbb{C}; [a_3, a_2] = a_1, [a_3, a_3] = a_1$.

 ISBN: 978-84-612-9727-6

(b) $[a_3, a_3] = a_1$.

(c) $[a_2, a_2] = a_1; [a_3, a_3] = a_1$.

(d) $[a_1, a_3] = a_1$.

(e) $[a_1, a_3] = \alpha \cdot a_1, \alpha \in \mathbb{C} - \{0\}; [a_2, a_3] = a_2; [a_3, a_2] = -a_2$.

(f) $[a_2, a_3] = a_2; [a_3, a_2] = -a_2; [a_3, a_3] = a_1$.

(g) $[a_1, a_3] = 2 \cdot a_1; [a_2, a_2] = a_1; [a_2, a_3] = a_2; [a_3, a_2] = -a_2; [a_3, a_3] = a_1$.

3. Case 3: $dim_{\mathbb{C}}(\mathfrak{g}^{\text{ann}}) = 2$ (non-Lie Leibniz algebras)

(a) $[a_1, a_3] = \beta \cdot a_1, \beta \in \mathbb{C} - \{0\}; [a_2, a_3] = a_2$.

(b) $[a_1, a_3] = a_1 + a_2; [a_2, a_3] = a_2$.

(c) $[a_1, a_3] = a_2; [a_3, a_3] = a_1$.

(d) $[a_1, a_3] = a_2; [a_2, a_3] = a_2; [a_3, a_3] = a_1$.

Remark 3.1.

1. *If $\gamma_1, \gamma_2 \in \mathbb{C}$ such that $\gamma_1 \neq \gamma_2$, then the corresponding two Leibniz algebras of the family 2 (a) are not isomorphic.*

2. *If $\alpha_1, \alpha_2 \in \mathbb{C} - \{0\}$ such that $\alpha_1 \neq \alpha_2$, then the corresponding two Leibniz algebras of the family 2 (e) are not isomorphic.*

3. *If $\beta_1, \beta_2 \in \mathbb{C} - \{0\}$ such that $\beta_1 \neq \beta_2$, then the corresponding two Leibniz algebras of the family 3 (a) are isomorphic if and only if, $\beta_1 = \frac{1}{\beta_2}$*

4. *If we choose two Leibniz algebras in different families, these algebras are not isomorphic.*

4 Computer program

In this section we describe a program running under Mathematica for implementing the algorithm discusses in the previous section. The program establishes the existence of a non singular matrix P satisfying the equation (2). The Mathematica code together with some examples are available in `http://web.usc.es/~mladra/research.html`.

```
(*########################################################################
 This program inform us about the existence of a non singular matrix P
which defines an isomorphism between two Leibniz algebras structures on
                    an underlying C-vector space.

########################################################################*)
```

 ISBN: 978-84-612-9727-6

```
(* Let (𝔤 =< a_1,...,a_n >,[-,-]_1) and (𝔤 =< a_1,...,a_n >,[-,-]_2) be two Leibniz
algebras of dimension n *)
(* Insert the Brackets represented by BracketOne[{i1,i2}] := {λ_1,...,λ_n} where
[a_i1,a_i2]_1 = λ_1 a_1+...+λ_n a_n and BracketTwo[{i1,i2}] := {μ_1,...,μ_n} where [a_i1,a_i2]_2 =
μ_1 a_1 + ... + μ_n a_n .*)
(* Insert the bilinear condition over the brackets *)

BracketOne[vectorX_, vectorY_] :=
 Sum[vectorX[[i]]*vectorY[[j]]*BracketOne[{i, j}],
     {j, 1,Length[vectorX]}, {i, 1, Length[vectorX]}]

BracketTwo[vectorX_, vectorY_] :=
 Sum[vectorX[[i]]*vectorY[[j]]*BracketTwo[{i, j}],
     {j, 1,Length[vectorX]}, {i, 1, Length[vectorX]}]


IsoLeibnizQ[n_] := Module[{res, eqs, P, G},

   P = Table[Table[p[irow, jcol], {jcol, 1, n}], {irow, 1, n}];

   eqs = Join[
              Flatten[
                 Table[
                   Table[
                         P.BracketTwo[{i, j}] -BracketOne[P.IdentityMatrix[n][[i]],
                                                          P.IdentityMatrix[n][[j]]]
                    , {j, 1, n}]
                  , {i, 1, n}]]
             ,
             {Y*Det[P] - 1}];

   G = GroebnerBasis[eqs, Join[{Y}, Flatten[P]]];

   If[G === {1},
    res = "Algebras are not isomorphic";
    ,
    res = "Algebras are isomorphic"
    ];

   Return[res];
   ];
```

Next example shows the application of the algorithm on two Leibniz algebras structures corresponding to the family 3 (a).

Examples 4.1. Let $(L =< \{a_1, a_2, a_3\} >, [-,-]_1)$ and $(L =< \{a_1, a_2, a_3\} >, [-,-]_2)$ be two Leibniz algebras such that $[a_1, a_3]_1 = 2 \cdot a_1$, $[a_2, a_3]_1 = a_2$ (0 otherwise) and $[a_1, a_3]_2 = \frac{1}{2} \cdot a_1$, $[a_2, a_3]_2 = a_2$ (0 otherwise).

```
BracketOne[{1,1}] := {0, 0, 0}
BracketOne[{1,2}] := {0, 0, 0}
BracketOne[{1,3}] := {2, 0, 0}
BracketOne[{2,1}] := {0, 0, 0}
BracketOne[{2,2}] := {0, 0, 0}
BracketOne[{2,3}] := {0, 1, 0}
BracketOne[{3,1}] := {0, 0, 0}
BracketOne[{3,2}] := {0, 0, 0}
BracketOne[{3,3}] := {0, 0, 0}

BracketTwo[{1,1}] := {0, 0, 0}
BracketTwo[{1,2}] := {0, 0, 0}
BracketTwo[{1,3}] := {1/2, 0, 0}
BracketTwo[{2,1}] := {0, 0, 0}
BracketTwo[{2,2}] := {0, 0, 0}
BracketTwo[{2,3}] := {0, 1, 0}
BracketTwo[{3,1}] := {0, 0, 0}
BracketTwo[{3,2}] := {0, 0, 0}
BracketTwo[{3,3}] := {0, 0, 0}

BracketOne[vectorX_, vectorY_] :=
 Sum[vectorX[[i]]*vectorY[[j]]*BracketOne[{i, j}],
     {j, 1,Length[vectorX]}, {i, 1, Length[vectorX]}]

BracketTwo[vectorX_, vectorY_] :=
 Sum[vectorX[[i]]*vectorY[[j]]*BracketTwo[{i, j}],
     {j, 1,Length[vectorX]}, {i, 1, Length[vectorX]}]

IsoLeibnizQ[3]
```

Algebras are isomorphic

Acknowledgements

First and third authors were supported by Ministerio de Educación y Ciencia, Grant MTM2006-15338-C02-02 (European FEDER support included) and by Xunta de Galicia, Grant PGIDITI06PXIB371128PR.

References

[1] S. Albeverio, Sh. A. Ayupov and B. A. Omirov, *On nilpotent and simple Leibniz algebras*, Comm. Algebra **33** (1) (2005) 159–172.

[2] S. Albeverio, B. A. Omirov and I. S. Rakhimov, *Varieties of nilpotent complex Leibniz algebras of dimension less than five*, Comm. Algebra **33** (2005) 1575–1585.

[3] S. Albeverio, B. A. Omirov and I. S. Rakhimov, *Classification of 4-dimensional nilpotent complex Leibniz algebras*, Extracta Math. **21** (3) (2006) 197–210.

[4] Sh. A. Ayupov and B. A. Omirov, *On Leibniz algebras*, *Algebra and Operator Theory (Tashkent, 1997), 1–12*, Kluwer Acad. Publ., Dordrecht, 1998.

[5] Sh. A. Ayupov and B. A. Omirov, *On a description of irreducible component in the set of nilpotent Leibniz algebras containing the algebra of maximal nilindex, and classification of graded filiform Leibniz algebras*, *Computer algebra in scientific computing (Samarkand, 2000), 21–34*, Springer, Berlin, 2000.

[6] Sh. A. Ayupov and B. A. Omirov, *On some classes of nilpotent Leibniz algebras*, Siberian Math. J. **42** (1) (2001) 15–24.

[7] J. M. Casas, M. A. Insua, M. Ladra and S. Ladra, *Algorithm for testing the Leibniz algebra structure*, Proceedings of EUROCAST (2009).

[8] D. Cox, J. Little and D. O'Shea, *Ideals, Varieties, and Algorithms*, 2nd ed., Springer-Verlag, New York, 1997.

[9] C. Cuvier, *Algèbres de Leibnitz: définitions, propriétés*, Ann. Scient. École Norm. Sup (4) **27** (1) (1994) 1–45.

[10] W. de Graaf, *Lie algebras: theory and algorithms*, North-Holland Publishing Co., Amsterdam, 2000.

[11] W. de Graaf, *Classification of solvable Lie algebras*, Experiment. Math. **14** (2005) 15–25.

[12] M. Goze and Y. Khakimdjanov, *Nilpotent Lie algebras*, Mathematics and its Applications 361, Kluwer Acad. Publ., Dordrecht, 1996.

[13] N. Jacobson, *Lie algebras*, Interscience Publ., New York-London, 1962.

[14] J.-L. Loday, *Une version non commutative des algèbres de Lie: les algèbres de Leibniz*, Enseign. Math. (2) **39** (1993) 269–293.

[15] J.-L. Loday, *Algèbres ayant deux opérations associatives (digèbres)*, C. R. Acad. Sci. Paris Sér. I Math. **321** (2) (1995) 141–146.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

A Spherical Interpolation Algorithm Using Zonal Basis Functions

Roberto Cavoretto[1] and Alessandra De Rossi[1]

[1] *Department of Mathematics, University of Turin*

emails: roberto.cavoretto@unito.it, alessandra.derossi@unito.it

Abstract

In this paper we present a new local algorithm for spherical interpolation of large scattered data sets. The method we implemented is a local Shepard's scheme using zonal basis functions as nodal functions. The algorithm is based on an optimized nearest-neighbour searching procedure. Experimental results show efficiency and accuracy of the algorithm.

Key words: zonal basis functions, spherical Shepard's formula, local algorithms, scattered data interpolation.
MSC 2000: 65D05, 65D15, 65D17

1 Introduction

Let $\mathbb{S}^{m-1} = \{x \in \mathbb{R}^m : \|x\|_2 = 1\}$ be the unit sphere in $\mathbb{R}^m$. We consider the problem of interpolating a function $f : \mathbb{S}^{m-1} \to \mathbb{R}$, $(m \geq 1)$, defined on a finite set $\mathcal{X}_n = \{x_i\}_{i=1}^n$ of distinct *data points* or *nodes* lying on $\mathbb{S}^{m-1}$. It consists of constructing a multivariate function F, which interpolates the *data values* or *function values* f_i at the nodes x_i, namely $F(x_i) = f_i$, $i = 1, \ldots, n$. Possible applications include modeling closed surfaces in CAGD and representing scalar functions which estimate temperature, rainfall, pressure, ozone, gravitational forces, etc. at all points on the surface of the earth based on a discrete sample of values taken at arbitrary locations.

Recently, in [1], an efficient algorithm was proposed for the interpolation of large scattered data sets in bidimensional domains. It is based on a very fast *strip method*, consisting in the use of particular data structures named *strips* in the nearest neighbour searching procedure. It allows to obtain a very fast algorithm for bivariate interpolation. In this paper we extend the algorithm to the spherical setting. We found good results also in this case. At the moment investigations in the direction of comparison of this algorithm with the one proposed in [3] are under consideration.

The paper is organized as follows. In section 2 the zonal basis function (ZBF) method is briefly recalled. Section 3 is devoted to the local spherical interpolation

method with ZBFs. In section 4 the spherical algorithm used is explained and in section 5 numerical results are given.

2 ZBF Interpolation

Since we will propose a local interpolation scheme involving a *zonal basis function* (or *spherical radial basis function* (SRBF)) interpolant, here we focus both on theoretical and computational aspects of the ZBF method (see, e.g. [6, 4, 5]), recalling some basic mathematical interpolation tools [8].

Definition 2.1. *Given a set of distinct data points $\mathcal{X}_n = \{x_i\}_{i=1}^n$ arbitrarily distributed on $\mathbb{S}^{m-1}$, and the associated function values $\mathcal{F}_n = \{f_i\}_{i=1}^n$ of a function $f : \mathbb{S}^{m-1} \to \mathbb{R}$, a zonal basis function interpolant $s : \mathbb{S}^{m-1} \to \mathbb{R}$ has the form*

$$s(x) = \sum_{j=1}^{n} a_j \psi(d(x, x_j)), \qquad x \in \mathbb{S}^{m-1}, \tag{1}$$

where $d(x, x_j) = \arccos\left(x^T x_j\right)$ denotes the geodesic distance, which is the length of the shorter part of the great circle arc joining x and x_j, $\psi : [0, \pi] \to \mathbb{R}$ is called zonal basis function, and s satisfies the interpolation conditions $s(x_i) = f_i$, $i = 1, \ldots, n$.

Although, as far as we know, there is no a complete characterization of the class of the functions ψ, a sufficient condition for nonsingularity is that the corresponding matrix

$$A_{i,j} = \psi(d(x_i, x_j)), \qquad 1 \le i, j \le n, \tag{2}$$

be positive definite (see [6]).

Definition 2.2. *A continuous function $\psi : [0, \pi] \to \mathbb{R}$ is said positive definite of order n on $\mathbb{S}^{m-1}$, if*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \psi(d(x_i, x_j)) \ge 0, \tag{3}$$

for any set of distinct data points $\mathcal{X}_n = \{x_i\}_{i=1}^n$, and any $a = [a_1, \ldots, a_n]^T \in \mathbb{R}^n$. If the inequality (3) holds strictly for any nontrivial a, ψ is called strictly positive definite of order n. If ψ is (strictly) positive definite for any n, then it is called (strictly) positive definite.

Therefore, if ψ is (strictly) positive definite, the interpolant (1) is unique, since the corresponding interpolation matrix (2) is positive definite and hence nonsingular.

Generally, one requires that an interpolant reproduces the low degree spherical harmonics (as polynomials for RBFs in the multivariate setting), but this property is not satisfied. Hence, it is often convenient to add to s a spherical harmonic, which can be defined in the following way.

 ISBN: 978-84-612-9727-6

Definition 2.3. *Let $H_d \equiv H_d(\mathbb{S}^{m-1})$, $d \in \mathbb{Z}^+$, be the space of homogeneneous harmonics of degree d restricted to $\mathbb{S}^{m-1}$. The linear space H_d is called the space of spherical harmonics of exact degree d.*

It is well known that the dimension of H_d is given by

$$N_{m,d} = \dim(H_d) = \begin{cases} 1, & \text{if } d = 0, \\ \frac{2d+m-2}{d}\binom{d+m-3}{d-1}, & \text{if } d \geq 1, \end{cases}$$

and $N_{m,d} = O(d^{m-2})$, for $d \to \infty$. Moreover, the spherical harmonics of different degrees are orthogonal with respect to the L_2-inner product on $\mathbb{S}^{m-1}$

$$(f, g)_{L_2(\mathbb{S}^{m-1})} = \int_{\mathbb{S}^{m-1}} f(x)g(x)d\mu(x),$$

where $d\mu(x)$ is the standard measure on the sphere.

Now, denoting by $\{Y_{d,k} : k = 1, \ldots, N_{m,d}\}$ a (fixed) orthonormal basis of H_d, we have that $\mathcal{H}_d = \oplus_{j=0}^{d} H_j$, $d \in \mathbb{Z}^+$, is the space of spherical harmonics of degree at most d. Moreover, it is also known that $\{Y_{d,k} : k = 1, \ldots, N_{m,d};\ d = 0, 1, \ldots\}$ is a complete orthonormal basis of $L_2(\mathbb{S}^{m-1})$. For more details, we refer to [8, 13].

Then, the drawback of the lacked reproduction of the low degree spherical harmonics can be overcome, adding to the ZBF interpolant s, given by (1), a spherical harmonic of degree d. It assumes the form

$$s(x) = \sum_{j=1}^{n} a_j \psi(d(x, x_j)) + \sum_{k=1}^{V} b_k Y_k(x), \qquad x \in \mathbb{S}^{m-1}, \tag{4}$$

where $V = \dim\mathcal{H}_d(\mathbb{S}^{m-1})$, and $\{Y_1, \ldots, Y_V\}$ is a basis for $\mathbb{S}^{m-1}$.

The analytic solution (4) is obtained by requiring that s satisfies the interpolation conditions

$$s(x_i) = f_i, \qquad i = 1, \ldots, n,$$

and the additional conditions

$$\sum_{i=1}^{n} a_i Y_k(x_i) = 0, \qquad \text{for } k = 1, \ldots, V. \tag{5}$$

To compute the coefficients $a = [a_1, \ldots, a_n]^T$ and $b = [b_1, \ldots, b_V]^T$ in (4), it is required to solve the system of n linear equations in $n + V$ unknowns. Thus, supposing that $n \geq V$, we have the linear system

$$\begin{bmatrix} A & Y \\ Y^T & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \tag{6}$$

where $A = \{\psi(d(x_i, x_j))\}_{i,j=1}^{n}$ is an $n \times n$ matrix (as in (2)), $Y = \{Y_k(x_i)\}$ is an $n \times V$ matrix, and f denotes the column vector of the k-th coordinate of the function f_i.

 ISBN: 978-84-612-9727-6

Definition 2.4. *A continuous function $\psi : [0, \pi] \to \mathbb{R}$ is said strictly conditionally positive definite of order $s \in \mathbb{N}$ on $\mathbb{S}^{m-1}$, if the quadratic form (3) is positive for any set of distinct data points $\mathcal{X}_n = \{x_i\}_{i=1}^n$ and any nonzero $a = [a_1, \ldots, a_n]^T \in \mathbb{R}^n$ satisfying (5).*

Definition 2.5. *Let s be a positive integer and let $V = \dim\mathcal{H}_{s-1}(\mathbb{S}^{m-1})$. A set of distinct data points $\{x_i\}_{i=1}^V$ is named $\mathcal{H}_{s-1}(\mathbb{S}^{m-1})$-unisolvent if the only element of $\mathcal{H}_{s-1}(\mathbb{S}^{m-1})$ to vanish at each x_i is the zero spherical harmonic.*

Any strictly conditionally positive function ψ of degree s can be used to provide an augmented ZBF interpolant (4) with $d = s - 1$. Nevertheless, in order to guarantee the solution uniqueness, we require also that the interpolation points contain an $\mathcal{H}_{s-1}(\mathbb{S}^{m-1})$-unisolvent subset. Then, the interpolant (4) is unique [10] (see also [9]).

Theorem 2.1. *Let ψ be a strictly conditionally positive definite on $\mathbb{S}^{m-1}$. Let $\mathcal{X}_n = \{x_i\}_{i=1}^n$ denote a set of n distinct data points in $\mathbb{S}^{m-1}$ such that $n \geq V = \dim\mathcal{H}_{s-1}(\mathbb{S}^{m-1})$, and $\mathcal{X}_n$ contains an $\mathcal{H}_{s-1}(\mathbb{S}^{m-1})$-unisolvent subset. Then the matrix of the linear system (6) is nonsingular.*

3 Local Spherical Interpolation by ZBFs

In this section we describe a local method for the multivariate interpolation of large scattered data sets lying on the sphere. The scheme is based on the local use of zonal basis functions, i.e. ZBF interpolants as nodal functions, and represents a further variant of the well-known modified Shepard's method. Hence, this local interpolation approach exploits the characteristic of accuracy of ZBFs, overcoming common disadvantages as the unstability due to the need of solving large linear systems (possibly, bad conditioned) and the inefficiency of the ZBF global interpolation method. A similar technique was already introduced at first by Pottmann and Eck [11] (MQ), and then by De Rossi [3] (ZBF).

As we present a local interpolation method, we need to define a ZBF interpolant of the form

$$Z(x) \equiv s_{|_{\mathcal{D}}}(x), \qquad x \in \mathcal{D} \subset \mathcal{X}_n,$$

where $\mathcal{D}$ is the restriction of the data point set $\mathcal{X}_n$.

Therefore we consider the following definition of the modified spherical Shepard's method.

Definition 3.1. *Given a set of distinct data points $\mathcal{X}_n = \{x_i\}_{i=1}^n$, arbitrarily distributed on the sphere $\mathbb{S}^{m-1}$, with associated the corresponding set of real values $\mathcal{F}_n = \{f_i\}_{i=1}^n$ of an unknown function $f : \mathbb{S}^{m-1} \to \mathbb{R}$, the modified spherical Shepard's interpolant $F : \mathbb{S}^{m-1} \to \mathbb{R}$ takes the form*

$$F(x) = \sum_{j=1}^{n} Z_j(x)\bar{W}_j(x), \tag{7}$$

 ISBN: 978-84-612-9727-6

where the nodal functions $Z_j(x)$, $j = 1, \ldots, n$, are local approximants to f at x_j, relative to the subset $\mathcal{D}$ of the n_Z data points closest to x_j, satisfying the interpolation conditions $Z_j(x_j) = f_j$, and $\bar{W}_j(x)$, $j = 1, \ldots, n$, are the weight functions

$$\bar{W}_j(x) = \left[\frac{W_j(x)}{\sum_{k=1}^{n} W_k(x)} \right]^p, \quad j = 1, \ldots, n, \quad (p > 0),$$

with

$$W_j(x) = \tau(x, x_j)/\alpha(x, x_j).$$

The localizing function $\tau(x, x_j)$, often called step function, is

$$\tau(x, x_j) = \begin{cases} 1, & \text{if } x_j \in \mathcal{C}(x; s), \\ 0, & \text{otherwise}, \end{cases}$$

where $\mathcal{C}(x; s)$ is a hypercube of centre at x and side s, whereas $\alpha(x, x_j) = \arccos\left(x^T x_j\right)$.

To control the localization of the ZBFs, a certain number n_W of nodes close to x must be considered.

4 Spherical Interpolation Algorithm

In this section we propose an efficient algorithm for the interpolation on the sphere $\mathbb{S}^2 \subset \mathbb{R}^3$. In practice, this is typically the most interesting case, since it represents some physical phenomena in many areas, including e.g. geophysics and meteorology where the sphere is taken as model of the earth.

Thus, we consider the problem of approximating a function $f : D \to \mathbb{R}$, defined only on a finite set of distinct and scattered data points $\mathcal{X}_n = \{(x_i, y_i, z_i)\}_{i=1}^n$, where $D = \mathbb{S}^2 \subseteq [0, 1] \times [0, 1] \times [0, 1] \subset \mathbb{R}^3$, and $\mathcal{F}_n = \{f_i\}_{i=1}^n$ is the set of corresponding values of the unknown function. We now describe a spherical interpolation algorithm, which is based on a strip searching procedure and a standard sorting procedure as `quicksort` routine. This requires on average a time complexity $O(M \log M)$, where M is the number of points to be sorted.

Moreover, the employment of such a strip structure allows some advantages: it optimizes the searching procedure of nodes making the interpolation algorithm efficient and guarantees a high parallelism.

Here is the algorithm in detail:

Step 1. Let data points $\mathcal{X}_n$, data values $\mathcal{F}_n$, evaluation points $\mathcal{G}_s$, and localizing parameters n_Z and n_W be given.

Step 2. Order the set $\mathcal{X}_n$ with respect to a common direction (e.g. the z-axis), by applying a *quicksort$_z$ procedure*.

Step 3. Construct a local (circle) neighbourhood, for each node (x_i, y_i, z_i), $i = 1, \ldots, n$. The half-size of the neighbourhoods depends on the dimension n, the considered value n_Z, and the positive integer k_1, i.e.

$$\delta_Z = \arccos\left(1 - 2\sqrt{k_1}\frac{n_Z}{n}\right), \quad k_1 = 1, 2 \ldots \tag{8}$$

 ISBN: 978-84-612-9727-6

Step 4. Find the number of strips

$$q = \left\lceil \frac{\pi}{\delta_Z} \right\rceil,$$

deriving directly by the length π of the shorter part of the great circle joining "north pole" and "south pole", and the neighbourhood half-size δ_Z.

Step 5. Construct a suitable family of q strips of equal width $\delta_{s_1} \equiv \delta_Z$ (with possible exception of one of them) and parallel to xy-plane (either, equivalently, parallel to xz-plane or yz-plane) on the domain D. The set $\mathcal{X}_n$ of nodes is partitioned by the strip structure into q subsets $\mathcal{X}_{n_k}$, whose n_k elements are (x_{k1}, y_{k1}, z_{k1}), ..., $(x_{kn_k}, y_{kn_k}, z_{kn_k})$, $k = 1, \ldots, q$.

This lead to the following *strip partitioning procedure*:

Procedure 1.

```
count := 0;
zs := -1;
for (k from 1 to q)
{
 n[k] := 0;
 i := count + 1;
 u[k] := k*delta_z;
 v[k] := zs - (1 + cos(u[k]));
 while (z[i] ≤ v[k] && i ≤ n)
 {
  n[k] := n[k] + 1;
  count := count + 1;
  i := i + 1;
  BS[k] := count - n[k] + 1;
  ES[k] := count;
  OUTPUT(n[k] data of the k-th strip).
 }
}
```

Step 6. For each node of $\mathcal{X}_{n_k}$, $k = 1, \ldots, q$, define the strips to be examined for determining all data points belonging to a (local) neighbourhood centred at (x_i, y_i, z_i), $i = 1, \ldots, n$, applying the *strip searching procedure* described below. The number of nodes of the neighbourhood is counted and stored in m_i.

Procedure 2.

```
for (k from 1 to q)
{
 B := k - 1;
 E := k + 1;
 if (B < 1)
   B := 1;
```

 ISBN: 978-84-612-9727-6

```
if (E > q)
  E := q;
for (h from BS[k] to ES[k])
{
 m[h] := 0;
 for (i from B to E)
 {
  for (j from BS[i] to ES[i])
  {
   if ((x[j],y[j],z[j]) belongs to the h-th neighbourhood of centre
       (x[h],y[h],z[h]) and spherical radius delta_z)
    m[h] := m[h] + 1;
    STORE[h][m[h]] ← (x[j],y[j],z[j],f[j]);
  }
 }
 OUTPUT(data set belonging to the h-th neighbourhood of centre
       (x[h],y[h],z[h]) and spherical radius delta_z).
}
}
```

Step 7. Order, and then reduce to n_Z, all the nodes belonging to a circle neighbourhood centred at x_i, $i = 1, \ldots, n$, by applying a based-distance sorting process, that is a *quicksort$_d$ procedure*.

Step 8. For each node (x_j, y_j, z_j), find a local interpolant Z_j, $j = 1, \ldots, n$, constructed on the n_Z data points closest to it.

Step 9. Order all the points of the set $\mathcal{G}_s$ with respect to the z-axis, by applying a *quicksort$_z$ procedure*.

Step 10. For each evaluation point $(x, y, z) \in D$, construct a circle neighbourhood, whose half-size depends on the dimension n, the parameter value n_W, and the (positive integer) number k_2, that is,

$$\delta_W = \arccos\left(1 - 2\sqrt{k_2}\frac{n_W}{n}\right), \quad k_2 = 1, 2, \ldots \tag{9}$$

Step 11. Determine the number of strips

$$r = \left\lceil \frac{\pi}{\delta_W} \right\rceil,$$

deriving directly by the length π of the shorter part of the great circle joining "north pole" and "south pole", and the neighbourhood half-size δ_W.

Step 12. Construct a second (suitable) family of r strips of equal width $\delta_{s_2} \equiv \delta_W$ (with possible exception of one of them), again parallel to xy-plane on the domain D. The set $\mathcal{G}_s = \{(x_i, y_i, z_i)\}_{i=1}^{s}$ of evaluation points is partitioned into r subsets $\mathcal{G}_{p_k}$, whose p_k evaluation points are (x_{k1}, y_{k1}, z_{k1}), ..., $(x_{kp_k}, y_{kp_k}, z_{kp_k})$, $k = 1, \ldots, r$.

 ISBN: 978-84-612-9727-6

The strip structure is similar to that presented in Procedure 1; in particular, we have that `q` ct `r`, `n[k]` ct `p[k]`, `n` ct `s`, and `delta_z` ct `delta_w`, where "ct" means "change to".

Step 13. For each evaluation point of $\mathcal{G}_{p_k}$, $k = 1, \ldots, r$, search all data points belonging to a (local) neighbourhood of centre (x_i, y_i, z_i) and geodesic radius δ_W, by applying a procedure based on strips. The number of nodes of the neighbourhood is counted and stored in s_i, $i = 1, \ldots, s$.

As regard to Procedure 2, the following changes are required: `q` ct `r`, `m[h]` ct `s[h]`, and `delta_z` ct `delta_w`.

Step 14. Order, and then reduce to n_W, the nodes of each neighbourhood by applying a *quicksort$_d$ procedure*.

Step 15. Find a local weight function $\bar{W}_j(x, y, z)$, $j = 1, \ldots, n$, considering only the n_W points closest to the evaluation point (x, y, z), where (x, y, z) denotes the generic evaluation point.

Step 16. Apply the modified spherical Shepard's formula (7), and evaluate the surface at each evaluation point $(x, y, z) \in D$.

Note that to localize the nodes closest to each strip point, we establish the minimal number of strips to be examined, which here is three, i.e. the strip on which the considered data point lies, the previous and the next strips. The reason of such a value follows from the choice of setting $\delta_{s_1} \equiv \delta_Z$ and $\delta_{s_2} \equiv \delta_W$. Indeed, a node belonging to a strip can be closer to data points that lie in nearby strips than those in the same strip. Therefore, the searching of nodes belonging to local neighbourhood must be extended to all the strips in which there is, at least, a possible candidate (point). Obviously, for all nodes of the "first" and "last" strip, we reduce the strips to be examined to two (see Procedure 2 in Step 6, and Step 13).

The size of circle neighbourhoods is carried out so that, supposing a uniform distribution of points on all the domain D, each local neighbourhood has a prefixed number of nodes. The condition is satisfied, taking into account the dimension n, the parameter n_Z (or n_W), and the positive integer k_1 (or k_2). In particular, the rule (8) in Step 3 (or (9) in Step 10) estimates for $k_1 = 1$ (or $k_2 = 1$), at least, n_Z (or n_W) points for each neighbourhood. However, the approch we propose is completely automatic, for which the procedure locates the minimal positive integer k_1 (or k_2) satisfying the request to having a sufficient number of data points on each neighbourhood. This means that the method works successfully also when the distribution of data points is not uniform.

5 Numerical Results

In this section we show the accuracy and efficiency of the proposed algorithm, which has been implemented in *C language*. All the numerical results we present are obtained on a Pentium IV computer (2.40 GHz). In particular, we are also interested to stress the effectiveness of the considered strip searching procedure which allow to reduce the execution CPU times. For this reason, we propose a comparison between the spherical

interpolation algorithm implemented by using the strip structure on the sphere (SA), and the classical algorithm (CA), where the sphere $\mathbb{S}^2$ is not partitioned in strips (see Table 1).

In the tests we consider a few data sets of $n = 2^i \cdot 500$, $i = 0, 1, \ldots, 7$, Halton points on the sphere [14] as scattered points to be interpolated, and a set of 600 spiral points as evaluation points which are generated by using the method of Saff and Kuijlaars [12]. Data values are taken by the restriction on $\mathbb{S}^2$ of the following four (trivariate) test functions [11, 7]:

$$f_1(x,y,z) = \frac{1+2x+3y+4z}{6},$$

$$f_2(x,y,z) = \frac{9x^3-2x^2y+3xy^2-4y^3+2z^3-xyz}{10},$$

$$f_3(x,y,z) = \frac{\mathrm{e}^x+2\mathrm{e}^{y+z}}{10},$$

$$f_4(x,y,z) = \sin x \sin y \sin z.$$

n	CPU Time	
	SA	CA
500	0.400	0.511
1000	0.441	0.671
2000	0.901	2.103
4000	2.303	9.484
8000	5.969	34.139
16000	15.001	192.037
32000	38.565	807.241
64000	175.753	3717.986

Table 1: CPU times (in seconds) obtained by SA and CA using ψ_2 for f_1.

The choice of the appropriate numbers n_Z and n_W is a non trivial problem, since it determines the accuracy of the local ZBF scheme. Numerical investigations pointed out that "good" values for these parameters are $n_Z = 15$ and $n_W = 10$. Moreover, among several tested ZBFs, we take the spherical inverse multiquadric (IMQ) [4] and the logarithmic spline [7]:

$$\psi_1(t) = \frac{1}{\sqrt{1+\gamma^2-2\gamma c}}, \qquad \text{spherical IMQ},$$

$$\psi_2(t) = \frac{1}{\beta}\log\left(1+\frac{2\beta}{\sqrt{1+\beta^2-2\beta c}+1-\beta}\right), \qquad \text{logarithmic spline},$$

where $\beta, \gamma \in (0,1)$, $c = \cos(t)$, and t is the geodesic distance on the sphere, namely

 ISBN: 978-84-612-9727-6

n	f_1	f_2	f_3	f_4
500	9.4916E − 3 1.6234E − 3	9.1391E − 3 1.6347E − 3	6.1264E − 3 9.6098E − 4	3.7898E − 3 5.5405E − 4
1000	2.5101E − 3 5.0926E − 4	3.7769E − 3 5.1455E − 4	2.0022E − 3 3.3969E − 4	2.0022E − 3 3.3969E − 4
2000	1.2924E − 3 2.0323E − 4	1.0688E − 3 1.7905E − 4	8.5904E − 4 1.4012E − 4	5.8907E − 4 7.3540E − 5
4000	2.4042E − 4 4.7891E − 5	3.1177E − 4 4.9201E − 5	1.9583E − 4 3.3424E − 5	1.3363E − 4 1.6935E − 5
8000	7.4362E − 5 1.2049E − 5	6.4579E − 5 1.0369E − 5	4.0073E − 5 7.8723E − 6	2.4360E − 5 4.0642E − 6
16000	4.5552E − 5 3.3560E − 6	2.2178E − 5 2.8248E − 6	1.8007E − 5 1.9938E − 6	4.9581E − 6 9.0486E − 7
32000	6.4436E − 6 7.2957E − 7	5.1260E − 6 6.8387E − 7	6.0991E − 6 4.8144E − 7	2.1046E − 6 2.3501E − 7
64000	1.6482E − 6 1.5209E − 7	1.1047E − 6 1.4685E − 7	7.1330E − 7 9.3410E − 8	4.5072E − 7 4.7949E − 8

Table 2: MAEs and RMSEs by using ψ_1.

n	f_1	f_2	f_3	f_4
500	4.1540E − 3 7.4688E − 4	5.2372E − 3 8.7653E − 4	2.1841E − 3 3.7548E − 4	2.6918E − 3 3.3230E − 4
1000	1.2445E − 3 2.0671E − 4	1.9831E − 3 2.4179E − 4	4.9512E − 4 1.0690E − 4	4.8661E − 4 1.0567E − 4
2000	3.0853E − 4 7.1861E − 5	5.0038E − 4 7.8670E − 5	2.5460E − 4 4.2001E − 5	2.6100E − 4 3.7240E − 5
4000	9.7610E − 5 1.7739E − 5	1.3159E − 4 2.1183E − 5	5.8100E − 5 9.8421E − 6	5.3480E − 5 8.1228E − 6
8000	2.7372E − 5 4.4705E − 6	2.7909E − 5 4.3736E − 6	1.4293E − 5 2.3327E − 6	1.1469E − 5 2.0051E − 6
16000	1.8184E − 5 1.2999E − 6	8.0451E − 6 1.1528E − 6	6.9378E − 6 6.3696E − 7	2.9160E − 6 4.4485E − 7
32000	1.4535E − 6 2.4965E − 7	2.2620E − 6 2.8757E − 7	1.0116E − 6 1.2674E − 7	9.7478E − 7 1.0951E − 7
64000	4.9573E − 7 6.0061E − 8	1.0710E − 6 7.9300E − 8	3.0403E − 7 3.1628E − 8	1.7742E − 7 2.3408E − 8

Table 3: MAEs and RMSEs by using ψ_2.

$t \in [0, \pi]$. These two functions are both (strictly) positive definite on $\mathbb{S}^2$, and their values of the shape parameters are chosen to be $\gamma = \beta = 0.7$. This is a good trade-off between accuracy and stability by varying the dimension n, taking into account that in

a local approach the number of points to be interpolated is small; hence, the condition numbers of the interpolation matrices are relatively small. Regarding the value of p in the weight functions, we took $p = 1$.

Finally, in Tables 2 and 3 we show the maximum absolute errors (MAEs) and the root mean square errors (RMSEs) achieved by using ψ_1 and ψ_2, respectively.

References

[1] G. Allasia, R. Besenghi, R. Cavoretto and A. De Rossi, *A strip method for continuous surface modelling from scattered and track data*, Quaderno del Dipartimento di Matematica, Università di Torino, n.2/2009, 1–23, submitted.

[2] M. Costanzo and A. De Rossi, *A parallel algorithm for scattered data fitting on the sphere*, in M. Primicerio, R. Spigler, V. Valente (eds.), Applied and Industrial Mathematics in Italy, World Scientific, 2005, 249–259.

[3] A. De Rossi, *Spherical interpolation of large scattered data sets using zonal basis functions*, in M. Dæhlen, K. Mørken and L. L. Schumaker (eds.), Mathematical methods for curves and surfaces: Tromsø 2004, Nashboro Press, Brentwood, TN, 2005, 125–134.

[4] G. E. Fasshauer and L. L. Schumaker, *Scattered data fitting on the sphere*, in M. Dæhlen, T. Lyche and L. L. Schumaker (eds.), Mathematical methods for curves and surfaces II, Vanderbilt Univ. Press, Nashville, TN, 1998, 117–166.

[5] W. Freeden, T. Gervens and M. Schreiner, *Constructive approximation on the sphere with applications to geomathematics*, Clarendon Press, Oxford, UK, 1998.

[6] S. Hubbert and B. J. C. Baxter, *Radial basis function for the sphere*, International Series of Numerical Mathematics, vol. 137, Birkhauser Verlag Basel, Switzerland, 2001, 33–47.

