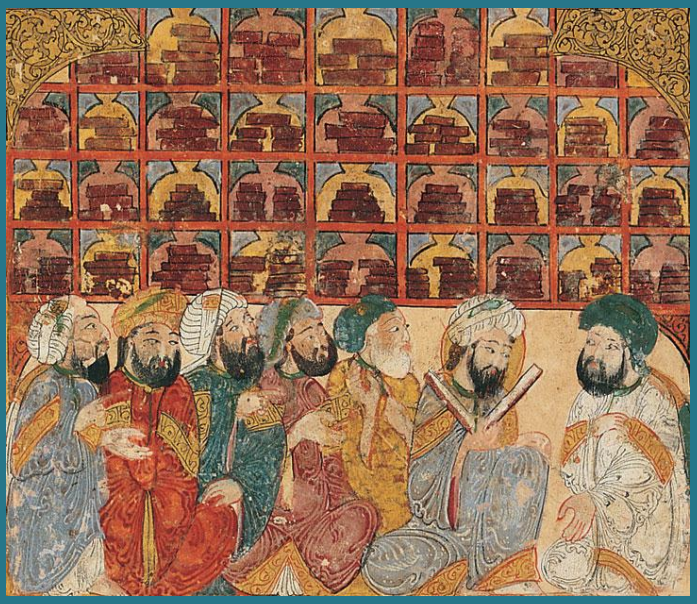
Proceedings of the 2012
International Conference on
Computational and Mathematical
Methods in Science and Engineering
Murcia, Spain
July 2-5, 2012



CMMSE

VOLUME I

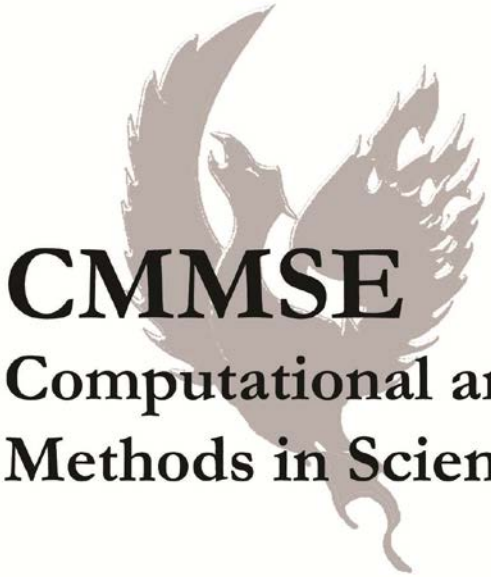
Editor: J. Vigo-Aguiar

Associate Editors:
A.P. Buslaev, A. Cordero, M. Demiralp,
I. P. Hamilton, E. Jeannot, V.V. Kozlov,
M.T. Monteiro, J.J. Moreno, J.C. Reboredo,
P. Schwerdtfeger, N. Stollenwerk, J.R. Torregrosa,
E. Venturino, J. Whiteman

Proceedings of the 2012 International Conference on Computational and Mathematical Methods in Science and Engineering

La Manga, Murcia, Spain

July 2-5, 2012



CMMSE

Computational and Mathematical Methods in Science and Engineering

Editor

J. Vigo-Aguiar

Associate Editors

A.P. Buslaev, A. Cordero, M. Demiralp,
I. P. Hamilton, E. Jeannot, V.V. Kozlov,
M.T. Monteiro, J.J. Moreno, J.C. Reboredo,
P. Schwerdtfeger, N. Stollenwerk, J.R. Torregrosa,
E. Venturino, J. Whiteman

Front cover: Arab anonymous painting
"The origin of Algebra"

ISBN 978-84-615-5392-1

Printed on acid-free paper
Volume I, II & III articles edited with LaTeX
Volume IV articles edited with Microsoft Word

Preface

We are honoured to present the proceedings from the ***"12th International Conference on Computational and Mathematical Methods in Science and Engineering"*** (CMMSE 2012), held at La Manga, Murcia, Spain, from July 2-5, 2012, consisting of the extended abstracts of the presented works at the conference.

From the first meeting at Milwaukee in 2000 until now CMMSE has proved to be a catalyst for computational mathematics in science and engineering. Many scientists have found a place to share their knowledge not only with colleagues from nearby branches, but also with researchers from different areas who have found a new link between their subjects.

CMMSE 2012 mini-symposia and special sessions cover a wide range: advances in the area of high performance computing applied to complex large-scale computational problems; mathematical models for the future Internet; numerical methods for partial differential equations, nonlinear problems and linear and nonlinear optimization; computational chemistry, physics and discrete mathematics; educational methodologies by using the new technologies; mathematical models and information intelligent systems on transport; the influence of algebraic and geometrical tools in cryptography and coding theory; bio-mathematics; orthogonal polynomials and applications; modelling physics phenomena; and, last but not least, industrial mathematics.

New problems with large-scale computing continually arise in many scientific and engineering applications. We are pleased to acknowledge the Spanish Network CAPAP-H2 "High Performance Computing on Heterogeneous Parallel Architectures," whose mini-symposium is supported by the Spanish Ministry of Science and Innovation (project TIN2010-12011-E & TIN2011-15739-E).

In today's "information society," it is essential to preserve the privacy and integrity of communications. Indeed, the internet is continuously changing and in the future the internet will involve a network with new structures that have greater requirements for broadband, scalability, mobility, ubiquity, and economical efficiency. This infrastructure will be united with communication, broadcasting, computing, and sensors. In this year's CMMSE two mini-symposia are devoted to these areas.

The development of new technologies is associated with improvements in human safety (traffic flows, pedestrian flows, ecology, etc), educational methodologies, and with the design of efficient algorithms in information technology. These are just some of the items analyzed and reported on at this conference.

The theory of computation and its applications is one of the most important developments in modern science. Two mini-symposia will be devoted to computational methods applied to chemistry, physics and discrete mathematics.

Many phenomena in science and engineering are modelled by partial or ordinary differential equations and nonlinear systems. They are usually treated numerically so it is necessary to improve the algorithms in terms of efficiency, applicability, and stability. Three mini-symposia and a special session are devoted to recent trends in these topics.

Finally, theoretical and practical applications pertaining to biological mathematics and orthogonal polynomials appear in two mini-symposia in which researchers present the latest results in these branches of investigation.

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 in setting the direction of this conference. Lastly, we thank all participants because without the interest and the input of so many individuals, there would be no conference.

All articles in these volumes are of significant computational interest and contain original and substantial mathematical analysis or development of computational methodology, and contain all the results of the Conference.

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

La Manga, Murcia, Spain, July 2, 2012

J. Vigo-Aguiar, A.P. Buslaev, A. Cordero,
M. Demiralp, I. P. Hamilton, E.Jeannot,
V.V. Kozlov, M.T. Monteiro, J.J. Moreno, J.C. Reboredo,
P. Schwerdtfeger, N. Stollenwerk, J.R. Torregrosa,
E. Venturino, J. Whiteman

CMMSE 2012 Special Sessions

Session Title	Organizers
High Performance Computing (HPC)	Diego Llanos & Enrique S. Quintana-Orti
Mathematically modelling the future Internet and developing future Internet security technology	
Numerical Methods for P.D.E.	Bruce A. Wade
Computational Chemistry and Physics: From clusters to the bulk	Ian P. Hamilton & Peter Schwerdtfeger
New Educational Methodologies Supported by New Technologies	Juan Antonio López-Ramos & José Antonio Pedra Fernández
Computational Discrete Mathematics	J. C. Valverde
Mathematical Models and Information-Intelligent Systems on Transport	Valerii V. Kozlov, Andreas Schadschneider & Alexander P. Buslaev
Computational Methods for Linear and Nonlinear Optimization	M.T. Monteiro
Numerical Methods for Solving Nonlinear Problems	Alicia Cordero & Juan R. Torregrosa
Crypto & Codes	Juan Antonio López-Ramos
Bio-mathematics	Ezio Venturino, Nico Stollenwerk & Maíra Aguiar
Recent Trends on Orthogonal Polynomials and Special Functions	Andrei Martínez Finkelshtein, Francisco Marcellán & Juan José Moreno Balcázar
Industrial Mathematics	Bruce A. Wade
Numerical solution of differential equations	J. Vigo Aguiar

Acknowledgements

We would like to express our gratitude to our sponsor Hewlett-Packard, for its assistance.

We also would like to thank all of the local organizers for their efforts devoted to the success of this conference:

P. Alonso Velázquez, R. Cortina Parajón, (*Universidad de Oviedo),* M.T. de Bustos, B. Martín García, A. Fernández (*Universidad de Salamanca),* J.A. López Ramos (*Universidad de Almería),* F.J. Martínez Zaldivar (*Universidad Politécnica de Valencia),* J.A. Vera, R. de Paco Gabarrón, T. Romera Navarro (*Universidad Politécnica de Cartagena).*

CMMSE 2012 Plenary Speakers

- M. Demiralp, Istanbul Technical University, Turkey
- E. Jeannot, INRIA, France
- P. Schwerdtfeger, Massey University, New Zealand
- N. Stollenwerk, Lisbon University, Portugal
- J. Whiteman, Brunel University London, United Kingdom
- E. Venturino, University of Torino, Italy

Volume I

Contents:

Volume I

Contents:

Volume II

Contents:

Volume III

Contents:

Volume IV

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Principal Logarithm of matrix by recursive methods

J. Abderramán Marrero[1], R. Ben Taher[2] and M. Rachidi[2]

[1] *Department of Mathematics Applied to Information Technologies, Telecommunication Engineering School, U.P.M. Technical University of Madrid, Spain*

[2] *Group of DEFA - Department of Mathematics and Informatics, Faculty of Sciences, University My Ismail, Meknes, Morocco*

emails: jc.abderraman@upm.es, bentaher89@hotmail.fr, mu.rachidi@hotmail.fr

Abstract

We propose new methods for producing explicit representations of the principal matrix logarithm, without to use the Jordan canonical form of the original matrix. They are based on the Hörner decomposition of the matrix and the Binet formula for the general solution of linear recurrence relations.
Key words: Binet formula, Recurrence relations, Matrix powers, Logarithm of a matrix.