[7] S. Hubbert, *Computing with radial basis functions on the sphere*, preprint, 2002.

[8] S. Hubbert and T. Morton, *L_p-error estimates for radial basis function interpolation on the sphere*, J. Approx. Theory **129** (2004) 58–77.

[9] J. Levesley, W. Light, D. Ragozin and X. Sun, *A simple approach to the variational theory for the interpolation on spheres*, in International Series of Numerical Mathematics, vol. 132, Birkhauser Verlag, Basel, 1999, 117–143.

[10] W. A. Light and H. Wayne, *Power functions and error estimates for radial basis function interpolation*, J. Approx. Theory **92** (1992) 245–267.

[11] H. Pottmann and M. Eck, *Modified multiquadric methods for scattered data interpolation over a sphere*, Comput. Aided Geom. Design **7** (1990) 313–321.

[12] E. Saff and A. B. J. Kuijlaars, *Distributing many points on a sphere*, Math. Intelligencer **19** (1997) 5–11.

[13] H. Wendland, *Scattered data approximation*, Cambridge Monogr. Appl. Comput. Math., vol. 17, Cambridge Univ. Press, 2005.

[14] T. Wong, W. Luk and P. Heng, *Sampling with Hammersley and Halton points*, J. Graphics Tools **2** (1997) 9–24.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Complete triangular structures and Lie algebras

Manuel Ceballos[1], Juan Núñez[1] and Ángel F. Tenorio[2]

[1] *Departamento de Geometría y Topología , Facultad de Matemáticas, Universidad de Sevilla.*

[2] *Dpto. de Economía, Métodos Cuantitativos e H.ª Económica , Escuela Politécnica Superior, Universidad Pablo de Olavide.*

emails: mceballos@us.es, jnvaldes@us.es, aftenorio@upo.es

Abstract

In this paper we study the families of n-dimensional Lie algebras associated with a combinatorial structure made up by n vertices whose edges form a complete graph. Moreover, we characterize some properties of these structures by using Lie theory and show some examples and representations, as well as studying the type of Lie algebras associated with them in order to get its classification. Finally, an implementation of the algorithmic method used to associate Lie algebras with complete triangular structures is also shown.

Key words: Triangular configurations, combinatorial structures, Lie algebras.
MSC 2000: 17B60, 05C99, 05C90, 17B20, 17B30.

1 Introduction

In this paper, we wish to find new links between different fields of Mathematics to solve old and new problems by using different techniques as well as improving known theories and revealing new ones. In this way, we deal with two mathematical fields: Lie algebras and Graph Theory.

The study of Graph Theory is running in a high level. This theory is nowadays being used as a tool to deal with the study of other subjects due to its many applications. Indeed, the main goal of this paper is to set up a link between Graph Theory and Lie algebras in a similar way as in [2] and [3], where a mapping was defined between Lie algebras and determined combinatorial structures as follows: Every Lie algebra was represented by a particular type of combinatorial structure, whose properties could be next translated to the language of Lie algebras.

Apart from that, the research on Lie Theory is also very extended. This is in part due to their applications to Engineering, Physics and Applied Mathematics above all;

apart from its own theoretical study. However, some aspects of Lie algebras are still unknown. In fact, the classification of nilpotent and solvable Lie algebras is still an open problem, although the classification of other types of Lie algebras (like semisimple and simple ones) was got in 1890.

Another classical use of Lie Theory corresponds to the study of symmetries [4, 5]. Nowadays, symmetries are not limited to the geometrical ones of space-time, because there are other new symmetries associated with "internal" degrees of freedom of particles and fields. The study of the combinatorial structure proposed here could give us some additional information about symmetries.

The structure of this paper is the following: After recalling some known results on Lie Theory in Section 2, the next section is devoted to expound the general method used to associate a Lie algebra with a combinatorial structure. In Section 4, we analyze the particular structures dealt in this paper: Complete triangular structures. Next, Section 5 shows how to determine the subclass made up by the Lie algebras associated with those structures. Finally, Section 6 shows an implementation of the algorithmic method used to associate Lie algebras with complete triangular structures.

In our opinion, the tools introduced in this paper are useful to give a little step forward to obtain improvements on Lie and Graph Theory by firstly getting the classification of the associated combinatorial structures. It could involve an easier method to solve the classification problem than considering Lie algebras by themselves.

2 Preliminaries on Lie algebras

Some preliminary concepts on Lie algebras are recalled in this section, bearing in mind that the reader can consult [6] for a general overview. In this paper, only finite-dimensional Lie algebras over the complex number field $\mathbb{C}$ are considered.

Definition 1 *A Lie algebra $\mathfrak{g}$ is a vector space with a second bilinear composition law $([,])$ called bracket product, which satisfies two conditions: $[X,X]=0$, $\forall X \in \mathfrak{g}$ and $[[X,Y],Z]+[[Y,Z],X]+[[Z,X],Y]=0$, $\forall X,Y,Z \in \mathfrak{g}$. The last condition will be denoted by $J(X,Y,Z)=0$.*

A basis $\{e_1,\ldots,e_n\}$ of $\mathfrak{g}$ is characterized by its structure constants or Maurer-Cartan constants: $[e_i,e_j]=\sum c_{i,j}^h e_h$, for $1 \leq i,j \leq n$.

Note 1 *The first condition in Definition 1 and the bilinear property of the bracket imply the skew-symmetry*

$$[X,Y]=-[Y,X] \quad \forall\, X,Y \in \mathfrak{g}.$$

In general, there exist three different types of Lie algebras: solvable ones, semisimple ones and direct sums of two Lie algebras, one of each two previous types.

Definition 2 *A Lie algebra $\mathfrak{g}$ is semisimple if $\mathfrak{g}$ does not contain any proper abelian ideal and a simple Lie algebra is a non-abelian Lie algebra with no non-trivial ideals.*

 ISBN: 978-84-612-9727-6

Definition 3 *The* commutator central series *and the* lower central series *of a Lie algebra* $\mathfrak{g}$ *are, respectively,*

$$\mathcal{C}_1(\mathfrak{g}) = \mathfrak{g},\ \mathcal{C}_2(\mathfrak{g}) = [\mathfrak{g},\mathfrak{g}],\ \ldots,\ \mathcal{C}_k(\mathfrak{g}) = [\mathcal{C}_{k-1}(\mathfrak{g}),\mathcal{C}_{k-1}(\mathfrak{g})],\ \ldots \quad \text{and}$$

$$\mathcal{C}^1(\mathfrak{g}) = \mathfrak{g},\ \mathcal{C}^2(\mathfrak{g}) = [\mathfrak{g},\mathfrak{g}],\ \ldots,\ \mathcal{C}^k(\mathfrak{g}) = [\mathcal{C}^{k-1}(\mathfrak{g}),\mathfrak{g}],\ \ldots$$

Hence, $\mathfrak{g}$ *is called* solvable *if there exists* $m \in \mathbb{N}$ *such that* $\mathcal{C}_m(\mathfrak{g}) \equiv \{0\}$ *and* nilpotent *if there exists* $m \in \mathbb{N}$ *such that* $\mathcal{C}^m(\mathfrak{g}) \equiv \{0\}$.

Note 2 *Let us note that, from both definitions, every nilpotent Lie algebra is solvable; because* $\mathcal{C}_i(\mathfrak{g}) \subseteq \mathcal{C}^i(\mathfrak{g})$, *for* $i = 1, \ldots, n$.

3 Associating triangular configurations with Lie algebras

Given a n-dimensional Lie algebra $\mathfrak{g}$ and a basis $\mathcal{B} = \{e_1, \ldots, e_n\}$ of $\mathfrak{g}$, the law of $\mathfrak{g}$ with respect to the basis $\mathcal{B}$ is given by $[e_i, e_j] = \sum_{k=1}^{n} c_{i,j}^k e_k$, where $c_{i,j}^k$ are the structure constants for $\mathcal{B}$. The pair $(\mathfrak{g}, \mathcal{B})$ can be associated with a combinatorial structure by using the following method, which was introduced in [2]:

a) For each $e_i \in \mathcal{B}$, one vertex labeled as index i is drawn.

b) Given three vertices $i < j < k$, the full triangle can be drawn. The weight assigned to the edges which connect vertices i, j and k are $c_{i,j}^k$, $c_{j,k}^i$, and $c_{i,k}^j$. An example can be seen in Figure 1.

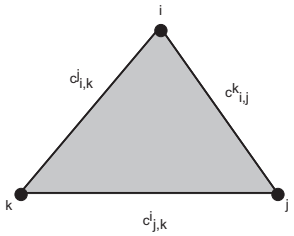


Figure 1: Full triangle.

From here on, we will suppose that:

b1) If $c_{i,j}^k = c_{j,k}^i = c_{i,k}^j = 0$, then the triangle is not drawn.

b2) If some structure constant is zero, the corresponding edge will be drawn by using a discontinuous line (called *ghost edge*).

b3) If two triangles of vertices $\{i, j, k\}$ and $\{i, j, l\}$ with $1 \leq i < j < k < l \leq n$ satisfy $c_{i,j}^k = c_{i,j}^l$, then the edge between the vertices i and j is shared.

 ISBN: 978-84-612-9727-6

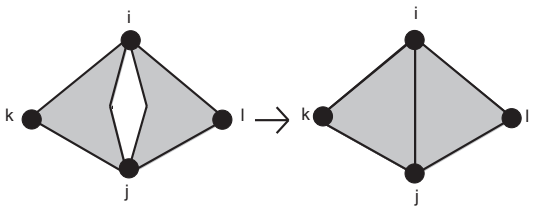


Figure 2: Triangles with a shared edge.

Consequently, every Lie algebra with a selected basis is associated with a triangular configuration. However, it is necessary to point out that this association depends on the selected basis.

Let us note that, if there appears an insolated vertex, the corresponding vector in $\mathcal{B}$ belongs to the center of $\mathfrak{g}$. The converse of this result is false; i.e. the center of $\mathfrak{g}$ can correspond to non-insolated vertices in the triangular configuration associated with $\mathfrak{g}$.

Let us recall a lemma that will be used in following sections and can be seen in [1].

Lemma 1 *Fixed and given a triangular configuration with n vertices having all of them degree r, the number of edges in this configuration is equal to* $\frac{nr}{2}$.

Definition 4 *Given three natural numbers* $i < j < k$ *corresponding to three vertices, if there exists a non-zero structure constant associated with these vertices, a full triangle is drawn. In this situation, if a structure constant is zero, a discontinuous line is drawn and called* ghost edge.

Definition 5 *The* degree d_i *or* $deg(i)$ *of a vertex i in a triangular configuration is the number of edges incident with i. In this definition, we are assuming both types of edges (including ghost edges).*

Consequently, when studying if two vertices are adjacent or if a vertex is incident with an edge, both types of edges are considered. Therefore, we have the following

Definition 6 *A graph G is said to be* complete *if each vertex in G is adjacent with all the remaining vertices by full or ghost edges.*

4 Complete triangular structures associated with Lie algebra

A complete triangular structure T is defined as a complete graph G with full triangles between their weighted and non-directed edges. It is possible that there exist several ghost edges in the structure T. We are going to study under what conditions this type of structure can be associated with a Lie algebra.

Let T be a complete triangular structure made up by n vertices. We label all the vertices consecutively by 1, 2, 3, ..., n, following the positive counterclockwise

 ISBN: 978-84-612-9727-6

orientation. Fixed and given two vertices i and j such that $1 \leq i < j \leq n$, the weight of the edge ij will be denoted by $c_{i,j}$ (i.e. the structure constant for the bracket $[e_i, e_j]$). Under this assumption, we can define a vector space V endowed with a basis $\{e_1, \ldots, e_n\}$ where e_i corresponds to the vertex i of T and whose law is given by the following nonzero brackets

$$[e_i, e_j] = c_{i,j}(\sum_{h \neq i,j} e_h). \tag{1}$$

Now we distinguish several cases, depending on the number of vertices in the complete triangular structure.

Case $n = 3$**:** The unique complete triangular given by three vertices is the one represented in Figure 3. The nonzero brackets are the following

$$[e_1, e_2] = c_{1,2}\, e_3, \quad [e_1, e_3] = c_{1,3}\, e_2, \quad [e_2, e_3] = c_{2,3}\, e_1.$$

In this case, $J(e_1, e_2, e_3) = 0$ and, hence, this structure is associated to a 3-dimensional Lie algebra.

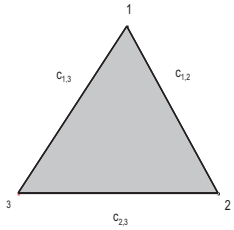


Figure 3: Complete triangular structure with 3 vertices.

Case $n \geq 4$**:** We must solve the system of equations given by all the Jacobi identities. To do it, we study this identity for three arbitrary vertices i, j and k verifying $1 \leq i < j < k \leq n$. When imposing $J(e_i, e_j, e_k) = 0$ and according to the law defined in (1), the following equation is obtained

$$(-c_{i,j} \sum_{p \neq i,j,k} c_{k,p} + c_{i,k} \sum_{q \neq i,j,k} c_{j,q})e_i + (-c_{i,j} \sum_{p \neq i,j,k} c_{k,p} - c_{j,k} \sum_{q \neq i,j,k} c_{i,q})e_j +$$
$$(c_{i,k} \sum_{p \neq i,j,k} c_{j,p} - c_{j,k} \sum_{q \neq i,j,k} c_{i,q})e_k + \sum_{l \neq i,j,k} (-c_{i,j} \sum_{p \neq i,j,k,l} c_{k,p} - c_{j,k} \sum_{q \neq i,j,k,l} c_{i,q} + c_{i,k} \sum_{r \neq i,j,k,l} c_{j,r})e_l = 0.$$

When we solve the system, we will obtain that some structure constants are equal to zero. There will be several solutions that will be classified according to the Lie algebra associated with. In order to choose a particular solution, the remaining constants will be nonzero, determining a weighted full edge. Therefore, a solution is chosen in a natural way.

Following this reasoning and using the previous notation, we can set the following

Theorem 1 *Let us consider a complete triangular structure T made up by $n \geq 4$ vertices. Then, T is associated with a n-dimensional Lie algebra if and only if one of the followings statement holds:*

- *There exists a unique vertex i such that $c_{i,j}$ is nonzero, for $1 \leq j \leq n$ and $i \neq j$. Besides, the remaining edges are ghost.*
- *There exist three vertices i, j and k such that $c_{i,j}, c_{j,k}$ and $c_{i,k}$ are nonzero and the remaining edges are ghost.*

Proof:

Let T be a complete triangular configuration made up by n vertices. We construct a vector space associated with T, according to the method expounded previously. By imposing the Jacobi identities for the vectors associated with the vertices i, j and k, the following system of equations is obtained for the structure constants from the vectors e_i, e_j and e_k

$$\left\{\begin{array}{l} -c_{i,j}\sum_{p\neq i,j,k} c_{k,p} + c_{i,k}\sum_{q\neq i,j,k} c_{j,q} = 0, \\ -c_{i,j}\sum_{p\neq i,j,k} c_{k,p} - c_{j,k}\sum_{q\neq i,j,k} c_{i,q} = 0, \\ c_{i,k}\sum_{p\neq i,j,k} c_{j,p} - c_{j,k}\sum_{q\neq i,j,k} c_{i,q} = 0. \end{array}\right.$$

Now, by introducing the notation

$$x = c_{i,j}\sum_{p\neq i,j,k} c_{k,p}, \quad y = c_{i,k}\sum_{q\neq i,j,k} c_{j,q}, \quad z = c_{j,k}\sum_{r\neq i,j,k} c_{i,r},$$

the following system is obtained

$$\left\{\begin{array}{l} -x + y = 0, \\ -x - z = 0, \\ y - z = 0. \end{array}\right.$$

which is homogeneous with a unique solution given by $(x, y, z) = (0, 0, 0)$. The equation for each vector $e_l \in \mathcal{B}$ (where $l \neq i, j, k$) is

$$-c_{i,j}\sum_{p\neq i,j,k,l} c_{k,p} - c_{j,k}\sum_{q\neq i,j,k,l} c_{i,q} + c_{i,k}\sum_{r\neq i,j,k,l} c_{j,r} = 0,$$

which is equivalent to $-(x - c_{i,j}c_{k,l}) - (z - c_{j,k}c_{i,l}) + (y - c_{i,k}c_{j,l}) = 0$ and, therefore, is equivalent to

$$c_{i,j}c_{k,l} + c_{j,k}c_{i,l} - c_{i,k}c_{j,l} = 0.$$

Consequently, we have to solve the following system

$$\left\{\begin{array}{l} c_{i,j}\sum_{p\neq i,j,k} c_{k,p} = 0,\ c_{i,k}\sum_{q\neq i,j,k} c_{j,q} = 0,\ c_{j,k}\sum_{r\neq i,j,k} c_{i,r} = 0, \\ c_{i,j}c_{k,l} + c_{j,k}c_{i,l} - c_{i,k}c_{j,l} = 0,\ \forall\, l \neq i, j, k. \end{array}\right.$$

Let us note that there exists a solution of this system if and only if the coefficients $c_{i,j}$, $c_{i,k}$ and $c_{j,k}$ are nonzero or there exists $\alpha \in \mathbb{N}$ verifying that $c_{\alpha,\beta}$ is nonzero

 ISBN: 978-84-612-9727-6

for $\beta \neq \alpha$. In the first case, it is easy to prove that $c_{i,l}$, $c_{j,l}$ and $c_{k,l}$ are zero for $l \neq i, j, k$. In the second case, the remaining structure constants are zero.

Consequently, we have the following

Corollary 1 *Let T be a complete triangular structure with $n \geq 4$ vertices. Then, T is associated with a n-dimensional Lie algebra if and only if one of the followings statement is satisfied:*

- *There are only $n-1$ full edges and all of them are incident with a unique vertex i.*
- *There are $n-3$ vertices incident only with ghost edges and a unique full triangle with full edges.*

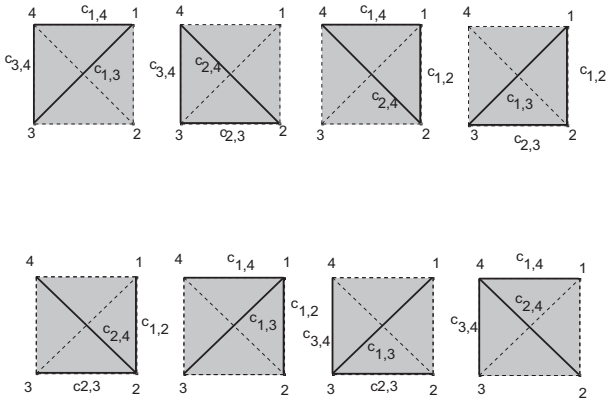
Now, we study a property for the complete triangular structures. More concretely, we are interested in computing the complementary of such a structure.

Definition 7 *Fixed and given a complete triangular structure T, its complement structure $\overline{T}$ is a complete triangular structure with the same vertices of T and verifying the following property: If the edge uv is full in T, the corresponding edge is ghost in $\overline{T}$. Analogously, if the edge pq is ghost in T, the corresponding edge in $\overline{T}$ is full.*

Proposition 1 *Let T be a complete triangular structure formed by $n = 4$ vertices. If T is associated with a Lie algebra, then $\overline{T}$ is associated with another Lie algebra. However, this is false for $n \geq 5$.*

Proof:

Let us consider a complete triangular structure T, formed by $n = 4$ vertices. By applying Theorem 1, we can obtain 8 different Lie algebras, associated with T, represented as follows



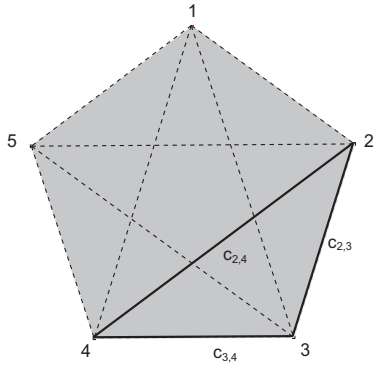
Each representation is the complement structure of the one placed above. In this way, we have four pairs of solutions.

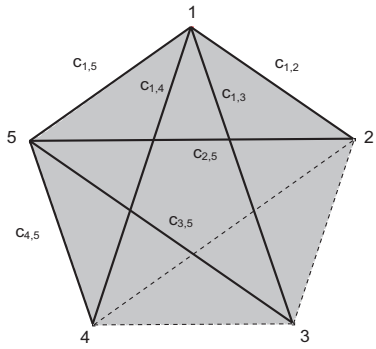
 ISBN: 978-84-612-9727-6

Now, we prove that the statement is false for $n \geq 5$. This is due to the fact that both solutions given in Theorem 1 are not complementary. □

We show a particular example:

Example 1 *Given a complete triangular structure T with 5 vertices, we fix a particular solution of the system obtained. The solution and its complementary are represented here*





In this case, from $J(e_1, e_2, e_3) = 0$, we deduce

$(-c_{1,2}c_{3,4} - c_{1,2}c_{3,5} + c_{1,3}c_{2,4} + c_{1,3}c_{2,5})e_1 + (-c_{1,2}c_{3,4} - c_{3,5}c_{1,2})e_2 + (c_{1,3}c_{2,4} + c_{2,5}c_{1,3})e_3 + (-c_{1,2}c_{3,5} + c_{1,3}c_{2,5})e_4 + (-c_{1,2}c_{3,4} + c_{1,3}c_{2,4})e_5 = 0.$

Since the structure constants above are nonzero, the following system

$$\begin{cases} -c_{1,2}c_{3,4} - c_{1,2}c_{3,5} + c_{1,3}c_{2,4} + c_{1,3}c_{2,5} = 0, \ -c_{1,2}c_{3,4} - c_{3,5}c_{1,2} = 0, \\ c_{1,3}c_{2,4} + c_{2,5}c_{1,3} = 0, -c_{1,2}c_{3,5} + c_{1,3}c_{2,5} = 0, \\ -c_{1,2}c_{3,4} + c_{1,3}c_{2,4} = 0, c_{4,5}c_{2,3} - c_{3,5}c_{2,4} = 0, \\ c_{4,5}c_{1,3} - c_{3,5}c_{1,4} = 0, -c_{3,5}c_{1,4} - c_{3,5}c_{2,4} = 0, \\ c_{4,5}c_{1,3} + c_{4,5}c_{2,3} = 0, c_{4,5}c_{1,3} + c_{4,5}c_{2,3} - c_{3,5}c_{1,4} - c_{3,5}c_{2,4} = 0. \end{cases}$$

has no solutions.

 ISBN: 978-84-612-9727-6

5 Types of Lie algebra and complete triangular structures

Proposition 2 *A complete triangular structure T made up by $n = 3$ vertices is associated to a simple Lie algebra.*

Proof:

Let T be a complete triangular structure with $n = 3$ vertices (see Figure 3). We will denote by $\{e_1, e_2, e_3\}$ the basis of the 3-dimensional space associated with T. We have the following brackets

$$[e_1, e_2] = c_{1,2}e_3, \quad [e_1, e_3] = c_{1,3}e_2, \quad [e_2, e_3] = c_{2,3}e_1.$$

It is easy to check that $J(e_1, e_2, e_3) = 0$. So T is associated with a 3-dimensional Lie algebra, which does not contain any proper abelian ideal. Then, this Lie algebra is simple. □

Theorem 2 *Let us consider a complete triangular structure T made up by $n \geq 4$ vertices verifying that there are only $n - 1$ full edges and all of them are incident with a unique vertex i. Then, T is associated with a non-nilpotent solvable Lie algebra.*

Proof:

Let T be a complete triangular structure formed by $n \geq 4$ vertices. We know that there exists a unique vertex i verifying that there are only $n - 1$ full edges and all of them are incident with i. In virtue of Theorem 1, T is associated with a n-dimensional Lie algebra $\mathfrak{g}$ with basis $\{e_1, \ldots, e_n\}$ and nonzero brackets

$$[e_i, e_j] = c_{i,j} \sum_{h \neq i,j} e_h, \quad \text{for } j \neq i.$$

In this way, the commutator central series is

$$\mathcal{C}_1(\mathfrak{g}) = \mathfrak{g},\ \mathcal{C}_2(\mathfrak{g}) = \langle \sum_{h \neq i,1} e_h, \ldots, \sum_{h \neq i,i-1} e_h, \sum_{h \neq i,i+1} e_h, \ldots, \sum_{h \neq i,n} e_h \rangle,\ \mathcal{C}_3(\mathfrak{g}) = \{0\},$$

and the lower central series is

$$\mathcal{C}^1(\mathfrak{g}) = \mathfrak{g},\ \mathcal{C}^2(\mathfrak{g}) = \langle \sum_{h \neq i,1} e_h, \ldots, \sum_{h \neq i,i-1} e_h, \sum_{h \neq i,i+1} e_h, \ldots, \sum_{h \neq i,n} e_h \rangle,\ \mathcal{C}^3(\mathfrak{g}) = \mathcal{C}^2(\mathfrak{g}).$$

Consequently, $\mathfrak{g}$ is a non-nilpotent solvable Lie algebra. □

Theorem 3 *Let us consider a complete triangular structure T made up by $n \geq 4$ vertices verifying that $n - 3$ vertices are only incident with ghost edges and there is a unique full triangle with full edges. Then, the structure T is associated with a non-solvable, non-semisimple Lie algebra.*

 ISBN: 978-84-612-9727-6

Proof:

Let T be a complete triangular structure formed by $n \geq 4$ vertices. We know that $n-3$ vertices are only incident with ghost edges and there is a unique full triangle made up by full edges. By applying Theorem 1, T is associated with a n-dimensional Lie algebra $\mathfrak{g}$ with a basis $\{e_1, \ldots, e_n\}$ verifying that there exist three natural numbers i, j and k such that

$$[e_i, e_j] = c_{i,j} \sum_{p \neq i,j} e_p, \ [e_i, e_k] = c_{i,k} \sum_{q \neq i,k} e_p, \ [e_j, e_k] = c_{j,k} \sum_{r \neq j,k} e_r$$

and the remaining brackets are zero.

In this way, the commutator central series is the following

$$\mathcal{C}_1(\mathfrak{g}) = \mathfrak{g}, \ \mathcal{C}_2(\mathfrak{g}) = \left\langle \sum_{p \neq i,j} e_p, \sum_{q \neq i,k} e_p, \sum_{r \neq j,k} e_r \right\rangle, \ \mathcal{C}_3(\mathfrak{g}) = \mathcal{C}_2(\mathfrak{g}),$$

and, consequently, $\mathfrak{g}$ is not solvable.

Besides, the Lie algebra $\mathfrak{g}$ contains proper abelian ideals, like the ideal $I = \langle \{e_l\}_{l \neq i,j,k} \rangle$. So, $\mathfrak{g}$ cannot be semisimple. □

6 Algorithm to compute the Lie algebra associated with a complete triangular structure

Let T be a complete triangular structure with n vertices. In this section, an algorithmic method is implemented for computing the complete triangular structure associated with a given n-dimensional Lie algebra. This algorithmic method consists in the following steps:

1. Implementing a subroutine which computes the summands necessary to generate the equations from the corresponding Jacobi identities.

2. Programming a subroutine to determine the coefficients of each basis vector, from a Jacobi identity formed by three arbitrary vectors.

3. Computing the complete triangular structure associated with a n-dimensional Lie algebra.

All the vertices are consecutively labelled by $1, 2, 3, \ldots, n$ and a vector space V is defined by considering a basis $\mathcal{B} = \{e_1, e_2, \ldots, e_n\}$, where the vector e_i corresponds to the vertex i. Now the implementation of the algorithm is shown by using the symbolic computation package MAPLE, giving a step-by-step explanation of this implementation.

First, the library `linalg` is load for commands related to Linear Algebra, since Lie algebras are vector spaces endowed with a second inner structure: The Lie bracket.

 ISBN: 978-84-612-9727-6

Besides, the library `combinat` is also loaded to apply commands related to Combinatorial Algebra. To implement the algorithm, several subroutines have been programmed to be called by the main routine.

The first subroutine is called `sum` and receives six natural numbers as its inputs. The four first numbers and the last one correspond to the subindexes of five basis vector in $\mathcal{B}$, the other number is the dimension of the Lie algebra. The output of this subroutine is a summand which is necessary to generate the expressions coming from the Jacobi identities in order to compute the structure constants and define the Lie algebra associated with the complete triangular structure. When implementing this subroutine, conditional sentences are used. To settle the skew-symmetry of the structure constants, conditional sentences are also considered.

```
> sum:=proc(i,j,k,l,n,y)
> local x;
> x:=0;
> for p from 1 to n do
> if p<>i then
> if p<>j then if p<>k then
> if p<>l then if y<p then  x:=x + c[y,p]; else x:=x-c[p,y];
> end if; end if; end if; end if; end if; end do;
> return x;
> end proc:
```

For the second subroutine, `eq`, the input is formed by four arguments, namely: The dimension `n` of $\mathfrak{g}$ and three subindexes, `i`, `j` and `k`, corresponding to the basis vectors of the Jacobi identity to be computed. First, a set `S` is defined as a local variable. Obviously, the subroutine `sum` is called by `eq` to compute the equations obtained from the Jacobi identity. In this way, the output for the subroutine `eq` is precisely the set `S` so obtained.

```
> eq:=proc(n,i,j,k)
> local S; S:={};
> S:={op(S),-c[i,j]*sum(i,j,k,k,n,k) + c[i,k]*sum(i,j,k,k,n,j)=0};
> S:={op(S),-c[i,j]*sum(i,j,k,k,n,k) - c[j,k]*sum(i,j,k,k,n,i)=0};
> S:={op(S),c[i,k]*sum(i,j,k,k,n,j)-c[j,k]*sum(i,j,k,k,n,i)=0};
> for x from 1 to n do
> if x<>i then if x<>j then if x<>k then
> S:={op(S),-c[i,j]*sum(i,j,k,x,n,k)-c[j,k]*sum(i,j,k,x,n,i)
  +c[i,k]*sum(i,j,k,x,n,j)=0};
> end if; end if; end if; end do;
> return S;
> end proc:
```

The routine `sys` receives as input a natural number corresponding to the dimension `n`. This subroutine computes the solution of the system of equations generated by the

 ISBN: 978-84-612-9727-6

subroutine `eq` in order to obtain the Lie algebra of dimension `n` associated with a complete triangular structure of `n` vertices. To implement this subroutine, two local variables `L` and `T` have been defined, where `L` is a list and `T` is a set. Next, these variables are explained: the list `L` is formed by all the combinations of the first `n` natural numbers taken three at a time. Finally, the set `T` is used to record the equations of the system to be solved for the routine. Now, we show the implementation of the subroutine `sys`:

```
> sys:=proc(n)
> local L,T;
> L:=choose(n,3); T:={};
> for p from 1 to nops(L) do
> T:={op(T),op(eq(n,L[p][1],L[p][2],L[p][3]))};
> end do;
> return solve(T);
> end proc:
```

References

[1] F. Harary, *Graph Theory*, Addison-Wesley, Reading, 1969.

[2] A. Carriazo, L.M. Fernández and J. Núñez, *Combinatorial structures associated with Lie algebras of finite dimension* , Linear Algebra and Applications **389** (2004), 43–61.

[3] L.M. Fernández and L. Martín-Martínez , *Lie algebras associated with triangular configurations*, Linear Algebra and Applications **407** (2005), 43–63.

[4] F. Iachello, *Lie algebras and Applications*, Lecture Notes in Physics **708**, Springer, Berlin, 2006.

[5] P.J. Olver, *Applications of Lie Groups to Differential Equations*, Springer, New York, 1986.

[6] V.S. Varadarajan, *Lie Groups, Lie Algebras and Their Representations*, Springer, New York, 1984.

 ISBN: 978-84-612-9727-6

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Reconstruction of the discontinuities in the parameters of quasi-linear elliptic problems

Ivan Cimrák[1]

[1] *NaM*^{2} *Research Group, Department of Mathematical Analysis, Ghent University, Galglaan 2, B-9000, Belgium*

emails: ivan.cimrak@ugent.be

Abstract

We solve an inverse problem of shape determination. In the physical setting, the computational domain contains an unknown object with different physical properties. The problem can be described by a quasi-linear elliptic PDE for the state variable where the parameter function has a discontinuity across the boundary of the unknown object. This discontinuity defines the object.

We aim at determination of the shape of this object from the measurements of the state variable. We employ the artificial time and we let the boundary of the unknown object evolve in such a matter that the cost functional based on the measurements decreases.

We study the sensitivity of the cost functional on the artificial time and we suggest the velocity function according to which the boundary of the object should evolve to decrease the cost functional. Finally, we suggest a generic algorithm that minimizes the cost functional. It is well-suited for the level set method.

Key words: sensitivity analysis, adjoint problem, shape optimization, level set method

1 Introduction

Consider the static case of the magnetic vector potential formulation of the Maxwell equations with planar symmetry. The only one nonzero component of the vector potential is denoted by A. In this formulation with planar symmetry, the magnetic induction has two nonzero components and can be expressed as $\mathbf{B} = (\frac{\partial A}{\partial y}, -\frac{\partial A}{\partial x}, 0)^{\mathrm{T}}$. Let $\Omega \in \mathbb{R}^2$ be a bounded object with the permeability μ and suppose that the magnetic potential vanishes on the boundary. Then A satisfies the following quasi-linear elliptic PDE which is a special case of the Maxwell equations

$$\nabla \cdot (\nu(\mathbf{x}, |\nabla A(\mathbf{x})|^2)\nabla A(\mathbf{x})) = J(\mathbf{x}) \quad \text{in } \Omega, \qquad A = 0 \quad \text{on } \partial\Omega. \tag{1}$$

where

$$\nu(\mathbf{x}, s) = \frac{1}{\mu(\mathbf{x}, s)},$$

and $J(\mathbf{x})$ represents the current density. For linear materials, the permeability is a scalar function leading to a simpler linear elliptic PDE. The same linear system describes the problem of EIT which has been thoroughly studied by many authors, we mention only few of them [1, 2, 5]. In EIT, one considers the conductivity instead of the permeability.

We consider more general case of nonlinear materials, where the permeability depends on the intensity of the magnetic induction $\mathbf{B}$. In the case of planar symmetry, we have $|\mathbf{B}| = |\nabla A|$. The permeability is thus a function depending on the location vector $\mathbf{x}$ and on the squared modulus $|\nabla A|^2$.

Physical systems often consist of several parts, each having different permeability. The permeability function $\mu(\mathbf{x}, s)$ defined over the whole Ω thus has discontinuities over the borders between those parts. We model the case where Ω contains one region represented by an open set D, on which the permeability has different properties than on the rest of the domain. For illustration see Figure 1. Set D can be a union of several open sets.

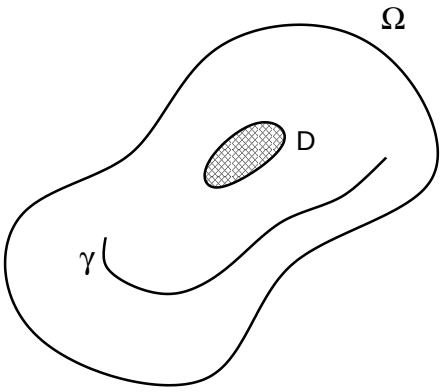


Figure 1: Model with piecewise permeability. The permeability is supposed to have a discontinuity across ∂D. The measurements of the magnetic induction are available along the line γ. After introduction of the artificial time variable t, the domain D will depend on t and will be denoted by D_t.

From now on we will work with the function ν only. Let us suppose that $\nu(\mathbf{x}, s)$ is of the following form

$$\nu(\mathbf{x}, s) = \begin{cases} \nu_1(s) & \text{for } \mathbf{x} \in D, \\ \nu_2(s) & \text{for } \mathbf{x} \in \Omega \setminus D. \end{cases} \tag{2}$$

Note, that the space dependence is expressed only by the splitting of the domain. Both functions ν_1, ν_2 are differentiable with well-defined derivatives ν_1', ν_2'

 ISBN: 978-84-612-9727-6

2 Preliminaries

Since the coefficient ν in the quasi-linear elliptic equation is discontinuous, we will further use weak formulations. For this we introduce some notations. Denote the scalar product in $L^2(\Omega)$ by $(\cdot,\cdot)$. The scalar product in $(L^2(\Omega))^3$ will be denoted by the same symbol since it is always obvious from the context which scalar product is meant. The norm in $L^2(\Omega)$ will be denoted by $\|\cdot\|_2$. The scalar product of two vectors $\mathbf{u},\mathbf{v}$ in $\mathbb{R}^d$ will be denoted by $\mathbf{u}\cdot\mathbf{v}$. The available measurements are the values of magnetic induction along the 1D line γ, therefore we introduce the notation

$$(\mathbf{u},\mathbf{v})_\gamma = \int_\gamma \mathbf{u}\cdot\mathbf{v},$$

for the line integral along γ. Analogous we denote the line integral over the boundary ∂D by

$$(\mathbf{u},\mathbf{v})_{\partial D} = \int_{\partial D} \mathbf{u}\cdot\mathbf{v}.$$

3 Inverse problem

We are interested in the reconstruction of the domain D for given ν_1,ν_2 and measurements $\mathbf{B}_0$ of magnetic induction along the given line γ. We already mentioned that $\mathbf{B} = (\frac{\partial A}{\partial y}, -\frac{\partial A}{\partial x}, 0)^{\mathrm{T}}$. We transform the three-dimensional measurements vector $\mathbf{B}_0 = (B_0^1, B_0^2, 0)^{\mathrm{T}}$ to a two-dimensional vector $\mathbf{b}_0$ by the following transform

$$\mathbf{b}_0 = (-B_0^2, B_0^1),$$

and in our inverse problem we try to reconstruct the domain D such that the gradient of the state variable ∇A fits the vector field $\mathbf{b}_0$.

We approach this problem by introducing an artificial time variable t and we let D evolve in time. Since D will not be static anymore we redefine D as D_t to emphasize the dependence on the time t. Correspondingly, the dependence on t will appear also in the potential. The governing equation for weak solution will be

$$(\nu(|\nabla A(t)|^2)\nabla A(t), \nabla\phi) = (J,\phi) \quad \text{for all } \phi \in H_0^1(\Omega), \tag{3}$$

where $\nu(\mathbf{x},s)$ is defined as in (2) with D_t instead of D. Notice that for brevity we omit the space variable $\mathbf{x}$ in the equations. Using the characteristic function of D_t we can express $\nu(\mathbf{x},s)$ as

$$\nu(\mathbf{x},s) = \chi_D(\mathbf{x},t)\nu_1(s) + (1-\chi_D(\mathbf{x},t))\nu_2(s), \tag{4}$$

where $\chi_D(\mathbf{x},t) = 1$ for $\mathbf{x}\in D_t$ and $\chi_D(\mathbf{x},t) = 0$ for $\mathbf{x}\in\Omega\setminus D_t$.

D_t should evolve in such a way that the corresponding solution $A(\mathbf{x},t)$ approaches the available measurements. To measure the fidelity of the current solution to the measurements we define the cost function

$$F(D_t) = \frac{1}{2}\int_\gamma |\nabla A(t) - \mathbf{b}_0|^2,$$

 ISBN: 978-84-612-9727-6

where $\mathbf{b}_0$ is the vector transformed from the measurements of the magnetic induction along the line γ.

Next step is to define a velocity function $\mathbf{v}(\mathbf{x},t)$ such that if D_t evolves according to this velocity, the functional decreases, that is

$$\frac{d}{dt}[F(D_t)] < 0.$$

Let us point out that only the normal component of the velocity is important for us, since we are interested in the shape of D_t only. Therefore the velocity function can be written as

$$\mathbf{v}(\mathbf{x},t) = v(\mathbf{x},t)\mathbf{n}(\mathbf{x},t),$$

where $\mathbf{n}(\mathbf{x},t)$ is the outward normal unit vector to ∂D_t.

4 Sensitivity analysis

Formal differentiation of F with respect to t gives

$$\frac{d}{dt}[F(D_t)] = \Big(\nabla\delta A, \nabla A(t) - \mathbf{b}_0\Big)_\gamma, \tag{5}$$

where $\delta A := \frac{d}{dt}[A(t)]$ denotes the Eulerian derivative of $A(t)$ [4]. To know the response of F to small changes in t, one needs to know the response of the potential A to small changes in t. Let us formally differentiate (3) with respect to t. Since $J(\mathbf{x})$ is t-independent, we obtain

$$\Big(\frac{d}{dt}[\nu(|\nabla A(t)|^2)]\nabla A(t), \nabla\phi\Big) + \Big(\nu(|\nabla A(t)|^2)\nabla\delta A, \nabla\phi\Big) = 0. \tag{6}$$

To handle the first term we need to differentiate ν. Using (4) we obtain

$$\begin{aligned}\frac{d}{dt}[\nu(|\nabla A(t)|^2)] &= \frac{d}{dt}[\chi_D(t)](\nu_1(|\nabla A(t)|^2) - \nu_2(|\nabla A(t)|^2)) \\ &\quad +2\chi_D(t)\nu_1'(|\nabla A(t)|^2)\nabla\delta A\cdot\nabla A(t) \\ &\quad +2\Big[1-\chi_D(t)\Big]\nu_2'(|\nabla A(t)|^2)\nabla\delta A\cdot\nabla A(t).\end{aligned} \tag{7}$$

To understand the expression $\frac{d}{dt}[\chi_D(t)]$ we need to stress that the boundary moves with the velocity $\mathbf{v}(\mathbf{x},t)$. Then, according to [4] we have

$$\frac{d}{dt}[\chi_D(t)] = \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t)\delta_{\partial D_t}(\mathbf{x}) = v(\mathbf{x},t)\delta_{\partial D_t}(\mathbf{x}), \tag{8}$$

where $\delta_{\partial D_t}$ is the 2-dimensional Dirac delta distribution concentrated on the boundary ∂D_t of the shape D_t.

The last two terms in (7) are functions defined either on D or on its complement $\Omega\setminus D$. To simplify the expressions let us define

$$\nu^{diff}(\mathbf{x},s) = \begin{cases}\nu_1'(s) & \text{for } \mathbf{x}\in D,\\ \nu_2'(s) & \text{for } \mathbf{x}\in\Omega\setminus D.\end{cases} \tag{9}$$

 ISBN: 978-84-612-9727-6

Using our preliminary results (7), (8) and (9) in (6) we get the following sensitivity equation for δA

$$\begin{aligned}&\left((\nu_1(|\nabla A(t)|^2) - \nu_2(|\nabla A(t)|^2))v(\mathbf{x},t)\ \nabla A(t), \nabla\phi\right)_{\partial D_t}\\ &= 2\left(\nu^{diff}(|\nabla A(t)|^2)\nabla\delta A\cdot\nabla A(t)\ \nabla A(t), \nabla\phi\right) + \left(\nu(|\nabla A(t)|^2)\nabla\delta A, \nabla\phi\right). \quad (10)\end{aligned}$$

Solving the previous PDE for given $A(t)$ gives us the Eulerian derivative $\delta A(t)$. Then, plugging this derivative into (5) gives us the derivative of F with respect to t. However, we still do not know what $\mathbf{v}(\mathbf{x},t)$ to use in order to get the derivative of F negative.

5 Adjoint problem

We use the Adjoint Variable Method [3, 6] to explicitly express $\frac{d}{dt}[F]$. This method introduces the adjoint variable, we denote it by b. It is a solution of the following adjoint problem

$$\begin{aligned}&\left(\nabla A(t) - \mathbf{b}_0, \nabla\psi\right)_\gamma\\ &= 2\left(\nu^{diff}(|\nabla A(t)|^2)\nabla A(t)\cdot\nabla b\ \nabla A(t), \nabla\psi\right) + \left(\nu(|\nabla A(t)|^2)\nabla b, \nabla\psi\right), \quad (11)\end{aligned}$$

for all $\psi \in H_0^1(\Omega)$. Notice that the operator part of this PDE is identical to the operator part of the sensitivity equation (10).

If we take $\phi = b$ in (10) and $\psi = \delta A$ in (11), we end up with the equality of the operator parts of both PDEs. This results in the equality of the functional parts

$$\frac{d}{dt}[F(D_t)] = \left(\nabla A(t) - \mathbf{b}_0, \nabla\delta A\right)_\gamma = \left((\nu_1(|\nabla A(t)|^2) - \nu_2(|\nabla A(t)|^2))v(\mathbf{x},t)\ \nabla A(t), \nabla b\right)_{\partial D_t}.$$

Now we see what is the velocity to get $\frac{d}{dt}[F(D_t)]$ negative. Setting

$$v(\mathbf{x},t) = -(\nu_1(|\nabla A(t)|^2) - \nu_2(|\nabla A(t)|^2))\nabla A(t)\cdot\nabla b, \quad (12)$$

we obtain

$$\frac{d}{dt}[F(D_t)] = -\left\|(\nu_1(|\nabla A(t)|^2) - \nu_2(|\nabla A(t)|^2))\nabla A(t)\cdot\nabla b\right\|^2_{\partial D_t} < 0.$$

6 Algorithm

Knowledge of the F-decreasing velocity $\mathbf{v}(\mathbf{x},t)$ entitles us to design an iterative algorithm that determine the shape D of the unknown object. The algorithm gives us a sequence D_i of the shapes that approximate the evolution of D_t. We start from an initial guess D_0 that can be a result of the more robust but less accurate parametric optimization.

1. Set $i = 0$, D_0 is given and set $F_{-1} = \infty$.

 ISBN: 978-84-612-9727-6

2. Set $D_t = D_i$ and compute $A(t)$ from (3). Evaluate $F_i = F(D_t)$. If the stopping criterion (e.g. $F_i - F_{i-1} > -\epsilon$, for some small positive ϵ) is fulfilled then stop.

3. For known D_t and $A(t)$ compute b from (11) and using (12) evaluate $\mathbf{v}(\mathbf{x}, t)$.

4. Determine the time step $\lambda > 0$ and move the boundary of D_i by $\lambda\mathbf{v}(\mathbf{x}, t)$. Set

$$D_{i+1} = \{\mathbf{x} + \lambda\mathbf{v}(\mathbf{x}, t), \mathbf{x} \in \partial D_i\},$$

shift the index $i := i + 1$ and go to step 2.

The presented algorithm is a general prototype for a concrete implementations. The correct choice of the time step λ in the step 4 depends on the implementation. The approach of artificial time dependence is well-suited for the level let method where the boundary of D_t is defined as a zero level of a higher dimensional function Φ. For the overview of the level set method used in EIT see [4].

Acknowledgements

Ivan Cimrák is supported by the Fund for Scientific Research - Flanders FWO, Belgium.

References

[1] M. Brühl and M. Hank, *Recent progress in electrical impedance tomography*, Inverse Problems, **19** (2003), 65–90.

[2] W. Chen, J. Cheng, J. Lin, and L. Wang, *A level set method to reconstruct the discontinuity of the conductivity in EIT*, Science in China Series A: Mathematics, **52** (2009), 29–44.

[3] I. Cimrák and V. Melicher, *Sensitivity analysis framework for micromagnetism with application to optimal shape design of magnetic random access memories*, Inverse Problems, **23** (2007), 563–588.

[4] O. Dorn, *Level set methods for inverse scattering*, Inverse Problems, **22** (2006), R67–R131.

[5] J. L. Muller and S. Sitanen, *Direct reconstruction of conductivities from boundary measurements*, SIAM J. Sci. Comput., **24** (2003), 1232–1266.

[6] P. Sergeant, I. Cimrák, V. Melicher, L. Dupré, and R. Van Keer, *Adjoint variable method for the study of combined active and passive magnetic shielding*, Math. Prob. Eng., **2008** (2008), Article ID 369125.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Computing minimal keys using lattice theory

Cordero, P.[1], Mora, A.[1], Enciso, M.[2] and de Guzmán, I.P.[1]

[1] *Department of Applied Mathematics, University of Málaga. Spain*

[2] *Department of Languages and Computer Science, University of Málaga. Spain*

emails: pcordero@uma.es, amora@ctima.uma.es, enciso@lcc.uma.es, guzman@ctima.uma.es

Abstract

The problem of finding all the minimal keys deduced from a given set of functional dependencies is NP-complete. There exists in the literature some classical algorithms used to deal with this problem. Their efficiency is measured by counting the number of closures applied to solve the problem. Thus, the only way to improve these algorithms is to reduce the number of closures needed to find all minimal keys. A deeper algebraic framework and a formal study of the information that appears in functional dependencies will render a set of result that may be applied to design more efficient algorithms. In this paper, we use lattice theory to develop a theoretical framework which allow us to obtain a larger reduction of the candidate key's lattice than others approaches of this problem.

We translate into the lattice theory the most important definition for functional dependencies, normal forms, minimal keys and maximal antikeys. The main element of this development is the concept of *nd-ideal operator*, which formalizes the notion of functional dependence in this theoretical framework. This development allows us to introduce the *KeyBox* algorithm which renders an important reduction in the number of closures. A first prototype of the *KeyBox* algorithm is implemented in Prolog and some illustrative examples are executed.

Key words: Lattice theory, functional dependencies, minimal keys, Prolog.

1 Introduction

The problem of finding keys in a well known problem in the relational theory. It has two different dimensions: we can look for all the keys which are satiffied in a concrete instance of a relation (the constraint which are hold by the data in the present instant) or we can infer the key from a set of Functional Dependencies belonging to the scheme of the relation (the constraint must be satisfied by any instance of the relation). The first situation is related with data mining and rough sets, while the second is usually treated with deduction method or algorithms.

Concerning the complexity of the first dimension of the problem, H. Mannila [14] cites as an open problem in Data Mining that of finding all minimal keys in a relational instance using only time that is polynomial in the number of relational attributes. C. Giannella and C. Wyss propose in [9] four algorithms to find minimal keys in a particular relation instance but they conclude that "they do not have the required time-complexity of a solution to Mannila's problem".

Recently, in [10] the authors remark that "identification of (composite) key attributes is of fundamental importance for many different data management tasks such as data modeling, data integration, anomaly detection, query formulation, query optimization, and indexing". Moreover, their GORDIAN algorithm is used to find keys in any collection of entities, e.g., relational tables or XML repositories. In [20] the idea of keys is generalized for the XML-like datamodel.

There exists a wide range of techniques used to deal with the instance version of the problem. J. Demetrovics [7] presents interesting results concerning the average length of keys in random databases, M. Albrecht [2] uses a Monte Carlo method for finding minimal keys, Pawlack's Rough set theory [18] can be used to discover knowledge which is latent in database relations (D.A. Bell and J.W. Guan use this guideline in [11]), H. Mannila [15] relates the problem of finding minimal keys with the problem of traversing hypergraphs and in [13] he proves new results about the existence and size of Armstrong relations for Boolean dependencies.

Although the former problem may be applied in a significant number of situations, the latter is required when we have to study the structure of a relational scheme which has not yet been instantiated. This situation appears in analysis and design issues, where the designer must construct a relational scheme fulfilling user requirements. The discovery of keys provides insights into the structure of data which are not easy to get by alternative means. Its benefits include to scale down storage requirements of the data, index creation (some platforms make necessary to specify table indexes to optimize queries), etc.

The second problem, to find *all* the minimal keys has also been studied. In the past C. Lucchesi and S. Osborn in [12] shown an algorithm to calculate all candidate keys. Thalheim in [21] shown that the number of keys is bounded by $\binom{n}{\lfloor \frac{n}{2} \rfloor}$ (n is the number of attributes). H. O. Saiedian and T. Spencer [19] presented and algorithm using attribute graphs to find all possible keys of a relational database schema. Another algorithm describing minimal keys as hypergraphs is presented by J. Demotrovics and V.D. Thi [8]. R. Wastl [22] propose an inference system for deriving all keys which can be inferred from a set of functional dependencies and establish that the number of keys of a relational schema is bounded by $e^{|\Gamma|/e}$ (Γ is the number of FDs). More recently, P. B. Wordlant [23] presents an algorithm to normalize to third normal form which includes a routine to find all the keys.

In order to focus on this problem with a formal basis we study functional dependencies over a set of attributes from lattice theory. Our work is based on a formal study of the concept of functional dependency within the general framework of the lattice theory presented in [5]. This theory allows us to have a uniform framework and to prove some important results which guided us an important reduction in the number of closures used to solve the input problem. The set of dependencies is deal in a deduction-style, allowing us to directly apply the result from the formal framework and some previous works on executable logics for functional dependencies.

In [1, 4] we present a new logic, named $\mathbf{L}_{DF}$ (Substitution Logic for FDs), which includes

two new *substitution rules* suitable for automated deduction. Substitution Logic was formally introduced in [4], providing correction and completion results. Its novel inference system has opened the door to the design of new efficient algorithms for the management of FDs: to remove redundancy [16] and to solve the implication problem [1].

In this work, we characterize the concept of f-family by means of a new concept which we call non-deterministic ideal operator (*nd.ideal-o.s*). The study of nd.ideal-o.s allows us to obtain results about functional dependencies as trivial particularizations, to clarify the semantics of the functional dependencies and to progress in their efficient use, and to extend the concept of *schema*. Moreover, the algebraic characterization of the concept of *Key of a schema* allows us to propose new formal definitions in the lattice framework for classical normal forms in relation schemes. We give a formal definition of the *normal forms* for functional dependencies more frequently used in the bibliography.

We emphasize that the framework of the lattice theory which allow us to obtain a larger reduction of the candidate key's lattice than others approaches of this problem. So, our results may be used to increase the efficiency of any finding key algorithms. Our method reduces both, the number of attributes and the number of FDs, and the bounds $\binom{n}{\lfloor \frac{n}{2} \rfloor}$, $e^{|\Gamma|/e}$ presented in the bibliography are strongly decreased. Finally, we present the *KeyBox* algorithm which computes all minimal keys. The algorithm is directly based on the theoretical results concerning lattice theory and FDs.

This work is organized as follows. First, in section 2, we introduce the formal basis to define (section 3) and manage (section 4) functional dependencies and keys. We present an algebraic study about minimal keys which is the basis of the *KeyBox* algorithm (section 5). We also introduce several examples which illustrate the execution of the minimal key search algorithm. To conclude, we give a brief presentation of a Prolog prototype of the KeyBox Algorithm executed over a set of examples extracted from the bibliography.

2 The formal framework

In this section, we summarize the theoretical framework [5] that has allowed to reduce the candidate key's lattice and develop a new algorithm for finding all minimal keys.

We assume that basic concepts of lattice theory are knows. An *ideal* in a lattice $(A, \leq)$ is a subset $I \subset A$ that is closed for supremum ($a, b \in I$ implies that $\sup\{a, b\} \in I$ and lower closed ($a \in I$ and $b \leq a$ implies that $b \in I$). An ideal is *principal* if it has a maximum element.

The concept of non-deterministic operation (also named multioperation, multivalued operation, polyoperation or hyperoperation depending of the authors) has been successfully used in literature to solve several algebraic and computational problems. They are operations in which the images are subsets instead of elements. A particular case is the hyperoperations in which the empty set is excluded. The theory of hyperstructures has been widely developed and has been used in several areas of computational sciences, for example in fuzzy set theory.

In this paper, we will only use unary non-deterministic operations. That is, a **unary non-deterministic operation** (henceforth, ndo) in a non empty set A is a map $F : A \to 2^A$. The unary case is the trivial one that corresponding with binary relations in A. However, using the idea of ndo and its notation several concepts and results become easier.

 ISBN: 978-84-612-9727-6

Note that every ndo F in A can be extended to an endofunction in 2^A ($F\colon 2^A \to 2^A$) defining $F(X) = \bigcup_{x\in X} F(x)$.

The domain of a binary relation, when we see them as ndo is named support. Given an ndo$F$ in A, we define the **support of** F as the set $Spp(F) = \{x \in A \mid F(x) \neq \varnothing\}$.

The study of functional dependencies and keys in databases required the following special type of ndo.

Definition 1 *Let F be an unary* ndo *in a lattice $(A, \leq)$. We say that F is a* **non-deterministic ideal operator** *(briefly* **nd.ideal-o***) if it is reflexive ($a \in F(a)$ for all $a \in A$), transitive ($F \circ F \subseteq F$) and $F(a)$ is an ideal of $(A, \leq)$, for all $a \in A$.*

Moreover, if $F(a)$ is a principal ideal, for all $a \in A$, then we say that F is **principal**.

In [5] we show that the intersection of nd.ideal-os is also an nd.ideal-o. So, given an ndo F, we can define the least nd.ideal-o that contain it and $\widehat{F}$ denote it. We can inductevely define it as $\widehat{F}(a) = \bigcup_{i\in\mathbb{N}} X_i$ where $X_0 = \{a\}$ and

$$X_{i+1} = X_i \cup \bigcup_{x\in X_i} F(x) \cup \{\sup(x, y) \mid x, y \in X_i\} \cup \bigcup_{x\in X_i} \{y \mid y \leq x\}$$

3 Keys and Functional Dependencies

In order to make this paper self-contained, we summarize some basic concepts concerning functional dependencies. Codd [6] conceives data stored in tables and labels each column of the table with *attributes*. Let a be an attribute, $dom(a)$ is the domain to which the values of the column determined by such attribute belong. Thus, if $\mathcal{A}$ is the finite set of attributes, we are interested in $R \subseteq \Pi_{a\in\mathcal{A}} dom(a)$ relations. Each $t \in R$, that is, each row, is denominated *tuple of the relation*. If t is a tuple of the relation, $a \in \mathcal{A}$ and $X \subseteq \mathcal{A}$, then $t(a)$ is the a-component of t and the *projection* of t over X, $t_{/X}$, is the restriction of t to X. That is, $t_{/X}(a) = t(a)$, for all $a \in X$.

Definition 2 (Functional Dependency) *Let R be a relation over $\mathcal{A}$. Any affirmation of the type $X \mapsto Y$, where $X, Y \subseteq \mathcal{A}$, is named* **functional dependency** *(henceforth FD) over R. We say that R* **satisfies** *$X \mapsto Y$ if, for all $t_1, t_2 \in R$ we have that: $t_{1/X} = t_{2/X}$ implies that $t_{1/Y} = t_{2/Y}$. If Γ is a set of functional dependencies in $\mathcal{A}$, we say that $s = (A, \Gamma)$ is a* **scheme** *and we say that R* **satisfies** *s if R satisfies $X \mapsto Y$ for all $X \mapsto Y \in \Gamma$.*

If $(\mathcal{A}, \Gamma)$ be a scheme, we denote by F_Γ the ndo in $2^{\mathcal{A}}$ given by $F_\Gamma(X) = \{Y \mid X \mapsto Y \in \Gamma\}$. Henceforth, we will represent schemes by a pair $(\mathcal{A}, \Gamma)$, where Γ is a set of functional dependencies over $\mathcal{A}$, or by a pair (A, F), where $A = 2^{\mathcal{A}}$ and F is a ndo in A.

It is immediate to prove that, if R is a relation over A, then the ndo FD_R in $2^{\mathcal{A}}$ given by $FD_R(X) = \{Y \mid X, Y \subseteq \mathcal{A}, R \text{ satisfies } X \mapsto Y\}$ is an nd.ideal-o. Conversely, given a non-empty finite set, U, for all nd.ideal-o, F, there exists a relation R such that $F = FD_R$.

Given an schema s, X^+ denotes the closure of a set of attributes X, that is, the biggest attributes set Y such that, if R is a relation that satisfies s then it also satisfies $X \mapsto Y$.

Proposition 3 *Let $(\mathcal{A}, \Gamma)$ be a scheme and $X \subseteq \mathcal{A}$. Then $X^+ = \sup \widehat{F_\Gamma}(X)$.*

 ISBN: 978-84-612-9727-6

Finally, we present some classical concepts of the relational model in terms of our formal framework. Given an scheme $s = (\mathcal{A}, \Gamma)$ and $X \subseteq \mathcal{A}$, we say that X is **key** if, for all relation R satisfying s and for any $t_1, t_2 \in R$, $t_{1/X} = t_{2/X}$ implies that $t_1 = t_2$. $\mathcal{K}_s$ denotes the set of keys in s. That is, $\mathcal{K}_s = \{a \in A \mid \widehat{F}(a) = A\}$.

We say that X is **minimal key** if it is a key and there not exists other key $X' \subsetneq X$. $\mathcal{K}_{s,min}$ denote the set of minimal keys in s. $\mathcal{K}_{s,min} = Minimals(\mathcal{K}_s)$.

X is an **antikey** if it is not a key and, for all $Y \supsetneq X$, Y is a key. $\mathcal{K}^{-1}_{s,max}$ denotes the set of antikeys in s. $\mathcal{K}^{-1}_{s,max} = Maximals(A\text{-}\mathcal{K}_s)$.

Example 1 *Let $\mathcal{A}$ be the set of attributes $\{a, b, c, d, e, f\}$. As usual, abc denotes the subset $\{a, b, c\}$. Let $s = (\mathcal{A}, \Gamma)$ be the scheme in which $\Gamma = \{ab \mapsto c, c \mapsto d, e \mapsto f, f \mapsto e\}$.*
$ab^+ = abcd$ and abe is a key because $abe^+ = abcdef = \mathcal{A}$.
$\mathcal{K}_s = \{abe, abce, abde, abcde, abf, abcf, abdf, abcdf, abef, abcef, abdef, abcdef\}$,
$\mathcal{K}_{s,min} = \{abe, abf\}$ and
$\mathcal{K}^{-1}_{s,max} = \{acdef, bcdef, abcd\}$.

4 Reducing the size of the boolean algebra to find all minimal keys

In [17] we present the theoretical results that will allow us to design the *KeyBox* algorithm.

Definition 4 *Let $(\mathcal{A}, \Gamma)$ be a scheme. We define the* **determinant** *and the* **determinate set of** Γ*, respectively, as follows:*

$$Dnt(\Gamma) = Spp(F_\Gamma) = \bigcup_{X \mapsto Y \in \Gamma} X \qquad \text{and} \qquad Dte(\Gamma) = \sup F_\Gamma(2^{\mathcal{A}}) = \bigcup_{X \mapsto Y \in \Gamma} Y$$

We define the **core of** s *as* $\texttt{core}_\texttt{s} = \mathcal{A} - Dte(\Gamma)$ *and we define the* **body of** s *as* $\texttt{body}_\texttt{s} = (Dnt(\Gamma) \cap (\mathcal{A} - \texttt{core}_\texttt{s}{}^+))$

Theorem 5 *Let $s = (\mathcal{A}, \Gamma)$ be a scheme. Then $\mathcal{K}_{s,min} \subseteq \left[\texttt{core}_\texttt{s}, \texttt{core}_\texttt{s} \cup \texttt{body}_\texttt{s}\right]$.* [1] *That is, if X is a minimal key then $\texttt{core}_\texttt{s} \subseteq X \subseteq (\texttt{core}_\texttt{s} \cup \texttt{body}_\texttt{s})$.*

Example 2 *Let $s = (\mathcal{A}, \Gamma)$ where $\mathcal{A}$ is the attributes set $\{a, b, c, d, e, f\}$ and Γ is the set of FDs $\{ab \mapsto c, a \mapsto g, g \mapsto c, b \mapsto h, bh \mapsto d, c \mapsto d, e \mapsto f, f \mapsto e\}$.*

$$\texttt{core}_\texttt{s} = ab \quad \text{and} \quad \texttt{body}_\texttt{s} = ef$$

Therefore, Theorem 5 ensures that $\mathcal{K}_{s,min} \subseteq [ab, abef]$ and, effectively, $\mathcal{K}_{s,min} = \{abe, abf\}$.

We remark that in literature we can find that the attributes present only in the determinants (ab) are in all keys, and the attributes only present in the determinates (d) are present in all antikeys. We translate this affirmation to our algebraic study: $\mathcal{K}_{s,min} \subseteq [ab, abcefgh]$. The important reduction our proposal brings is made evident. The algebraic study succeeds the minimal lattice for finding all minimal keys.

Our next step is going to be, to reduce the size of the Boolean Algebra A, we work with, so as to reduce even more the calculation of minimal keys.

The following definition allow us to reduce the lattice of the candidate keys. In figure 1 we show the algorithm to reduce the relational scheme.

[1] $\texttt{core}_\texttt{s}$ is the set of attributes present in all keys[11] and $\texttt{body}_\texttt{s}$ is the set of significative attributes.

 ISBN: 978-84-612-9727-6

Reduce Algorithm
Input: $s = (\ \mathcal{A}, \Gamma)$, *Output*: $s_{red} = (\ \mathcal{A}', \quad \Gamma'), \texttt{body}_\texttt{s}, \texttt{core}_\texttt{s}$
Begin
1. Calculate $\texttt{body}_\texttt{s}$ and $\texttt{core}_\texttt{s}$ of s
   1.1 $Dnt(\Gamma) = \emptyset$, $Dte(\Gamma) = \emptyset$
   1.2 *For each* $X \mapsto Y \in \Gamma$
   $Dnt(\Gamma) = Dnt(\Gamma) \bigcup X$
   $Dte(\Gamma) = Dte(\Gamma) \bigcup Y$
   1.3 $\texttt{core}_\texttt{s} = \mathcal{A} - Dte(\Gamma)$
   1.4 $\texttt{body}_\texttt{s} = (Dnt(\Gamma) \cap (\ \mathcal{A} - \texttt{core}_\texttt{s}{}^{+}))$
2. $\mathcal{A}' = \texttt{body}_\texttt{s}$
3. $\Gamma' = \emptyset$
4. *For each* $X \mapsto Y \in \Gamma$
   $\Gamma' = \Gamma' \bigcup \{X \cap \texttt{body}_\texttt{s} \mapsto Y \cap \texttt{body}_\texttt{s}\}$
End

Figure 1: Reduce the scheme.

Definition 6 *Let $(\mathcal{A}, \Gamma)$ to be a relational scheme. We define the* **reduced scheme** *as the scheme $s_{red} = \big(\texttt{body}_\texttt{s}, \mathbf{Reduc}(\Gamma)\big)$ where*

$$\mathbf{Reduc}(\Gamma) = \{(X \cap \texttt{body}_\texttt{s}) \mapsto (Y \cap \texttt{body}_\texttt{s}) \mid X \mapsto Y \in \Gamma \text{ and } Y \cap \texttt{body}_\texttt{s} \neq \varnothing\}$$

When we calculate the minimal keys of the **reduced scheme**, we render the minimal keys of the **source scheme**, using the following definition.

Definition 7 *Let $s = (\mathcal{A}, \Gamma)$ be a scheme. Then*

$$\mathcal{K}_{s,min} = \{X \cup \texttt{core}_\texttt{s} \mid X \in \mathcal{K}_{s_{red},min}\}$$

Example 3 *Given $s = (\mathcal{A}, \Gamma)$ a relational scheme with $\mathcal{A} = \{a, b, c, d, e, f, g, l, m, n\}$ and $\Gamma = \{a \mapsto bc, b \mapsto def, d \mapsto e, g \mapsto be, l \mapsto m, am \mapsto ln, bl \mapsto n\}$.*

We find that $\texttt{core}_\texttt{s} = ag$, $Dnt(\Gamma) = abdglm$, $\texttt{core}_\texttt{s}{}^{+} = abcdefg$ and $\texttt{body}_\texttt{s} = lm$

Therefore, we obtain $\mathbf{Reduc}(\Gamma) = \{l \mapsto m, m \mapsto l\}$ and the reduced scheme of s is $s_{red} = (\{l, m\}, \{l \mapsto m, m \mapsto l\})$.
Since $\mathcal{K}_{s_{red},min} = \{m, l\}$, we have that $\mathcal{K}_{s,min} = \{agm, agl\}$

5 *KeyBox*: an algorithm to find all minimal keys and maximal antikeys

The *ideal-ond* notion allow us the study of scheme reduction and, thereby, to obtain in an efficient way the minimal keys in a theoretical general framework.

 ISBN: 978-84-612-9727-6

Initial Keys Algorithm
Input: $s = (\ \mathcal{A}, \Gamma)$, *Output*: $\mathcal{K}_{s,min}$, $\mathcal{K}_s$, $\mathcal{K}^{-1}_{s,max}$, $\mathcal{K}^{-1}_s$
Begin
1. $\mathcal{K}_{s,min}$, $\mathcal{K}_s = \varnothing$, $\mathcal{K}^{-1}_{s,max}$, $\mathcal{K}^{-1}_s = \varnothing$
2. Sort $\Gamma' = \{X_i \mapsto Y_i\}$ in lexicography order of X_i
4. *For each* $X \mapsto Y \in \Gamma$ *do*
   If $X \notin (\ \mathcal{K}_{s,min} \cup \mathcal{K}_s)\uparrow$ *then*
      If $X^+ = \mathcal{A}$ *then*
         If $X \in Atom(\ \mathcal{A})$ *then*
            X is minimal key, $\mathcal{K}_{s,min} = \mathcal{K}_{s,min} \cup \{X\}$
         else X is key, $\mathcal{K}_s = \mathcal{K}_s \cup \{X\}$
   If $X \notin (\ \mathcal{K}^{-1}_{s,max} \cup \mathcal{K}^{-1}_{s,max})\downarrow$ *then*
      If $X^+ \neq \mathcal{A}$ *then*
         If $X \in Coatom(\ \mathcal{A})$ *then*
            X is maximal antikey, $\mathcal{K}^{-1}_{s,max} = \mathcal{K}^{-1}_{s,max} \cup \{X\}$
         else
            X is antikey, $\mathcal{K}^{-1}_s = \mathcal{K}^{-1}_s \cup \{X\}$
   End for each
End

Figure 2: Calculate initial keys.

When we have obtained the reduced scheme $s_{red} = (\mathcal{A}', \Gamma')$, our goal is to obtain the minimal keys in an efficient way. In order to squeeze the full information presents in the FD set, our study renders the convenience of checking if the determinants of Γ' are contained in $\mathcal{K}_{s_{red},min}$, $\mathcal{K}_{s_{red},min}\uparrow^*$, $\mathcal{K}^{-1}_{s_{red},max}$, either in $\mathcal{K}^{-1}_{s_{red},max}\downarrow^*$. The algorithm shown in the figure 2 sorts in a first step the FD set, because if by example X is in $\mathcal{K}_{s_{red},min}$ a superset of X doesn't check, etc. Then, cross out the FD set and checks the information of the determinants of the FDs. The algorithm renders a set of initial keys obtained with the maximum information contained in the FDs and allow us to reduce the number of closures necessaries to find in the lattice of candidate keys (the Boole algebra $(2^{\mathcal{A}'}, \subseteq)$) the minimal keys of the scheme.

The following example illustrates the execution of the Reduce and Initial Keys Algorithms:

Example 4 *Given* $s = (\mathcal{A}, \Gamma)$ *with* $\mathcal{A} = \{a, b, c, d, e, f, g, h\}, \Gamma = \{c \mapsto abd, d \mapsto b, ab \mapsto c, e \mapsto f, f \mapsto gh\}$. *The execution of* Reduce and Initial Keys Algorithms *renders the following results:*

Reduce Algorithm
$s_{red} = (\ \mathcal{A}',\ \Gamma') = (\mathtt{body_s},\ \ \mathbf{Reduc}(\Gamma))$
$\mathtt{core_s} = \{e\}$
$\mathcal{A}' = \mathtt{body_s} = \{a, b, c, d\}$, $\Gamma' = \{c \mapsto abd,\ d \mapsto b, ab \mapsto c\}$
$\mathcal{K}_{s,min} \subseteq \ [\mathtt{core_s},\ \mathtt{core_s} \cup \mathtt{body_s}] = [\{e\},\ \{a, b, c, d, e\}]$

Initial Keys and Antikeys

$\{c\}$, $\{c\} \nsubseteq (\ \mathcal{K}_{s',min} \cup \mathcal{K}_{s'})\uparrow$ y $\{c\} \nsubseteq (\ \mathcal{K}^{-1}_{s',max} \cup \mathcal{K}^{-1}_{s'})\downarrow$

 ISBN: 978-84-612-9727-6

$(\{c\})^+ = \mathcal{A}' \Rightarrow$ YES, c is minimal key
$\mathcal{K}_{s',min} = \{c\},\ \mathcal{K}^{-1}_{s',max} = \varnothing,\ \mathcal{K}_{s'} = \varnothing,\ \mathcal{K}^{-1}_{s'} = \varnothing$
$\underbrace{\{d\}}, \{d\} \not\subseteq (\ \mathcal{K}_{s',min} \cup\ \mathcal{K}_{s'}) \uparrow$ y $\{d\} \not\subseteq (\ \mathcal{K}^{-1}_{s',max} \cup\ \mathcal{K}^{-1}_{s'}) \downarrow$

$(\{d\})^+ \neq \mathcal{A}' \Rightarrow$ NO key
$\mathcal{K}_{s',min} = \{c\},\quad \mathcal{K}^{-1}_{s',max} = \varnothing,\quad \mathcal{K}_{s'} = \varnothing,\ \mathcal{K}^{-1}_{s'} = \{d\}$
$\underbrace{\{ab\}}, \{ab\} \not\subseteq (\ \mathcal{K}_{s',min} \cup\ \mathcal{K}_{s'}) \uparrow$ y $\{ab\} \not\subseteq (\ \mathcal{K}^{-1}_{s',max} \cup\ \mathcal{K}^{-1}_{s}) \downarrow$

$(\{ab\})^+ = \mathcal{A}' \Rightarrow$ YES, c is key
$\mathcal{K}_{s',min} = \{c\},\ \mathcal{K}^{-1}_{s',max} = \varnothing,\ \mathcal{K}_{s'} = \{ab\},\ \mathcal{K}^{-1}_{s'} = \{d\}$

$\mathcal{A}' = \{a,b,c,d\}\ \ \Gamma' = \{c \mapsto\ abd, d\ \mapsto\ b, ab \mapsto\ c\}$

Note that our algebraic study (reducing scheme algorithm) succeeds to reduce the lattice size from 2^8 to 2^4. Moreover the scheme is considerably reduced and the exponential nature of the problem has been reduced.

Now we are able to propose an approximation to an algorithm for finding all minimal keys in a scheme. The algorithm we present in fig. 3 takes a scheme $s = (\Gamma, \mathcal{A})$, execute the previous algorithms to prune the scheme and calculate the initial keys and antikeys and renders $\mathcal{K}_{s,min}$ and $\mathcal{K}^{-1}_{s,max}$. We use the lattice structure of the set of possible minimal keys to reduce the number of necessary closures. We will show the way the algorithm works and how it prevents unnecessary closures.

Example 5 *In the example 4 we have reduced the set of FDs to $s' = (\mathcal{A}', \Gamma')$, with $\mathcal{A}' = \{a,b,c,d\}, \Gamma' = \{c \mapsto abd, d \mapsto b, ab \mapsto c\}$, the initial set of keys and antikeys are $\mathcal{K}_{s',min} = \{c\}, \mathcal{K}^{-1}_{s',max} = \varnothing, \mathcal{K}_{s'} = \{ab\}, \mathcal{K}^{-1}_{s'} = \{d\}$. Then, we apply the* **KeyBox Algorithm** *that seize the work of the previous algorithms. In this point of the process, we have a reduced scheme* [2] *and important information about a set of initial keys, minimal keys, antikeys, and maximal antikeys that will be used in* **KeyBox Algorithm** *to minimize the number of closures necessaries to calculate all minimal keys. Obviously, we can't eliminate the exponential nature of the problem but, we are exploiting all possible information to reduce quite a lot of its temporal complexity. A comparative study of our approach and the classical finding key algorithms is actually in process.*

1. $\mathcal{A}' = \{a,b,c,d\}$,
   $\Gamma' = \{c \mapsto abd, d \mapsto b, ab \mapsto c\}$
2. $\mathtt{core_s} = \{e\}$
3. $\mathcal{K}_{s',min} = \{c\}, \mathcal{K}^{-1}_{s',max} = \varnothing$,
   $\mathcal{K}_{s'} = \{ab\}, \mathcal{K}^{-1}_{s'} = \{d\}$
4. **Generate candidates**
   (Step i) $B = \{a\}$, $\overline{B} = \{b,c,d\}$

B fulfills Condition 1, not fulfills Condition 2, and not fulfills Condition 3 ($B^+ \neq \mathcal{A}'$) then $B = \{a\}$ is NOT A KEY

$\overline{B}$ not fulfills Condition 1, $\overline{B} \in \mathcal{K}_{s',min} \uparrow$ then $\overline{B} = \{b,c,d\}$ can not be an antikey.