MSC 2000: Primary 15A99, 40A05 Secondary 40A25, 45M05, 15A18.

1 Decomposition of the principal logarithm of matrix

The logarithm of matrix occurs in various fields of mathematics, applied sciences and engineering; e.g. see [8]. Particularly, computing logarithms of real matrices turns out to be crucial in some class of problems of medical imaging and system of identification [1]. Various approaches, methods and algorithms are expanded in producing representations of the matrix logarithm (see [4, 6, 8]), and its computation still an exciting area. On the other hand, the numerical aspect of the matrix logarithm is also an interesting research subject (see [8]).

Originally, for a matrix B in $M_d(\mathbb{C})$, the algebra of square matrices, the problem consists in finding $X \in M_d(\mathbb{K})$, satisfying the equation $e^X = T$. Any solution of this equation, denoted $X = log(B)$, is called *logarithm of* B. It was shown in [6] that a matrix B has a logarithm (not necessary real) if and only if B is invertible. Moreover, the matrix equation $\exp(X) = B$ may have infinitely many solutions. Meanwhile, if $B \in GL(d, \mathbb{C})$ have no

eigenvalues on the closed negative real axis, there exists a unique logarithm X of B, called the *principal logarithm* of B and denoted by $X = Log(B)$, e.g. see [4, 6, 8]. This unique matrix logarithm has all its eigenvalues into the horizontal strip determined by the condition $\{\lambda_i(X) \in \mathbb{C} : |Im(\lambda_i(X))| < \pi\}$. In addition, if B is a real matrix then its principal logarithm is real.

The simplest way to define $Log(B)$ is the Taylor series $Log(B) = \sum_{n\geq 0}(-1)^n \frac{(B-I_d)^{n+1}}{n+1}$, which makes sense when $A = I_d - B$ (I_d is the identity matrix), satisfies $\| A \| < 1$, for any matrix norm $\| \cdot \|$, or the spectral radius verifies $\rho(A) = \max\{|\lambda|; \lambda \in \sigma(A)\} < 1$, $\sigma(A)$ is the spectrum of A. More precisely, the computation of $Log(B)$ was based in various papers (see [4, 6]) on the powers series $Log(I_d - tA) = -\sum_{n\geq 1} \frac{t^n}{n} A^n$, when it converges. For $A = I_d - B$ and $t = 1$ we recover the series of $Log(B)$. Let A be in $M_d(\mathbb{C})$ such that $\| A \| < 1$ and $R(z) = z^r - a_0 z^{r-1} - \cdots - a_{r-1}$ ($a_{r-1} \neq 0$) such that $R(A) = \Theta_d$ (zero matrix). The decomposition of A^n ($n \geq r$) in the Hörner basis $A_0 = I_d$, $A_1 = A - a_0 I_d$, $\cdots$, $A_{r-1} = A^{r-1} - a_0 A^{r-2} - \cdots - a_{r-2} I_d$, is

$$A^n = u_n A_0 + u_{n-1} A_1 + ... + u_{n-r+1} A_{r-1}, \quad \text{for} \quad n \geq r \tag{1}$$

(see [2, 3]) where $u_0 = 1$, $u_{-1} = \cdots = u_{-r+1} = 0$, and for $n \geq 1$ we show that the term u_n verifies the linear recursive relation of order r,

$$u_{n+1} = a_0 u_n + \cdots + a_{r-1} u_{n-r+1}, \tag{2}$$

where $a_0, a_1, \cdots, a_{r-1}$ are specified as the coefficients of $\{u_n\}_{n\geq -r+1}$ (see [5]). Using (1) we derive that, for $t \in \mathbb{R}$ with $\sigma(I_d - tA) \cap (\mathbb{R}^- \cup \{0\}) = \emptyset$ and $|t| < 1/\| A \|$, we have

Theorem 1 [Hörner decomposition]. *Under the preceding data, we obtain*

$$Log(I_d - tA) = \sum_{s=0}^{r-1} (-\varphi_s(t))\, A_s, \ \textit{where}\ \varphi_s(t) = \sum_{n=1}^{\infty} u_{n-s} \frac{t^n}{n}. \tag{3}$$

For the polynomial decomposition of A^n we obtain $A^n = \sum_{p=0}^{r-1} \left(\sum_{j=0}^{p} a_{r-p+j-1} u_{n-j} \right) A^p$, for $n \geq r$. Therefore, the polynomial decomposition of $Log(I_d - tA)$ can also be provided.

2 Binet formula for the principal logarithm of matrix

The solutions of sequence (2) can be written using the Binet formula $u_n = \sum_{i=1}^{l} \sum_{j=0}^{m_i - 1} c_{i,j} n^j \lambda_i^n$, for all $n \in \mathbb{N}$, where the λ_i ($1 \leq i \leq l$) are the roots of the polynomial $R(z)$, of multiplicities m_i ($1 \leq i \leq l$). The coefficients $c_{i,j}$ are obtained by solving the linear system $\sum_{i=1}^{l} \sum_{j=1}^{m_i} c_{i,j} n^j \lambda_i^n = u_n (= \delta_{0,-n})$, $n = -r+1, ..., -1, 0$ (see [5]). Set $\Delta_1 = \{i\, (1 \leq i \leq$

 ISBN:978-84-615-5392-1

$l) : m_i = 1\}$ and $\Delta_2 = \{i\,(1 \leq i \leq l); m_i > 1\}$. Then, the Binet formula takes the form $u_n = \sum_{i\in\Delta_1} c_i\lambda_i^n + \sum_{i\in\Delta_2}\sum_{k=0}^{m_i-1} c_{i,k}n^k\lambda_i^n$. For reason of generality we suppose that $\Delta_1 \neq \emptyset$ and $\Delta_2 \neq \emptyset$. Combining (3) of Theorem 1 with the Binet formula of (2), we derive the following main results on the new explicit representation for the principal logarithm of matrix.

Theorem 2 *Let A be in $M_d(\mathbb{C})$ such that $\| A \| < 1$ and $P(A) = \Theta_d$, where $P(z) = z^r - a_0z^{r-1} - \cdots - a_{r-1}$ $(a_{r-1} \neq 0)$. Then, for $t \in \mathbb{R}$ with $\sigma(I_d - tA) \cap (\mathbb{R}^- \cup \{0\}) = \emptyset$, we have $Log(I_d - tA) = \sum_{s=0}^{r-1}(Q_s(t) + \Phi_s(t) + \Psi_s(t))A_s$, such that $Q_s(t)$ is a polynomial of degree $\leq r-1$, given by*

$$Q_s(t) = \sum_{n=1}^{s-1}\left(\sum_{i\in\Delta_1} c_i\lambda_i^{n-s} + \sum_{i\in\Delta_2}\sum_{j=0}^{m_i-1} c_{ij}(n-s)^j\lambda_i^{n-s}\right)\frac{t^n}{n}. \tag{4}$$

The function $\Phi_s(t)$ is

$$\Phi_s(t) = \sum_{i\in\Delta_1}\frac{c_i}{\lambda_i^s}Log(1-\lambda_i t) + \sum_{i\in\Delta_2}\sum_{j=0}^{m_i-1}\frac{c_{ij}}{\lambda_i^s}(-s)^j Log(1-\lambda_i t), \tag{5}$$

and finally the rational function $\Psi_s(t)$ is

$$\begin{aligned}\Psi_s(t) &= \sum_{i\in\Delta_2}\sum_{j=1}^{m_i-1}\frac{c_{ij}}{\lambda_i^s}\sum_{k=1}^{j}\binom{j}{k}(-s)^{j-k}D^k\left(Log(1-\lambda_i t)\right)\\ &= \sum_{i\in\Delta_2}\sum_{j=1}^{m_i-1}\frac{c_{ij}}{\lambda_i^s}\left(\frac{-(j)\lambda_i t}{(1-\lambda_i t)} + \sum_{k=2}^{j}\binom{j}{k}(-s)^{j-k}\frac{P_{k-1}(\lambda_i t)}{(1-\lambda_i t)^k}\right)\end{aligned} \tag{6}$$

where $D = t\dfrac{d}{dt}$ (degree derivation) is a differential operator and the P_n are polynomials satisfying $P_{n+1}(t) = t(1+t)\dfrac{dP_n}{dt}(t) - ntP_n(t)$, for $n \geq 1$.

Example 1 *Let consider $B = \begin{pmatrix} 0 & \frac{5}{16} & -\frac{1}{32} \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$. Then, $A = I_3 - B = \begin{pmatrix} 1 & -\frac{5}{16} & \frac{1}{32} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ is a (real) companion matrix. Binet formula yields $u_n = 4\left(\frac{1}{2}\right)^n - (3+n)\left(\frac{1}{4}\right)^n$. Formulas of Theorems 1 and 2 provide us the principal logarithm of matrix B; $Log(B) = \sum_{s=0}^{2}(-\varphi_s(1))\,A_s$, where $(-\varphi_0(1)) = 2\ln(2) - 3\ln(3) + \frac{1}{3}$, $(-\varphi_1(1)) = 8\ln(\frac{2}{3}) + \frac{4}{3}$, and $(-\varphi_2(1)) = 16\ln(\frac{2}{3}) + \frac{16}{3}$. Matrices $A_0 = I_3$, A_1, and A_2 are the components of the Hörner basis of the matrix A. A direct computation shows that,*

$$Log(B) = \begin{pmatrix} 2\ln(2) - 3\ln(3) + \frac{1}{3} & 2\ln(\frac{3}{2}) - \frac{1}{4} & \frac{1}{4}\ln(\frac{2}{3}) + \frac{1}{24} \\ 8\ln(\frac{2}{3}) + \frac{4}{3} & 5\ln(3) - 6\ln(2) - 1 & \frac{1}{2}\ln(\frac{2}{3}) + \frac{1}{6} \\ 16\ln(\frac{2}{3}) + \frac{16}{3} & 8\ln(\frac{3}{2}) - 4 & \frac{2}{3} - \ln(2) \end{pmatrix}.$$

 ISBN:978-84-615-5392-1

3 Concluding remarks and perspective

In the best of our knowledge expressions (4), (5) and (6) are not current in the literature. Generally in various studies the Jordan canonical form of the matrix B plays an important role for determining $Log(I_d - tA)$ (or $Log(B)$) (see [4, 8]). Meanwhile, the Jordan canonical form is not necessary for exhibiting the Hörner (or polynomial) decomposition of $Log(I_d - tA)$, and also for the usage of Binet formula in Theorem 2 and Example 1.

Our methods have some interesting perspective. We have already obtained some results. Particularly, the problem of computing the principal logarithm of matrices of order 2 and 3 are detailed. Moreover, for $\sigma(A) \nsubseteq \mathbb{D} = \{z \in \mathbb{C} : |z - 1| < 1\}$, our methods work by considering some matrix transformations.

References

[1] V. Arsigny, X. Pennec and N. Ayache, *Polyrigid and polyaffine transformations: a novel geometrical tool to deal with non-rigid deformations application to the registration of histological slices*, *Medical Image Analysis*, **9** (2005), 507–523.

[2] R. Ben Taher and M. Rachidi, *On the matrix powers and exponential by r-generalized Fibonacci sequences methods: the companion matrix case*, *Linear Algebra and Its Applications*, **370** (2003) 341–353.

[3] R. Ben Taher, M. Mouline and M. Rachidi, *Fibonacci-Horner decomposition of the matrix exponential and the fundamental system of solutions*, *Electron. J. Linear Algebra*, **15** (2006) 178–190.

[4] W. J. Culver, *On the existence and uniqueness of the real logarithm of matrix*, *Proc. of the Amer. Math. Soc.*, **17**, No. 5 (1966) 1146–1151.

[5] F. Dubeau, W. Motta, M. Rachidi and O. Saeki, *On weighted r-generalized Fibonacci sequences*, *Fibonacci Quarterly*, **35** (1997) 102–110.

[6] F. R. Gantmacher, *Theory of Matrices*, Vol. I, Chelsea, New York, 1960.

[7] G. Golub and C. Van Loan, *Matrix Computations*, 3rd Ed., Johns Hopkins Press, Baltimore, 1996.

[8] N. Higham, *Functions of Matrices: Theory and Computation*. SIAM Philadelphia, 2008.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

An extension of the Ikebe algorithm for the inversion of Hessenberg matrices

J. Abderramán Marrero[1] and V. Tomeo[2]

[1] *Department of Mathematics Applied to Information Technologies, Telecommunication Engineering School, U.P.M. Technical University of Madrid, Spain*

[2] *Department of Algebra, School of Statistics, U.C.M. Complutense University of Madrid, Spain*

emails: jc.abderraman@upm.es, tomeo@estad.ucm.es

Abstract

Ikebe algorithm for computing the lower half of the inverse of any (unreduced) upper Hessenberg matrix is extended here to compute the entries of the superdiagonal. It gives rise to an algorithm of inversion based on the factorization $\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}$. The lower Hessenberg matrix $\mathbf{H}_L$ is a quasiseparable one and $\mathbf{U}^{-1}$ is upper triangular, with diagonal entries $u_{i,i} = 1$. Its computational complexity, $O(n^3)$, is connected with back substitution for the inversion of the matrix $\mathbf{U}$. Moreover, the inverses of quasiseparable Hessenberg matrices are obtained in $O(n^2)$ times. Numerical comparisons with other specialized algorithms of inversion are also introduced.

Key words: Computational complexity, Hessenberg matrix, Ikebe algorithm, inverse matrix, matrix factorization
MSC 2000: 15A09, 15A15, 15A23, 65F05, 65Y20

1 An algorithm for computing and factorizing the inverses of unreduced Hessenberg matrices

Hessenberg matrices have a main role in numerical linear algebra, in particular for the eigenvalue problem of a general matrix. Furthermore the search of fast and simple algorithms for the inverses of such structured matrices is of current interest. Ikebe algorithm [6], yields the entries of the upper half of the inverse of any (unreduced) lower Hessenberg matrix with $O(n^2)$ complexity. Two algorithms with $O(n^3)$ complexity have been recently

introduced, [3, 4], for computing the inverse matrix and the determinant of (unreduced) lower Hessenberg matrices.

Our aim here is to propose a procedure to compute the factorization $\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}$, of the inverse of an unreduced Hessenberg matrix $\mathbf{H}$. Matrix $\mathbf{H}_L$ is a quasiseparable one, i.e. $rank\,(\mathbf{H}_L(i+1:n, 1:i)) \leq 1$, $rank\,(\mathbf{H}_L(1:i, i+1:n)) \leq 1$, $i = 1, 2, ..., n-1$; e.g. see [2]. Matrix $\mathbf{U}$ is an upper triangular matrix, with ones on its main diagonal. We take upper Hessenberg matrices without loss of generality. The lower Hessenberg matrix $\mathbf{H}_L$ is directly obtained by a simple extension of Ikebe algorithm to the superdiagonal entries of $\mathbf{H}^{-1}$.

We recall the Ikebe algorithm, adapted here to an (unreduced) upper Hessenberg matrix $\mathbf{H}$ of order n. It provides the lower half of the inverse matrix $\mathbf{H}^{-1}$, i.e. $h_{i,j}^{(-1)}$ with $i \geq j$. Following [6], we have

$$h_{i,j}^{(-1)} = y(i) \cdot x(j); \qquad i \geq j, \tag{1}$$

where $y(i)$ and $x(j)$ are the components ith and jth of the vectors $\vec{y}$ and $\vec{x}$, respectively. The components of vector $\vec{x}$ were achieved in the following recursive way, with $h_{j,j-1}^{-1} = 1/h_{j,j-1}$,

$$\begin{aligned} x(1) &= \lambda \neq 0 \qquad \text{(an arbitrary constant)}, \\ x(j) &= -h_{j,j-1}^{-1} \sum_{k=1}^{j-1} h_{k,j-1} x(k) \qquad (j = 2, 3, ..., n). \end{aligned} \tag{2}$$

The entries of vector $\vec{y}$ were also given by the following recurrence,

$$\begin{aligned} y(n) &= \left(\sum_{k=1}^{n} h_{k,n} x(k) \right)^{-1}, \\ y(i) &= -h_{i+1,i}^{-1} \sum_{k=i+1}^{n} h_{i+1,k} y(k) \qquad (i = n-1, n-2, ..., 1). \end{aligned} \tag{3}$$

Now we compare equations (2)-(3) with the closed representation for the entries of the inverses of upper Hessenberg matrices; see [1], Corollary 1. Therefore, we can extend the Ikebe algorithm to obtain the entries of the superdiagonal, $h_{i,i+1}^{(-1)}$, $i = 1, ..., n-1$, of $\mathbf{H}^{-1}$.

Proposition 1 *The entries for the superdiagonal of the inverse of an (unreduced) upper Hessenberg matrix* $\mathbf{H}$ *can be represented as*

$$h_{i,i+1}^{(-1)} = y(i) \cdot x(i+1) + h_{i+1,i}^{-1}; \qquad 1 \leq i \leq n-1, \tag{4}$$

where $y(i)$ *and* $x(i+1)$ *given by (3) and (2), respectively, are obtained from the Ikebe algorithm.*

In addition, we can recover from $y(n)$ the value of $\det \mathbf{H}$, the determinant of the matrix $\mathbf{H}$, with the convention $\det \mathbf{H}_0^{(n)} = 1$,

$$\det \mathbf{H} \quad = \quad (-1)^{n-1} \frac{(\prod_{k=2}^{n} h_{k,k-1})}{\lambda \cdot y(n)}.$$

Taking into consideration that the resulting matrix $\mathbf{H}_L$ contains the lower half plus the superdiagonal of the inverse matrix $\mathbf{H}^{-1}$, and $\mathbf{H} \cdot \mathbf{H}^{-1} = \mathbf{I}_n$, we obtain the product $\mathbf{H} \cdot \mathbf{H}_L = \mathbf{U}$, an upper triangular matrix with ones in its main diagonal. Therefore both matrices $\mathbf{H}_L$ and $\mathbf{U}$ are nonsingular. It gives rise to our main result; a particular factorization for the inverses of (nonsingular) upper Hessenberg matrices.

Theorem 1 *Let $\mathbf{H}$ be a nonsingular matrix of order n. Then the following statements are equivalent:*

1. *$\mathbf{H}$ is an upper Hessenberg matrix.*
2. *The inverse matrix $\mathbf{H}^{-1}$ has a factorization of the form $\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}$, where the lower Hessenberg matrix $\mathbf{H}_L$ is a quasiseparable matrix, and $\mathbf{U}^{-1}$ is an upper triangular matrix with ones in its main diagonal.*

An equivalent result can be proposed for (nonsingular) lower Hessenberg matrices.

From these results, since in the unreduced case matrix $\mathbf{H}_L$ is obtained from the Ikebe expanded algorithm, we propose a constructive method for computing and factorizing the inverses of unreduced Hessenberg matrices. Note as its computational complexity, $O(n^3)$ for general nonsingular Hessenberg matrices, is connected with back substitution for the inversion of the upper triangular matrix $\mathbf{U}$, with $u_{i,i} = 1$ $(1 \leq i \leq n)$.

2 Numerical examples

To begin with the numerical examples, we use *Matlab*® commercial package in a computer of 1.80 GHz. First, we handle a quasiseparable upper Hessenberg matrix with an order $n = 5$, transpose of a customary example given in [3, 4]. Its entries are $h_{ij} = 1$ for $1 \leq i \leq j \leq 5$, $h_{ij} = -1$ for $2 \leq i = j + 1 \leq 5$, and $h_{ij} = 0$, otherwise. Our constructive procedure gives the outcomes for the entries of the inverse in $O(n^2)$ times. Nevertheless, the algorithms from [3, 4] provide results in $O(n^3)$ times.

```
>> H^(-1) = 0.5000   -0.5000         0         0         0
            0.2500    0.2500   -0.5000         0         0
            0.1250    0.1250    0.2500   -0.5000         0
            0.0625    0.0625    0.1250    0.2500   -0.5000
            0.0625    0.0625    0.1250    0.2500    0.5000
```

 ISBN:978-84-615-5392-1

Order	Elouafi - Hadj	Chen - Yu	Ikebe expanded
15	1.75e-13	1.67e-13	1.68e-14
35	7.37e-10	1.54e-10	5.34e-14
55	2.45e-06	1.31e-06	8.65e-14
75	1.36e-02	5.58e-03	2.57e-13
95	4.78e+01	2.62e+01	1.49e-13
115	1.02e+05	6.15e+04	2.57e-13
135	3.09e+08	2.29e+08	7.21e-13
155	2.55e+12	1.02e+12	2.03e-12

Table 1: Values of $norm(\mathbf{H}^{-1}\mathbf{H} - \mathbf{I})$ for the accuracy of the three algorithms in the computation of the inverse of a quasiseparable Hessenberg matrix.