(Step ii) $B = \{b\}$, $\overline{B} = \{a,c,d\}$

[2] Note that our theoretical study gets the most reduction of the lattice of the candidate keys that appears in the literature.

 ISBN: 978-84-612-9727-6

KeyBox Algorithm

Input: $s=(\mathcal{A},\Gamma)$, *Output*: $\mathcal{K}_{s,min}$

Begin

[s', body_s, core_s]=Reduce(s)

[$\mathcal{K}_{s',min}$, $\mathcal{K}_{s'}$, $\mathcal{K}^{-1}_{s',max}$, $\mathcal{K}^{-1}_{s'}$]=Initial Keys(s')

visitednodes=1

While visitednodes $\leq 2^{|\mathcal{A}|-1}$

For each $B \in 2^{\mathcal{A}}$ *do* (Generate Candidates)

(1)*If* $B \notin \mathcal{K}_{s',min}\uparrow$ and $B \notin (\mathcal{K}^{-1}_{s',max} \cup \mathcal{K}^{-1}_{s'})\downarrow$ *then*

(2) *If* $B \in \mathcal{K}_{s'}$, $B \notin \mathcal{K}^{-1}_{s'}\downarrow$ and $B \notin \mathcal{K}_{s',min}\uparrow$ *then*

$\mathcal{K}_{s',min} = \mathcal{K}_{s',min} \cup B$, $\mathcal{K}_{s'} = \mathcal{K}_{s'} - B$

(3)*elsif* $B^+ = \mathcal{A}$ *then*

$\mathcal{K}_{s',min} = \mathcal{K}_{s',min} \cup \{B\}$

(4)*If* $\overline{B} \notin (\mathcal{K}_{s'} \cup \mathcal{K}_{s',min})\uparrow$ and $\overline{B} \notin \mathcal{K}^{-1}_{s',max}\downarrow$ *then*

(5) *If* $\overline{B} \in \mathcal{K}^{-1}_{s'}$ $B \notin \mathcal{K}_{s'}\uparrow$ and $\overline{B} \notin \mathcal{K}^{-1}_{s',max}\downarrow$ *then*

$\mathcal{K}^{-1}_{s',max} = \mathcal{K}^{-1}_{s',max} \cup \overline{B}$, $\mathcal{K}^{-1}_{s'} = \mathcal{K}^{-1}_{s'} - \overline{B}$

(6) *elsif* $\overline{B}^+ \neq \mathcal{A}$ *then*

$\mathcal{K}^{-1}_{s',max} = \mathcal{K}^{-1}_{s',max} \cup \overline{B}$

else

$\mathcal{K}_{s'} = \mathcal{K}_{s'} \cup \overline{B}$

visitednodes=visitednodes+1

End While

Find the minimal keys in $\mathcal{K}_{s'}$ and add to $\mathcal{K}_{s',min}$

$\mathcal{K}_{s,min} = \mathcal{K}_{s',min} \times \text{core}_s$

End

Figure 3: Calculate all minimal keys.

B fulfills Condition 1, not fulfills Condition 2, and not fulfills Condition 3 ($B^+ \neq \mathcal{A}'$) then $B=\{a\}$ is NOT A KEY

$\overline{B}$ not fulfills Condition 4,$\overline{B} \in \mathcal{K}_{s',min}\uparrow$ then $\overline{B} = \{b,c,d\}$ can not be an antikey.

(Step iii) $B=\{c\}$, $\overline{B}=\{a,b,d\}$

B not fulfills Condition 1, B is a minimal key yet.

$\overline{B}$ not fulfills Condition 4,$\overline{B} \in \mathcal{K}_{s',min}\uparrow$ then $\overline{B} = \{b,c,d\}$ can not be an antikey.

(Step iv) $B=\{d\}$, $\overline{B}=\{a,b,c\}$

$B=\{d\} \in$ not fulfills Condition 1, B is antikey, thus can not be a key.

$\overline{B}$ not fulfills Condition 4, $\{c\}$ is minimal key, thus $\overline{B}$ can not be an antikey.

(Step v) $B=\{a,b\}$, $\overline{B}=\{c,d\}$

$B=\{a,b\} \in$ fulfills Condition 1, and fulfills Condition 2, then B is minimal key.

ACTUALIZE THE SETS $\mathcal{K}_{s',min} = \{c,ab\}, \mathcal{K}_{s'}=\varnothing$

$\overline{B}$ not fulfills Condition 4, $\{c\}$ is minimal key, thus $\overline{B}$ can not be an antikey.

(Step v) $B=\{a,c\}$, $\overline{B}=\{b,d\}$

$B = \{a,c\} \in$ not fulfills Condition 1, $\{c\}$ is minimal key, thus B can not be a minimal key.

$\overline{B}$ fulfills Condition 4, not fulfills Condition 5, and fullfils

 ISBN: 978-84-612-9727-6

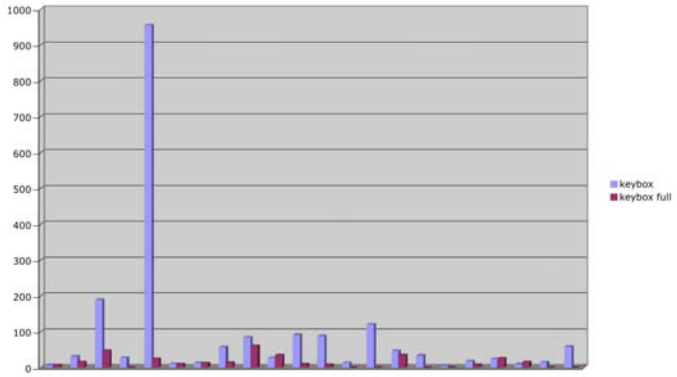


Figure 4: Execution of illustrative examples.

`Condition 6, thus` $\overline{B}$ `is a maximal antikey.`
`ACTUALIZE THE SETS` $\mathcal{K}^{-1}_{s',max} = \{b,d\}, \mathcal{K}_{s'} = \varnothing$

`...`

5. **The algorithm renders all minimal keys and maximal antikeys:**
$\mathcal{K}_{s,min} = \{abe, ade, ce\}, \mathcal{K}^{-1}_{s,max} = \{b,d\}$.

Note that our algebraic study (reducing scheme algorithm) succeeds to reduce the lattice size from 2^8 *to* 2^4. *And the KeyBox algorithm reduce considerably the number of closures.*

Prolog has been used in many areas of Artificial Intelligence. The Prolog prototypes provides a declarative and pedagogical point of departure and illustrated the behavior of the new techniques in a very fast and easy way. Now, we show the result of execution of a first Prolog prototype of *KeyBox algorithm* [3] over a set of examples selected from several books where the key problem was presented.

We execute the Prolog `KeyBox(Scheme)` and `KeyBoxFull(Scheme)` to prove the great reduction obtained with out theoretical study. The result are depicted in figure 4 where each example are executed with and without our reduction algorithm and the number of closures executed are compared. The main result is that the number of closures is decreased in 19 examples when using the full version of the Keybox method, and it is increased in 3 cases. The average percentage of reduction is 70,52% and the average augmentation ratio is 18,98%. There exists 6 examples where the Keybox method reduces the number of closures to 0.

In the three examples in which our algorithm increase the number of closures, the extra closures are due to the algorithm 4.

[3]For referees the full implementation with a lot of examples and a small statistic is available in http://enciso.lcc.uma.es/keys/

6 Conclusions

In this work, we emphasize the notion of *nd-ideal operator* which formalizes the notion of functional dependence in the lattice theory framework. The study of this algebraic tool allows us to design new algorithms to find all minimal keys. These algorithms are directly based on this theoretical study and render an important reduction in the number of closures used to solve the problem.

The *KeyBox* and *KeyBoxFull* algorithms have been executed over a set of FDs to illustrate the reduction obtained. The test shows the promising results of the finding minimal key and maximal antikeys algorithm.

We have not found in the bibliography any work focussed in the design of a benchmark for the problem of finding minimal keys. To design the experiment of this work we have recollected illustrative examples which appear in some papers and books where this problem is presented. We are now interesting in the design of a benchmark that systematizes the characteristics of this problem.

Acknowledgements

This work is partially supported by the Spanish research project TIN07-65819.

References

[1] G. Aguilera, P. Cordero, M. Enciso, A. Mora, and I. P. de Guzmán. A non-explosive treatment of Functional dependencies using rewriting logic. *Lecture Notes - LNAI*, 3171, 31–40, 2007.

[2] M. Albrecht, M. Altus, B. Buchholz, A. Dusterhoft, K. Schewe, and B. Thalheim. Die intelligente tool box zum datenbankenwurf rad. *Datenbank-Rundbrief, 13. F.G. 2.5 der GI, Kassel*, 1994.

[3] William W. Armstrong. Dependency structures of data base relationships. *Proc. IFIP Congress. North Holland, Amsterdam*, pages 580–583, 1974.

[4] P. Cordero, M. Enciso, M., I.P. de Guzmán, and A. Mora. SLFD logic: Elimination of data redundancy in knowledge representation. *Lecture Notes - LNAI*, 2527, 141–150, 2002.

[5] P. Cordero, A. Mora, I.P. de Guzmán, and M. Enciso. Non-deterministic ideal operators: An adequate tool for formalization in Data Bases. *Discrete Applied Mathematics. doi: 10.1016/j.dam.2007.02.014*, 141–150, 2007.

[6] Edgar F. Codd. The relational model for database management: Version 2. reading, mass. *Addison Wesley*, 1990.

[7] Janos Demetrovics and Vu Duc Thi. Family of functional dependencies and its equivalent descriptions. *Computers Math. Applic.*, 29 (4):101–109, 1995.

[8] János Demetrovics and Vu Duc Thi. Describing Candidate Keys by Hypergraphs. *Computers and Artificial Intelligence*, 18 (2), 1999.

[9] C. Giannella and C.M. Wyss. Finding Minimal Keys in a Relation Instance. *citeseer.comp.nus.edu.sg/279058.html*.

[10] Yannis Sismanis, Paul Brown, Peter J. Haas, and Berthold Reinwald. Gordian: efficient and scalable discovery of composite keys. *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pages 691–702, 2006.

[11] J.W. Guan and D.A. Bell. Rough computational methods for information systems. *Artificial Intelligence*, 105:77–103, 1998.

[12] C. Lucchesi and S. Osborn. Candidate keys for relations. *JJCSS*, 17(2), pages 270–279, 1978.

[13] R. Khardon and D. Mannila, H. an Roth. Reasoning with examples: Propositional formulae and database dependencies. *Acta Informatica*, 36(4):267–286, 1999.

[14] Heikki Mannila. Methods and problems in data mining. *Proceedings of International Conference on Database Theory. Afrati, Kolaitis (ed.).*, 17 (2), 1997.

[15] Heikki Mannila and Kari-Jouko Raiha. Algorithms for inferring functional dependencies from relations. *Data and Knowledge Engineering*, 12:83–99, 1994.

[16] A. Mora, M. Enciso, P. Cordero, and I.P. de Guzmán. The functional dependence implication problem: optimality and minimality. an efficient preprocessing transformation based on the substitution paradigm. *Lecture Notes - LNAI*, 3040, 136–146, 2004.

[17] A. Mora, P. Cordero, M. Enciso, G. Aguilera, and I.P. de Guzmán. Ideal non-deterministic operators as a formal framework to reduce the key finding problem. *Submitted to IJCS. Technical Report at http://enciso.lcc.uma.es/keys/IdealKey.pdf*.

[18] Z. PAWLAK, *Rough Set: theoretical aspects of reasoning about data*, Kluwer. Dordercht, Netherlands, 1991

[19] Hossein Saiedian and Thomas Spencer. An Efficient Algorithm to Compute the Candidate Keys of a Relational Database Schema. *Comput. J.*, 39 (2) , pp. 124–132, 1996.

[20] Klaus-Dieter Schewe. Redundancy, dependencies and normal forms for xml databases. pages 7–16, 2005.

[21] B. Thalheim. On the number of keys in relational and nested relational databases. *Discrete Applied Mathematics*, 40 , pp. 265–282, 1992.

[22] Ralf Wastl. On the Number of Keys of a Relational Database Schema. *Journal of Universal Compurter Science*, 4 (5), pages 547–559, 1998.

[23] Peter Worland. An efficient algorithm for 3NF determination. *Inf. Sci. Inf. Comput. Sci.*, 167 (1-4), pages 177–192, 2004.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Incorporating a Four-Dimensional Filter Line Search Method into an Interior Point Framework

M. Fernanda P. Costa[1] and Edite M. G. P. Fernandes[2]

[1] *Department of Mathematics for Science and Technology, University of Minho, 4800 Azurem, Portugal*

[2] *Department of Production and Systems, University of Minho, 4710-057 Braga, Portugal*

emails: mfc@mct.uminho.pt, emgpf@dps.uminho.pt

Abstract

Here we incorporate a four-dimensional filter line search method into an infeasible primal-dual interior point framework for nonlinear programming. Each entry in the filter has four components measuring dual feasibility, complementarity, primal feasibility and optimality. Three measures arise directly from the first order optimality conditions of the problem and the fourth is the objective function, so that convergence to a stationary point that is a minimizer is guaranteed. The primary assessment of the method has been done with a well-known collection of small problems.

Key words: Nonlinear optimization, interior point, filter method
MSC 2000: 90C51, 90C30

1 Introduction

In this paper we consider a nonlinear constrained optimization problem in the following form:

$$\begin{array}{l} \min_{x\in\mathbb{R}^n} \ F(x) \\ \text{s.t.} \ \ h(x) \geq 0 \end{array} \tag{1}$$

where $h_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, m$ and $F : \mathbb{R}^n \to \mathbb{R}$ are nonlinear and twice continuously differentiable functions. Interior point methods based on a logarithmic barrier function have been widely used for nonlinear programming [9, 11, 12]. To allow convergence from poor starting points, barrier and augmented Lagrangian merit functions may be used [7]. Some line search frameworks use penalty merit functions to enforce progress toward the solution. As an alternative to merit functions, Fletcher and Leyffer [4] proposed a filter method as a tool to guarantee global convergence in

algorithms for nonlinear optimization. This technique incorporates the concept of non-dominance to build a filter that is able to accept trial points if they improve either the objective function or the constraints violation, instead of a combination of those two measures defined by a merit function. The filter replaces the use of merit functions, so avoiding the update of penalty parameters that are associated with the penalization of the constraints in a merit function. The filter technique has already been adapted to interior point methods. In [13, 14, 15], a filter line search strategy incorporated in a barrier type method is used. The two components of each entry in the filter are the barrier objective function and the constraints violation. In [10], a two-dimensional filter is used in a primal-dual interior point method context. The two entries, measuring quasi-centrality and optimality, combine the three criteria of the first order optimality conditions. A three-dimensional filter based line search strategy has already been tested in [2, 3]. The three components of the filter measure feasibility, centrality and optimality and are present in the first order KKT conditions of the barrier problem associated with the problem (1). The optimality measure relies on the norm of the gradient of the Lagrangian function. Convergence to stationary points may been proved, although convergence to a local minimizer is not guaranteed.

In this paper we propose a four-dimensional filter line search method to incorporate into a primal-dual interior point framework. The three criteria of the first order optimality conditions are used separately to define three measures, and the objective function, $F(x)$, is the other so that convergence to a stationary point that is a minimizer is guaranteed.

The paper is organized as follows. Section 2 presents the interior point paradigm and Section 3 introduces the novel filter line search method that relies on four components and presents the acceptance conditions used to accept a point in the filter. The experimental results and the conclusions make Section 4.

2 The primal-dual interior point paradigm

In this interior point paradigm, problem (1) is reformulated as an equality constrained problem by using nonnegative slack variables w, as follows:

$$\begin{array}{rl} \min_{x\in\mathbb{R}^n, w\in\mathbb{R}^m} & F(x) \\ \text{s.t.} & h(x) - w = 0 \\ & w \geq 0, \end{array} \tag{2}$$

and the first order or Karush-Kuhn-Tucker (KKT) optimality conditions for a minimum of (2) are written as

$$\begin{array}{rl} \nabla_x \mathcal{L}(x, w, y, v) & = 0 \\ y - v & = 0 \\ Wy & = 0 \\ h(x) - w & = 0 \\ w \geq 0,\ v & \geq 0 \end{array} \tag{3}$$

where $y, v \in \mathbb{R}^m$ are the vector of Lagrange multipliers, $W = diag(w_i)$ is a diagonal matrix, and $\nabla_x \mathcal{L}$ is the gradient with respect to x of the Lagrangian function defined

 ISBN: 978-84-612-9727-6

by

$$\mathcal{L}(x, w, y, v) = F(x) - y^T(h(x) - w) - v^T w.$$

The system (3) is equivalent to the system

$$\begin{array}{rl} \nabla F(x) - A(x)^T y & = 0 \\ Wy & = 0 \\ h(x) - w & = 0 \\ w \geq 0,\ y & \geq 0 \end{array} \tag{4}$$

where $A(x)$ is the Jacobian matrix of the constraint functions $h(x)$. If the second equation of conditions (4) is perturbed, we get the KKT perturbed system of equations

$$\begin{array}{rl} \nabla F(x) - A(x)^T y & = 0 \\ Wy - \mu e & = 0 \\ h(x) - w & = 0 \\ w \geq 0,\ y & \geq 0 \end{array} \tag{5}$$

where e is a vector of unit m elements and μ is a positive parameter called barrier parameter [9, 12]. This perturbed system is equivalent to the KKT conditions of the barrier problem associated with problem (2), in the sense that they have the same solution,

$$\begin{array}{rl} \min_{x\in\mathbb{R}^n, w\in\mathbb{R}^m} & \varphi_\mu(x, w) \\ \text{s.t.} & h(x) - w = 0 \end{array} \tag{6}$$

where $\varphi_\mu(x, w) \equiv F(x) - \mu \sum_{i=1}^{m} \log(w_i)$ is the logarithmic barrier function. Applying the Newton's method to solve (5), the following system, after symmetrization, arises

$$\begin{bmatrix} -H & 0 & A(x)^T \\ 0 & -W^{-1}Y & -I \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla F(x) - A(x)^T y \\ -\mu W^{-1} e + y \\ w - h(x) \end{bmatrix} \tag{7}$$

where $Y = diag(y_i)$ is a diagonal matrix,

$$H = \nabla^2 F(x) - \sum_{i=1}^{m} y_i \nabla^2 h_i(x)$$

is the Hessian matrix of the Lagrangian function.

Since the second equation in (7) can be used to eliminate Δw without producing any off-diagonal fill-in in the remaining system, one obtains

$$\Delta w = WY^{-1}\left(\mu W^{-1} e - y - \Delta y\right), \tag{8}$$

and the resulting reduced KKT system

$$\begin{bmatrix} -H & A(x)^T \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla F(x) - A(c)^T y \\ w - h(x) + WY^{-1}(\mu W^{-1} e - y) \end{bmatrix} \tag{9}$$

 ISBN: 978-84-612-9727-6

to compute the search directions Δx, Δw, Δy. This interior point based method implements a line search procedure combined with a backtracking strategy to compute a step size α_k, at each iteration k, and define a new approximation by

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k \Delta x_k \\ w_{k+1} &= w_k + \alpha_k \Delta w_k \\ y_{k+1} &= y_k + \alpha_k \Delta y_k \end{aligned}$$

where equal step sizes are used with primal and dual directions. The choice of the step size α_k is a very important issue in nonconvex optimization and in the interior point context aims:

1. to ensure the nonnegativity of the slack and dual variables;

2. to enforce progress towards feasibility, complementarity and optimality.

Here we propose a four-dimensional filter method combined with a backtracking strategy to define new approximations to the primal, slack and dual variables that give a sufficient reduction in one of the filter measures. The backtracking strategy defines a decreasing sequence of step sizes

$$\alpha_{k,l} \in (0, \alpha_k^{\max}], l = 0, 1, \ldots,$$

with $lim_l \alpha_{k,l} = 0$, until a set of acceptance conditions are satisfied. Here, the index l denotes the iteration counter for the inner loop. The parameter $\alpha_k^{\max}$ represents the longest step size that can be taken along the direction before violating the nonnegativity conditions $w_k \geq 0, y_k \geq 0$. If the initial approximations for the slack and dual variables satisfy $w_0 > 0, y_0 > 0$, the maximal step size $\alpha_k^{\max} \in (0, 1]$ is defined by

$$\alpha_k^{\max} = \min\left\{1, \varepsilon \min\{-w_k^i (\Delta w_k^i)^{-1}, -y_k^i (\Delta y_k^i)^{-1}\}\right\} \tag{10}$$

for all i such that $\Delta w_k^i < 0$ and $\Delta y_k^i < 0$, and $\varepsilon \in (0, 1)$ is a fixed parameter.

3 Four-dimensional filter line search method

In order to define the components of each entry in the filter and the corresponding acceptance conditions, the following notation is used:

$$\begin{aligned} u &= (x, w, y), & \Delta &= (\Delta x, \Delta w, \Delta y), \\ u^1 &= (x, w), & \Delta^1 &= (\Delta x, \Delta w), \\ u^2 &= (w, y), & \Delta^2 &= (\Delta w, \Delta y), \\ u^3 &= (x, y), & \Delta^3 &= (\Delta x, \Delta y). \end{aligned}$$

The optimality conditions (4) define a set of natural measures to assess the algorithm progress. Some combinations of these measures may be used to define the components of each entry in the filter, see for example [10]. We use the three conditions separately. Further, to be able to guarantee convergence to stationary points that are minimizers, we introduce F as the fourth measure in the filter [5]. Table 1 lists the four components for the herein proposed filter.

Table 1: Components of the four-dimensional filter

measure	
primal feasibility	$\theta_{pf}(u^1) \equiv \|h(x) - w\|_2$
complementarity	$\theta_c(u^2) \equiv \|Wy\|_2$
dual feasibility	$\theta_{df}(u^3) \equiv \left\|\nabla F(x) - A(x)^T y\right\|_2$
optimality	$F(x)$

3.1 The acceptance conditions

In this algorithm, the trial point $u_k(\alpha_{k,l}) = u_k + \alpha_{k,l}\Delta_k$ is acceptable by the filter, if it leads to sufficient progress in one of the four measures compared to the current iterate,

$$\begin{array}{l} \theta_{pf}(u_k^1(\alpha_{k,l})) \leq (1-\gamma_1)\,\theta_{pf}(u_k^1) \;\text{ or }\; \theta_c(u_k^2(\alpha_{k,l})) \leq (1-\gamma_2)\,\theta_c(u_k^2) \\ \text{or }\; \theta_{df}(u_k^3(\alpha_{k,l})) \leq (1-\gamma_3)\,\theta_{df}(u_k^3) \;\text{ or }\; F(x_k(\alpha_{k,l})) \leq F(x_k) - \gamma_4\theta_{pf}(u_k^1) \end{array} \tag{11}$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in (0,1)$ are fixed constants.

However, to prevent convergence to a point that is nonoptimal, and whenever for the trial step size $\alpha_{k,l}$, the following switching conditions

$$\begin{array}{l} m_k(\alpha_{k,l}) < 0 \text{ and } [-m_k(\alpha_{k,l})]^{s_o}\,[\alpha_{k,l}]^{1-s_o} > \delta\left[\theta_{pf}(u_k^1)\right]^{s_1} \\ \text{and } [-m_k(\alpha_{k,l})]^{s_o}\,[\alpha_{k,l}]^{1-s_o} > \delta\left[\theta_c(u_k^2)\right]^{s_2} \text{ and } [-m_k(\alpha_{k,l})]^{s_o}\,[\alpha_{k,l}]^{1-s_o} > \delta\left[\theta_{df}(u_k^3)\right]^{s_3} \end{array} \tag{12}$$

hold, with fixed constants $\delta > 0$, $s_1, s_2, s_3 > 1$, $s_o \geq 1$, where

$$m_k(\alpha) = \alpha\nabla F(x_k)^T\Delta x_k,$$

then the trial point must satisfy the Armijo condition with respect to the optimality measure

$$F(x_k(\alpha_{k,l})) \leq F(x_k) + \eta_1\, m_k(\alpha_{k,l}), \tag{13}$$

instead of (11) to be acceptable. Here, $\eta_1 \in (0, 0.5)$ is a constant.

According to previous publications on filter methods (for example [13]), a trial step size $\alpha_{k,l}$ is called a F-step if (13) holds. Similarly, if a F-step is accepted as the final step size α_k in iteration k, then k is referred to as a F-type iteration.

3.2 The four-dimensional filter

The filter is a set that contains combinations of the four measures θ_{pf}, θ_c, θ_{df} and F that are prohibited for a successful trial point and is initialized to

$$\overline{F}_0 \subseteq \left\{(\theta_{pf}, \theta_c, \theta_{df}, F) \in \mathbb{R}^4 : \theta_{pf} \geq \theta_{pf}^{\max}, \theta_c \geq \theta_c^{\max}, \theta_{df} \geq \theta_{df}^{\max}, F \geq F^{\max}\right\}, \tag{14}$$

for the nonnegative constants $\theta_{pf}^{\max}$, $\theta_c^{\max}$, $\theta_{df}^{\max}$ and $F^{\max}$. The filter is updated according to

$$\begin{array}{rl} \overline{F}_{k+1} & = \overline{F}_k \cup \left\{(\theta_{pf}, \theta_c, \theta_{pf}, F) \in \mathbb{R}^4 : \theta_{pf} \geq (1-\gamma_1)\,\theta_{pf}(u_k^1) \text{ and } \theta_c \geq (1-\gamma_2)\,\theta_c(u_k^2)\right. \\ & \left. \text{and } \theta_{df} \geq (1-\gamma_3)\,\theta_{df}(u_k^3) \text{ and } F \geq F(x_k) - \gamma_4\theta_{pf}(u_k^1)\right\}, \end{array} \tag{15}$$

 ISBN: 978-84-612-9727-6

whenever the accepted step size satisfies (11). However, when for the accepted step size the conditions (12) and (13) hold, the filter remains unchanged.

Finally, when the backtracking line search cannot find a trial step size $\alpha_{k,l}$ that satisfies the above criteria, we define a minimum desired step size $\alpha_k^{\min}$, using linear models of the involved functions,

$$\alpha_k^{\min} = \xi \begin{cases} \min\{\gamma_1, \pi_1, \pi_2, \pi_3, \pi_4\}, \text{ if } m_k(\alpha_{k,l}) < 0 \\ \qquad \text{and } (\theta_{pf}(u_k^1) \leq \theta_{pf}^{\min} \text{ or } \theta_c(u_k^2) \leq \theta_c^{\min} \text{ or } \theta_{df}(u_k^3) \leq \theta_{df}^{\min}) \\ \min\{\gamma_1, \pi_1\}, \text{ if } m_k(\alpha_{k,l}) < 0 \\ \qquad \text{and } (\theta_{pf}(u_k^1) > \theta_{pf}^{\min} \text{ and } \theta_c(u_k^2) > \theta_c^{\min} \text{ and } \theta_{df}(u_k^3) > \theta_{df}^{\min}) \\ \gamma_1, \text{ otherwise} \end{cases} \tag{16}$$

where

$$\pi_1 = \frac{\gamma_4 \theta_{pf}(u_k^1)}{-m_k(\alpha_{k,l})}, \quad \pi_2 = \frac{\delta\left[\theta_{pf}(u_k^1)\right]^{s_1}}{[-m_k(\alpha_{k,l})]^{s_o}}, \quad \pi_3 = \frac{\delta\left[\theta_c(u_k^2)\right]^{s_2}}{[-m_k(\alpha_{k,l})]^{s_o}}, \quad \pi_4 = \frac{\delta\left[\theta_{df}(u_k^3)\right]^{s_3}}{[-m_k(\alpha_{k,l})]^{s_o}}$$

for positive constants $\theta_{pf}^{\min}, \theta_c^{\min}, \theta_{df}^{\min}$ and a safety factor $\xi \in (0, 1]$.

Like in [15] and whenever the backtracking line search finds a trial step size $\alpha_{k,l} < \alpha_k^{\min}$, the algorithm reverts to a restoration phase. Here, the algorithm tries to find a new iterate u_{k+1} that is acceptable to the current filter, *i.e.*, (11) holds, by reducing either the primal feasibility measure or the complementarity within an iterative process.

3.3 Restoration phase

The task of the restoration phase is to compute a new iterate acceptable to the filter by decreasing either the primal feasibility or the complementarity, whenever the backtracking line search procedure cannot make sufficient progress and the step size becomes too small. Thus, the restoration algorithm works with the new functions

$$\theta_{pf}^2(u^1) = \frac{1}{2}\|h(x) - w\|_2^2 \text{ and } \theta_c^2(u^2) = \frac{1}{2}\|Wy\|_2^2$$

and the steps Δ^1 and Δ^2 that are descent directions for $\theta_{pf}^2(u^1)$ and $\theta_c^2(u^2)$, respectively.

Using a backtracking strategy, the algorithm selects, at each iteration k, a step size $\alpha_k \in (0, \alpha_k^{\max}]$ to define a new trial point $u_k(\alpha_k) = u_k + \alpha_k \Delta_k$ that satisfies either

$$\theta_{pf}^2(u_k^1(\alpha_k)) \leq \theta_{pf}^2(u_k^1) + \alpha_k \eta_2 \nabla \theta_{pf}^2(u_k^1)^T \Delta_k^1$$

or

$$\theta_c^2(u_k^2(\alpha_k)) \leq \theta_c^2(u_k^2) + \alpha_k \eta_3 \nabla \theta_c^2(u_k^2)^T \Delta_k^2$$

for constants η_2 and η_3 in the set $(0, 0.5)$.

3.4 Setting the barrier parameter

To guarantee a positive decreasing sequence of μ values, the barrier parameter is updated by a formula that couples the theoretical requirement defined on the first order KKT conditions (5) with a simple heuristic. Thus, μ is updated by

$$\mu_{k+1} = \max\left\{\epsilon, \min\left\{\kappa_\mu \mu_k, \delta_\mu \frac{w_{k+1}^T y_{k+1}}{m}\right\}\right\} \tag{17}$$

where the constants $\kappa_\mu, \delta_\mu \in (0,1)$ and the tolerance ϵ is used to prevent μ from becoming too small so avoiding numerical difficulties at the end of the iterative process.

3.5 Termination criteria

The termination criteria consider dual and primal feasibility and complementarity measures

$$\max\left\{\frac{\|\nabla F(x) - A(x)^T y\|_\infty}{s}, \|h(x) - w\|_\infty, \frac{\|Wy\|_\infty}{s}\right\} \leq \epsilon_{tol}, \tag{18}$$

where

$$s = \max\left\{1, 0.01\frac{\|y\|_1}{m}\right\}$$

and $\epsilon_{tol} > 0$ is the error tolerance.

4 Experimental results and conclusions

To test this interior point framework with the herein proposed four-dimensional filter line search technique we selected 109 constrained problems from the Hock and Schittkowski (HS) collection [8]. This preliminary selection aims to consider small and simple to code problems. The tests were done in double precision arithmetic with a Pentium 4. The algorithm is coded in the C programming language and includes an interface to AMPL to read the problems that are coded in the AMPL modeling language [6].

Our algorithm is a quasi-Newton based method in the sense that a symmetric positive definite quasi-Newton BFGS approximation, B_k, is used to approximate the Hessian of the Lagrangian H, at each iteration k. In the first iteration, we may set $B_0 = I$ or $B_0 =$ positive definite modification of $\nabla^2 F(x_0)$, depending on the characteristics of the problem to be solved.

4.1 Initial approximations

The algorithm implements two alternatives to initialize the primal and the dual variables. One uses the usual published initial values, x_0, as mentioned in [8], and sets all the dual variables to one. The other uses the published x_0 to define the initial dual variables, y_0, and new primal variables, $\tilde{x}_0$, by solving the simplified reduced system:

$$\begin{bmatrix} -(B_0 + I) & A^T(x_0) \\ A(x_0) & I \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \nabla F(x_0) \\ 0 \end{bmatrix}.$$

 ISBN: 978-84-612-9727-6

Further, if $\|y_0\|_\infty > 10^3$ then y_0 is component by component set to one. However, if $\|\tilde{x}_0\|_\infty > 10^3\|x_0\|_\infty$ then $\tilde{x}_0 = x_0$.

The nonnegativity of the initial slack variables are ensured by computing $w_0 = \max\{|h(x_0)|, \epsilon_w\}$, for the previously defined x_0, and a fixed positive constant ϵ_w.

4.2 Setting user defined parameters

The chosen values for some of the constants are similar to the ones proposed in [15]: $\theta_{pf}^{\max} = 10^4 \max\left\{1, \theta_{pf}(u_0^1)\right\}$, $\theta_{pf}^{\min} = 10^{-4} \max\left\{1, \theta_{pf}(u_0^1)\right\}$, $\theta_c^{\max} = 10^4 \max\left\{1, \theta_c(u_0^2)\right\}$, $\theta_c^{\min} = 10^{-4} \max\left\{1, \theta_c(u_0^2)\right\}$, $\theta_{df}^{\max} = 10^4 \max\left\{1, \theta_{df}(u_0^3)\right\}$, $\theta_{df}^{\min} = 10^{-4} \max\left\{1, \theta_{df}(u_0^3)\right\}$, $F^{\max} = 10^4 \max\left\{1, F(x_0)\right\}$, $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 10^{-5}$, $\delta = 1$, $s_1 = s_2 = s_3 = 1.1$, $s_o = 2.3$, $\eta_1 = \eta_2 = \eta_3 = 10^{-4}$, $\xi = 0.05$.

The other parameters are set as follows: $\varepsilon = 0.95$, $\delta_\mu = \kappa_\mu = 0.1$, $\epsilon = 10^{-9}$, $\epsilon_w = 0.01$ and $\epsilon_{tol} = 10^{-6}$.

4.3 Comparative results

Table 2 summarizes the results obtained with the herein proposed four-dimensional filter line search interior point method. The table reports the number of iterations required to obtain a solution according to the termination criteria in (18), Nit, and the objective function value, $F(x^*)$. Except in 11 problems, the number of function evaluations was $Nit+1$. Results inside parentheses were obtained with the parameter ϵ_{tol} set to $= 10^{-4}$. In all problems, our algorithm converges to the solution within a reasonable number of iterations.

For a comparative purpose we compare our results with the IPOPT, a filter line search barrier based method [13, 14, 15]. The results obtained by IPOPT are reported in the file "Ipopt-table.pdf" under
`http://www.research.ibm.com/people/a/andreasw/papers/Ipopt-table.pdf`.

We noticed differences, some are rather small, in the objective function value in 32 problems. They are listed in Table 3. For the remaining problems used in this study, the herein proposed filter line search interior point method converges to the solutions reported in "Ipopt-table.pdf". The table reveals that we were able to get better solutions in eight problems. They are *emphasized* in the table. We may then conclude that the four-dimensional filter line search interior point based method is effective in reaching the solution of small nonlinear constrained optimization problems. In the future, different combinations of the criteria involved in the first order optimality conditions (4) will be analyzed, tested and compared with the present proposal.

References

[1] H.Y. Benson, R.J. Vanderbei and D.F. Shanno, *Interior-point methods for nonconvex nonlinear programming: filter methods and merit functions*, Computational Optimization and Applications, **23** (2002) 257–272.