To check the accuracy, we compare in Table 1 the outcomes for the $norm(\mathbf{H}^{-1}\mathbf{H} - \mathbf{I})$, with $\mathbf{I}$ the identity matrix, in the inversion of a Hessenberg matrix, with entries $h_{i,j} = -1$ for $i = j + 1$, $h_{i,j} = -2.5$ for $i \leq j$, and $h_{i,j} = 0$, otherwise. Our procedure also gives the outcomes in $O(n^2)$ times. Algorithms given in [3, 4] produce inaccurate outcomes in $O(n^3)$ times, for the entries of the inverse matrix of this quasiseparable Hessenberg matrix.

When handling more general Hessenberg matrices the three algorithms given outcomes for the inverses in $O(n^3)$ times. However, our procedure produces shorter elapsed times with respect to the given by the algorithms from [3, 4].

References

[1] J. Abderramán Marrero and V. Tomeo, *On the closed representation for the inverses of Hessenberg matrices*, J. Comp. Appl. Math. **236** (2012) 2962–2970.

[2] R. Bevilaqua, E. Bozzo, and G. M. Del Corso, *qd-type methods for quasiseparable matrices*, SIAM J. Matrix Anal. Appl. **32** (2011) 722–747.

[3] Y. H. Chen and C. Y. Yu, *A new algorithm for computing the inverse and the determinant of a Hessenberg matrix*, Appl. Math. Comput. **218** (2011) 4433–4436.

[4] M. Elouafi and A.D. Aiat Hadj, *A new recursive algorithm for inverting Hessenberg matrices*, Appl. Math. Comput. **214** (2009) 497–499.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations, third ed.*, Johns Hopkins University Press, Baltimore, Maryland, USA, 1996.

[6] Y. Ikebe, *On inverses of Hessenberg matrices*, Linear Algebra Appl. **24** (1979) 93–97.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Skeletal based programming for Dynamic Programming on GPUs

Alejandro Acosta[1] **and Francisco Almeida**[1]

[1] *Dpt. Statistics and Computer Science, La Laguna University. Spain.*

emails: aacostad@ull.com, falmeida@ull.com

Abstract

Current parallel systems composed of mixed multi/manycore systems and GPUs become more complex due to their heterogeneous nature. The programmability barrier inherent to parallel systems increases almost with each new architecture delivery. The development of libraries, languages and tools that allow an easy and efficient use in this new scenario is mandatory. Among the proposals found to broach this problem, skeletal programming appeared as a natural alternative to easy the programmability of parallel systems in general but also the GPU programming in particular. In this paper we develop a programming skeleton for Dynamic Programming on GPUs. The skeleton, implemented in CUDA, allows the user to execute parallel codes for GPUs just by providing sequential C++ specifications of her problems. The performance and easy of use of this skeleton has been tested on several optimization problems. The experimental results obtained over a Nvidia Fermi prove the advantages of the approach.

Key words: Skeleton, GPU, Dynamic programming.

1 Introduction

Today's generation of computers is based on an architecture with identical multiple processing units consisting of several cores (multicores). The number of cores per processor is expected to increase every year. It is also well known that the current generation of compilers is incapable of automatically exploiting the ability this architecture affords applications.

The situation is further complicated by the fact that current architectures are heterogeneous by nature, which offers the possibility of combining these multicore with GPU-based systems, for example, in a general purpose architecture. The programmability of these systems, however, poses a barrier that hampers efficient access to its exploitation.

Many proposals have been put forth to facilitate the job of programmers. Leaving aside proposals based on the development of new programming languages due to the effort this represents for the user (effort to learn and reuse code), the remaining proposals are based on transforming sequential code into parallel code, or on transforming parallel code designed for one architecture into parallel code designed for another.

Skeletal programming for GPUs on domain specific applications have been provided by several authors, Patus [2] for stencil computations or Delite [1] for Machine Learning problems among others. More general approaches have been presented by SkelCL [5] or SkePU [3] to make easier the programming of GPU architectures.

In this paper also we propose the use of skeletal based programming to exploit GPUs. An advantage of the paradigm is that the user provides sequential specifications of her problem and the skeleton implements the parallelization of the algorithm to solve it. We instantiate the method over the Dynamic Programming technique. In [4] we proposed the use of DPSKEL skeletons to offset the dearth of general software dynamic programming (sequential and parallel) tools. Our aim was to bridge the obvious gap existing between general methods and DP applications. The goal of DPSKEL is to minimize the user effort required to work with the tool by conforming as much as possible to the use of standard methodologies. In this paper we have expanded the original version of DPSKEL to adapt it to new architectures. On this occasion we developed the solution engine for GPU architecture using CUDA. The proposed implementation shows several advantages, it allows the easy development and fast prototyping of Dynamic Programming problems on GPUs since it hides the parallel traversing of the Dynamic Programming Table and also hides the difficulty of CUDA programming. Another advantage is that the skeleton can be adapted to changes in the architecture and to the programming interface without altering the Dynamic Programming code for the specific problem provide by the user. As a proof of the easy of use of our tool, four combinatorial optimization problems have been instantiated: the 0/1 Knapsack problem, the Resource Allocation problem, the Triangulation of Convex Polygons problem and the Guillotine Cutting Stock problem. Computational results over a Nvidia Fermi C2050 have been provided for all test problems and a comparative analysis of the performance when compared with shared memory skeletons.

The paper is structured as follows, we present the GPU skeleton developed and its software architecture in section 2, and section 3 describes the expansive computational experiment undertaken as a result of applying the method developed. The ease of development and the increase in productivity are substantial. We conclude the paper with section 4, in which we outline the key findings and propose futures lines of research.

2 A GPU Skeleton for Dynamic Programming

As stated in DPSKEL [4] developing software skeletons for DP implies analysing and determining those elements that can be extracted from a specific case and whose elements depend on the application. Assuming that the user is capable of obtaining the functional equations herself, in DPSKEL the user provides the structure of a state and its evaluation through the functional equations, and the DP table is abstracted as a state table. DPSKEL provides the table and several methods for accessing it during the state evaluation process. These methods allow for different traversing (by rows, columns, diagonally), with the user choosing the best one based on the dependencies of the functional equations. In the sequential case, the traversing chosen for the table indicates that the states of the row (column or diagonal, respectively) will be processed sequentially, while in the GPU case, a set of rows (columns or diagonals, respectively) will be assigned to a set of threads to be processed simultaneously. This approach allows us to introduce any of the algorithm parallelization strategies devised for DP.

DPSKEL adheres to the classes model described in figure 1. The concepts of State and Decision are abstracted to the user in C++ classes (required). The user describes the problem, solution and methods for evaluating the state (the functional equation) and for obtaining the optimal solution. DPSKEL provides the classes (provided) for assigning and evaluating the DP table, making available the methods needed to yield the solution. The implementation details are hidden from the user. The initial versions featured solution engines for managing the sequential and parallel executions on shared and distributed platforms. In this paper we have developed the engines for GPU systems. Each solution engine implements different ways of accessing the DP table. We will now present some basic classes in DPSKEL.

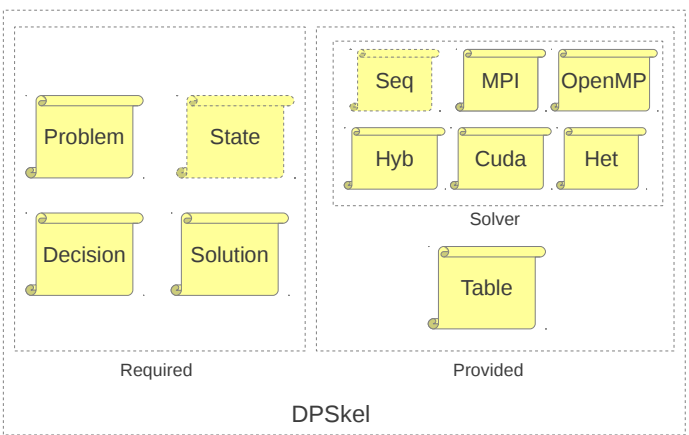


Figure 1: DPSkel classes model

 ISBN:978-84-615-5392-1

To develop the proposed GPU skeleton, we have used the CUDA SDK. We have chosen CUDA (instead of OpenCL) since it allows to execute C++ code inside a kernel, what enables the use the model of classes imposed in DPSKEL. We evaluated the use of OpenCL, however current versions are restricted to kernels implemented in C. That would impose a new design of the C++ DPSKEL tool and some lose of the generality provided by the object oriented programming.

2.1 The State class

The State class holds the information associated with a DP state. This class stores and calculates the optimal value in the state and the decision associated with the optimal evaluation. The evaluation of a state implies accessing information on other states in the DP table. DPSKEL provides an object of the `Table` class hidden in each instance of the `Solver` class.

Listing 1: Definition of the `State` class. Implementation of the Evalua method of the State class for the 0/1 knapsack problem.

```
1  void State::Evalua(int stage, int index) {
2     Decision dec;
3     int val;
4
5     if (index < w[stage]) {
6        val = 0;
7        dec.setDecision(0);
8     }
9     else if (stage == 0) {
10       val = (p[stage]);
11       dec.setDecision(1);
12    }
13    else {
14       val = max(table->getState((stage-1),index),0,
15                 (table->getState(stage-1,index-w[stage])+p[stage]),
16                 1,dec);
17    }
18
19    setValue(val);
20    setDecision(dec);
21    if ((stage == sol->getRowSol()) && (index == sol->getColSol()))
22       sol->setSolucion(this);
23 }
```

The code shown in Listing 1 defines the state class for the knapsack problem. A problem (`pbm`), a solution (`sol`), a decision (`d`) and the DP table (`table`) are defined. These variables can be regarded as generic variables, since they must be present in any problem to be solved. The `value` variable stores the optimal profit. We should mention a particular method in this class, the `Evalua` method, which implements the functional equation. The `Evalua` function

receives the indices for a state from the DP table. Any of the recurrences in Table 1 can be expressed with this prototype. If the functional equation for a specific problem requires a different prototype, the skeleton is open to method overloading using the polymorphism present in C++.