Table 2: Number of iterations and function values of our study

Prob	Nit	$F(x^*)$	Prob	Nit	$F(x^*)$	Prob	Nit	$F(x^*)$
hs001	36	1.77130e-14	hs038	54	2.24116e-15	hs077	(39)	2.41505e-01
hs002	10	5.04261e-02	hs039	13	-1.00000e+00	hs078	12	-2.91970e+00
hs003	4	9.91001e-07	hs040	9	-2.50000e-01	hs079	10	7.87768e-02
hs004	6	2.66667e+00	hs041	15	1.92593e+00	hs080	11	5.39498e-02
hs005	8	-1.91322e+00	hs042	9	1.38579e+01	hs081	14	5.39498e-02
hs006	10	1.01337e-15	hs043	18	-4.40000e+01	hs083	16	-3.06655e+04
hs007	9	-1.73205e+00	hs044	24	-1.50000e+01	hs084	(21)	-5.28034e+06
hs008	10	-1.00000e+00	hs045	10	1.00000e+00	hs086	12	-3.23487e+01
hs009	15	-5.00000e-01	hs046	(15)	1.50567e-08	hs087	43	8.82760e+03
hs010	11	-1.00000e+00	hs047	24	1.98162e-09	hs088	35	1.36266e+00
hs011	10	-8.49846e+00	hs048	9	5.75184e-15	hs089	45	1.36266e+00
hs012	17	-3.00000e+01	hs049	(18)	4.98460e-07	hs090	39	1.36266e+00
hs014	9	1.39346e+00	hs050	9	6.20897e-15	hs091	45	1.36266e+00
hs015	12	3.06500e+02	hs051	9	2.88300e-15	hs092	40	1.36266e+00
hs016	11	2.50000e-01	hs052	11	5.32665e+00	hs093	19	1.35076e+02
hs017	10	1.00000e+00	hs053	12	4.09302e+00	hs095	14	1.56195e-02
hs018	12	5.00000e+00	hs054	13	1.92857e-01	hs096	13	1.56199e-02
hs019	35	-6.96181e+03	hs055	14	6.66667e+00	hs097	16	3.13581e+00
hs020	11	3.81987e+01	hs056	(9)	-8.88237e-10	hs098	17	3.13581e+00
hs021	9	-1.00000e+02	hs057	9	3.06476e-02	hs099	28	-8.31080e+08
hs022	8	1.00000e+00	hs059	12	-6.74950e+00	hs100	37	6.80630e+02
hs023	11	2.00000e+00	hs060	13	3.25682e-02	hs101	45	1.80976e+03
hs024	11	-1.00000e+00	hs061	16	-1.43646e+02	hs102	36	9.11880e+02
hs025	29	1.27017e-16	hs062	26	-2.62725e+04	hs103	36	5.43668e+02
hs026	(21)	1.83530e-07	hs063	13	9.61715e+02	hs104	15	3.95116e+00
hs027	21	4.00000e-02	hs064	51	6.29984e+03	hs105	48	1.13630e+03
hs028	7	1.38756e-13	hs065	9	9.53529e-01	hs106	60	7.04925e+03
hs029	14	-2.26274e+01	hs066	10	5.18164e-01	hs107	83	5.05501e+03
hs030	8	1.00000e+00	hs067	23	-1.16203e+03	hs108	25	-8.66025e-01
hs031	11	6.00000e+00	hs070	30	8.92318e-03	hs110	7	-4.57785e+01
hs032	14	1.00000e+00	hs071	13	1.70140e+01	hs112	60	-4.77611e+01
hs033	12	-4.58579e+00	hs072	20	7.27679e+02	hs113	17	2.43062e+01
hs034	12	-8.34032e-01	hs073	11	2.98944e+01	hs114	24	-1.76880e+03
hs035	8	1.11111e-01	hs074	18	5.12650e+03	hs116	89	9.75875e+01
hs036	14	-3.30000e+03	hs075	18	5.17441e+03	hs117	21	3.23487e+01
hs037	12	-3.45600e+03	hs076	8	-4.68182e+00	hs118	17	6.64820e+02
						hs119	23	2.44900e+02

 ISBN: 978-84-612-9727-6

Table 3: Function values obtained by IPOPT

Prob	$F(x^*)$	Prob	$F(x^*)$	Prob	$F(x^*)$
hs001	5.82781e-16	hs048	7.88861e-31	hs090	1.36265e+00
hs002	*4.94123e+00*	hs049	1.06000e-11	hs091	1.36265e+00
hs003	-7.49410e-09	hs050	0.00000e+00	hs092	1.36265e+00
hs016	2.31447e+01	hs051	4.93038e-32	hs095	1.56177e-02
hs025	*1.03460e-15*	hs054	-9.08075e-01	hs096	1.56177e-02
hs026	1.29138e-16	hs055	*6.77451e+00*	hs097	*4.07124e+00*
hs028	3.08149e-31	hs056	-3.45600e+00	hs098	*4.07124e+00*
hs038	3.34111e-19	hs059	*-7.80279e+00*	hs105	1.04461e+03
hs044	*-1.30000e+01*	hs070	7.49846e-03	hs108	*-6.74981e-01*
hs046	8.55335e-16	hs088	1.36265e+00	hs110	-9.96e+39
hs047	6.57516e-14	hs089	1.36265e+00		

[2] M.F.P. Costa and E.M.G.P. Fernandes, *Comparison of interior point filter line search strategies for constrained optimization by performance profiles*, International Journal of Mathematics Models and Methods in Applied Sciences, **1** (2007) 111–116.

[3] M.F.P. Costa and E.M.G.P. Fernandes, *Practical implementation of an interior point nonmonotone line search filter method*, International Journal of Computer Mathematics **85** (2008) 397–409.

[4] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Mathematical Programming **91** (2002) 239–269.

[5] R. Fletcher, S. Leyffer and P. Toint, *A brief history of filter methods*, Report ANL/MCS-P1372-0906, Argonne National Laboratory 2006.

[6] R. Fourer, D.M. Gay and B. Kernighan, *A modeling language for mathematical programming*, Management Science **36** (1990) 519–554.

[7] N.I.M. Gould, D. Orban, A. Sartenaer and P. L. Toint, *Superlinear convergence of primal-dual interior point algorithms for nonlinear programming*, SIAM Journal on Optimization **11** (2001) 974–1002.

[8] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming*, Springer-Verlag, 1981.

[9] D.F. Shanno and R.J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods*, Mathematical Programming B, **87** (2000) 303–316.

[10] M. Ulbrich, S. Ulbrich and L.N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*, Mathematical Programming, **100** (2004) 379–410.

[11] R.J. Vanderbei, *LOQO: An interior-code for quadratic programming*, Technical report SOR-94-15, (1998) Princeton University, Statistics and Operations Research.

[12] R.J. Vanderbei and D.F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications, **13** (1999) 231–252.

[13] A. Wächter and L.T. Biegler, *Line search filter methods for nonlinear programming: motivation and global convergence*, SIAM Journal on Optimization, **16** (2005) 1–31.

[14] A. Wächter and L.T. Biegler, *Line search filter methods for nonlinear programming: local convergence*, SIAM Journal on Optimization, **16** (2005) 32–48.

[15] A. Wächter and L.T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming, **106** (2007) 25–57.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Harvesting in an ecoepidemic model with species-barrier crossing

Andrea Costamagna[1] **and Ezio Venturino**[1]

[1] *Dipartimento di Matematica "Giuseppe Peano", Università di Torino, Italia*

emails: rui84_mathe@yahoo.it, ezio.venturino@unito.it

Abstract

In this paper we consider predator-prey ecoepidemic models in which the disease can spread from one population to the other one. Either one or both are assumed to be subject to harvesting.

Key words: ecoepidemics, predator-prey, diseases, harvesting
MSC 2000: AMS 92D25, 92D30, 92D40

1 Introduction

The paper is organized as follows. In order to explain what ecoepidemic models are, we present at first a brief outline of population theory and its recent developments. Therefore in the three subsections of the Introduction, we review single and interacting population models and the basis of mathematical epidemic theory. In Section 2 we present the basic ecoepidemic model with the disease breaking the species barrier. In the following Section we discuss the simplest ecoepidemic system with linear harvesting being carried out only on the prey. Section 4 contains instead the analysis of a linear harvesting term of the predators. We then examine the results of a diminishing return harvesting policy applied only to sound prey. Section 6 contains instead the same term for the infected prey. A final discussion of the results concludes the paper.

1.1 Single population models

The basic single population models, [16], are the Malthus model, [17]

$$\frac{dP}{dt} = R(P) = rP$$

and its logistic correction, due to Verhulst, [29, 30, 31] also sometimes called Holling type I:

$$\frac{dP}{dt} = R(P) = r(P)P = rP\left(1 - \frac{P}{K}\right)$$

to distinguish it from the similar model with a bounded return function, known as Michaelis-Menten or Holling type II dynamics,

$$r(P) = \frac{P}{K+P} \ .$$

In all these and the subsequent models, the parameters have to be understood as being nonnegative unless otherwise specified.

1.2 Interacting population models

We need at least two populations in these systems, which for simplicity we denote as P, the predators and Q the prey. Other types of interactions can be specified, but in general they basically entail only changes in the signs of the interaction terms. They originated long ago, [3, 15, 32]. Assuming logistic growth for the prey and a mass action type of interaction with the predators, the model reads

$$Q' = rQ(1-\frac{Q}{K}) - aQP, \quad P' = gP + aeQP - bP^2 \ .$$

Note that also quadratic mortality at rate $b \neq 0$ is here assumed for the predators, indicating intraspecific competition for resources. Moreover, $g > 0$ indicates that other food sources are available to the predators, while for $g < 0$ they are not. Assuming instead a Holling type II return function for instance, the system would change to model the fact that the too much abundant food is eventually disregarded,

$$Q' = rQ(1-\frac{Q}{K}) - aP\frac{Q}{c+Q} \ , \quad P' = gP + aeP\frac{Q}{c+Q} - bP^2 \ .$$

More complicated food chains can be formed, with several trophic levels in which the populations in the intermediate levels are at the same time predators for those at the lower level and prey of those on top of them. The bottom layer, the phytoplankton in the ocean and the vegetables on land, and a top predator, in general ultimately this can be regarded as the man, complete the picture. For some such or more complicated situations, see [19, 20].

1.3 Epidemic models

The purpose of mathematical epidemiology is the study of disease propagation, with the possible aim to fight, control and eradicate them. The basic problem is in fact that newly infected individuals are in general able to spread the disease much earlier than the time at which their symptoms appear. The first such model is [14]. These infected individuals are therefore not recognizable, and this fact entails the need of mathematical modelling and simulation for investigating the possible outcomes of an epidemic.

Here a single population, of fixed size in the classical models, is partitioned among several classes according to their respective status with respect to an epidemic spreading by contact among individuals. In the simplest case, let S denote the susceptibles, I the

 ISBN: 978-84-612-9727-6

infectious individuals and let $N = S + I$ be the fixed population size. Let $\beta(I)$ denote the contact rate; in the linear case, $\beta(I) = bI$. The trivial SI model is then, [5]

$$S' = -bIS \ , \quad I' = bIS \ .$$

The standard incidence model contains a nonlinear contact rate $\beta(I) = b\frac{I}{N}$; let us introduce also recovery of the disease at rate γ, to get the SIS model

$$S' = -\frac{b}{N}SI + \gamma I \ , \quad I' = \frac{b}{N}SI - \gamma I \ .$$

More general models can be obtained by introducting also the class of the recovered, or quarantined, individuals R,

$$S' = -\frac{b}{N}SI + \gamma R \ , \quad I' = \frac{b}{N}SI - \delta I \ , \quad R' = \delta I - \gamma R \ .$$

More advanced models then relax the fixed population size assumption [6, 18] and investigate much more complex phenomena, such as epidemic waves, [21] or prophilactic measure, [22]. For a fairly recent account, see for instance the review [10].

2 The ecoepidemic model

Ecoepidemic models merge the two previous aspects, [7, 1, 23, 13, 28]. The basic model of interest here can be written in a general way as follows, see [2]

$$\begin{aligned} \dot{p} &= p(r_1 - \frac{p+u}{K} - b_1(q+v) - \gamma u - \beta v) \\ \dot{u} &= u(r_2 - \frac{p+u}{K} - b_2(q+v) + \gamma p) + \beta p v \\ \dot{q} &= q(-m + e(b_1 p + b_2 u) - \alpha u - \eta v) \\ \dot{v} &= v(-m + e(b_1 p + b_2 u) + \eta q) + \alpha u q \end{aligned} \tag{1}$$

where in each of its respective term the first equation describes the evolution of sound prey p, accounting for logistic reproduction, infection both by infected prey and infected predators and finally predation by sound and infected predators. The second one contains the dynamics of infected prey u, which are also able to reproduce logistically, though at a different rate than for sound individuals; they are hunted by both types of predators, and enter this class from the sound class via interactions with diseased prey or predator individuals. The third equation describes the sound predators q, with natural mortality m, feeding on both susceptible and infected prey and being subject to catching the disease via contacts with infected prey and predators. Finally the last equation shows the evolution of infected predators v, which die at the same rate as for sound ones, i.e. we do not take into consideration a disease-related mortality here, hunt both sound and infected prey, and enter into this class via "successful" contacts with infected predators and prey.

The meaning of the remaining parameters is as follows. The net reproduction rates for sound and infected prey are r_1 and r_2 respectively, K denotes their environment

 ISBN: 978-84-612-9727-6

capacity, b_1 and b_2 are the rates at which they are hunted, in general we can assume that $b_2 \geq b_1$, since infected prey should be weaker and therefore easier to catch. Further γ is the disease incidence for intraspecific contacts and β is the one related to contacts with infected predators, e is the conversion factor for predators, η and α are the incidence rates for predators, respectively for intraspecific contacts as well as for contacts with infected prey.

Various other situations can be described in this framework, when the disease cannot spread from one population to the other one, and have already been modeled, see for instance [11, 12, 8, 4, 24, 25, 26, 9], while others can be found in the cited references of [9, 27, 16]. Note that classical epidemic models would be modeled by the first or the last pair of equations, and classical predator-prey models would not contain the second and fourth equations.

In order to minimize the number of parameters, the model can be suitably rescaled as follows. Let $y_1 = \theta p$, $y_2 = \phi u$, $y_3 = \psi q$, $y_4 = \omega v$, $\tau = \sigma t$, and furthermore using the notations $\sigma = m$, $\phi = \frac{\gamma}{m}$, $\omega = \frac{\beta}{m}$, $\theta = \frac{eb_1}{m}$, $\psi = \frac{b_1}{m}$ and $\frac{\gamma}{eb_1} = A$, $\frac{1}{K\gamma} = B$, $\frac{b_1}{\beta} = C$, $\frac{b_2}{\beta} = D$, $\frac{\alpha}{\gamma} = E$, $\frac{\eta}{\beta} = F$ and finally $y_1 = p$, $y_2 = u$, $y_3 = q$, $y_4 = v$, we arrive at the rescaled model

$$\begin{aligned} \dot{p} &= \frac{r_1 p}{m} - ABp^2 - Bpu - pu - pv - pq - Cpv \ , \\ \dot{u} &= \frac{r_2 u}{m} - ABpu - Bu^2 + Apu - \frac{Duq}{C} - Duv + mApv \ , \\ \dot{q} &= -q - Euq - Fqv + pq + \frac{ADuq}{C} \ , \\ \dot{v} &= -v + \frac{Fqv}{C} + pv + \frac{ADuv}{C} + \frac{Euq}{C} \ . \end{aligned} \tag{2}$$

3 Linear harvesting of prey

We will introduce into (1) harvesting terms. We begin by adding a linear such term in the prey equation, namely $-hp$ e $-hu$ where h represents the harvesting effort. After suitable rescaling, as done for (2) the model becomes

$$\begin{aligned} \dot{p} &= \frac{r_1 p}{m} - ABp^2 - Bpu - pu - pv - pq - Cpv - \frac{hp}{m} \ , \\ \dot{u} &= \frac{r_2 u}{m} - ABpu - Bu^2 + Apu - \frac{Duq}{C} - Duv + mApv - \frac{hu}{m} \ , \\ \dot{q} &= -q - Euq - Fqv + pq + \frac{ADuq}{C} \ , \\ \dot{v} &= -v + \frac{Fqv}{C} + pv + \frac{ADuv}{C} + \frac{Euq}{C} \ . \end{aligned} \tag{3}$$

 ISBN: 978-84-612-9727-6

Its Jacobian matrix is

$$\begin{pmatrix} J_{11} & -p(B+1) & -p & -p(1+C) \\ u(-AB+A)+mAv & J_{22} & -\frac{Du}{C} & -Du+mAp \\ q & q(-E+\frac{AD}{C}) & J_{33} & -Fq \\ v & \frac{ADv+Eq}{C} & \frac{Fv+Eu}{C} & J_{44} \end{pmatrix} \tag{4}$$

where

$$J_{11}=\frac{r_1}{m}-2ABp-Bu-u-v-q-Cv-\frac{h}{m}\ , \quad J_{33}=-1-Eu-Fv+p+\frac{ADu}{C}\ ,$$
$$J_{22}=\frac{r_2}{m}-ABp-2Bu+Ap-\frac{Dq}{C}-Dv-\frac{h}{m}\ , \quad J_{44}=\frac{Fq}{C}-1+p+\frac{ADu}{C}\ .$$

The equilibria are found as follows. The origin has eigenvalues $\frac{r_1-h}{m}, \frac{r_2-h}{m}, -1, -1$ and therefore it is locally asymptotically stable if

$$h>\max(r_1,r_2). \tag{5}$$

The second one contains only the infected prey, $B2=\left(0,\frac{r_2-h}{Bm},0,0\right)$, feasible for

$$H_{B2}=r_2-h\geq 0\ . \tag{6}$$

The eigenvalues of (4) evaluated at $B2$ are

$$\frac{Br_1-Br_2-r_2+h}{Bm}, \quad -\frac{r_2-h}{m}, \quad \frac{(r_2-h)(AD-EC)-BmC}{BmC}, \quad \frac{(r_2-h)AD-BmC}{BmC},$$

so that its stability conditions reduce to

$$B(r_1-r_2)<H_{B2}; \qquad ADH_{B2}<BmC\ . \tag{7}$$

The equilibrium $B3=(\frac{r_1-h}{ABm},0,0,0)$ contains only sound prey. It is feasible if

$$H_{B3}=r_1-h\geq 0\ . \tag{8}$$

From the eigenvalues of the Jacobian (4) evaluated at $B3$,

$$-\frac{r_1-h}{m}\ , \quad \frac{Br_2-Br_1+r_1-h}{Bm}\ , \quad \frac{r_1-ABm-h}{ABm}\ , \quad \frac{r_1-ABm-h}{ABm}\ ,$$

we find the stability conditions to be

$$H_{B3}-ABm<0; \qquad H_{B3}+B(r_2-r_1)<0\ . \tag{9}$$

The predator-free equilibrium is $B4=\left(\frac{Br_1-Br_2-r_2+h}{Am},\frac{Br_2-Br_1+r_1-h}{m},0,0\right)$. The feasibility conditions are

$$H_{B4_1}=Br_2-Br_1+r_1-h\geq 0, \quad H_{B4_2}=Br_1-Br_2-r_2+h\geq 0\ . \tag{10}$$

 ISBN: 978-84-612-9727-6

Since in this case the Jacobian matrix can be block factorized, the characteristic equation has the explicit form

$$(\lambda - J_{33})(\lambda - J_{44})(\lambda^2 - (J_{11} + J_{22})\lambda + J_{11}J_{22} - J_{21}J_{12}) = 0 \ .$$

To find purely imaginary eigenvalues, we need to impose $J_{11} + J_{22} = -\frac{B(r_1 - r_2)}{m} = 0$; it follows then $B = 0$ since $r_1 = r_2$ does not lead to suitable conditions. Substituting the former into the Jacobian we find the characteristic equation

$$\begin{aligned}&[\lambda CAm + ECA(r_1 - h) + C(r_2 - h) + A^2D(h - r_1 + ACm)][\lambda CAm \\ &+C(r_2 - h) + A^2D(h - r_1) + CAm](\lambda^2 m^2 - (-r_1 + h)(-r_2 + h)) = 0 \ .\end{aligned}$$

If we require

$$\begin{aligned}ECA(r_1 - h) + C(r_2 - h) + A^2D(h - r_1) + ACm > 0 \ , \qquad (11)\\ C(r_2 - h) + A^2D(h - r_1) + CAm > 0 \ , \quad -(-r_1 + h)(-r_2 + h) > 0 \ ,\end{aligned}$$

then a Hopf bifurcation arises. To verify it also numerically, we consider the following parameter values $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.6$, $h = 0.65$, for which both $B = 0$ and (11) hold. In this case the equilibrium attains the value $B4 = (0.1, 0.1, 0, 0)$. The simulations are run with $\{0.3, 0.2, 0.4, 0.3\}$ as initial conditions, see Figure 1.

The pandemic coexistence equilibrium $B5 = \left(0, \frac{C}{AD}, 0, \frac{AD(r_2 - h) - BmC}{AD^2 m}\right)$ with no sound survivors has the feasibility condition

$$H_{B5} = AD(r_2 - h) - BmC \geq 0 \ . \qquad (12)$$

The eigenvalues in this case are

$$\begin{aligned}-\frac{CBm \pm \sqrt{(CBm)^2 - 4mADH_{B5}}}{2mAD}, \quad \frac{-ECDm - FH_{B5}}{AD^2 m},\\ \frac{ADh(-D + 1 + C) - (1 + C)(-CBm + ADr_2) - D(-r_1 AD + CBm + Cm)}{AD^2 m}\end{aligned}$$

so that the stability condition is

$$(1 + C)(-CBm + ADr_2) + D(-r_1 AD + CBm + Cm) > ADh(-D + 1 + C) \ . \quad (13)$$

To obtain oscillations, we need two purely imaginary eigenvalues and the other two with negative real parts. Again in view of the structure of the Jacobian at $B5$, the characteristic polynomial is $(J_{11} - \lambda)(J_{33} - \lambda)(\lambda^2 - J_{22}\lambda - J_{42}J_{24})$. We thus need to impose $J_{22} = \frac{BC}{AD} = 0$; thus the first condition is $B = 0$, since $C = 0$ cannot be taken as it appears in some denominators. The characteristic polynomial then simplifies to $(\lambda mDA - r_1 DA + mC + Ar_2 - Ah + CAr_2 - ACh + ADh)(\lambda mDA + ECm + r_2 AF - AFh)(\lambda^2 m + r_2 - h)$ for which the roots are as required provided that

$$-r_1 DA + mC + Ar_2 - Ah + CAr_2 - ACh + ADh > 0, \quad r_2 - h > 0 \ . \qquad (14)$$

 ISBN: 978-84-612-9727-6

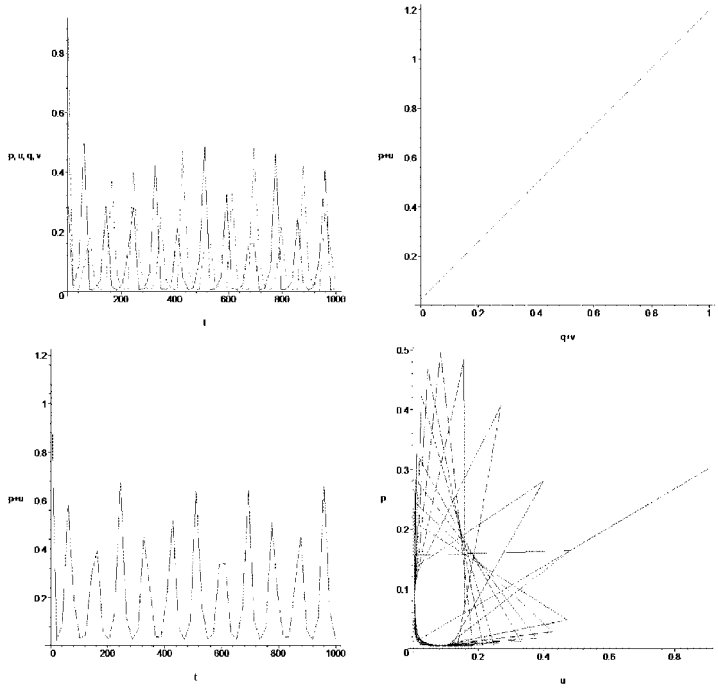


Figure 1: Solutions near $B4$ for the parameter values $A = 1$, $B = 0$, $C = 0.9$,$E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.6$, $h = 0.65$, with initial condition $(0.3, 0.2, 0.4, 0.3)$. Top left: the solutions as function of time; Top right: solution projections onto the total predator-prey phase plane, $q+v/p+u$; Bottom left: the total prey $p + u$ as function of time; Bottom right: solution projections onto the infected prey—sound prey phase plane, u/p.

These results are verified by simulations, see Figure 2 for the parameter values $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.6$, $h = 0.5$, which satisfy both $B = 0$ and (14). The equilibrium is $B5 = (0, 0.75, 0, 0.1\overline{6})$. We take the initial conditions $(0.3, 0.9, 0.4, 0.6)$.

In the next equilibrium only sound populations survive $B6 = (1, 0, \frac{r_1 - ABm - h}{m}, 0)$, with feasibility condition

$$H_{B6} = -r_1 + ABm + h \leq 0 \ . \tag{15}$$

The eigenvalues in this case are

$$-\frac{ABm \pm \sqrt{(ABm)^2 + 4mH_{B6}}}{2m}, \quad \frac{P_{B6} \pm \sqrt{P_{B6}^2 - 4Q_{B6}}}{J_{B6}}$$

where P_{B6}, Q_{B6}, J_{B6} are complicated expressions that we omit. The stability condition becomes

$$(D - F)H_{B6} - C(ABm - r_2 + h - Am) < 0 \ ;$$
$$ACEm^2 - DFH_{B6} + CF(ABm - r_2 + h - Am) < 0 \ .$$

 ISBN: 978-84-612-9727-6

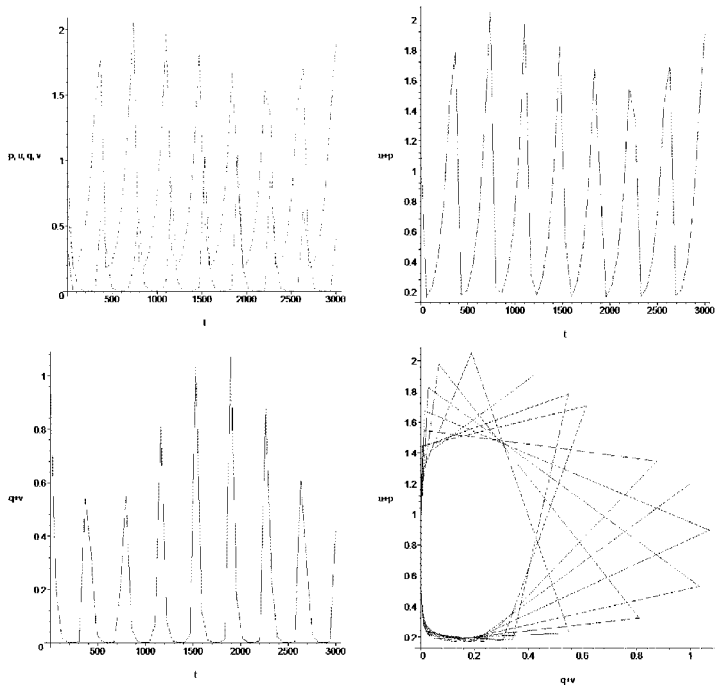


Figure 2: Solutions around $B5$ for the parameters $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.6$, $h = 0.5$ and initial condition $(0.3, 0.9, 0.4, 0.6)$. Top left: solutions as function of time; Top right: total prey population $p+u$ as function of time; Bottom left: total predator population $q+v$ as function of time; Bottom right: solution projection onto the total predator-prey population phase plane, $q + v/p + u$.

But from the second one, we find $ABm - r_2 + h - Am < 0$ for which from the first one $DH_{B6} - C(ABm - r_2 + h - Am) < 0$ follows, i.e. the second condition can never be satisfied. In conclusion $B6$ is always unstable.

There is another equilibrium, namely $B7 = (p_{B7}, u_{B7}, 0, v_{B7})$, in which the disease remains endemic in the prey and the latter survive together with the sound predators. The populations levels are very much complicated to write down analytically. Finally there is also the coexistence of the whole ecosystem, $B8 = (p_{B8}, u_{B8}, -v_{B8}, v_{B8})$, but it is clearly infeasible.

Remarks. Contrary to the basic model with no harvesting, [2], here the origin can be rendered stable. Denoting by Q_i the equilibria corresponding in case of absence of harvesting to the various B_i here found, [2], we can make the following considerations. The equilibria $B2$ and $B3$ are not always feasible, as it occurs instead for $Q2$ and $Q3$. In particular the first one is infeasible for $h > r_2$, while the second one for $h > r_1$. Further, the coexistence equilibrium here is unfeasible, a fact which also occurs in the model without harvesting, [2].

4 Linear harvesting of predators

Using once again the harvesting rate h, the model in this case reads

$$\begin{aligned} \dot{p} &= \frac{r_1 p}{m} - ABp^2 - Bpu - pu - pv - pq - Cpv \ , \\ \dot{u} &= \frac{r_2 u}{m} - ABpu - Bu^2 + Apu - \frac{Duq}{C} - Duv + mApv \ , \\ \dot{q} &= -q - Euq - Fqv + pq + \frac{ADuq}{C} - \frac{hq}{m} \ , \\ \dot{v} &= -v + \frac{Fqv}{C} + pv + \frac{ADuv}{C} + \frac{Euq}{C} - \frac{hv}{m} \ , \end{aligned} \tag{16}$$

with Jacobian

$$\begin{pmatrix} J_{11} & -p(B+1) & -p & -p(1+C) \\ u(-AB+A)+mAv & J_{22} & -\frac{Du}{C} & -Du+mAp \\ q & q(-E+\frac{AD}{C}) & J_{33} & -Fq \\ v & \frac{ADv+Eq}{C} & \frac{Fv+Eu}{C} & J_{44} \end{pmatrix} \tag{17}$$

where

$$J_{11} = \frac{r_1}{m} - 2ABp - Bu - u - v - q - Cv \ , \quad J_{22} = \frac{r_2}{m} - ABp - 2Bu + Ap - \frac{Dq}{C} - Dv \ ,$$
$$J_{33} = -1 - Eu - Fv + p + \frac{ADu}{C} - \frac{h}{m} \ , \quad J_{44} = \frac{Fq}{C} - 1 + p + \frac{ADu}{C} - \frac{h}{m}$$

The equilibria are the origin, $C1 = (0,0,0,0)$, with eigenvalues

$$\frac{r_1}{m} \ , \frac{r_2}{m} \ , -1 - \frac{h}{m} \ , -1 - \frac{h}{m} \ ,$$

from which its inconditionate instability.

We then find the infected-prey only point $C2 = \left(0, \frac{r_2}{Bm}, 0, 0\right)$, whose eigenvalues are

$$\frac{Br_1 - Br_2 - r_2}{Bm} \ , \quad -\frac{r_2}{m} \ , \quad \frac{(r_2)(AD - EC) - CB(m+h)}{BmC} \ , \quad \frac{r_2 AD - BC(m+h)}{BmC}$$

giving the stability conditions

$$B(r_1 - r_2) < 0 \ , \qquad ADr_2 < BC(m+h) \ . \tag{18}$$

There is then the sound-prey only equilibrium $C3 = \left(\frac{r_1}{ABm}, 0, 0, 0\right)$, for which the eigenvalues are

$$-\frac{r_1}{m} \ , \quad \frac{Br_2 - Br_1 + r_1}{Bm} \ , \quad \frac{r_1 - AB(m+h)}{ABm} \ , \quad \frac{r_1 - AB(m+h)}{ABm} \ ,$$

 ISBN: 978-84-612-9727-6

with stability conditions

$$r_1 - AB(m+h) < 0 \; ; \qquad r_1 + B(r_2 - r_1) < 0 \; . \tag{19}$$

The equilibrium $C4 = \left(\frac{Br_1 - Br_2 - r_2}{Am}, \frac{Br_2 - Br_1 + r_1}{m}, 0, 0\right)$, in which the whole prey population survives, is feasible for

$$H_{C4_1} = Br_2 - Br_1 + r_1 \geq 0 \; , \quad H_{C4_2} = Br_1 - Br_2 - r_2 \geq 0 \; . \tag{20}$$

Note that from $H_{C4_1} > 0$, and $H_{C4_2} > 0$ it follows $r_1 - r_2 > 0$. The eigenvalues are

$$\frac{1}{AmC}[-AmC - EACH_{C4_1} + CH_{C4_2} + A^2DH_{C4_1} - hCA] \; ,$$
$$\frac{1}{AmC}[-AmC + CH_{C4_2} + A^2DH_{C4_1} - hCA] \; ,$$
$$\frac{1}{2m}\left[r_2B - r_1B \pm \sqrt{5(r_2B - r_1B)^2 - 4(Br_1^2 - r_1r_2 - Br_2^2}\right] \; .$$

The first two are nonnegative if $Z_{C4} = H_{C4_2}C + A^2DH_{C4_1} - AC(m+h) < 0$ while the last two are if $|B(r_2 - r_1)| > \sqrt{5B^2(r_2 - r_1)^2 - 4X_{C4}}$ where $X_{C4} = Br_1^2 - r_1r_2 - Br_2^2 = r_1H_{C4_2} + r_2H_{C4_2} + r_2^2$. It follows then that $X_{C4} > 0$ and the stability condition reduces to $B^2(r_2 - r_1)^2 > 5B^2(r_2 - r_1)^2 - 4X$. From the latter it follows $B^2(r_2 - r_1)^2 - X < 0$. Since $B^2(r_2 - r_1)^2 - X = -H_{C4_1}H_{C4_2}$ the latter is always negative, so that the only stability condition is

$$Z_{C4} < 0 \; . \tag{21}$$

The next equilibrium is the one in which the disease is endemic in both species, $C5 = \left(0, \frac{C(m+h)}{ADm}, 0, \frac{ADr_2 - BC(m+h)}{AD^2m}\right)$. The feasibility condition in this case is

$$H_{C5} = ADr_2 - BC(m+h) \geq 0 \; . \tag{22}$$

In what follows, we consider only the strict inequality (22). The eigenvalues are

$$\frac{1}{AD^2m}[r_1AD^2 - BCmD - BCDh - DmC - hCD - (1+C)H_{C5}] \; ,$$
$$\frac{1}{AD^2m}[-r_2DAF + FBCm + FBhC - ECmD - ECDh] \; ,$$
$$-\frac{1}{2mAD}\left[BC(m+h) \pm \sqrt{(BC(m+h))^2 - \frac{4A(m+h)H_{C5}}{m}}\right] \; ,$$

thus providing a single stability condition,

$$(1+C)H_{C5} + D(-r_1AD + CBm + Cm) < hC(-D - BD) \; . \tag{23}$$

To obtain a Hopf bifurcation, we need a pair of purely imaginary eigenvalues, while the remaining ones need to have a negative real part. From the form of the Jacobian at this equilibrium, the characteristic polynomial reads $(J_{11} - \lambda)(J_{33} - \lambda)(\lambda^2 - J_{22}\lambda - J_{42}J_{24})$. We therefore need to impose $J_{22} = -\frac{BC(m+h)}{ADm} = 0$ from which necessarily $B = 0$, since

 ISBN: 978-84-612-9727-6

C cannot vanish since it appears in the denominators of the Jacobian. Using this result in the Jacobian at $C5$, the characteristic polynomial simplifies to $(\lambda mDA - r_1DA + Cr_2A + mC + hC + Ar_2)(\lambda mDA + ECm + CEh + r_2AF)(\lambda^2m^2 + r_2m + r_2h)$, and the latter has purely imaginary roots if

$$-r_1DA + Cr_2A + mC + hC + Ar_2 > 0 \tag{24}$$

To verify numerically this result, we have run simulations with parameter values $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.5$, $h = 1$, which verify both the condition $B = 0$ as well as (24) and with initial values $(0.3, 0.9, 0.4, 0.6)$. The equilibrium is $C5 = (0, 2.25, 0, 0.8\overline{3})$. Figure 2 shows the results.

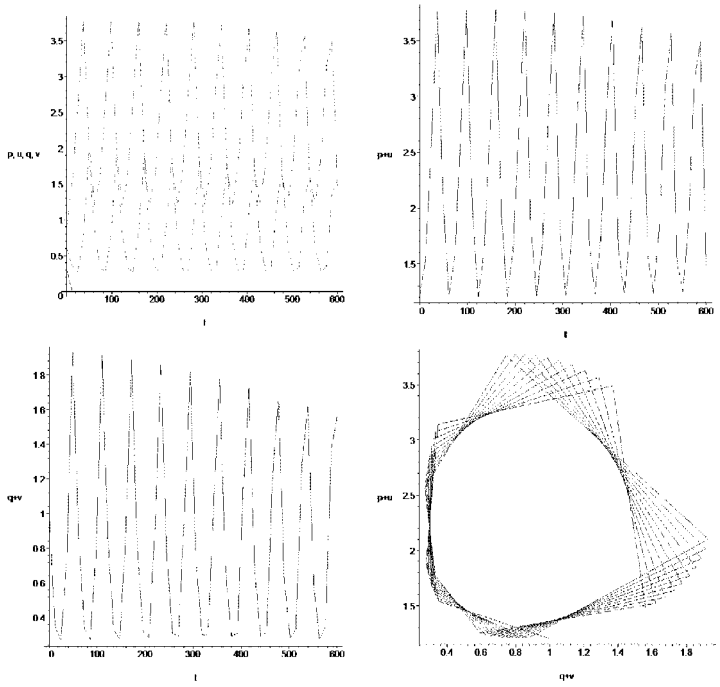


Figure 3: Simulations results near $C5$ with parameter values: $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.5$, $h = 1$, and initial values $(0.3, 0.9, 0.4, 0.6)$. Top left: solutions as function of time; Top right: total prey population $p + u$ as function of time; Bottom left: total predator population $q + v$ as function of time; Bottom right: solution projection onto the total predator-prey population phase plane, $q + v/p + u$.

The sixth equilibrium is $C6 = \left(\frac{m+h}{m}, 0, \frac{r_1 - AB(m+h)}{m}, 0\right)$ with the disease eradicated from both populations, has the feasibility condition

$$H_{C6} = r_1 - AB(m + h) \geq 0 \ , \tag{25}$$

 ISBN: 978-84-612-9727-6

and eigenvalues given by

$$-\frac{AB(m+h)\pm\sqrt{(AB(m+h))^2-4(m+h)H_{C6}}}{2m}\ ,\quad \frac{P_{C6}\pm\sqrt{P_{C6}^2-4Q_{C6}}}{J_{C6}}\ .$$

In what follows we consider only the strict inequality (25). To obtain stability, we need the following conditions verified,

$$\begin{aligned}(F-D)H_{C6}-C(A(m+h)(B-1)-r_2)<0;\\ DFH_{C6}+CF(A(m+h)(B-1)-r_2)+AhCEm+ACEm^2<0.\end{aligned}$$

But from the second one we obtain $-hA+ABm+hAB-Am-r_2<0$ and from the first one it follows $-DH_{C6}-C(-hA+ABm+hAB-r_2-Am)<0$, which entails that the second condition is positive, so that $C6$ is always unstable.

There are finally the sound predators-free equilibrium $C7=(p_{C7},u_{C7},0,v_{C7})$, which is very complicated to analyze and the coexistence of all subpopulations, $C8=(p_{C8},u_{C8},-v_{C8},v_{C8})$, which is easily seen to be always infeasible.

Remarks. Note that the stability of $C2$, $C3$ e $C4$ is easier to ensure than respectively the stability conditions of $Q2$, $Q3$ e $Q4$. Also, the two coexistence equilibria in this case are infeasible, as it also happens formerly. Looking at the results for C_6, the disease cannot be removed from the ecosystem leaving both populations thriving.