Listing 2 shows how the attribute `__host__ __device__` is added in the header to allow the methods of this class to be executed both in the GPU and in the host system.

Listing 2: Header of the `State` class.

```
requires class State{
    int _value;
    Decision decision;
    Problem* pbm;
    Solution* sol;
    Table* table;
public:
    __host__ __device__ void init(Problem* pbm, Solution* sol, Table* table);
    __host__ __device__ void Evalua(int stage, int index);
    ...
};
```

2.2 The Table class

The class `Table` holds the set of `States` that configure the problem. Each entry in the DP table stores all of the information associated with a state. It holds methods to get (`getState(i,j)`) and put (`putState(i,j)`) states from the table. The class `Solver` takes charge of building the table at the beginning of the execution. Note that to create the table, we used the Unified Virtual Addressing (UVA) provided by CUDA, that unifies the system memory and GPU memory into a single address space. We experimentally tested the benefits of using the UVA for our skeleton, better performances and an important increase of the size of the problem. Listing 3 shows the method to allocate the memory for the Dynamic Programming table and to initialize the states involved in it.

Listing 3: Definition of the `Table` class. Initialized method.

```
void Table::init(const Setup &setup, Problem* pbm, Solution* sol){
   NumStages = setup.getNumStages();
   NumStates = setup.getNumStates();

   cudaMallocHost(&cTABLE, Num_Stages*Num_States*sizeof(State));
   for(int i=0; i<Num_Stages*Num_States; i++) {
      cTABLE[i].init(pbm,sol,this);
   }
}
```

 ISBN:978-84-615-5392-1

2.3 The Solver class

The solver class provides solution engines for different platforms. This class contains the data structures and methods needed to carry out a DP execution in keeping with the specifications. In practice, this is a virtual class, with the solver classes provided being defined as a sub-class of this main class. To the already known solution algorithms in DPSKEL:

- Solver_seq. A sequential solver.
- Solver_OpenMP. A solver for shared-memory systems.
- Solver_MPI. A solver for distributed-memory systems.
- Solver_hybrid. A solver for hybrid distributed and shared memory systems.
- Solver_heterogen. A solver for heterogeneous environments.

in the current design, we added a new solver for the GPU:

- Solver_cuda. A solver for gpu systems.

When a Solver class object is instanced, it is created dynamically in the DP table according to the configuration parameters. Listing 4 shows how the DP table is accessed by the runByRows method of the Solver_cuda class. First, we obtain the number of threads to solve the problem, in this case a thread per collumn of the table is generated. Next the sequential traversing of the table is performed calling the kernel (`kernelRun`, see Listing 5). This kernel takes chare of calling the methods to evaluate a row that have been provided by the end user. Each one of the threads evaluates a `state` using the Evalua method supplied by the user. Several Kernels implementing different traversing modes have been implemented:

- KernelrunByRows. Traverses the DP table by rows, one thread per column.
- KernelrunByDiag. Traverses the DP table by diagonally, starting from the main diagonal upward.
- KernelrunByDiag2. Traverses the DP table by diagonally, downward until the secondary diagonal.

As usual in skeletons, the proposed implementation shows several advantages, the parallelism is hidden, and allows the end user to express her problem as a sequential code, moreover it also hides the complexity of the CUDA programming. The user just implements her problem using sequential C++ and added the headers that enable the methods to be used by the GPU. Another important advantage is that new changes in the architecture and in the programming interface, can be faced by adapting the skeleton without introducing any change in the code provided by the end user.

Listing 4: Implementation of the runByRows function of the solution engine for GPU systems.

```
void Solver_cuda::runByRows() {
   int ls = setup.getNumStates();
   int numBlock = (ls % NTHREAD == 0) ? ls/NTHREAD:(ls/NTHREAD)+1;
   dim3 dimGrid(numBlock,1,1);
   dim3 dimBlock(NTHREAD,1,1);

   for (int i = 0; i < setup.getNumStages(); i++) {
       kernelRunByRows<<<dimGrid,dimBlock>>>(i, ls, table);
   }
   cudaDeviceSynchronize();
}
```

Listing 5: Implementation of the KernelrunByRows kernel.

```
__global__ void kernelRunByRows(int i, int lastState, Table *table) {
    int myid = blockIdx.x*blockDim.x+threadIdx.x;
    if (myid < lastState) {
        table->getState(i, myid)->Evalua(i, myid);
    }
}
```

3 Computational Results

In order to validate the skeleton developed, we tested them on several dynamic programming problems (Table 1). We must remember that the data dependencies are different in most of the formulas considered. This means that we have to use different parallel traversing in the DP table. The GPU skeleton used for RAP and KP access the table by rows, those for TCP access it diagonally, starting from the main diagonal upward, and those for GCP move diagonally downward until the secondary diagonal is reached.

The parallel platform used to execute the GPU skeleton is a Nvidia C2050 GPU installed on an Intel i7 host processor. We compare this new skeleton with skeletons developed for system consisting of four AMD Opteron 6128 processors with eight cores each. We used the sequential skeleton and the parallel skeletons that fit better to each problem. For the KP we used the shared-memory skeleton, for the RAP the heterogeneous skeleton and for the TCP and GCP we used the MPI version. The parallel executions on the AMD were developed using 4, 16 and 32 processes. To simplify the experiment, the tests were carried out using square matrices of order 1000, 2000 and 5000.

Table 2 shows the execution times of all the problems proposed using the GPU skeleton. This table provides an overview of each problem's granularity. All of the times are expressed in seconds. We can see how the KP problem have the finest granularity, while the GCP problem exhibits the coarsest granularity.

Problem	Recurrence
0/1 Knapsack KP	$f_{i,j} = max\{f_{i-1,j}, f_{i-1,j-w_i} + p_i\}$
Resource Allocation RAP	$f_{i,j} = p_{1,j}\ if\ i = 1\ and\ j > 0$ $f_{i,j} = \max_{0 \le k < j}\{f_{i-1,j-k} + p_{i,k}\}\ if\ (i > 1)\ and\ (j > 0)$
Triangulation Convex Polygons TCP	$f_{i,j} = cost_i.cost_{i+1}.cost_{i+2}\ if\ (i = (j-2))$ $f_{i,j} = \min_{i<k<j}\{f_{i,k} + f_{k,j} + (cost_i.cost_k.cost_j)\}$
Guillotine Cut GCP	$f_{i,j} = \max \begin{cases} \max_{0 \le k < object}\{profit_k\} \\ \max_{0 \le z \le i/2}\{f_{z,j} + f_{i-z,j}\} \\ \max_{0 \le y \le j/2}\{f_{i,y} + f_{i,j-y}\} \end{cases}$

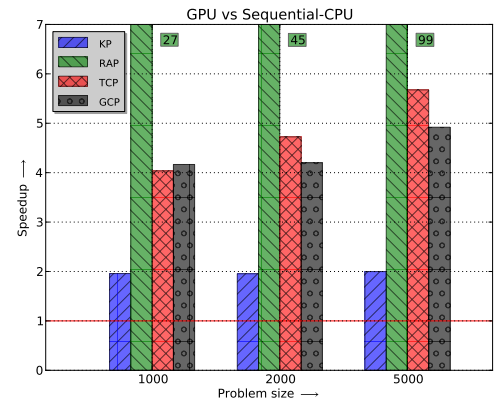
Table 1: Dynamic programming problems analysed

GPU				
Size	KP	RAP	TCP	GCP
1000	0.20446	0.778735	7.60743	20.4213
2000	0.826242	3.75642	58.0168	182.774
5000	5.15905	27.0818	905.123	2864.28

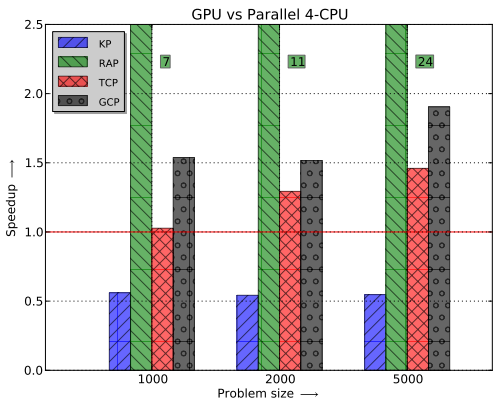
Table 2: Execution Time of the problems analysed

Figure 2 shows the speedup for each one of the problems when compared against the sequential and parallel skeletons. We first observe that the RAP performance is over the remaining problems. One of the reasons for this good behaviour resides in the granularity of the problem. In the case of the RAP, with non-fine grain granularity, the reduction in computational time overpass the cost associated to the data memory movements. In the KP, with a lower speedup, we find a fine grain problem where the total time is coerced by the memory accesses. In both former problems, each iteration launches a thread to compute a column of the table. In the case of the TCP the table is traversed by diagonals starting from the main diagonal. The number of threads is different on each iteration and when the number of iterations increases, the number of threads decreases. In the GCP we have a similar situation, in this case the number of threads increases and later decreases. This kind of traversing where the number of threads becomes very low, negatively influences the performance in GPU architectures.