5 Diminishing return harvesting policy of sound prey

In the sound prey, harvesting is modelled now via a new term, a concave function of the harvested population as follows

$$\begin{aligned}\dot p&=\frac{r_1p}{m}-ABp^2-Bpu-pu-pv-pq-Cpv-\frac{hp}{m(u+p)}\ ,\\ \dot u&=\frac{r_2u}{m}-ABpu-Bu^2+Apu-\frac{Duq}{C}-Duv+mApv\ ,\\ \dot q&=-q-Euq-Fqv+pq+\frac{ADuq}{C}\ ,\\ \dot v&=-v+\frac{Fqv}{C}+pv+\frac{ADuv}{C}+\frac{Euq}{C}\ ,\end{aligned}\tag{26}$$

with Jacobian

$$\begin{pmatrix} J_{11} & -p(B+1)+\frac{hp}{m(u+p)^2} & -p & -p(1+C)\\ u(-AB+A)+mAv & J_{22} & -\frac{Du}{C} & -Du+mAp\\ q & q(-E+\frac{AD}{C}) & J_{33} & -Fq\\ v & \frac{ADv+Eq}{C} & \frac{Fv+Eu}{C} & J_{44}\end{pmatrix}\tag{27}$$

 ISBN: 978-84-612-9727-6

where

$$J_{11} = \frac{r_1}{m} - 2ABp - Bu - u - v - q - Cv - \frac{h}{m(u+p)} + \frac{hp}{m(u+p)^2} \ ,$$
$$J_{22} = \frac{r_2}{m} - ABp - 2Bu + Ap - \frac{Dq}{C} - Dv \ ,$$
$$J_{33} = -1 - Eu - Fv + p + \frac{ADu}{C} \ , \quad J_{44} = \frac{Fq}{C} - 1 + p + \frac{ADu}{C}$$

The equilibrium $A1 = \left(0, \frac{r_2}{Bm}, 0, 0\right)$, where only infected prey survive has eigenvalues

$$\frac{ADr_2 - BmC}{BmC} \ , \quad -\frac{r_2}{m} \ , \quad \frac{ADr_2 - BmC - ECr_2}{BmC} \ , \quad \frac{r_2(Br_1 - Br_2 - r_2) - hmB^2}{Bmr_2} \ ,$$

so that its stability conditions reduce to

$$ADr_2 - BmC < 0 \qquad r_2(Br_1 - Br_2 - r_2) - hmB^2 < 0 \ . \tag{28}$$

We find then the point with only sound prey, $A2 = \left(\frac{r_1 \pm \sqrt{r_1^2 - 4BmAh}}{2BmA}, 0, 0, 0\right)$, which is actually a double equilibrium, since the prey level comes from solving a quadratic equation. The feasibility condition requires its discriminant to be nonnegative, namely $H_{A2} = r_1^2 - 4BmAh \geq 0$. For the following discussion, we consider only the strict inequality for the discriminant. Then, $A2_+ = (\frac{r_1 + \sqrt{H_{A2}}}{2BmA}, 0, 0, 0)$ and $A2_- = (\frac{r_1 - \sqrt{H_{A2}}}{2BmA}, 0, 0, 0)$ are the two roots, with respective eigenvalues

$$-\frac{2BmA - (r_1 + \sqrt{H_{A2}})}{2BmA} \ , \quad -\frac{\sqrt{r_1^2 - 4BmAh}}{m} \ , \quad -\frac{-2Br_2 + (B-1)(r_1 + \sqrt{H_{A2}})}{2Bm}$$

and

$$-\frac{2BmA - (r_1 - \sqrt{H_{A2}})}{2BmA} \ , \quad \frac{\sqrt{r_1^2 - 4BmAh}}{m} \ , \quad \frac{2Br_2 - (B-1)(r_1 - \sqrt{H_{A2}})}{2Bm} \ ,$$

where in both cases the first one is a double root. From the latter, $A2_-$ is always unstable, while the former give the stability conditions for $A2_+$

$$-2Br_2 + (B-1)(r_1 + \sqrt{H_{A2}}) > 0 \qquad 2BmA - (r_1 + \sqrt{H_{A2}}) > 0 \ . \tag{29}$$

The whole predators-free equilibrium $A3 = (p_{A3}, u_{A3}, 0, 0)$ is actually a four-tuple of equilibria since the nonvanishing components come each from solving a different quadratic equation. The analysis is quite complicated to be fully carried out.

The equilibrium $A4 = \left(0, \frac{C}{AD}, 0, \frac{ADr_2 - BmC}{AD^2m}\right)$, i.e. pandemic disease in both populations, is feasible for

$$H_{A4} = ADr_2 - BmC \geq 0 \ . \tag{30}$$

Its eigenvalues are

$$-\frac{1}{mAD^2C}[C((1+C)H_{A4} + D(CBm + Cm - r_1AD)) + D^3hA^2] \ ,$$
$$-\frac{1}{AD^2m}[FH_{A4} + ECmD] \ , \quad -\frac{1}{2DAm}[BmC \pm \sqrt{B^2m^2C^2 - 4ADmH_{A4}}] \ ,$$

 ISBN: 978-84-612-9727-6

which provide the only stability condition

$$(1+C)(ADr_2 - BmC) + D(CBm + Cm - r_1AD) > -\frac{D^3hA^2}{C} \ . \tag{31}$$

In what follows, we consider only the strict inequality (30). From the structure of the Jacobian, the characteristic polynomial is found as $(\lambda - J_{33})(\lambda - J_{11})(\lambda^2 - \lambda J_{22} - J_{42}J_{24})$. To find a Hopf bifurcation, we need to impose $J_{22} = -\frac{BC}{AD} = 0$ from which it follows $B = 0$ since C cannot vanish since it appears in some denominators in the Jacobian. This simplifies the characteristic polynomial to

$$(\lambda CAmD - r_1ADC + C^2m + r_2AC + r_2C^2A + hA^2D^2)(\lambda DAm + CEm + r_2FA)(\lambda^2 m + r_2)$$

which has two purely imaginary roots and the other two with negative real parts if

$$-r_1ADC + C^2m + r_2AC + r_2C^2A + hA^2D^2 > 0 \ . \tag{32}$$

Using the initial conditions $(0.3, 0.9, 0.4, 0.6)$ we simulated in these conditions the system's behavior and the parameter values $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.5$, $h = 1$ satisfying $B = 0$ and (32) and giving the equilibrium $A4 = (0, 0.75, 0, 0.8\overline{3})$, see Figure 4.

In the next equilibrium $A5 = \left(1, 0, \frac{r_1 - BmA - h}{m}, 0\right)$, the disease is eradicated from the ecosystem. It is feasible for

$$H_{A5} = -r_1 + BmA + h \leq 0 \ . \tag{33}$$

Its eigenvalues have negative real part if the following conditions hold

$$-C(-mA - r_2 + BmA) + (D - F)H_{A5} < 0 \ , \quad h - BmA < 0 \ ,$$
$$CF(-mA - r_2 + BmA) + EACm^2 - FDH_{A5} < 0 \ .$$

Assuming the strict inequality (33), in view of the fact that $H_{A5} < 0$, the last condition gives $-mA - r_2 + BmA < 0$, for which $DH_{A5} - C(-mA - r_2 + BmA) < 0$ which is a contradiction. Thus $A5$ is always unstable, thus meaning that the disease-free environment cannot be restored once a disease invades such environment.

The equilibrium without sound predators, $A6 = (p_{A6}u_{A6}, 0, v_{A6})$ is rather complicated to analyze. The coexistence equilibrium $A7 = \left(p_{A7}, u_{A7}, q_{A7}, -\frac{1}{C}q_{A7}\right)$ is anew always infeasible. In principle there are also other coexistence equilibria of the type $A8 = \left(p_{A8}, -\frac{F}{E}v_{A8}, q_{A8}, v_{A8}\right)$, but again the infected prey and the infected predators levels cannot both be positive, so that also this one is always infeasible.

Remarks. Note that $A1$ is the same as $Q2$ but with a simpler stability condition, and similarly for $A4$ and $Q5$. $A2$ differs radically from $Q3$, since it gives two feasible equilibria only if $H_{A2} \geq 0$. Similarly $A3$, is much more complicated than $Q4$ and depends on the roots of a quadratic. The interior coexistence equilibria are once again never feasible. In this case the origin cannot be an equilibrium. We have also observed that if a disease enters in an ecosystem, the disease-free environment cannot be restored.

 ISBN: 978-84-612-9727-6

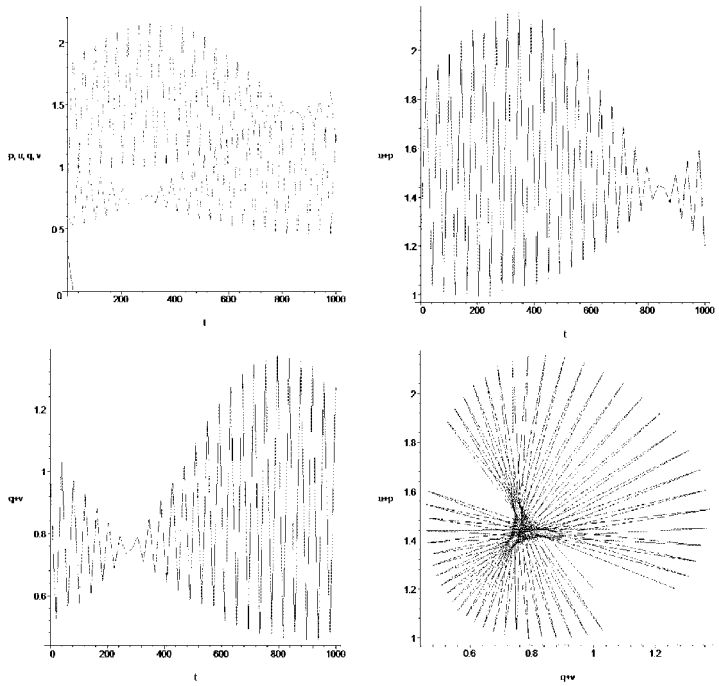


Figure 4: Solutions around $A4$ for the parameter values $A = 1$, $B = 0$, $C = 0.9$, $E = 1.1$, $F = 1$, $D = 1.2$, $m = 0.5$, $r_1 = 0.7$, $r_2 = 0.5$, $h = 1$ and initial condition $(0.3, 0.9, 0.4, 0.6)$. Top left: solutions as functions of time; Top right: total prey population $p+u$ as function of time; Bottom left: total predators population $q+v$ as function of time; Bottom right: solution projection onto the total predators—total prey phase plane $q + v/p + u$.

6 Diminishing return harvesting of infected prey

This assumption is made thinking of the diseased prey as more easily catchable than the sound ones. The model is

$$\begin{aligned}
\dot{p} &= \frac{r_1 p}{m} - ABp^2 - Bpu - pu - pv - pq - Cpv \ , \qquad (34)\\
\dot{u} &= \frac{r_2 u}{m} - ABpu - Bu^2 + Apu - \frac{Duq}{C} - Duv + mApv - \frac{hu}{m(u+p)} \ ,\\
\dot{q} &= -q - Euq - Fqv + pq + \frac{ADuq}{C} \ ,\\
\dot{v} &= -v + \frac{Fqv}{C} + pv + \frac{ADuv}{C} + \frac{Euq}{C} \ ,
\end{aligned}$$

 ISBN: 978-84-612-9727-6

with Jacobian

$$\begin{pmatrix} J_{11} & -p(B+1) & -p & -p(1+C) \\ u(-AB+A)+mAv+\frac{hu}{m(u+p)^2} & J_{22} & -\frac{Du}{C} & -Du+mAp \\ q & q(-E+\frac{AD}{C}) & J_{33} & -Fq \\ v & \frac{ADv+Eq}{C} & \frac{Fv+Eu}{C} & J_{44} \end{pmatrix} \tag{35}$$

where $J_{11}=\frac{r_1}{m}-2ABp-Bu-u-v-q-Cv$, $J_{22}=\frac{r_2}{m}-ABp-2Bu+Ap-\frac{Dq}{C}-Dv-\frac{h}{m(u+p)}+\frac{hu}{m(u+p)^2}$, $J_{33}=-1-Eu-Fv+p+\frac{ADu}{C}$, $J_{44}=\frac{Fq}{C}-1+p+\frac{ADu}{C}$.

The equilibrium $D1=\left(0,\frac{r_2\pm\sqrt{r_2^2-4Bmh}}{2Bm},0,0\right)$, with only infected prey, comes from solving a quadratic equation, for which the roots are real if $H_{D1}=r_2^2-4Bmh\geq 0$. Taking the stric inequality, for the first root, $D1_+=\left(0,\frac{r_2+\sqrt{H_{D1}}}{2Bm},0,0\right)$, the eigenvalues turn out then to be

$$-\frac{1}{2Bm}[-2r_1B+(B+1)(r_2+\sqrt{H_{D1}})]\ ,\quad \frac{1}{2Bm}[-2CBm+AD(r_2+\sqrt{H_{D1}})]\ ,$$
$$-\frac{1}{m}\sqrt{H_{D1}}\ ,\quad \frac{1}{2CBm}[-2CBm+(AD-EC)(r_2+\sqrt{H_{D1}})]$$

giving the stability conditions

$$-2r_1B+(B+1)(r_2+\sqrt{H_{D1}})>0 \qquad -2CBm+AD(r_2+\sqrt{H_{D1}})<0\ . \tag{36}$$

As for the second root, $D1_-=\left(0,\frac{r_2-\sqrt{H_{D1}}}{2Bm},0,0\right)$, the eigenvalues are

$$\frac{1}{m}\sqrt{H_{D1}}\ ,\quad -\frac{1}{2CBm}[2CBm+(EC-AD)(r_2+\sqrt{H_{D1}})]\ ,$$
$$\frac{1}{2Bm}[2r_1B-(B+1)(r_2-\sqrt{H_{D1}})]\ ,\quad -\frac{1}{2CBm}[2CBm-AD(r_2-\sqrt{H_{D1}})]$$

so that $D1_-$ is always unstable.

The second equilibrium contains only the sound prey, $D2=\left(\frac{r_1}{ABm},0,0,0\right)$. It is always feasible with eigenvalues

$$-\frac{1}{ABm}[ABm-r_1]\ ,\quad -\frac{r_1}{m}\ ,\quad -\frac{1}{mBr_1}[Br_1(r_1-r_2)-r_1^2+hB^2Am]\ ,$$

where the first one is a double eigenvalue, giving the stability conditions

$$r_1-ABm<0 \qquad r_1^2+Br_1(r_2-r_1)-hAB^2m<0\ . \tag{37}$$

The predator-free equilibrium exists also, $D3=(p_{D3},u_{D3},0,0)$, but is very difficult to analyze in detail. Its nonvanishing components are roots of a quadratic equation.

 ISBN: 978-84-612-9727-6

The fourth equilibrium $D4 = \left(0, \frac{C}{AD}, 0, \frac{ADr_2C - BmC^2 - hA^2D^2}{AD^2mC}\right)$ has the pandemic in both populations. The feasibility condition is

$$H_{D4} = ADr_2C - BmC^2 - hA^2D^2 \geq 0 \ . \tag{38}$$

For the case when (38) is strictly verified, its eigenvalues are found to be

$$\frac{hA^2D^2 - BC^2m \pm \sqrt{(hA^2D^2 - BC^2m)^2 - 4DAmCH_{D4}}}{2DAmC} \ ,$$
$$-\frac{FH_{D4} + DmC^2E}{AD^2mC} \ , \quad \frac{-(1+C)H_{D4} + CD(Dr_1A - CBm - Cm)}{AD^2mC}$$

giving the stability condition

$$-(1+C)H_{D4} + CD(Dr_1A - CBm - Cm) < 0 \ . \tag{39}$$

The fifth equilibrium is completely disease-free, $D5 = \left(1, 0, \frac{r_1 - BmA}{m}, 0\right)$, and is feasible for

$$H_{D5} = r_1 - BmA \geq 0 \ . \tag{40}$$

For the strict (40) the eigenvalues have negative real part if

$$(F-D)H_{D5} - C(ABm - mA + h - r_2) < 0 \ ,$$
$$CF(ABm - mA + h - r_2) + DFH_{D5} + EACm^2 < 0 \ ,$$

but from the second condition it follows $ABm - mA + h - r_2 < 0$ so that $-DH_{D5} - C(ABm - mA + h - r_2) < 0$ which entails a contradiction with the second condition. Therefore $D5$ is always unstable.

There are further more complicated equilibria, the one in which sound predators disappear, $D6 = (p_{D6}, u_{D6}, 0, v_{D6})$, and the one in which only the sound prey vanish, $D7 = (0, u_{D7}, q_{D7}, v_{D7})$, whose nonzero components arise as roots of a quadratic equation. The theoretical analysis proves to be hopeless.

Finally there are again potentially two types of interior equilibria where all subpopulations coexist. The first one, $D8 = \left(p_{D8}, -\frac{F}{E}v_{D8}, q_{D8}, v_{D8}\right)$, has the components of diseased prey and infected predators that cannot be both positive, so it must be discarded. The second type of coexistence equilibria is of the form $D9 = (p_{D9}, u_{D9}, -v_{D9}, v_{D9})$, and here the infeasibility stems from the two predators subpopulations, which cannot both be positive. Therefore it is always infeasible.

Remarks Here we find that $D2$ concides with $Q3$ but possesses much simpler stability condifions. $D1$ differs greatly from $Q2$ since it is actually a double equilibrium, and both are feasible only if $H_{D1} \geq 0$ holds. $D4$ shows stability conditions harder to obtain than those of the corrisponding equilibrium $Q5$. Here also the new equilibrium $D7$ arises, with no sound prey, which does not appear in the basic model.

 ISBN: 978-84-612-9727-6

7 Discussion

The novel feature of the ecoepidemic models introduced here is the role of harvesting, played not by one of the two populations described in the ecosystem, but by an external agent. With respect to the model in which the disease is able to spread to both populations by contact, [28, 2], harvesting has some influence in the resulting system's equilibria. The boundedness of the system trajectories has not been even mentioned, as it is an easy corollary of the same result obtained for the similar model without harvesting, since the latter contributes negatively to each right hand side term of the system. The subpopulations levels at equilibria are certainly influenced by harvesting.

The origin can be stabilized, the ecosystem may collapse, in case of strong linear harvesting of prey (5), while this cannot happen for linear harvesting only of predators. Thus the role of selective harvesting may be vital in ecosystems. A high harvesting rate h renders the infected-prey-only equilibrium feasible when linear harvesting on prey is exploited, while it is necessary for its stability in case of linear harvesting of predators and unexpectedly also for the diminishing return policy for sound prey, compare (6), (18) and (28). The sound prey-only equilibrium needs a low h for feasibility of $B3$ and $A2$, as the latter needs to have a real value, while a high one for stability of $C2$ and $D2$. An intermediate value of linear harvesting wipes out the predators and leaves the disease endemic in the prey, (10), but we have seen that this may trigger persistent oscillations in the ecosystem. For linear harvesting on predators, a relatively high rate is needed for stability of the corresponding equilibrium, (21). For the pandemic equilibria with no sound survivors, a low h is needed for feasibility and stability in the linear harvesting of prey, (12) and (13), while an intermediate one ensures the same for linear harvesting of predators, (22) and (23). A high h ensures stability of the similar situation for the diminishing returns case (31), and also the feasibility and stability for $D4$, (38) and (39). Finally, as already remarked, once the disease enters in such an ecosystem, it cannot be wiped out, both in absence as in presence of harvesting, since all the disease-free equilibria are shown always to be unstable.

Acknowledgements

This paper is dedicated to Professor Giampietro Allasia on the occasion of his 70th birthday, for his friendship and strong support over many long years of hard work both at the home Institution and away from it.

References

[1] E. Beltrami, T. O. Carroll, *Modelling the role of viral disease in recurrent phytoplankton blooms*, J. Math. Biol. **32** (1994) 857–863.

[2] A. Costamagna, E. Venturino, *Intermingling of disease-affected populations*, Proceedings Mathmod 09 Full Papers CD Volume, Argesim Report, **35** (2009) Vienna, Austria, 1808–1820.

[3]) U. D'Ancona, *The struggle for existence*, Brill, Leiden, 1954.

[4] M. Delgado, M. Molina-Becerra, A., Suarez, *Relating disease and predation: equilibria of an epidemic model*, Math. Methods Appl. Sci. **28** (2005) 349–362.

[5] J. C. Frauenthal, *Mathematical modeling in epidemiology*, Springer, Berlin, 1980.

[6] L. Q. Gao, H. W. Hethcote, *Disease transmission models with density-dependent demographics*, J. of Math. Biology **30** (1992) 717–731.

[7] K. P. Hadeler, H. I. Freedman, *Predator-prey populations with parasitic infection*, Journal of Mathematical Biology **27** (1989) 609–631.

[8] L. Han, Z. Ma, H. W. Hethcote, *Four predator prey models with infectious diseases*, Math. Comp. Modelling **30** (2001) 849–858.

[9] M. Haque, E. Venturino, *The role of transmissible diseases in Holling-Tanner predator-prey model*, Theoretical Population Biology **70** (2006) 273–288

[10] H. W.Hethcote, *The mathematics of infectious diseases*, SIAM Review **42** (2000) 599–653.

[11] H. W.Hethcote, Z. Ma, S. Liao, *Effects of quarantine in six endemic models for infectious diseases*, Math. Biosci. **180** (2002) 141–160.

[12] H. W. Hethcote, W. Wang, Z. Ma, *A predator prey model with infected prey*, Theoretical Population Biology **66** (2004) 259–268.

[13] Hsieh Y.H., Hsiao C.K. *Predator-prey model with disease infection in both populations*, Mathematical Medicine and Biology, **25** (2008), 247–266.

[14] W. O. Kermack, A. G. McKendrick, *A Contribution to the Mathematical Theory of Epidemics*, Proceedings of the Royal Society of London A **115** (1927) 700–721.

[15] A. J. Lotka, *Elements of Mathematical Biology*, Dover, New York, 1956.

[16] H. Malchow, S. Petrovskii, E. Venturino, *Spatiotemporal patterns in Ecology and Epidemiology* CRC, Boca Raton, FL, 2008.

[17] T. R. Malthus, *An essay on the principle of population* J. Johnson in St. Paul's Churchyard, London, 1798.

[18] J. Mena-Lorca, H. W. Hethcote, *Dynamic models of infectious diseases as regulator of population sizes*, J. Math. Biology **30** (1992) 693–716.

[19] A. Morozov, E. Arashkevic, *Patterns of zooplankton functional response in communities with vertical heterogeneity: a model study*, Math. Model. Nat. Phenom. **3**(3) (2008) 131–148.

[20] J. C. Poggiale, M. Gauduchon, P. Auger, *Enrichment paradox induced by spatial heterogeneity in a phytoplankton-zooplankton system*, Math. Model. Nat. Phenom. **3**(3) (2008) 87–102.

[21] I. Sazonov, M. Kelbert, M. B. Gravenor, *The speed of epidemic waves in a one-dimensional lattice of SIR models*, Math. Model. Nat. Phenom. **3**(4) (2008) 28–47.

[22] H. R. Thieme, A. Tridane, Y. Kuang, *An epidemic model with post-contact prophilaxis of distributed length II. Stability and oscillations if treatment is fully effective*, Math. Model. Nat. Phenom. **3**(7) (2008) 267–293.

[23] E. Venturino, *Epidemics in predator-prey models: disease among the prey*, in O. Arino, D. Axelrod, M. Kimmel, M. Langlais, Editors: *Mathematical Population Dynamics: Analysis of Heterogeneity, Vol. one: Theory of Epidemics* (1995) Wuertz Publishing Ltd, Winnipeg, Canada, 381–393.

[24] E. Venturino, *The effects of diseases on competing species*, Math. Biosc. **174** (2001) 111–131.

[25] E. Venturino, *Epidemics in predator-prey models: disease in the predators*, IMA Journal of Mathematics Applied in Medicine and Biology **19** (2002) 185–205.

[26] E. Venturino, *How diseases affect symbiotic communities*, Math. Biosc. **206** (2007) 11–30.

[27] E. Venturino, *Ecoepidemiology 15 years later: a review*, in T. Simos (Editor), *Numerical Analysis and Applied Mathematics, Proceedings of ICNAAM 2007*, AIP **936** (2007) 31–34.

[28] Venturino E. *On epidemics crossing the species barrier in interacting population models*, Varahmihir J. Math. Sci., **6** (2006), 247–263.

[29] P. F. Verhulst, *Notice sur la loi que la population suit dans son accroissement*, in *Correspondance Mathématique et Physique* Publiée par A. Quételet **10** (1838) 113-121.

[30] P. F. Verhulst, *Recherches mathématiques sur la loi d'accroissement de la population*, Mém. Acad. Roy. Bruxelles **18** (1845).

[31] P. F. Verhulst, *Recherches mathématiques sur la loi d'accroissement de la population*, Mém. Acad. Roy. Bruxelles **20** (1847).

[32] V. Volterra, U. D'Ancona, *La concorrenza vitale tra le specie dell'ambiente marino*, VIIe Congr. Int. acquicult et de pêche, Paris (1931) 1-14.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Determining all indecomposable codes over some Hopf algebras

J. Cuadra Díaz[1], J.M. García Rubira[1] and J.A. López-Ramos[2]

[1] *Department of Algebra and Analysis, University of Almería*

[2] *Department of Algebra and Analysis, University of Almería*

emails: jcdiaz@ual.es, jgr836@ual.es, jlopez@ual.es

Abstract

In this paper we determine all indecomposable codes over a family of Hopf Algebras known as Taft Algebras. We calculate dual and tensor products of such codes.

Key words: Taft Hopf Algebra, ideal, cyclic code
MSC 2000: AMS codes (optional)

1 Introduction

The Theory of Codes has its mathematical foundations on the study of vector spaces over fields (usually finite fields) as we can see in Hamming's and MacWilliams' works [4] and [5]. The two main and classical results are the Extension Theorem and the Identities. The first one states that if two codes are equivalent under a vector space isomorphism that preserves the Hamming weight, then this isomorphism can be extended to a monomial transformation. Under the same hypothesis, the second one gives the relation between the weight enumerator of a code and its dual code.

As Berman in [1] observed, cyclic codes and Reed-Muller codes can be seen as ideals in the group ring $\mathbb{K}G$ (where $\mathbb{K}$ is a finite field and G is a finite cyclic group). This fact has led many authors to the study of codes from a point of view of Ring Theory. More recently, Wood [8] remarked the suitability of the Frobenius rings for the Coding Theory when he extend the Extension Theorem and MacWilliams' Identities to the case of finite Frobenius rings. In fact the importance of Frobenius rings has been completely remarked also by Wood in [9] where he states that a finite Frobenius ring is characterized by the fact of allowing the Extension Theorem for linear codes.

In [3] the authors extend these results to the case of codes over Quasi-Frobenius modules.

A special case of Frobenius ring is the group algebra $\mathbb{K}G$. This algebra is also a Hopf algebra. These algebras provide a natural framework for codes as stated in [8] and [9]. It is important to remark that techniques used on Hopf algebras have been useful in the study of $\mathbb{K}G$. Therefore, by the precedent, our aim is to study codes in a non-commutative setting over Hopf algebras, so we will treat with a class of non-semisimple Hopf algebras that are known as Taft's Hopf algebras. The duality existing in Hopf algebras, besides of the many number of computations that can be made using it will allows us to study new codes from some others given.

2 Taft Codes

Let $n \in \mathbb{N}$ and consider $\omega \in \mathbb{K}$ an n-th primitive root of unity. Then, the algebra given by

$$H = H_{n^2} = \mathbb{K} < \ g, x : g^n = 1, \ x^n = 0, \ xg = \omega gx \ >$$

is a Hopf algebra with comultiplication $\Delta(g) = g \otimes g$ and $\Delta(x) = 1 \otimes x + x \otimes g$, counit $\varepsilon(g) = 1$ and $\varepsilon(x) = 0$ and whose antipode is given by $S(g) = g^{-1}$ and $S(x) = -xg^{-1}$ (cf. [6] or [2]).

the dimension of H is n^2 and a basis as a vector space for this algebra is given by the set

$$B = \{1, g, \cdots, g^{n-1}, x, gx, \cdots, g^{n-1}x, \cdots g^{n-1}x^{n-1}\}$$

Since the Jacobson radical of H is the ideal generated by x (referencia), we have the following short exact sequence

$$0 \to J(H) = (x) \to H \to H/(x) \cong \mathbb{K}\mathbb{Z}_n \to 0$$

Therefore, the idempotents of $\mathbb{K}\mathbb{Z}_n$ are candidates to be idempotents in H by lifting. It is easy to prove that the set $\{e_l, \ 0 \leq l \leq n\}$, where

$$e_l = \frac{1}{n} \sum_{i=0}^{n-1} \omega^{il} g^i, \ 0 \leq l \leq n,$$

is a system of orthogonal idempotents.

This allows us to compute the indecomposable projective codes. Let denote by $P_l = He_l$ the projective cover of $(H/J(H))e_l$. To compute the elements from P_l we have to multiply e_l by any element in B. We have the following composition series:

$$\{0\} = J^n P_l \subseteq J^{n-1} P_l \subseteq \cdots \subseteq J^2 P_l \subseteq J P_l \subseteq P_l = He_l = P(T_l)$$

where P_l is the projective cover of the simple $T_l = \mathbb{K}v_l$, given by the actions $g \cdot v_l = \omega^{-l} v_l$ and $x \cdot v_l = 0$ and $e_l = \frac{1}{n} \sum_{j=0}^{n-1} \omega^{jl} g^j$ is the idempotent of the decomposition corresponding to the indecomposable P_l.

It is easy to show that

 ISBN: 978-84-612-9727-6

$$g^i x^j e_l = g^i e_{l+j} x^j = \omega^{-i(l+j)} e_{l+j} x^j = \omega^{-i(l+j)} x^j e_l$$

so $He_l = I\!K\{e_l, xe_l, \cdots, x^{n-1}e_l\}$

Then we can write each element of the basis of He_l as a coordinate vector in the basis of H as follows:

$$x^j e_l = \frac{1}{n}(0, \cdots, 0|1, \omega^{(l-j)}, \cdots, \omega^{(n-1)(l-j)}, |0, \cdots, 0),\ 0 \leq j < n$$

i.e., a vector formed by n blocks of length n, all of them zero excepting the $(j+1)$-th block. Therefore dimension and length of P_l are n and n^2 respectively.

Proposition 2.1 *a) The minimum distance of P_l is n.*

b) The weight enumerator is given by

$$W(in) = \#\{\textit{words of weigth } in\} = |I\!K^*|^i \begin{pmatrix} n \\ i \end{pmatrix}$$

Proof. a) Follows from the fact that non-zero blocks do not overlap.

b) By the distribution of non-zero blocks we get that there can only exist words of weight in, with i in $\{0, \cdots, n\}$. □

Proposition 2.2 *The codes P_l are equivalent.*

Proof. Consider the code $P_0 = I\!K < e_0, xe_0, \cdots, x^{n-1}e_0 >$ and let $P_l = I\!K < e_0, xe_0, \cdots, x^{n-1}e_0 >$. Let us show that both codes are equivalent.

Let us fix $j \in \{1, \cdots, n-1\}$ and consider the elements of the basis of the codes in its vectorial form

$$x^i e_0 = \frac{1}{n}(0, \cdots, 0|1, \omega^{-i}, \cdots, \omega^{-(n-1)i}, |0, \cdots, 0)$$

$$x^j e_l = \frac{1}{n}(0, \cdots, 0|1, \omega^{(l-j)}, \cdots, \omega^{(n-1)(l-j)}, |0, \cdots, 0)$$

If we impose that non-zero blocks of the generic elements are equal, then we get that this is equivalent to the fact that $\omega^i = \omega^{l-j}$. Since ω is an n-th primitive root of unity, this can happen if and only if $i \equiv_n j - l$, which let us to define the bijection $x^j e_l \to x^{j-l} e_0$ between the elements of the basis. More precisely, if we consider by blocks the elements in its vector form $(B_0|B_1|\cdots|B_{n-1})$, we have an application $P_l \to P_0$ sending block B_j in P_l to block $B_{(j-l) mod\ n}$. Now let B_j^l be the non-zero block of $x^j e_l$, $(1, \omega^{(l-j)}, \cdots, \omega^{(n-1)(l-j)})$. This block corresponds to B_{j-l}^0, the non-zero block of $x^{j-l}e_0$. Since B_j^l begins with coordinate $(j-l)n + 1 = jn - ln + 1$ and ends with coordinate $(j - l + 1)n = jn - ln + jn$, the permutation of the elements given by

$$\text{position } i \to \text{position } (i - ln)\ mod\ n^2,\ i = 1, \cdots, n^2$$

gives us the equivalence between P_l and P_0. □

3 Dual codes of the indecomposable Taft codes

Let us consider the indecomposable projective codes $P_l = I\!K\{v_0, v_1, \cdots, v_{n-1}\},\ l = 0, \cdots, n-1$ with actions

$$\begin{array}{lcl} g \cdot v_i & = & \omega^{-(i+l)} v_i \\ x \cdot v_i & = & v_i + 1, \quad i = 0, \cdots, n-2 \\ x \cdot v_{n-1} & = & 0 \end{array}$$

Then, the dual of P_l is given by $P_l^* = I\!K\{v_0^*, \cdots, v_{n-1}^*\},\ l = 0, \cdots, n-1$, where $v_i^*(v_j) = \delta_{ij}$.

Theorem 3.1 *P_l^* is isomorphic to $P_{-(l-1)}$*

Proof. Previously to check the action of g and x on the elements of the canonical basis of P_l^*, we get that,

$$\begin{array}{rcll} g^{-1} \cdot v_j & = & g^{n-1} \cdot v_j & = \\ & = & \omega^{-(n-1)(j+l)} v_j & = \\ & = & \omega^{(j+l)} v_j \quad j = 0, \cdots, n-1; & \\ (-xg^{-1}) \cdot v_j & = & -x \cdot (\omega^{(j+l)} v_j) & = \\ & = & -\omega^{(j+l)} x \cdot v_j & = \\ & = & -\omega^{(j+l)} v_{j+1} \quad j = 0, \cdots, n-2; & \\ (-xg^{-1}) \cdot v_{n-1} & = & -x \cdot (\omega^{(n-1+l)} v_{n-1}) & = \\ & = & -\omega^{(n-1+l)} x \cdot v_{n-1} & = \\ & = & 0 & \end{array}$$

Now, since $< g \cdot v_i^*, v_j > = < v_i^*, S(g) \cdot v_j >$ and $< x \cdot v_i^*, v_j > = < v_i^*, S(x) \cdot v_j >$, on one hand we get that,

$$\begin{array}{rcll} < g \cdot v_i^*, v_j > & = & < v_i^*, S(g) \cdot v_j > & = \\ & = & < v_i^*, g^{-1} \cdot v_j > & = \\ & = & < v_i^*, \omega^{(j+l)} v_j > & = \\ & = & \omega^{(j+l)} < v_i^*, v_j > & = \\ & = & \omega^{(j+l)} \delta_{ij} \quad j = 0, \cdots n-1 & \end{array}$$

so $g \cdot v_i^* = \omega^{(i+l)} v_i^*$.

On the other hand,

$$\begin{array}{rcll} < x \cdot v_i^*, v_j > & = & < v_i^*, S(x) \cdot v_j > & = \\ & = & < v_i^*, (-xg^{-1}) \cdot v_j > & = \\ & = & < v_i^*, \omega^{(j+l)} v_{j+1} > & = \\ & = & \omega^{(j+l)} < v_i^*, v_{j+1} > & = \\ & = & \omega^{(j+l)} \delta_{ij+1} \quad j = 0, \cdots n-2 & \end{array}$$

and analogously, $< x \cdot v_i^*, v_{n-1} > = 0$ and hence $x \cdot v_0^* = 0$ and $x \cdot v_i^* = -\omega^{(i-1+l)} v_{i-1}^*$ for $i = 1, \cdots, n-1$.

Then, considering the basis

$$\{u_0 = v^*_{n-1},\ u_1 = \omega x v^*_{n-1}, u_2 = \omega^2 x^2 v^*_{n-1}, \cdots, u_{n-1} = \omega^{n-1} x^{n-1} v^*_0\}$$

we obtain that

$$\begin{array}{rcllll} g \cdot u_i & = & g \cdot (x^i v^*_{n-1}) & = \\ & = & \omega^{-i} x^i (g \cdot v^*_{n-1}) & = \\ & = & x^i (\omega^{n-(i+1)+l} v^*_{n-1} & = \\ & = & \omega^{-i+l-1} x^i v^*_{n-1} & = \\ & = & \omega^{-(i-(l-1))} u_i \end{array}$$

$$x \cdot u_i = x \cdot (x^i v^*_{n-1}) = x^{i+1} v^*_{n-1} = u_{i+1},\ \text{for } i = 0, \cdots, n-2$$

and

$$x \cdot u_{n-1} = x \cdot (x^{n-1} v^*_{n-1}) = x^n v^*_{n-1} = 0$$

and so we get that $P^*_l \cong P_{-(l-1)}$ with the isomorphism $r_i \to v_i$. □

Let $J^j P_i = N_{i,j} \subseteq P_i$ be an indecomposable code in the Taft Hopf algebra H_{n^2}. Then we have the following:

Corollary 3.2 $N^*_{i,j} \cong N_{-(i-1),j}$

Proof. $N_{i,j}$ is embedded into P_i, so we get that P^*_i is projected on the dual of this indecomposable $N^*_{i,j}$. Therefore, since $P^*_i \cong P_{-(i-1)}$, $N^*_{i,j}$ must be isomorphic with the corresponding indecomposable of dimension $n-j$ in $P_{-(i-1)}$, i.e., $N^*_{i,j} \cong N_{-(i-1),j}$ □

4 Tensor products of indecomposables

Let us consider, as in the previous section $N_{i,j} = J^j P_i$ any indecomposable in the composition series

$$\{0\} = J^n P_l \subseteq J^{n-1} P_l \subseteq \cdots \subseteq J^2 P_l \subseteq J P_l \subseteq P_l = H e_l = P(T_l)$$

Theorem 4.1 *$N_{i,n-r} \otimes N_{j,n-s} \cong \bigoplus_{l=1}^{m} N_{i+j-l,n-t}$ where $m = min\{r,s\}$ and $t = max\{r,s\}$.*

Proof. Let us note first that the socle of P_i, i.e., the biggest semisimple submodule in P_i is $Soc(P_i) = N_{i,n-1} \cong T_i$. Therefore, $N_{i,n-1} = J^{n-1} P_i = J^{n-1} e_i = I\!K w_i$, where $w_i = x^{n-1} e_i$. Then,

$$\begin{array}{rcllllll} g \cdot w_i & = & g \cdot (x^{n-1} e_i) & = & (g x^{n-1}) e_i & = \\ & = & (\omega x^{n-1} g) e_i & = & (\omega x^{n-1})(g e_i) & = \\ & = & \omega x^{n-1} \omega^{-i} e_i & = & \omega^{-(i-1)} (x^{n-1} e_i) & = \\ & = & \omega^{-(i-1)} w_i \end{array}$$

 ISBN: 978-84-612-9727-6

and

$$x \cdot w_i = x(x^{n-1}e_i) = x^n e_i = 0$$

Therefore, the actions of g and x are preserved and so, the isomorphism given above is correct.