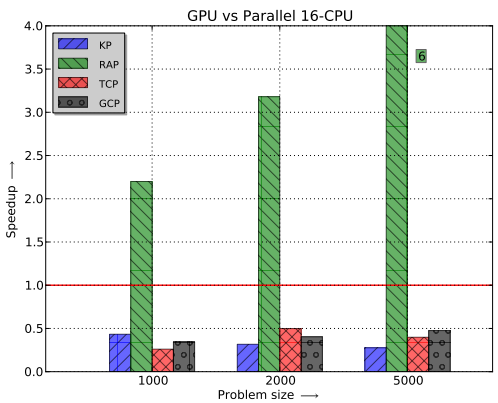
In figure 2(a) we show the speedup of the GPU skeleton compared with the sequential skeleton. In this case all problems show a positive speedup. In the case of the KP, this speedup is constant with the size of the problem, while in the other problems, the speedp increases when the size of the problem increases. This behaviour remains when we increment



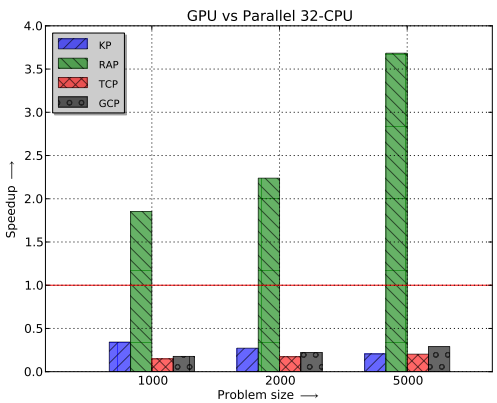
(a) Speedups: CUDA-GPU vs Sequential-CPU



(b) Speedups: CUDA-GPU vs Parallel 4-CPU



(c) Speedups: CUDA-GPU vs Parallel 16-CPU



(d) Speedups: CUDA-GPU vs Parallel 32-CPU

Figure 2: Speedup of the GPU skeleton

the number of CPUs, however in figure 2(b) we compare against the parallel version using 4 CPUs, in this case, the GPU skeleton for the KP is slower than parallel one using 4 CPUs. Figures 2(c) and 2(d) shows the performance using 16 and 32 CPUs, where we can see as only the RAP has a positive speedup. As a consequence of a non-efficient use of the threads in the GPU for the TCP and GCP, they are overpassed by the parallel CPU skeletons. For this classes of recurrences the GPU skeleton should be launched as a hybrid skeleton that dynamically on each iteration choose the use of GPU or CPU when be more efficient.

4 Conclusion

We developed a Dynamic Programming Skeleton for GPUs. The skeleton developed shows the known advantages of skeletal programming of ease of use, programmability and efficiency while hiding the parallelism at the same time. The high productivity of the approach is tested by using four combinatorial optimization problems. The GPU skeleton is compared against parallel skeletons developed for CPUs and the performance is analysed. Although resource allocation problems show very good performances, we observe that some other problems could take advantage of the future development of skeletons that, on each iteration, dynamically select the use of the CPUs or the GPU.

Acknowledgements

This work has been supported by the EC (FEDER) and the Spanish MEC with the I+D+I contract number: TIN2008-06570-C04-03 and TIN2011-24598

References

[1] K. J. Brown, A. K. Sujeeth, H. J. Lee, T. Rompf, H. Chafi, M. Odersky, and K. Olukotun. A heterogeneous parallel framework for domain-specific languages. In *Proceedings of the 2011 International Conference on Parallel Architectures and Compilation Techniques*, PACT '11, pages 89–100, Washington, DC, USA, 2011. IEEE Computer Society.

[2] M. Christen, O. Schenk, and H. Burkhart. Automatic code generation and tuning for stencil kernels on modern shared memory architectures. *Comput. Sci.*, 26(3-4):205–210, June 2011.

[3] J. Enmyren and C. W. Kessler. Skepu: a multi-backend skeleton programming library for multi-gpu systems. In *Proceedings of the fourth international workshop on High-level parallel programming and applications*, HLPP '10, pages 5–14, New York, NY, USA, 2010. ACM.

[4] I. Peláez, F. Almeida, and F. Suárez. Dpskel: A skeleton based tool for parallel dynamic programming. In *Seventh International Conference on Parallel Processing and Applied Mathematics, PPAM2007*, 2007.

[5] M. Steuwer, P. Kegel, and S. Gorlatch. Skelcl - a portable skeleton library for high-level gpu programming. In *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, IPDPSW '11, pages 1176–1182, Washington, DC, USA, 2011. IEEE Computer Society.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Descriptive and Predictive models of dengue epidemiology: an overview

Maíra Aguiar[1], Rick Paul[2], Aanavaj Sakuntabhai[2], Nico Stollenwerk[1] and Sumonmal Uttayamakul[3]

[1] *Centro de Matemática e Aplicações Fundamentais, Universidade de Lisboa Avenida Prof. Gama Pinto 2, 1649-003 Lisboa, Portugal*

[2] *Institut Pasteur, Paris, France*

[3] *Bamrasnaradura Infectious Diseases Institute, Department of Disease Control, Ministry of Public Health, Nonthaburi, Thailand Mahidol University, Bangkok, Thailand*

emails: mairа@ptmat.fc.ul.pt, relpaul@hotmail.com, anavaj.sakuntabhai@pasteur.fr, nico@ptmat.fc.ul.pt, sumonmal@health.moph.go.th

Abstract

In the DENFREE (Dengue Research Framework for Resisting Epidemics in Europe) project, with large emphasis on medical research under the heading of the Pasteur Institute, Paris, the Bio-mathemathics group at CMAF, will manage a work packed on mathematical modeling of spreading of dengue fever and its data analysis of potentially complex dynamics. The main objectives of DENFREE are (1) to identify key factors determining dengue transmission, outcome of infection and epidemics; (2) the development of less-invasive diagnostic tools to detect asymptomatic infections; (3) to evaluate the risk of dengue emergence into non-endemic areas by using Asian communities as a model. In this manuscript we present the official notification dengue hemorrhagic fever (DHF) data from the Ministry of Public Health in Bangkok, Thailand. The multi-strain dengue model, where the notion of at least two different strains is needed to describe differences between primary infection, mainly leading to mild dengue fever (DF) and secondary dengue infections with high risk of DHF, is also described. We present the properties of the minimalistic multi-strain model with a summary of the analysis of the dynamics. The goal is to introduce notation, terminology, and results that will be used in future for formal parameter estimation procedures.

Key words: dengue fever, mathematical models, complex dynamics, data analysis

1 Introduction

Epidemiology of infectious diseases studies the incidence of disease cases and factors that influence disease emergence, transmission and spread in a well-defined population. Epidemiological models are a formal framework to convey ideas about the components of a host-parasite interaction and can act as a tool to predict, understand and develop strategies to control the spread of infectious diseases by helping to understand the behavior of the system under various conditions. They can also aid data collection, data interpretation and parameter estimation. The purpose of epidemiological models is to take different aspects of the disease as inputs and to make predictions about the numbers of infected and susceptible people over time as output.

Dengue fever is a viral mosquito-borne infection, a major international public health concern with more than 55% of the world population at risk of acquiring the infection. Two variants of the disease exist: dengue fever (DF), a non-fatal form of illness, and dengue hemorrhagic fever (DHF), which may evolve toward a severe form known as dengue shock syndrome (DSS). Epidemiological studies support the association of DHF with secondary dengue infection due to a process described as antibody-dependent enhancement (ADE), where the pre-existing antibodies to previous dengue infection cannot neutralize but rather enhance the new infection. Treatment of uncomplicated dengue cases is only supportive, and severe dengue cases require hospitalization and proactive treatment of hemorrhagic symptoms [1]. A vaccine against dengue is not yet available, although several candidates of vaccines are at various stages of development [2].

The dynamics of dengue fever epidemiology shows large fluctuations of disease incidences and mathematical models describing the transmission of dengue viruses appeared in the literature as early as 1970. Retrospective data and the possibility to estimate hidden states in dengue models from such data [3, 4] have been discussed, specially, primary versus secondary infection, and symptomatic versus asymptomatic cases that can be studied via the first already available models [3, 5, 6].

One of the main objectives of the DENFREE project will be the implementation of a new theoretical framework for modelling and preventing dengue. A work packed on descriptive and predictive models of dengue epidemiology will contribute to estimate the risk of spreading DENV to uninfected areas, especially in Southern Europe where susceptible vector exists. The major tools generated will be predictive models that enable specific interventions, whether concerning the environment, mosquito or human, to be made and that can prevent an epidemic. Field epidemiological data collection programs in DENFREE, as part of already established programs and those to be established within the framework of the project proposal, will provide the fine-scale relevant information needed to fine-tune the top-down epidemiological models to be built.

From the Ministry of Public Health MoPH in Bangkok, Thailand, extensions of the already existing time series data sets were obtained up to the present months of 2012. The

time series can now be extended to 30 years.

In this manuscript we present the official notification DHF data from the Ministry of Public Health in Bangkok, Thailand. The minimalistic multi-strain model, motivated by dengue fever epidemiology, where the notion of at least two different strains is needed to describe differences between primary (DF) and secondary dengue infections (DHF) is also described and the validation of this models with a formal parameter estimation procedure is discussed.

2 Modeling dengue fever epidemiology

For effective disease management, whether control, prediction or risk mapping, a better understanding of the transmission dynamics of the virus is a key. In dengue fever epidemiology there are four antigenically related but distinct serotypes (DEN-1 to DEN-4). The occurrence of the virus as four distinct serotypes raises many complications in the analysis and interpretation of serological data. Antibodies generated by exposure to any one strain are known to be cross-reactive for other strains, but they are believed only to provide strain-specific lifelong immunity to reinfection [11]. The immunological response on exposure to a second strain is complex and depends on factors such as patient age, strain type and the interval between exposure to one serotype and exposure to the second serotype. The high antibody titers attained after primary infection appear to generate a degree of cross-protection for a while, but if secondary exposure occurs after antibody levels begin to decline, cross-reactivity appears to act to enhance the growth rate of the new invading viral strain. This is called antibody-dependent enhancement (ADE) and its occurrence in dengue has been used to explain the etiology of serious disease (DHF and DSS) [1, 15, 14, 16, 17, 12, 13].