Let us first assume that $m = r$ and $t = s$, i.e., $r \leq s$ and consider two any indecomposables
$N_{i,n-r} = I\!K\{v_1, v_2, \cdots, v_r\}$, $v_k = x^{n-k}e_i$ with actions

$$\begin{array}{rcccccl} g \cdot v_k & = & g \cdot (x^{n-k}e_i) & = & (gx^{n-k})e_i & = \\ & = & (\omega^k x^{n-k} g)e_i & = & (\omega^k x^{n-k})(ge_i) & = \\ & = & \omega^k x^{n-k}\omega^{-i}e_i & = & \omega^{-(i+k)}(x^{n-k}e_i) & = \\ & = & \omega^{-(i-k)}v_k \end{array}$$

$$\begin{array}{rcccccl} x \cdot v_k & = & x \cdot (x^{n-k}e_i) & = & x^{n-k+1}e_i & = \\ & = & x^{n-(k-1)}e_i & = & v_{k-1} \end{array}$$

having into account that if $k = 1$ then $x \cdot v_1 = x^n e_i = 0$, hence $v_0 = 0$.

Analogously, $N_{j,n-s} = I\!K\{w_1, w_2, \cdots, w_r\}$, $w_k = x^{n-k}e_i$ with actions $g \cdot w_l = \omega^{-(j-l)}w_l$, $x \cdot w_l = w_{l-1}$ $(w_0 = 0)$.

Applying the tensor product, we get

$$N_{i,n-r} \otimes N_{j,n-s} = I\!K\{z_{11}, \cdots, z_{1s}, z_{21}, \cdots, z_{2s}, \cdots, z_{r1}, \cdots, z_{rs}\}$$

where $z_{kl} = v_k \otimes w_l$ with actions

$$\begin{array}{rcccccl} g \cdot z_{k,l} & = & g \cdot (v_k \otimes w_l) & = & gv_k \otimes g \otimes w_l & = \\ & = & \omega^{-(i-k)}v_k \otimes \omega^{-(j-l)}w_l & = & \omega^{-(i-k)-(j-l)}v_k \otimes w_l & = \\ & = & \omega^{-(i+j)-(k+l)}z_{k,l} \end{array}$$

$$\begin{array}{rcccccl} x \cdot z_{k,l} & = & x \cdot (v_k \otimes w_l) & = & 1v_k \otimes xw_l + xv_k \otimes gw_l & = \\ & = & v_k \otimes w_{l-1} + v_{k-1} \otimes \omega^{-(j-l)}w_l & = \\ & = & z_{k,l-1} + \omega^{-(j-l)}z_{k-1,l} \end{array}$$

Let us fix $k = 1$ and consider $M = I\!K\{z_{1,1}, \cdots, z_{1,s}\}$ with $g \cdot z_{1,l} = \omega^{-(i+j)-(1+l)}z_{1l}$ and $x \cdot z_{1,l} = z_{1,l-1} + \omega^{-(j-l)}z_{0,l} = z_{1,l-1}$ since $z_{0,l} = v_0 \otimes w_l = 0 \otimes w_L = 0$. Then $g \cdots z_{1,1} = \omega^{-(i+j-2)}z_{1,1}$ and $x \cdot z_{1,1} = z_{1,0} = 0$. Therefore, the simple $T_{i+j-2} \cong N_{i+j-1,n-1}$ lays in M. Now we are going to show that, in fact, the corresponding s-dimensional indecomposable in the composition series also lays in M. So we will have that $M \cong N_{i+j-1,n-s}$. But comparing the actions of g and x on $N_{i+j-1,n-s} = I\!K\{w_1, \cdots, w_s\}$, $g \cdot w_l = \omega^{-(i+j-1-l)}w_l = \omega^{-(i+j)-(1+l)}w_l$ and $x \cdot w_l = w_{l-1}$ with the actions of g and x on M and having into account the dimensions, the isomorphism is clear.

Now let us fix $k = 2$ and consider the quotient

$$(N_{i,n-r} \otimes N_{j,n-s})/M = I\!K\{\overline{z_{2,1}}, \cdots, \overline{z_{2,s}}, \cdots, \overline{z_{r,1}}, \cdots, \overline{z_{r,1}}\}$$

Then we have that $g \cdot \overline{z_{2,l}} = \overline{g \cdot z_{2,l}} = \omega^{-(i+j)-(2+l)}\overline{z_{2,l}}$ and $x \cdot \overline{z_{2,l}} = \overline{x \cdot z_{2,l}} = \overline{z_{2,l-1}} + \omega^{-(j-l)}\overline{z_{1,l}} = \overline{z_{2,l-1}}$ since $\overline{z_{1,l}} = \overline{0}$. Again, comparing with the corresponding indecomposable $N_{i+j-2,n-s} = I\!K\{w_1, \cdots, w_s\}$, where the actions of g and x are given as before, we get that

$$I\!K\{\overline{z_{2,1}}, \cdots, \overline{z_{2,s}}, \cdots, \overline{z_{r,1}}, \cdots, \overline{z_{r,1}}\} \cong N_{i+j-2,n-s}$$

and therefore,

$$N_{i,n-r} \otimes N_{j,n-s} \cong N_{i+j-1,n-s} \oplus N_{i+j-2,n-s} \otimes M'$$

Suppose now, as induction hypothesis that

$$N_{i,n-r} \otimes N_{j,n-s} \cong \left(\bigoplus_{l=1}^{k-1} N_{i+j-l,n-s}\right) \oplus M''$$

with

$$I\!K\{\overline{z_{\hat{k},1}}, \cdots, \overline{z_{\hat{k},s}}\} \cong N_{i+j-\hat{k},n-s},\ \hat{k} = 1, \cdots, k-1 \qquad (*)$$

taking coset modulo $I\!K\{z_{1,1}, \cdots z_{1,s}, z_{2,1}, \cdots, z_{2,s}, \cdots, z_{k-1,1}, \cdots z_{k-1,s}\}$.

Then, we have that $g \cdot \overline{z_{k,l}} = \overline{g \cdot z_{kl}} = \omega^{-(i+j-(k+l))}\overline{z_{k,l}}$ and $x \cdot \overline{z_{k,l}} = \overline{x \cdot z_{kl}} = \overline{z_{k,l-1}} + \omega^{-(j-l)}\overline{z_{k-1,l}} = \overline{z_{k,l-1}}$, since $\overline{z_{k-1,l}} = \overline{0}$.

Comparing with the action on $N_{i+j-k,n-s} = I\!K\{w_1, \cdots, w_s\}$, $g \cdot w_l = \omega^{-(i+j-k-l)}w_l$ $= \omega^{-(i+j-(k+l))}w_l$ and $x \cdot w_l = w_{l-1}$ it is clear the isomorphism $(*)$ (with $\hat{k} = k$). Hence

$$N_{i,n-r} \otimes N_{j,n-s} \cong \left(\bigoplus_{l=1}^{k-1} N_{i+j-l,n-s}\right) \oplus N_{i+j-k,n-s} \oplus M''$$

and since $1 \leq k \leq r$ we have that

$$N_{i,n-r} \otimes N_{j,n-s} \cong \left(\bigoplus_{l=1}^{r} N_{i+j-l,n-s}\right)$$

Let us assume now that $r > s$ and consider the tensor product of two any indecomposables with the basis rearranged in the following way:

$$N_{i,n-r} \otimes N_{j,n-s} = I\!K\{z_{1,1}, \cdots, z_{r,1}, z_{1,2}, \cdots z_{r,2}, \cdots z_{1,s}, \cdots, z_{r,s}\}$$

We will check that the s indecomposables of dimension r in the direct sum lay in the tensor product. Again we will proceed by induction.

Let us fix $l = 1$ and consider $M = I\!K\{z_{1,1}, \cdots, z_{r,1}\}$, where the actions of g and x are given by $g \cdot z_{k,1} = \omega^{-(i+j-(k+l))}z_{k,1}$ and $x \cdot z_{k,1} = z_{k,0} + \omega^{-(j-1)}z_{k-1,1} = \omega^{-(j-1)}z_{k-1,1}$ respectively. Comparing with the actions of g and x on $N_{i+j-1,n-1} = I\!K\{v_1, \cdots, v_r\}$, $g \cdot v_k = \omega^{-(i+j-1-k)}v_k = \omega^{-(i+j-(k+1))}v_k$ and $x \cdot v_k = v_{k-1}$, we can see that the action of x is not preserved. However, if we consider $\hat{M} = I\!K\{\hat{z}_{1,1}, \cdots, \hat{z}_{r,1}\}$, where $\hat{z}_{k,1} = \omega^{(k-1)(j-1)}z_{k,1},\ k = 1, \cdots, r$, we have that

 ISBN: 978-84-612-9727-6

$$\begin{aligned} g\cdot\hat{z}_{k,l} &= g\cdot(\omega^{(k-1)(j-1)}z_{k,1}) &&= \omega^{(k-1)(j-1)}(g\cdot z_{k,1}) &&= \\ &= \omega^{(k-1)(j-1)}\omega^{-(i+j-(k+1))}z_{k,1} &&= \\ &= \omega^{-(i+j)-(k+l)}\hat{z}_{k,1} \end{aligned}$$

$$\begin{aligned} x\cdot\hat{z}_{k,1} &= x\cdot(\omega^{(k-1)(j-1)}z_{k,1}) &&= \omega^{(k-1)(j-1)}(x\cdot z_{k,1}) &&= \\ &= \omega^{(k-1)(j-1)}\omega^{-(j-1)}z_{k-1,1} &&= \omega^{(k-2)(j-1)}z_{k-1,1} &&= \\ &= \hat{z}_{k-1,1} \end{aligned}$$

Thus $\hat{M}\cong N_{i+j-1,n-r}(\subseteq N_{i,n-r}\otimes N_{j,n-s})$. Now let $l=2$ and consider

$$(N_{i,n-r}\otimes N_{j,n-s})/\hat{M} = I\!K\{\overline{z_{1,2}},\cdots,\overline{z_{r,2}},\cdots,\overline{z_{1,s}},\cdots,\overline{z_{r,s}}\}$$

Then $g\cdot\overline{z_{k,2}}=\overline{g\cdot z_{k,2}}=\omega^{-(i+j-(k+2))}\overline{z_{k,2}}$ and $x\cdot\overline{z_{k,2}}=\overline{x\cdot z_{k,2}}=\overline{z_{k,1}}+\omega^{-(j-2)}\overline{z_{k-1,2}}$ $=\omega^{-(j-2)}\overline{z_{k-1,2}}$.

So, if we modify the basis and consider $\hat{z}_{k,2}=\omega^{(k-1)(j-2)}z_{k,2},\ 1\le k\le r$, then we have

$$\begin{aligned} g\cdot\overline{\hat{z}_{k,2}} &= \overline{g\cdot\hat{z}_{k,2}} &&= \overline{g\cdot\omega^{(k-1)(j-2)}z_{k,2}} &&= \\ &= \overline{\omega^{(k-1)(j-2)}g\cdot z_{k,2}} &&= \overline{\omega^{(k-1)(j-2)}\omega^{-(i+j-(k+2))}z_{k,2}} &&= \\ &= \omega^{-(i+j)-(k+2)}\overline{\hat{z}_{k,2}} \end{aligned}$$

$$\begin{aligned} x\cdot\overline{\hat{z}_{k,2}} &= \overline{x\cdot\hat{z}_{k,2}} &&= \overline{x\cdot\omega^{(k-1)(j-2)}z_{k,2}} &&= \\ &= \overline{\omega^{(k-1)(j-2)}x\cdot z_{k,2}} &&= \overline{\omega^{(k-1)(j-2)}\omega^{-(j-2)}z_{k-1,2}} &&= \\ &= \overline{\omega^{(k-2)(j-2)}z_{k-1,2}} &&= \overline{\hat{z}_{k-1,2}} \end{aligned}$$

Therefore, $I\!K\{\overline{\hat{z}_{1,2}},\cdots\overline{\hat{z}_{r,2}}\}\cong N_{i+j-2,n-r}$ and hence

$$N_{i,n-r}\otimes N_{j,n-s}\cong\left(\bigoplus_{k=1}^{l-1}N_{i+j-k,n-r}\right)\oplus M''$$

with

$$I\!K\{\overline{\hat{z}_{1,\hat{l}}},\cdots,\overline{\hat{z}_{r,\hat{l}}}\}\cong N_{i+j-\hat{l},n-s},\ \hat{l}=1,\cdots,l-1 \qquad (**)$$

where $\hat{z}_{k,l}=\omega^{(k-1)(j-l)}z_{k,l}$, taking coset modulo

$$I\!K\{z_{1,1},\cdots,z_{r,1},z_{1,2},\cdots,z_{r,2},\cdots,z_{1,l-1},\cdots,z_{r,l-1}\}$$

So we have

$$\begin{aligned} g\cdot\overline{\hat{z}_{k,l}} &= \overline{g\cdot\hat{z}_{k,l}} &&= \overline{g\cdot\omega^{(k-1)(j-l)}z_{k,l}} &&= \\ &= \overline{\omega^{(k-1)(j-l)}g\cdot z_{k,l}} &&= \overline{\omega^{(k-1)(j-l)}\omega^{-(i+j-(k+l))}z_{k,l}} &&= \\ &= \omega^{-(i+j)-(k+l)}\overline{\hat{z}_{k,l}} \end{aligned}$$

$$\begin{aligned} x\cdot\overline{\hat{z}_{k,l}} &= \overline{x\cdot\hat{z}_{k,l}} &&= \overline{x\cdot\omega^{(k-1)(j-l)}z_{k,l}} &&= \\ &= \overline{\omega^{(k-1)(j-l)}x\cdot z_{k,l}} &&= \overline{\omega^{(k-1)(j-l)}\omega^{-(j-l)}z_{k-1,l}} &&= \\ &= \overline{\omega^{(k-2)(j-l)}z_{k-1,l}} &&= \overline{\hat{z}_{k-1,l}} \end{aligned}$$

that corresponds clearly with $N_{i+j-l,n-r} = \mathbb{K}\{v_1, \cdots, v_r\}$ where the actions are given by $g \cdot v_k = \omega^{-(i+j-l-k)} v_k = \omega^{-(i+j-(l+k))} v_k$ and $x \cdot v_k = v_{k-1}$.

Therefore, the isomorphism $(**)$ is correct and so

$$N_{i,n-r} \otimes N_{j,n-s} \cong \left(\bigoplus_{k=1}^{l-1} N_{i+j-l,n-r} \right) \oplus N_{i+j-l,n-r} \oplus M''$$

and since $1 \leq l \leq s$, we can conclude that

$$N_{i,n-r} \otimes N_{j,n-s} \cong \bigoplus_{k=1}^{s} N_{i+j-l,n-r}$$

5 Aplications

We can observe (cf. Section 2) that duals of the codes in the Taft's Hopf algebra are concatenation of cyclic codes in $\mathbb{K}\mathbb{Z}_n$. Recall that a generator matrix of P_l ($l = 0, \cdots, n-1$) is

$$\frac{1}{n} \begin{pmatrix} 1\,1\cdots 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0\,0\cdots 0 & 1 & \omega^{(l-1)} & \cdots & \omega^{(n-1)(l-1)} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0\,0\cdots 0 & 0 & 0 & \cdots & 0 & \cdots & 1 & \omega^{(l-n)} & \cdots & \omega^{(n-1)(l-n)} \end{pmatrix}$$

One of its interest lays in the possibility of identifying the concatenation of ideal / codes with an ideal/code in a bigger algebra.

Therefore, a good option for encoding/decoding algorithms for these codes is to make it block by block, i.e., using n standard encoders/decoders circuits in parallel (cf. [5, Ch.7]).

The main advantage of this is that we can increase the capability of error correction of these duals under some circumstances. From the above matrix it is immediate that the minimum distance of each block is n. If we decode block by block, then we are capable of correcting $E(\frac{n-1}{2})$ errors in each block and if we have the certainty that no more errors than $E(\frac{n-1}{2})$ in such a block occur, excepting the first one, which is a repetition code, then we are increasing this capability to $(n-1) \cdot E(\frac{n-1}{2})$. If we want to get better codes then we can puncture the original code by deleting the first n coordinates. Then we would get a new code where the minimum distance and the number of words do not decrease and we still have the possibility of correcting $(n-1) \cdot E(\frac{n-1}{2})$ errors without assuming that no error occurs in one of the blocks. The same applies for the original code P_l, whose error correcting capability is $E(\frac{n-1}{2})$.

In a fast memory framework, the error on one chip or module where a portion of word is stored in, can only affect to that piece (cf. Figure 1). Each portion can be considered as one of the blocks since all block have the same length. Then we could certify a good functioning of the memory whenever the state of each chip is over a

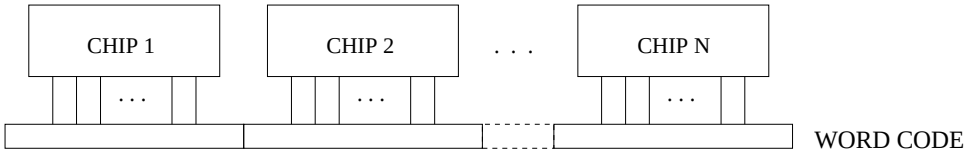


Figure 1: Fast Memory Scheme

certain critic level. This kind of codes are called byte error correcting codes and some examples can be found in [7].

Acknowledgements

This work has been partially supported by Junta de Andalucía FQM 0211 and TEC2006-12211-CO2-02

References

[1] J. Berman, *On the Theory of Group Codes,* Kibernetika (Kiev) **1** (1967), 31-39 (Russian): translated as Cybernetics **1** (1969) 25–39.

[2] S. DăscăLescu, C. NăstăSescu and S. Raianu, *Hopf Algebras. An Introduction,* Marcel Dekker, Inc., New York, 2001.

[3] M. Greferath, A. Nechaev and R. Wisbauer, *Finite Quasi-Frobenius Modules and Linear Codes,* Journal of Algebra and its Applications **3**(3) (2004) 247–272.

[4] R.W. Hamming, *Error detecting and error cofrrecting codes,* Bell System Tech. J. **29** (1950) 147–160.

[5] F.J. MacWilliams, N.J.A. Sloane, *The Theroy of Error-Correcting Codes,* North-Holland mathematical Library, Vol. 16, North-Holland Publishing Co., Amsterdam-New-York-Oxford (1977).

[6] S. Montgomery, *Hopf Algebras and Their Actions on Rings,* CBMS 82 A.M.S., Chicago, 1982.

[7] Rao, T. R. N., Fujiwara, E., *Error-Control Coding for Computer Systems*, Prentice Hall Inc, 1989.

[8] J.A. Wood, *Duality for modules over finite rings and applications to coding theory,* Amer. J. Math. **121**(3) (1999) 555–575.

[9] J.A. Wood, *Code equivalence characterizes finite Frobenius rings,* Proc. Amer. Math. Soc. **136**(2) (2008) 699–706 (electronic).

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

Comparing the behaviour of basic linear algebra routines on multicore platforms

Javier Cuenca[1], Luis P. García[2], Domingo Giménez[2] and Manuel Quesada[2]

[1] *Departamento de Ingeniería y Tecnología de Computadores, University of Murcia*

[2] *Departamento de Informática y Sistemas, University of Murcia*

emails: jcuenca@um.es, luis.garcia@sait.upct.es, domingo@dif.um.es, manuel.quesada@alu.um.es

Abstract

The use of an OpenMP compiler optimized for a multicore system could contribute to obtain programs with satisfactory execution time, but it is possible to have access in a system to more than one compiler, and different compilers optimize different parts of the code at different levels. In this paper, the influence of the compiler used on the performance of linear algebra routines is studied. From the results of the experiments carried out, we conclude a poly-compiling approach is necessary to decide automatically the best compiler.

Key words: multicore, linear algebra, performance

1 Introduction

Recently a new class of architecture has appeared, called chip-multiprocessor (CMP) or multicore. This new class of architecture means that as many threads as cores (or even more threads) can be executed in parallel. Nowadays, multicore architectures are everywhere and can be found in all market segments [3, 9]. In particular, they constitute the CPU of many embedded systems (for example, the last generation of video game consoles), personal computers (for example, the latest developments from Intel and AMD), servers (the IBM Power5 or Sun UltraSPARC T1, among others) and even supercomputers (for example, the CPU chips used as building blocks in the IBM Blue-Gene/L).

The arrival of multicores made parallel computing more available to the scientific groups: today it is possible to have parallel systems not only in the form of clusters, but also in a personal computer or a laptop. Thus, there is great interest in the scientific community in using multicore systems efficiently to solve their problems, but without the great effort of reprogramming existing sequential codes, which work well [4]. In those multicore platforms, it is easier

to develop a parallel code from a sequential one using shared memory paradigm than to transform the code into a message-passing code using, for example, MPI [1]. Using the shared memory paradigm, it is possible to express data parallelism with explicit threading techniques like Pthreads [10], but it is an invasive process. The computation must be separated into a function that can be mapped to threads, within which the work must be manually divided among the threads. Another possibility is to describe parallelism to the compiler using OpenMP [2], with a directive-based syntax. In the case of a compiler that does not understand OpenMP, the OpenMP directives are ignored, and the code is compiled without error. This is a key advantage of OpenMP over other parallel programming methods: it is incremental and relatively non-invasive. Up to now, there are no standardized OpenMP compilers and it is usual in a multicore system to have access to more than one OpenMP compiler, and it is not clear which is the best compiler. In some cases one compiler generates good executable code from a sequential code, but another optimizes more the use of multiple threads. It is also possible to obtain executables which perform well with a number of threads lower than the number of cores in the node, although when the number of threads increases, the performance greatly decreases. So, if we consider an OpenMP code generated with small modifications (typically including some `pragmas`) from a sequential code, the parallel code may not be optimized for the system in which it will be run, and to obtain good executable codes is a task of the compiler.

In previous works [7, 11], we developed an Automatically Tuning System (ATS) for linear algebra routines, mainly parallel ones. Linear algebra functions are widely used in the solution of scientific and engineering problems, which means that there is much interest in the development of highly efficient linear algebra libraries, and these libraries are used by non-experts in parallel programming. In recent years several groups have been working on the design of highly efficient parallel linear algebra libraries. These groups work in different ways: optimizing the library in the installation process for shared memory machines [16] or for message-passing systems [6, 13], analyzing the adaptation of the routines to the conditions of the system at a particular moment [8, 12, 14], using poly-algorithms [15] and by means of poly-libraries [5]. For each routine, the kernel of our ATS consists of an execution time model of the routine. In this model the characteristics of the platform (hardware + basic installed libraries) are included as System Parameters (SP), and a set of Algorithmic Parameters (AP), whose values should be appropriately chosen by the ATS in order to reduce the execution time of the routine.

Our goal now is to build a Poly-Compilation Engine (PCE) that generates different executables of each routine (one for each compiler in the system), and, when a particular problem has to be solved, it selects the executable which best fits the problem characteristics (the idea is a generalization of other methods to accelerate the solution of computationally demanding problems: poly-algorithms and poly-libraries). In this way, the PCE makes the ATS capable of managing a new AP: the selection of the most appropriate compiled version of the routine for each situation.

In this work, we compare the behaviour of linear algebra routines on multicore platforms when different compilers are used. Section 2 details the comparison performed with different basic linear algebra routines, and section 3 summarizes the conclusions and outlines future research.

For the experiments performed in this work, the multicore platforms used and their corresponding compilers have been:

- **P2c**: Intel Pentium, 2.8 GHz, with 2 cores. Compilers: icc 10.1 and gcc 4.3.2.
- **A4c**: Alpha EV68CB, 1 GHz, with 4 cores. Compilers: cc 6.3 and gcc 4.3.
- **X4c**: Intel Xeon, 3 GHz, with 4 cores. Compilers: icc 10.1 and gcc 4.2.3.
- **X8c**: Intel Xeon, 2 GHz, with 8 cores. Compilers: icc 10.1 and gcc 3.4.6.

2 Comparison the behaviour of Linear Algebra Routines

The routines used for this comparison have different computational costs (from $O(n^2)$ to $O(n^3)$) and different schemes for creation of threads and use of memory.

The first routine, called **R-mvomp**, is a matrix-vector multiplication. It consists basically of the distribution between the threads of the iterations of a `for` loop through the rows of a matrix $M \epsilon R^{n \times n}$, multiplying each row by a vector $v \epsilon R^n$. Each thread accesses a contiguous space of M (a set of rows), and the vector v (also a contiguous space). The cost of the routine is $O(n^2)$. Figure 1 shows the results obtained for a problem size $n = 2000$. The general impression is similar on platforms P2c, X4c and X8c: when the number of threads is less than or equal to the number of available cores, both versions have a similar behaviour, but after this point, the performance of the `icc` versions decreases notably. However, on A4c the performance with the `gcc` version is clearly inferior until 4 threads (the number of cores); but from this number the situation is completely the inverse, with the `cc` version being the best.

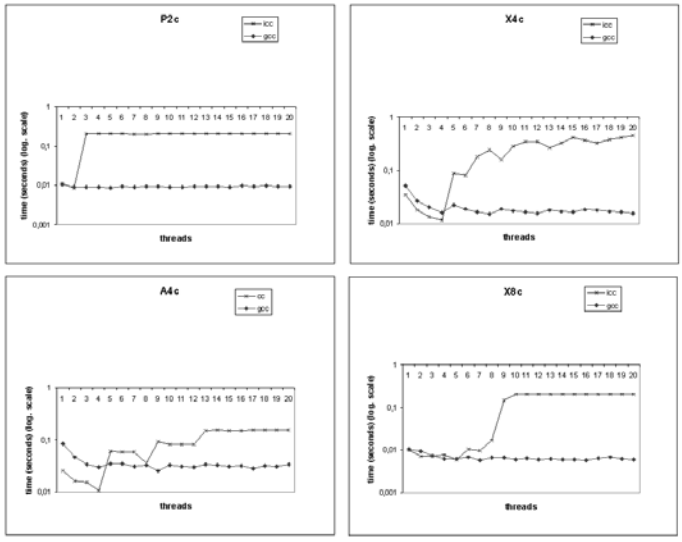


Figure 1: Comparison of the execution time (logarithmic scale) of the R-mvomp routine on the different plataforms (Problem size = 2000).

The next linear algebra routine (**R-Jacobi**) consists of a single iteration (in order to take more comparable times) of the Jacobi relaxation method for a 2D mesh of $n \times n$ points. The cost of this routine (one iteration of the algorithm) is $O(n^2)$. Figure 2 shows the results obtained for

 ISBN: 978-84-612-9727-6

a problem size of 1000. Like those with the previous routine, when the number of threads is less than or equal to the number of available cores, both compiled versions have a very similar behaviour on platforms P2c, X4c and X8c; and for more threads the performance with `gcc` is the same, whereas it decreases with the other compiler. However, on A4c the performance with the `gcc` version is clearly inferior up to 4 threads (the number of cores); and from this number on the `gcc` version is clearly the best. If we compare the behaviour in the two four-core platforms (figure 3), we can see how on X4c the differences between the versions become clearer, but, on A4c this difference reduces considerably.

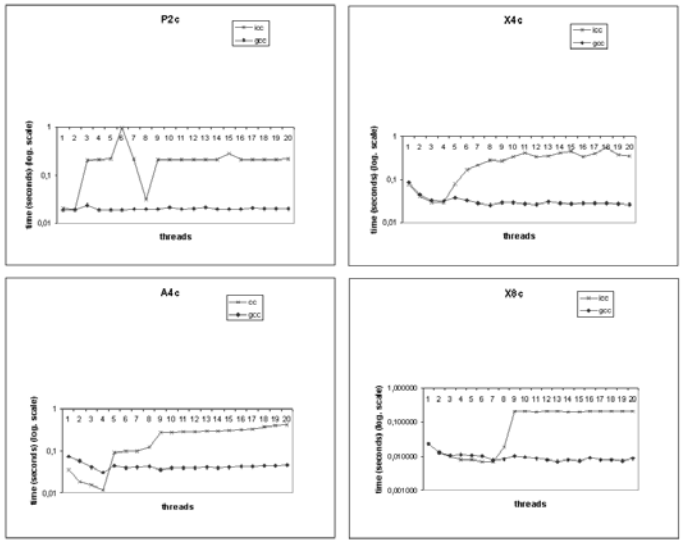


Figure 2: Comparison of the execution time (logarithmic scale) of the R-jacobi routine on the different plataforms (Problem size = 1000).

The routine **R-mmomp** is a matrix-matrix multiplication, $AB = C$, where A, B and C $\epsilon R^{n \times n}$. It consists of the three typical nested `for` loops, where the external one has been parallelized and its iterations are distributed between the threads. Each thread accesses a contiguous space of A (a set of rows), but in the access to B by columns the elements are not contiguous in memory (they are stored by rows). The cost of this routine is $O(n^3)$. Figure 4 shows the results obtained with this routine for a problem size $n = 500$. We can see that on the platforms P2c and A4c the behaviour of both compiled versions are very similar. However, in Xeon platforms (X4c and X8c) the execution times remain very similar just for a number of threads less than or equal to the number of cores, but when the number of threads exceedes this number, the performance with `gcc` is better.

If we compare the behaviour in the two four-core platforms (figure 5), we can see how on X4c both versions tend to have similiar execution times, but on A4c this difference increases slightly with the problem size.

On comparing these results with those obtained with the R-mvomp routine, we can appreciate that when the quantity of work to perform with the data increases (from $O(n^2)$ to $O(n^3)$) and the problem size grows, the features obtained with both versions are similar. This could be due to a reduction of the relative weight of creation and management of threads in the total

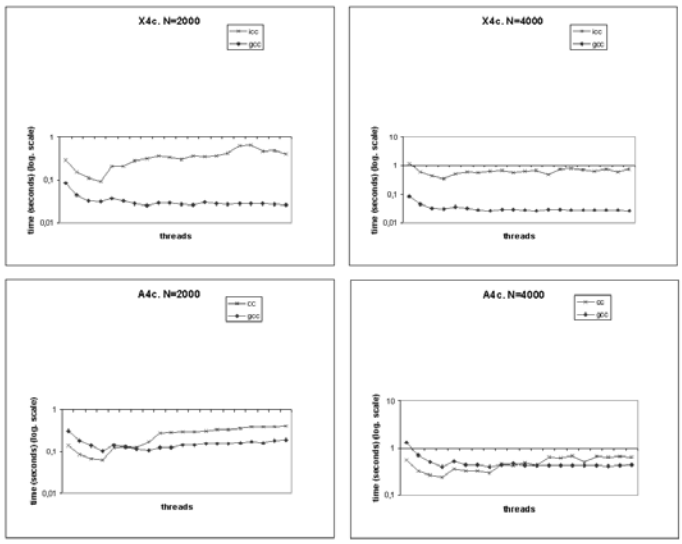


Figure 3: Evolution of the execution time (logarithmic scale) of the R-jacobi routine on the four-core platforms (Problem size = 2000, 4000).

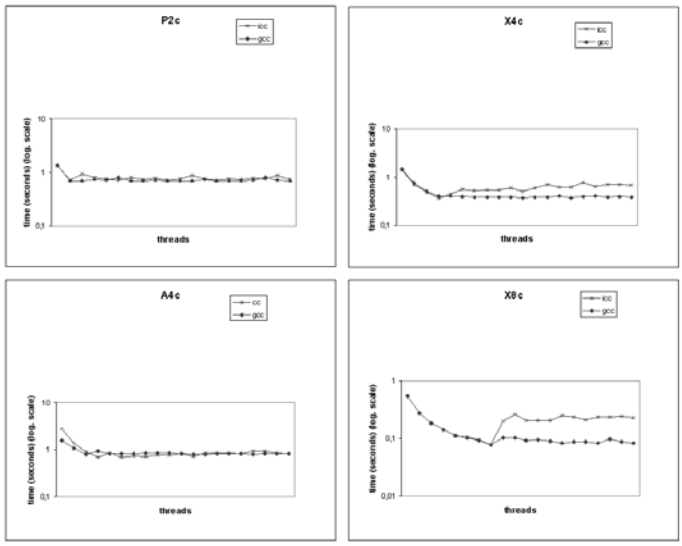


Figure 4: Comparison of the execution time (logarithmic scale) of the R-mmomp routine on the different platforms (Problem size = 500).

execution time, which has $O(n^3)$ arithmetic operations.

Finally, the last linear algebra routine (**R-strassen**) is a implementation of the well-known Strassen algorithm for matrices multiplication. The implementation is prepared to apply a distribution of the work to do between the threads in the first and in the second levels of recursion of the routine, with a maximum of 49 generated threads. The cost of the routine is $O(n^{2.807})$. The results obtained for a problem size $n = 1000$ are shown in figure 6, where the execution

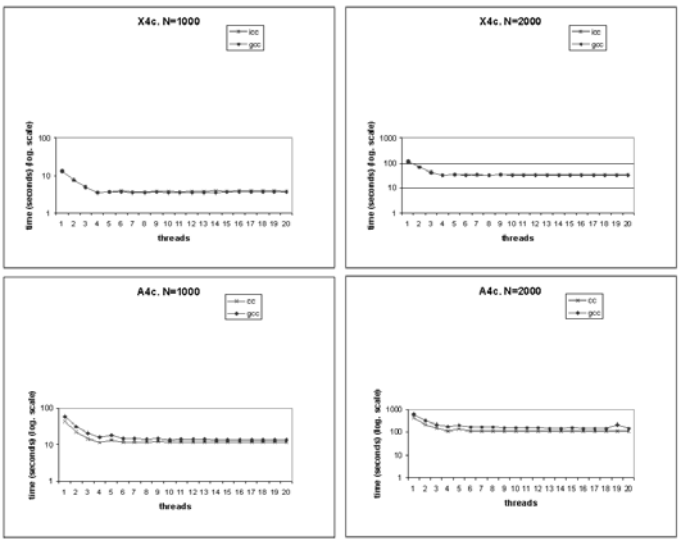


Figure 5: Evolution of the execution time (logarithmic scale) of the R-mmomp routine on the four-core platforms (Problem size = 1000, 2000).

times obtained for different number of threads for each of the two levels of recursion are compared (from one thread per level, 1×1, until seven per level, 7×7). We can see that for each version, the optimal execution time is obtained on each platform with a different combination of number of threads for the first and for the second recursion level. So, for a problem to be solved on a given platform, the decisions to be taken are: the compiled version, the number of threads and the recursion level.

From the obtained experimental information we can conclude that, in general, for small problem sizes the behaviour is clearly better with the `gcc` versions when there are more threads than cores[1], which entails frequent context exchanges among those threads that share a core, but when the quantity of work and the problem size increase, the features obtained with different compilers are close. This could be due to a reduction of the relative weight of creating and managing threads in the total execution time. However, when creating and managing threads is more complicated (like in the Strassen recursive routine) it is not trivial to determine either the best compiled version or the most appropriate number of threads to generate for each recursion level. Therefore, an ATS which would take the appropriate decisions (for example, the PCE would decide which compiled version to use, taking into account the number of available cores, the architecture of these cores and the problem size.) is necessary if we want to obtain near optimal performance with linear algebra routines on multicore platforms.

[1]The existence of more threads than cores could occur when the algorithm of the routine needs an appropriate quantity of threads for its correct working or for its improvement, like in the Strassen algorithm. Other situation of overloaded cores could happen when the platform is shared by different routines, with the total number of threads being higher than the quantity of cores.

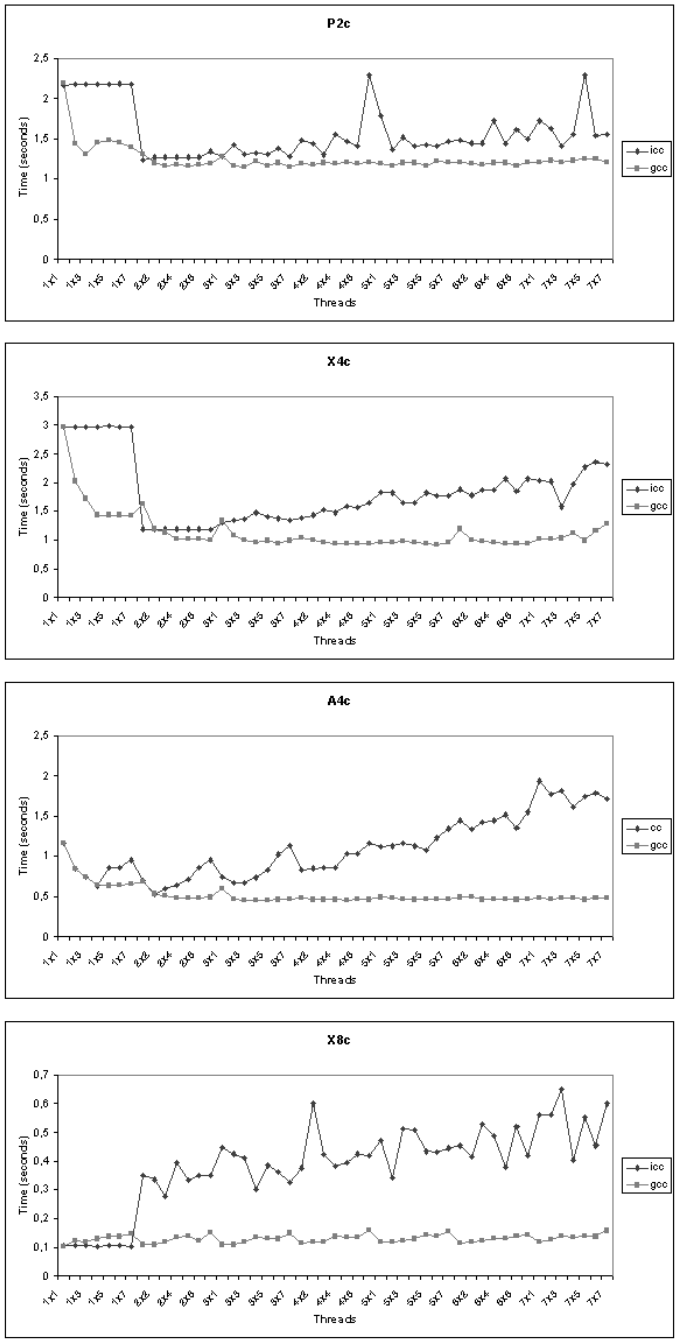


Figure 6: Comparison of the execution time (logarithmic scale) of the R-strassen routine on the different platforms, for different number of threads for each one of the two recursion levels (Problem size = 1000).