Epidemic models have been important in understanding the spread of infectious diseases and to evaluate the introduction of intervention strategies like vector control and vaccination. Infectious disease dynamics are by nature nonlinear. To understand such nonlinear epidemiological processes is vital for any modern society from the human as well as the economic perspective, but intrinsically mathematically difficult. Significant progress has been achieved in modelling childhood diseases, because they are so extraordinarily contagious, whereas other often more harmful diseases lag behind in terms of modelling. To make the urgently needed progress in improving our understanding of the dynamics of such diseases, concepts from other fields of mathematics are needed. Although the multi-strain interaction leading to deterministic chaos via ADE has been described previously, e.g. [18, 19, 20], the role of temporary cross-immunity has been neglected leading to unrealistic biological parameter estimation. More recently, despite incorporation of temporary cross immunity in rather complicated models, the possible dynamical structures were not deeply analyzed [21, 22, 23]. When including temporary cross immunity into the ADE models, a rich dy-

 ISBN:978-84-615-5392-1

namic structure including chaos in wider and more biologically realistic parameter regions was found [5, 6]. The model described in [5] is a basic two-strain SIR-type model for the host population motivated by modeling dengue fever epidemiology with its peculiar ADE phenomenology, being associated with secondary infection after primary infection with a different strain. The model consists of a set of classes representing parts of the population to be susceptible, infected and recovered individuals and two virus strains. The processes of infection are described in uniform mixing approximation and immunity to a particular strain is assumed to be lifelong. The demography of the host population is also included, assuming constant population size.

In the following sections we present the available DHF data and the model framework that has shown qualitatively very good results when comparing empirical DHF data and model simulation, offering a promising perspective on inference of parameter values from dengue case notifications.

2.1 Data

The first recorded epidemic of DHF in Thailand (see Fig.1a), population of approximately 66 million people [7]) was in 1958 [1]. The co-circulation of all four dengue serotypes and their capacity to produce severe dengue disease was demonstrated as early as 1960 in Bangkok, Thailand [8]. A system for reporting communicable diseases including DHF/DSS was considered fully installed in 1974 and the data bank of DHF and DSS is available at the Ministry of Public Health, Bangkok [10]. Reasonable data from all provinces exist since the beginning of the 1980s.

In figure 1b), a time series for the DHF cases in Thailand is shown. Note the yearly fluctuations, suggesting that the disease incidence is seasonal. This data come from the Ministry of Public Health of Thailand and consists in monthly incidences of DHF cases.

Large part of the data available to theoretical epidemiologists consists of time series tracking the evolution of a subset of states variables of an underlying dynamical system through a surveillance system. Partially-observed nonlinear stochastic dynamical systems (also known as hidden Markov models are a natural tool to simulate the processes potentially giving rise to such data sets. A hiden Markov model consists of a process model, governing the evolution of states variable along with an observation process. The formulation of various mechanistic process models through nonlinear stochastic dynamical system have been a major focus of theoretical epidemiology. However, difficulties associated with inference from time-series data of the unknown parameters of these models have been a constraint on their applications.

Frequently, the time series of empirical data are used as a qualitative check on model output, however, fitting every detail of the chaotic model to that of the empirical data is not possible. Parameter estimation based on empirical data to estimate initial conditions and model parameters have received great attention and is notoriously difficult for chaotic time

a) 

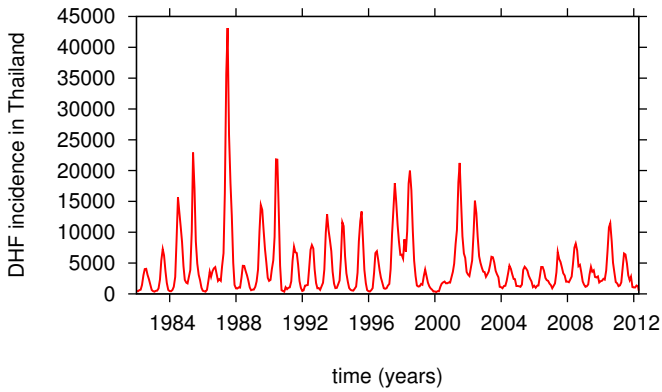
b) 

Figure 1: In a) The provinces of Thailand. DHF case data are available for all provinces. [24]. In b) Time series of monthly DHF cases for the whole country of Thailand.

series. Temporally local approaches are possible using iterated filtering algorithms [25, 26, 27], and at the moment only minimalistic models would have a chance to be qualitatively understood well and eventually tested against existing data.

2.2 Multi-strain model

The basic n-strain epidemiological model with primary and secondary infections can be written as follows

$$\dot{S} = \mu(N-S) - \sum_{i=1}^{n} \frac{\beta}{N} S \left(I_i + \rho \cdot N + \phi \left(\sum_{j=1, j\neq i}^{n} I_{ji} \right) \right) \quad (1)$$

and for $i = 1, ..., n$

$$\dot{I}_i = \frac{\beta}{N}\left(I_i + \rho \cdot N + \phi\left(\sum_{j=1, j\neq i}^{n} I_{ji}\right)\right) - (\gamma + \mu) I_i \tag{2}$$

$$\dot{R}_i = \gamma I_i - (\alpha + \mu) R_i \tag{3}$$

$$\dot{S}_i = \alpha R_i - \sum_{j=1, j\neq i}^{n} \frac{\beta}{N} S_i \left(I_j + \rho \cdot N + \phi\left(\sum_{k=1, k\neq j}^{n} I_{kj}\right)\right) - \mu S_i \tag{4}$$

and for $i = 1, ..., n$ and $j = 1, ..., n$ with $j \neq i$

$$\dot{I_{ij}} = \frac{\beta}{N} S_i \left(I_j + \rho \cdot N + \phi\left(\sum_{k=1, k\neq j}^{n} I_{kj}\right)\right) - (\gamma + \mu) I_{ij} \tag{5}$$

and finally

$$\dot{R} = \gamma\left(\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} I_{ij}\right) - \mu R \quad , \tag{6}$$

dividing the population into 6 groups: susceptible to all existing strains (S), primarily infected with one of the strains (I_i), recovered from the first infection (R_i), susceptible with a previous infection (S_i), secondarily infected with a different strain than the one acquired during the previous infection (I_{ij}) and recovered and life-long immune against all strains (R).

The four-strain epidemiological model can be written as an Eq. system of 26 ODEs (for the complete ODE system). It can be simplified to a three or two-strain model just by neglecting the existence of specific strains. A three-strain model can be obtained by putting $I_4 = 0$ at t_0 and initially no secondary infected for example, where the complete system of ODEs would be reduced to a system of 17 ODEs, and by putting $I_3 = 0, I_4 = 0$ at t_0 and initially no secondary infected, we get back the original two-strain model without any loss of generality, a system of 10 ODEs, once the respective import terms are set to zero. It turns out that the distinction between primary infection and secondary infection is dynamically more important than the exact number of strains to be considered in the model [28], so for the modeling construction at least 5 ingredients are used, here given by the models parameters. Parameter β is the infection rate, describing the transmissibility of the disease. Individuals can become infected with two different infection rates, depending from whom they are getting the infection, β when the infection is coming form an individual in its first infection and $\phi\beta$ when the infection is coming from an individual in its secondary

infection. γ is the recovery rate, μ is the demographic rate, α is the waning immunity rate related to temporary cross-immunity, and ϕ is the ADE ratio which describes the secondary infection contribution to the force of infection. In the models the ADE effect modifies the transmissibility of secondary infections. When ϕ is larger than one, individuals in their secondary infection transmit the disease more than individuals in their first infection, and when ϕ is smaller than one, the individuals in their secondary infection are transmitting less than individuals in their first infection. The parameter ρ is the import factor, related to the possibility of an individual to get infected outside the studied population and then bring the infection into the population to which this individual belongs to, mimicking the imported cases of the disease in a defined population.

Dengue models including multi-strain interactions via ADE but without temporary cross-immunity period e.g. [18, 19, 20] have shown deterministic chaos when strong infectivity on secondary infection was assumed.

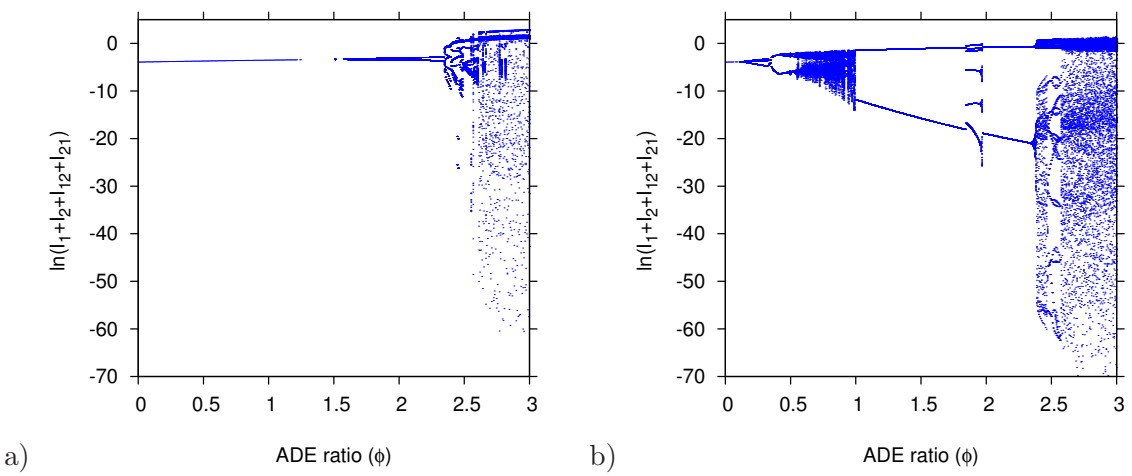


Figure 2: Bifurcation diagram. In a) model neglecting temporary cross-immunity ($\alpha = 52y^{-1}$). In b) model assuming temporary cross-immunity ($\alpha = 2y^{-1}$).

The different forces of infection combined with the temporary cross immunity period bring complex behavior in a unexpected and biologically more realistic parameter region, $\phi > 1$ [5, 6, 3], i.e. deterministic chaos was found in a wider parameter regions. In Fig. 2 we show bifurcation diagrams were the total number of infected individuals in a log scale is plotted over the ADE ratio parameter. Fixed points appear as one dot per parameter value, limit cycles appear as two dots, double-limit cycles as four dots, more complicated limit cycles as more dots, and chaotic attractors as continuously distributed dots for a single ϕ value. We see that when temporary cross immunity is assumed, a new chaotic window appears, and the ADE effect does not need to be restricted to one or another region in

parameter space.