3 Conclusions

We have shown how a good choice of compiler in a multicore system could contribute to accelerating the solution of scientific problems. Which the best compiler is depends on a

number of factors: the type of routine, the number of threads that it uses, the problem size,... So, according all the information collected in this work about the behaviour of the different compiled versions, we consider a poly-compiling approach is necessary to decide automatically the best compiler .

Therefore, we are currently working on the design and implementation of the Poly-Compilation Engine (PCE). For each routine, the PCE, using the information obtained with a Poly-Compilation Benchmarking tool (PCB) (the PCB is constituted with simple routines formed by OpenMP primitives basically, in order to extract information about creating and managing simple threads), calculates the basic System Parameters values for each pair platform-compiler. After that, using the theoretical model of the execution time of the routine, the PCE selects the appropriate compiled version (an Algorithmic Parameter) in each situation.

4 Acknowledgment

This work has been partially supported by the Conserjería de Educación de la Región de Murcia (Fundación Séneca, 08763/PI/08), and by the Ministerio de Ciencia e Innovación (TIN2008-06570-C04-02/TIN).

References

[1] The Message Passing Interface (MPI) standard. http://www-unix.mcs.anl.gov/mpi/.

[2] The OpenMP API specification for parallel programming. http://openmp.org/wp/.

[3] J. L. Abellán, J. Fernández, and M. E. Acacio. CellStats: a tool to evaluate the basic synchronization and communication. In *16th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, pages 261–268, 2008.

[4] S. Akhter and J. Roberts. *Multi-Core Programming*. Intel Press, 2006.

[5] P. Alberti, P. Alonso, A. Vidal, J. Cuenca, L. P. García, and D. Giménez. Designing polylibraries to speed up linear algebra computations. *International Journal of High Performance Computing and Networking (IJHPCN)*, 1(1-2-3):75–84, 2004.

[6] J. Cuenca, D. Giménez, and J. González. Towards the design of an automatically tuned linear algebra library. In *10th EUROMICRO Workshop on Parallel, Distributed and Networked Processing*, pages 201–208. IEEE, 2002.

[7] J. Cuenca, D. Giménez, and J. González. Architecture of an automatic tuned linear algebra library. *Parallel computing*, 30(2):187–220, 2004.

[8] J. Cuenca, D. Giménez, J. González, J. Dongarra, and K. Roche. Automatic optimisation of parallel linear algebra routines in systems with variable load. In *11th EUROMICRO Workshop on Parallel,Distributed and Networked Processing*, pages 409–416. IEEE, 2003.

[9] K. De Bosschere, W. Luk, X. Martorell, N. Navarro, M. O´ Boyle, D. Pnevmatikatos, A. Ramírez, P. Sainrat, A. Seznec, P. Stenstrőm, and O. Temam. High-performance embedded architecture and compilation roadmap. *Transactions on High-Performance Embedded Architecture and Compilation*, 1(1):5–29, 2007.

[10] R. S. Engelschall. GNU portable threads. http://www.gnu.org/software/pth/.

[11] L. P. García, J. Cuenca, and D. Giménez. Using experimental data to improve the performance modelling of parallel linear algebra routines. *PPAM'07 conference. Lecture Notes in Computer Science*, 4967:1150–1159, 2008.

[12] A. Kalinov and A. Lastovetsky. Heterogeneous distribution of computations while solving linear algebra problems on networks of heterogeneous computers. *Lecture Notes in Computer Science*, 1593:191–200, 1999.

[13] T. Katagiri and Y. Kanada. An efficient implementation of parallel eigenvalue computation for massively parallel processing. *Parallel Computing*, 27(14):1831–1845, 2001.

[14] A. Petitet, S. Blackford, J. Dongarra, B. Ellis, G. Fagg, K. Roche, and S. Vadhiyar. Numerical libraries and the grid. *International Journal of High Performance Applications and Supercomputing*, 15:359–374, 2001.

[15] A. Skjellum and P. V. Bangalore. Driving issues in scalable libraries: Polyalgorithms, data distribution independence, redistribution, local storage schemes. In *7th SIAM Conference on Parallel Processing for Scientific Computing*, pages 734–737, 1995.

[16] R. Whaley, A. Petitet, and J. Dongarra. Optimization of software and the ATLAS project. *Parallel Computing*, 27(1-2):3–35, 2001.

Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2009
30 June, 1–3 July 2009.

On Unequally Smooth Bivariate Quadratic Spline Spaces

Catterina Dagnino[1], Paola Lamberti[1] and Sara Remogna[1]

[1] *Department of Mathematics, University of Torino, via C. Alberto, 10 - 10123 Torino, Italy*

emails: catterina.dagnino@unito.it, paola.lamberti@unito.it, sara.remogna@unito.it

Abstract

In this paper we consider spaces of unequally smooth local bivariate quadratic splines, defined on criss-cross triangulations of a rectangular domain.
For such spaces we present some results on the dimension and on a local basis.
Finally an application to B-spline surface generation is provided.

Key words: bivariate spline approximation, unequally smooth bivariate spline space, B-spline basis
MSC 2000: 65D07; 41A15

1 Introduction

Aim of this paper is the investigation of bivariate quadratic spline spaces with less than maximum C^1 smoothness on criss-cross triangulations of a rectangular domain, with particular reference to their dimension and to the construction of a local basis. Indeed, in many practical applications, piecewise polynomial surfaces need to be connected by using different smoothness degrees and, in literature, tensor product spline surfaces of such a kind have already been investigated (see e.g. [1, 5]). In [2] the dimension and a B-spline basis for the space of all quadratic C^1 splines on a criss-cross triangulation are obtained. Since some supports of such B-splines are not completely contained in the rectangular domain, in [7] a new B-spline basis for such space is proposed, with all supports included in the domain.

The paper is organized as follows. In Section 2 we present some results on the dimension of the unequally smooth spline space and on the construction of a B-spline basis with different types of smoothness. In Section 3 an application to B-spline surface generation is presented.

2 Bases of unequally smooth bivariate quadratic spline spaces

Let $\Omega = [a, b] \times [c, d]$ be a rectangle decomposed into $(m + 1)(n + 1)$ subrectangles by two partitions

$$\bar{\xi} = \{\xi_i, \quad i = 0, \ldots, m + 1\},$$
$$\bar{\eta} = \{\eta_j, \quad j = 0, \ldots, n + 1\},$$

of the segments $[a, b] = [\xi_0, \xi_{m+1}]$ and $[c, d] = [\eta_0, \eta_{n+1}]$, respectively. Let $\mathcal{T}_{mn}$ be the criss-cross triangulation associated with the partition $\bar{\xi} \times \bar{\eta}$ of the domain Ω.

Given two sets $\bar{m}^\xi = \{m_i^\xi\}_{i=1}^m$, $\bar{m}^\eta = \{m_j^\eta\}_{j=1}^n$, with m_i^ξ, $m_j^\eta = 1, 2$ for all i, j, we set

$$M = 3 + \sum_{i=1}^{m} m_i^\xi, \quad N = 3 + \sum_{j=1}^{n} m_j^\eta \tag{1}$$

and let $\bar{u} = \{u_i\}_{i=-2}^M$, $\bar{v} = \{v_j\}_{j=-2}^N$ be the nondecreasing sequences of knots, obtained from $\bar{\xi}$ and $\bar{\eta}$ by the following two requirements:

(i) $u_{-2} = u_{-1} = u_0 = \xi_0 = a, \qquad b = \xi_{m+1} = u_{M-2} = u_{M-1} = u_M,$
$v_{-2} = v_{-1} = v_0 = \eta_0 = c, \qquad d = \eta_{n+1} = v_{N-2} = v_{N-1} = v_N;$

(ii) for $i = 1, \ldots, m$, the number ξ_i occurs exactly m_i^ξ times in $\bar{u}$ and for $j = 1, \ldots, n$, the number η_j occurs exactly m_j^η times in $\bar{v}$.

For $0 \le i \le M - 1$ and $0 \le j \le N - 1$, we set $h_i = u_i - u_{i-1}$, $k_j = v_j - v_{j-1}$ and $h_{-1} = h_M = k_{-1} = k_N = 0$. In the whole paper we use the following notations

$$\begin{aligned} \sigma_{i+1} &= \tfrac{h_{i+1}}{h_i + h_{i+1}}, \quad \sigma_i' = \tfrac{h_{i-1}}{h_{i-1} + h_i}, \\ \tau_{j+1} &= \tfrac{k_{j+1}}{k_j + k_{j+1}}, \quad \tau_j' = \tfrac{k_{j-1}}{k_{j-1} + k_j}. \end{aligned} \tag{2}$$

When in (2) we have $\frac{0}{0}$, we set the corresponding value equal to zero.

On the triangulation $\mathcal{T}_{mn}$ we can consider the spline space of all functions s, whose restriction to any triangular cell of $\mathcal{T}_{mn}$ is a polynomial in two variables of total degree two. The smoothness of s is related to the multiplicity of knots in $\bar{u}$ and $\bar{v}$ [4]. Indeed let m_i^ξ (m_j^η) be the multiplicity of ξ_i (η_j), then

$$\begin{aligned} m_i^\xi \ \ (m_j^\eta) \ \ + \ \ &\text{degree of smoothness for } s \text{ crossing the line } u = \xi_i \quad (v = \eta_j) \\ &= 2. \end{aligned}$$

We call such space $\mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn})$. We can prove [4] that

$$\dim \mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn}) = 8 - mn + m + n + (2 + n)\sum_{i=1}^{m} m_i^\xi + (2 + m)\sum_{j=1}^{n} m_j^\eta. \tag{3}$$

Now we denote by

$$\mathcal{B}_{MN} = \{B_{ij}(u, v)\}_{(i,j)\in\mathcal{K}_{MN}}, \quad \mathcal{K}_{MN} = \{(i, j) : 0 \le i \le M - 1, 0 \le j \le N - 1\}, \tag{4}$$

 ISBN: 978-84-612-9727-6

the collection of $M \cdot N$ quadratic B-splines defined in [4], that we know to span $\mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn})$. In $\mathcal{B}_{MN}$ we find different types of B-splines. There are $(M-2)(N-2)$ inner B-splines associated with the set of indices $\widehat{\mathcal{K}}_{MN} = \{(i,j) : 1 \le i \le M-2, 1 \le j \le N-2\}$, whose restrictions to the boundary $\partial\Omega$ of Ω are equal to zero.

To the latter, we add $2M + 2N - 4$ boundary B-splines, associated with

$$\widetilde{\mathcal{K}}_{MN} := \{(i,0),(i,N-1), 0 \le i \le M-1; (0,j),(M-1,j), 0 \le j \le N-1\},$$

whose restrictions to the boundary of Ω are univariate B-splines [7].

Any B_{ij} in $\mathcal{B}_{MN}$ is given in Bernstein-Bézier form. Its support is obtained from the one of the quadratic C^1 B-spline $\bar{B}_{ij}$, with octagonal support (Fig. 1) [2, 7], by conveniently setting h_i and/or k_j equal to zero in Fig. 1, when there are double (or triple) knots in its support. The B_{ij}'s BB-coefficients different from zero are computed by using Table 1, evaluating the corresponding ones related to the new support [3]. The symbol "O" denotes a zero BB-coefficient.

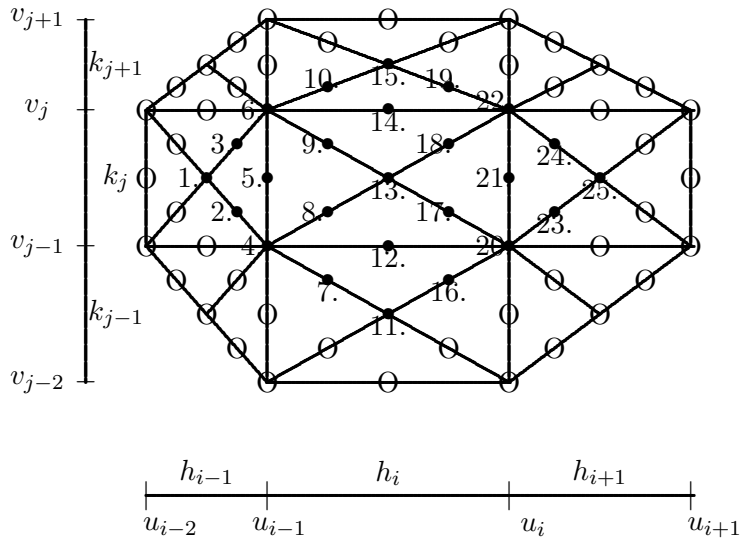


Figure 1: Support of the C^1 B-spline $\bar{B}_{ij}(u,v)$.

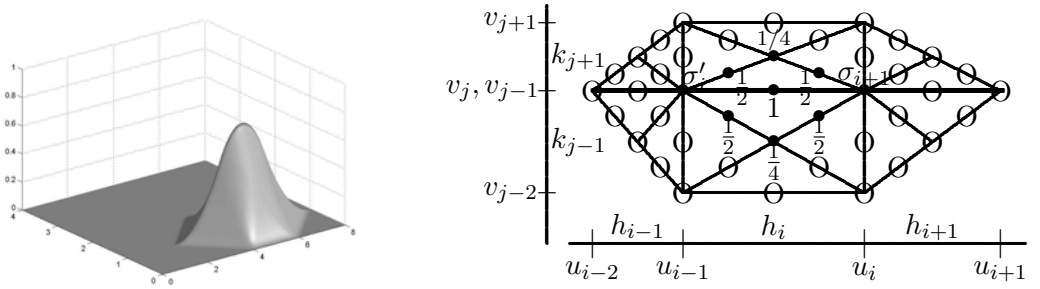
Since $\bar{u}$ and $\bar{v}$ can have multiple knots, then the B_{ij} smoothness changes and the B-spline support changes as well, because the number of triangular cells on which the function is nonzero is reduced. For example, in Fig. 2 we propose: (a) the graph of a B-spline B_{ij}, with the double knot $v_{j-1} = v_j$, (b) its support with its BB-coefficients different from zero, computed by setting $k_j = 0$ in Fig. 1 and Table 1. Analogously in Figs. 3÷6 we propose some other multiple knot B-splines. In Figs. $2(b)$÷$6(b)$ a thin line means that the B-spline is C^1 across it, while a thick line means that the function is continuous across it, but not C^1 and a dotted line means that the function has a jump across it.

All B_{ij}'s are non negative and form a partition of unity.

 ISBN: 978-84-612-9727-6

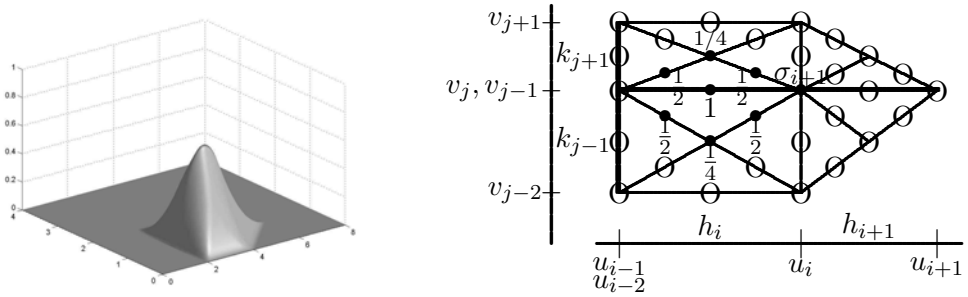
1. $\frac{\sigma_i'}{4}$,	2. $\frac{\sigma_i'}{2}$,	3. $\frac{\sigma_i'}{2}$,	4. $\sigma_i'\tau_j'$,	5. σ_i',
6. $\sigma_i'\tau_{j+1}$,	7. $\frac{\tau_j'}{2}$,	8. $\frac{\sigma_i'+\tau_j'}{2}$,	9. $\frac{\sigma_i'+\tau_{j+1}}{2}$,	10. $\frac{\tau_{j+1}}{2}$,
11. $\frac{\tau_j'}{4}$,	12. τ_j',	13. $\frac{\sigma_i'+\sigma_{i+1}+\tau_j'+\tau_{j+1}}{4}$,	14. τ_{j+1},	15. $\frac{\tau_{j+1}}{4}$,
16. $\frac{\tau_j'}{2}$,	17. $\frac{\sigma_{i+1}+\tau_j'}{2}$,	18. $\frac{\sigma_{i+1}+\tau_{j+1}}{2}$,	19. $\frac{\tau_{j+1}}{2}$,	20. $\sigma_{i+1}\tau_j'$,
21. σ_{i+1},	22. $\sigma_{i+1}\tau_{j+1}$,	23. $\frac{\sigma_{i+1}}{2}$,	24. $\frac{\sigma_{i+1}}{2}$,	25. $\frac{\sigma_{i+1}}{4}$,

Table 1: B-net of the C^1 B-spline $\bar{B}_{ij}(u,v)$.



(a) (b)

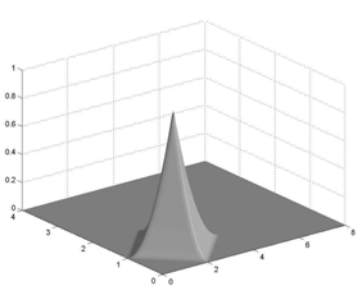
Figure 2: A double knot quadratic C^0 B-spline B_{ij} with $v_{j-1} = v_j$ and its support.



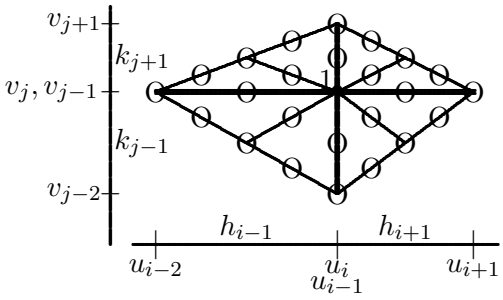
(a) (b)

Figure 3: A double knot quadratic C^0 B-spline B_{ij} with $u_{i-2} = u_{i-1}$, $v_{j-1} = v_j$ and its support.

Since $\sharp\mathcal{B}_{MN} = M \cdot N$, from (3) and (1) it results that $\sharp\mathcal{B}_{MN} > \dim \mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn})$. Therefore the set $\mathcal{B}_{MN}$ is linearly dependent and we can prove [4] that the number of linearly independent B-splines in $\mathcal{B}_{MN}$ coincides with dim $\mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn})$. Then we can

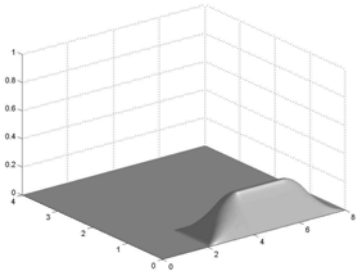


(a)

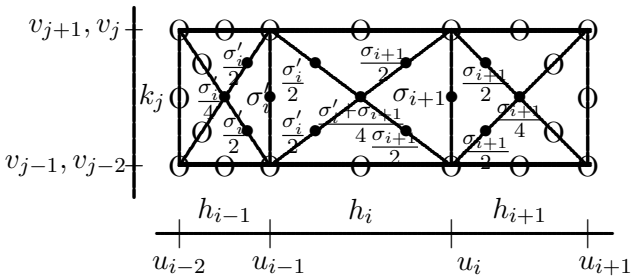


(b)

Figure 4: A double knot quadratic C^0 B-spline B_{ij} with $u_{i-1} = u_i$, $v_{j-1} = v_j$ and its support.

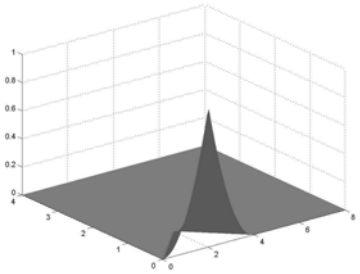


(a)

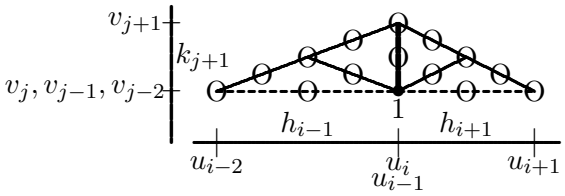


(b)

Figure 5: A double knot quadratic C^0 B-spline B_{ij} with $v_{j-2} = v_{j-1}$, $v_j = v_{j+1}$ and its support.



(a)



(b)

Figure 6: A triple knot quadratic B-spline B_{ij} with $u_{i-1} = u_i$, $v_{j-2} = v_{j-1} = v_j$ and its support.

conclude that the algebraic span of $\mathcal{B}_{MN}$ is all $\mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn})$.

 ISBN: 978-84-612-9727-6

3 An application to surface generation

In this section we propose an application of the above obtained results to the construction of unequally smooth quadratic B-spline surfaces.

An unequally smooth B-spline surface can be obtained by taking a bidirectional net of control points $\mathbf{P}_{ij}$, two knot vectors $\bar{u}$ and $\bar{v}$ in the parametric domain Ω, as in Section 2, and assuming the B_{ij}'s (4) as blending functions. It has the following form

$$\mathbf{S}(u,v) = \sum_{(i,j)\in\mathcal{K}_{MN}} \mathbf{P}_{ij}\ B_{ij}(u,v), \quad (u,v)\in\Omega. \tag{5}$$

Here we assume $(s_i,t_j)\in\Omega$ as the pre-image of $\mathbf{P}_{ij}$, with $s_i = \frac{u_{i-1}+u_i}{2}$ and $t_j = \frac{v_{j-1}+v_j}{2}$.

We remark that in case of functional parametrization, $\mathbf{S}(u,v)$ is the spline function defined by the well known bivariate Schoenberg-Marsden operator (see e.g. [6, 9]), which is "variation diminishing" and reproduces bilinear functions.

Since the B-splines in $\mathcal{B}_{MN}$ are non negative and satisfy the property of unity partition, the surface (5) has both the convex hull property and the affine transformation invariance one.

Moreover $\mathbf{S}(u,v)$ has C^1 smoothness when both parameters $\bar{u}$ and $\bar{v}$ have no double knots. When both/either $\bar{u}$ and/or $\bar{v}$ have/has double knots, then the surface is only continuous at such knots [8].

Finally, from the B-spline locality property, the surface interpolates both the four points $\mathbf{P}_{00}$, $\mathbf{P}_{M-1,0}$, $\mathbf{P}_{0,N-1}$, $\mathbf{P}_{M-1,N-1}$ and the control points $\mathbf{P}_{ij}$ if both u_i and v_j occur at least twice in $\bar{u}$ and $\bar{v}$, respectively.

Example 1.

We consider a test surface, given by the following functional parametrization:

$$\begin{cases} x = u \\ y = v \\ z = f(u,v) \end{cases},$$

with

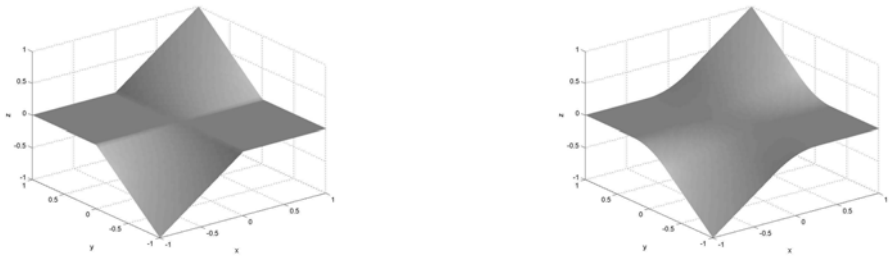
$$f(u,v) = \begin{cases} \mid u \mid v & \text{if } uv > 0 \\ 0 & \text{elsewhere} \end{cases}.$$

We assume $\Omega = [-1,1]\times[-1,1]$ as parameter domain and $m = n = 5$. Moreover we set $\bar{\xi} = \{-1,-0.5,-0.25,0,0.25,0.5,1\}$ and $\bar{\eta} = \bar{\xi}$. We choose $\bar{m}^{\xi} = \{1,1,2,1,1\}$ and $\bar{m}^{\eta} = \bar{m}^{\xi}$. Therefore we have $M = N = 9$ and

$$\bar{u} = \{-1,-1,-1,-0.5,-0.25,0,0,0.25,0.5,1,1,1\}, \quad \bar{v} = \bar{u}.$$

In this case $\mathbf{P}_{ij} = f(s_i,t_j)$. The graph of the corresponding surface (5) is reported in Fig. 7(a). It is obtained by evaluating $\mathbf{S}$ on a 55×55 uniform rectangular grid of points in the domain Ω. In Fig. 7(b) we present the quadratic C^1 B-spline surface, obtained if all knots in $\bar{u}$ and $\bar{v}$, inside Ω, are assumed simple.

 ISBN: 978-84-612-9727-6



(a) (b)

Figure 7: $\mathbf{S}$ with double (a) and simple (b) knots at $\xi_3 = \eta_3 = 0$.

We remark how the presence of double knots allows to well simulate a discontinuity of the first partial derivatives across the lines $u = 0$ and $v = 0$.

Example 2.

We want to reconstruct the spinning top in Fig. 8 by a non uniform quadratic B-spline surface (5).

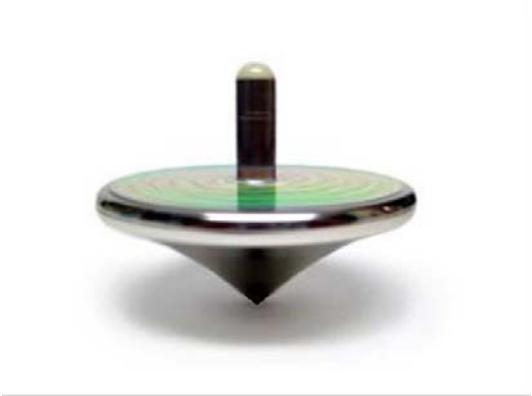


Figure 8: A spinning top.

In order to do it we consider the following control points

$$\mathbf{P}_{00} = \mathbf{P}_{10} = \mathbf{P}_{20} = \mathbf{P}_{30} = \mathbf{P}_{40} = \mathbf{P}_{50} = (0, 0, 0),$$

$$\begin{array}{lll} \mathbf{P}_{01} = (0, \frac{1}{2}, \frac{1}{2}), & \mathbf{P}_{11} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}), & \mathbf{P}_{21} = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}), \\ \mathbf{P}_{31} = (-\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}), & \mathbf{P}_{41} = (-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}), & \mathbf{P}_{51} = \mathbf{P}_{01}, \end{array}$$

$$\begin{array}{lll} \mathbf{P}_{02} = (0, \frac{3}{4}, \frac{7}{12}), & \mathbf{P}_{12} = (\frac{3}{4}, \frac{3}{4}, \frac{7}{12}), & \mathbf{P}_{22} = (\frac{3}{4}, -\frac{3}{4}, \frac{7}{12}), \\ \mathbf{P}_{32} = (-\frac{3}{4}, -\frac{3}{4}, \frac{7}{12}), & \mathbf{P}_{42} = (-\frac{3}{4}, \frac{3}{4}, \frac{7}{12}), & \mathbf{P}_{52} = \mathbf{P}_{02}, \end{array}$$

$$\begin{array}{lll} \mathbf{P}_{03} = (0, \frac{13}{10}, \frac{5}{6}), & \mathbf{P}_{13} = (\frac{13}{10}, \frac{13}{10}, \frac{5}{6}), & \mathbf{P}_{23} = (\frac{13}{10}, -\frac{13}{10}, \frac{5}{6}), \\ \mathbf{P}_{33} = (-\frac{13}{10}, -\frac{13}{10}, \frac{5}{6}), & \mathbf{P}_{43} = (-\frac{13}{10}, \frac{13}{10}, \frac{5}{6}), & \mathbf{P}_{53} = \mathbf{P}_{03}, \end{array}$$

 ISBN: 978-84-612-9727-6

$$\begin{array}{lll}
\mathbf{P}_{04}=(0,1,1), & \mathbf{P}_{14}=(1,1,1), & \mathbf{P}_{24}=(1,-1,1),\\
\mathbf{P}_{34}=(-1,-1,1), & \mathbf{P}_{44}=(-1,1,1) & \mathbf{P}_{54}=\mathbf{P}_{04},\\
\mathbf{P}_{05}=(0,\frac{1}{2},1), & \mathbf{P}_{15}=(\frac{1}{2},\frac{1}{2},1), & \mathbf{P}_{25}=(\frac{1}{2},-\frac{1}{2},1),\\
\mathbf{P}_{35}=(-\frac{1}{2},-\frac{1}{2},1), & \mathbf{P}_{45}=(-\frac{1}{2},\frac{1}{2},1) & \mathbf{P}_{55}=\mathbf{P}_{05},\\
\mathbf{P}_{06}=(0,\frac{1}{8},1), & \mathbf{P}_{16}=(\frac{1}{8},\frac{1}{8},1), & \mathbf{P}_{26}=(\frac{1}{8},-\frac{1}{8},1),\\
\mathbf{P}_{36}=(-\frac{1}{8},-\frac{1}{8},1), & \mathbf{P}_{46}=(-\frac{1}{8},\frac{1}{8},1) & \mathbf{P}_{56}=\mathbf{P}_{06},\\
\mathbf{P}_{07}=(0,\frac{1}{8},\frac{3}{2}), & \mathbf{P}_{17}=(\frac{1}{8},\frac{1}{8},\frac{3}{2}), & \mathbf{P}_{27}=(\frac{1}{8},-\frac{1}{8},\frac{3}{2}),\\
\mathbf{P}_{37}=(-\frac{1}{8},-\frac{1}{8},\frac{3}{2}), & \mathbf{P}_{47}=(-\frac{1}{8},\frac{1}{8},\frac{3}{2}) & \mathbf{P}_{57}=\mathbf{P}_{07},\\
\mathbf{P}_{08}=(0,\frac{1}{8},2), & \mathbf{P}_{18}=(\frac{1}{8},\frac{1}{8},2), & \mathbf{P}_{28}=(\frac{1}{8},-\frac{1}{8},2),\\
\mathbf{P}_{38}=(-\frac{1}{8},-\frac{1}{8},2), & \mathbf{P}_{48}=(-\frac{1}{8},\frac{1}{8},2) & \mathbf{P}_{58}=\mathbf{P}_{08},
\end{array}$$

$$\mathbf{P}_{09}=\mathbf{P}_{19}=\mathbf{P}_{29}=\mathbf{P}_{39}=\mathbf{P}_{49}=\mathbf{P}_{59}=(0,0,2),$$

defining the control net in Fig. 9. Here $M=6$ and $N=10$.

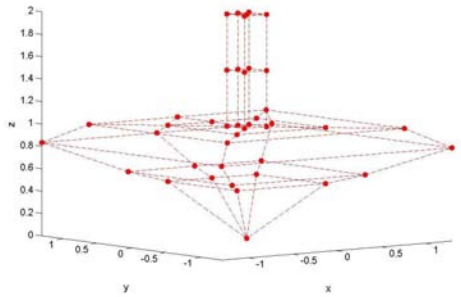


Figure 9: The control net corresponding to $\{\mathbf{P}_{ij}\}_{(i,j)\in\mathcal{K}_{6,10}}$.

Then, to well model our object, we assume $\bar{u}=\{0,0,0,1,2,3,4,4,4\}$ and $\bar{v}=\{0,0,0,1,2,3,3,4,4,5,6,6,6\}$. The graph of the B-spline surface of type (5) is reported in Fig. 10(a), while in Fig. 10(b) the corresponding criss-cross triangulation of the parameter domain is given.

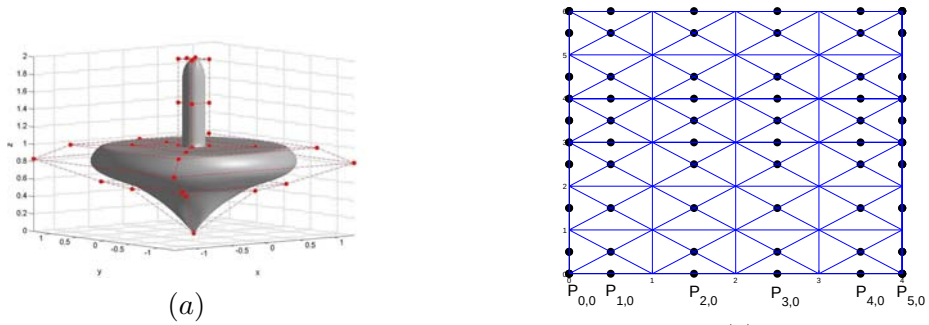
In Fig. 11 we present the quadratic C^1 B-spline surface based on the same control points and obtained if all knots in $\bar{u}$ and $\bar{v}$, inside Ω, are assumed simple, i.e.

$$\bar{u}=\{0,0,0,1,2,3,4,4,4\},\quad \bar{v}=\{0,0,0,1,2,3,4,5,6,7,8,8,8\}.$$

In Fig. 12(a) and (b) the effects of multiple knots are emphasized. We remark that in such a way we can better model the real object.

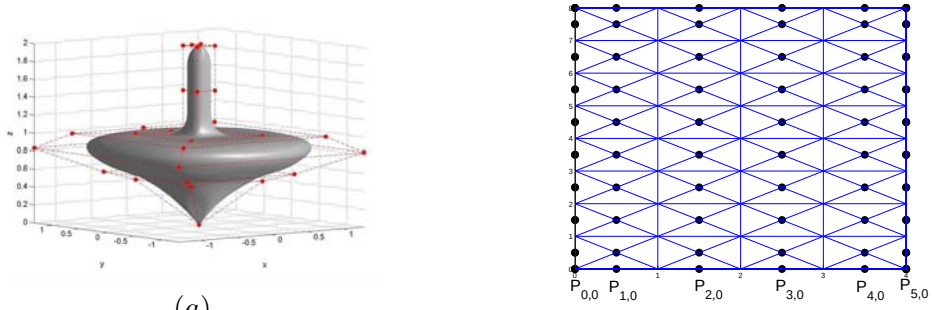
The construction of the B-spline basis and the B-spline surfaces has been realized by Matlab codes.

 ISBN: 978-84-612-9727-6



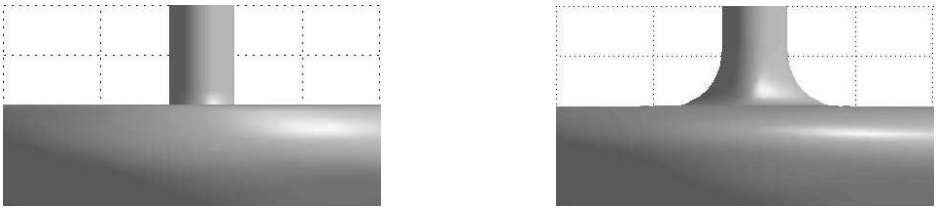
(a) (b)

Figure 10: The surface $\mathbf{S}(u, v)$ with double knots in $\bar{v}$ and its parameter domain.



(a) (b)

Figure 11: The surface $\mathbf{S}(u, v)$ with simple knots inside Ω and its parameter domain.



(a) (b)

Figure 12: In (a) zoom of Fig. 10(a) and in (b) zoom of Fig. 11(a).

4 Conclusions

In this paper we have presented some results on the dimension of the unequally smooth spline space $\mathcal{S}_2^{\bar{\mu}}(\mathcal{T}_{mn})$ and on the construction of a B-spline basis with different types of smoothness.

We plan to use these results in the construction of blending functions for multiple knot NURBS surfaces with a criss-cross triangulation as parameter domain. Moreover such results could be also applied in reverse-engineering techniques, by using surfaces based on spline operators reproducing higher degree polynomial spaces [6, 9].

 ISBN: 978-84-612-9727-6

References

[1] C. de Boor, *A Practical Guide to Splines, Revised Edition*, Springer 2001.

[2] C.K. Chui and R.H. Wang, *Concerning C^1 B-splines on triangulations of non-uniform rectangular partition*, Approx. Theory Appl. **1** (1984) 11–18.

[3] C. Dagnino, P. Lamberti and S. Remogna, *BB-coefficients of unequally smooth quadratic B-splines on non uniform criss-cross triangulations*, Quaderni Scientifici del Dipartimento di Matematica, Università di Torino **24** (2008) http://www.aperto.unito.it/bitstream/2318/434/1/Quaderno+24-08.pdf.

[4] C. Dagnino, P. Lamberti and S. Remogna, *Unequally smooth quadratic spline spaces on type 2 triangulations*, Quaderni Scientifici del Dipartimento di Matematica, Università di Torino **23** (2008) http://www.aperto.unito.it/bitstream/2318/433/1/Quaderno+23-08.pdf.

[5] L. Piegl and W. Tiller, *The NURBS Book, Second Edition*, Springer, Berlin/ Heidelberg/ New York/ Barcelona/ Hong Kong/ London/ Milan/ Paris, 1995.

[6] P. Sablonnière, *On some multivariate quadratic spline quasi-interpolants on bounded domains*, In: W. Hausmann & al. (Eds.), Modern developments in multivariate approximations, ISNM **145**, Birkhäuser Verlag, Basel (2003) 263–278.

[7] P. Sablonnière, *Quadratic B-splines on non-uniform criss-cross triangulations of bounded rectangular domains of the plane*, Prépublication IRMAR (Institut de Recherche Mathématique de Rennes) **03-14** (2003).

[8] R. H. Wang and C. J. Li, *A kind of multivariate NURBS surfaces*, J. Comp. Math. **22** (2004) 137–144.

[9] R. H. Wang, *Multivariate Spline Functions and Their Application*, Science Press, Beijing/New York, Kluwer Academic Publishers, Dordrecht/Boston/London, 2001.