The seasonal model with import shows complex dynamics and qualitatively a very good result when comparing empirical DHF and simulations (see Fig. 3) [3]. However, the extended model needs to be parametrized on data referring to incidence of severe disease.

A qualitatively a very good result when comparing empirical DHF data and simulation suggest that this parameter set could be the starting set for a more detailed parameter estimation procedure. In Fig. 3 the model simulation is matched with the DHF data for the Chiang Mai province in Thailand.

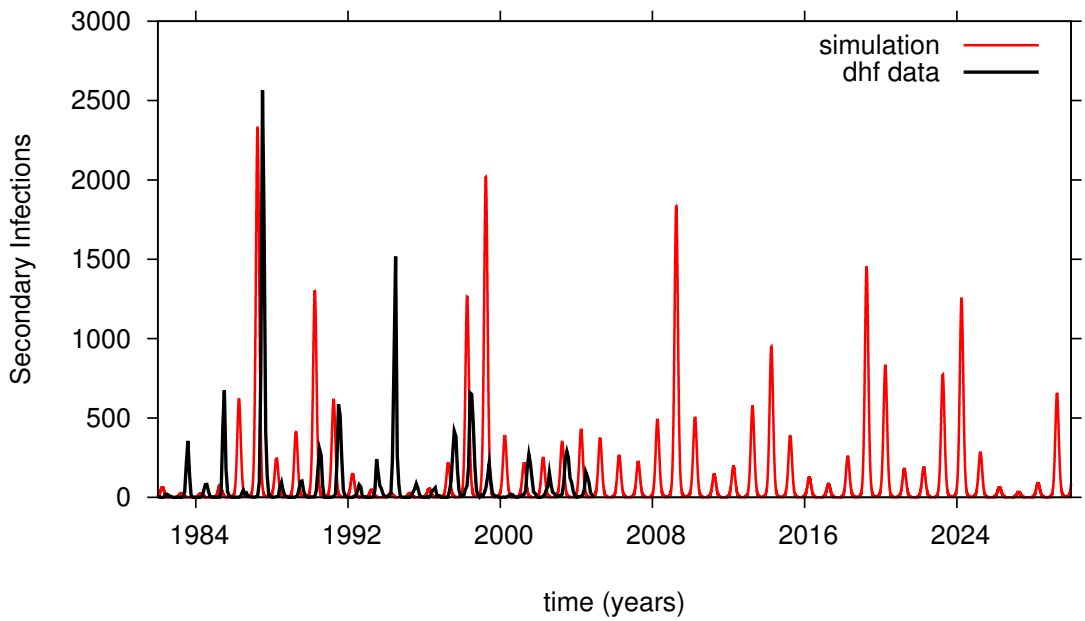


Figure 3: For the following parameter set: $\beta_0 = 2\gamma$, $\gamma = 52y^{-1}$, $\alpha = 2y^{-1}$, $\phi = 0.9$ and $\mu = 1/65y^{-1}$ empirical DHF incidence data (in black) for the province of Chiang Mai in the North of Thailand are matched with simulations (in red) for the seasonal multi-strain models with import of infected. The parameter β is given explicitly by $\beta(t) = \beta_0 \cdot (1 + \eta \cdot \cos(\omega \cdot t))$ where β_0 is the infection rate and η is the degree of seasonality. Here, the degree of seasonality is $\eta = 0.35$ and the import factor $\rho = 10^{-10}$.

However, more stochasticity is needed to get even better agreement for some of the available data sets, as in the case for Bangkok, where the available data is very noisy linked with a low endemicity of DHF cases [29].

The two-strain model in its simplicity is a good model to be analyzed, giving the expected complex behavior to explain the fluctuations observed in empirical data. It is minimalistic in the sense that it can capture the essential differences of primary versus

secondary infection without needing to restrict the ADE effect to one or another region in parameter space. For future parameter estimation only the two-strain model could attempt to estimate all initial conditions as well as the few model parameters.

3 Prospective work and discussion

In this manuscript we presented the official notification DHF data from the Ministry of Public Health in Bangkok, Thailand. The representation of the biological processes using mathematical modeling approach were improved in comparison to previous models based on the following aspects: temporary cross-immunity, antibody dependent enhancement, and seasonality. We now propose to extend the previously existing dengue models, by including vector dynamics, and the transmission contribution of an asymptomatic host and data analysis.

A model which can be fully parametrized on data referring to incidence of disease can become a predictive tool to guide the policies of prevention and control of the dengue virus transmission, including the implementation of vaccination programs when the candidate dengue fever vaccines will be available.

After the expansion of previous multi-strain dengue models, the basic parameters of transmission, infectivity and disease severity (ADE parameter) will be affected. In such a way the model will be parametrized on data referring to incidence of severe disease and prevalence of infection from different endemic countries with different intensities of dengue transmission. Technical parameter estimation is notoriously difficult for chaotic time series but temporally local approaches are possible [27], for example. In the EU project Dengue Research Framework for Resisting Epidemics in Europe (DENFREE), with a work-package on descriptive and predictive models of dengue epidemiology, various sources of data will be accessed to test the models which are able to provide a valuable tool to guide policies of prevention and control of the dengue virus transmission.

Acknowledgements

This work has been supported by the EU project DENFREE under Framework Program 7 and has been further supported by the Portuguese FCT project PTDC/MAT/115168/2009. We thank Bernard Cazelles, Ecole Normale Supérieure, France, and Yoshiro Nagao, Osaka University Graduate School of Medicine in Japan, for enabling us with parts of the existing data sets for Thailand prior to our access of the full data in collaboration with the MoPh in Bangkok.

References

[1] World Health Organization. (2009). *Dengue and Dengue Hemorrhagic Fever, Fact sheet 117*. Retrieved from http://www.who.int/mediacentre/factsheets/fs117/en/

[2] World Health Organization – Programs and Projects: Initiative for Vaccine Research (2011). *Vector borne infections*. Retrieved from http://www.who.int/vaccine_research/diseases/vector/en/index1.html#virology

[3] Aguiar, M., Ballesteros, S., Kooi, B.W., & Stollenwerk, N. (2011) The role of seasonality and import in a minimalistic multi-strain dengue model capturing differences between primary and secondary infections: complex dynamics and its implications for data analysis, *Jounal of Theoretical Biology*, **289**, 181–196.

[4] Stollenwerk, N., Aguiar, M., Ballesteros, S., Boto, J., Kooi, W. B., & Mateus, L. (2012). Dynamic noise, chaos and parameter estimation in population biology, *Interface Focus*, **2**, 156–169.

[5] Aguiar, M., Kooi, B., & Stollenwerk, N. (2008) 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**, 48–70.

[6] Aguiar, M., Stollenwerk, N., & Kooi, B. (2009) Torus bifurcations, isolas and chaotic attractors in a simple dengue fever model with ADE and temporary cross immunity, *Intern. Journal of Computer Mathematics* **86**, 1867–77.

[7] Wikipedia contributors. Wikipedia, The Free Encyclopedia. *Provinces of Thailand*. Retrieved from http://en.wikipedia.org/wiki/Provinces_of_Thailand

[8] Halstead S. B., et al. (1969). Dengue and chikungunya virus infection in man in Thailand, 1962–1964. V. Epidemiologic observations outside Bangkok. *Am. J. Trop. Med. Hyg.* 18, 1022–33.

[9] Gubler D. J., (2002). Epidemic dengue/dengue hemorrhagic fever as a public health, social and economic problem in the 21st century. *Trends in Microbiology*, 10, 100–103.

[10] Chareonsook, O.et al. (1999). Changing epidemiology of dengue hemorrhagic fever in Thailand. *Epidemiol. Infect.*, 122, 161–166.

[11] S. Matheus, X. Deparis, B. Labeau, J. Lelarge, J. Morvan, P. Dussart. *Discrimination between Primary and Secondary Dengue Virus Infection by an Immunoglobulin G Avidity Test Using a Single Acute-Phase Serum Sample.* Journal of Clinical Microbiology (2005), No. 43, 2793–97.

[12] Dejnirattisai, W. et al. (2010). Cross-Reacting Antibodies Enhance Dengue Virus Infection in Humans. *Science*, 328, 745–748.

[13] Guzmán, M.G. et al. (2010). Dengue: a continuing global threat. *Nature Reviews Microbiology*, 8, S7–S16, ISSN : 1740-1526.

[14] Halstead, S.B. (1982). Immune enhancement of viral infection. *Progress in Allergy*, 31, 301–364,ISSN 0079-6034.

[15] Halstead, S. B. (1994). Antibody-dependent Enhancement of Infection: A Mechanism for Indirect Virus Entry into Cells. *Cellular Receptors for Animal Viruses*, 28, Chapter 25, 493–516, ISBN 0-87969-429-7. (Cold Spring Harbor Laboratory Press).

[16] Halstead, S.B. (2003). Neutralization and antibody-dependent enhancement of dengue viruses. *Advances in Virus Research*, 60, 421–467.

[17] Mackenzie, J. S., Gubler, D. J. & Petersen, L. R. (2004). Emerging flaviviruses: the spread and resurgence of Japanese encephalitis, West Nile and dengue viruses. *Nature Medicine Review*, 12, S98–S109.

[18] Ferguson, N., Anderson, R. and Gupta, S. (1999). The effect of antibody-dependent enhancement on the transmission dynamics and persistence of multiple-strain pathogens. *Proc. Natl. Acad. Sci. USA*, 96, 790–94.

[19] Schwartz, I. B., et al. (2005). Chaotic desynchronization of multi-strain diseases. *Physical Review*, E 72, 066201–6.

[20] Billings, L., et al. (2007). Instabilities in multiserotype disease models with antibody-dependent enhancement. *Journal of Theoretical Biology*, 246, 18–27.

[21] Wearing, H.J. & Rohani, P. (2006). Ecological and immunological determinants of dengue epidemics *Proc. Natl. Acad. Sci. USA* , 103, 11802–11807.

[22] Nagao, Y. & Koelle, K.(2008). Decreases in dengue transmission may act to increase the incidence of dengue hemorrhagic fever. *Proc. Natl. Acad. Sci. USA*, 105, 2238–2243.

[23] Recker, M. et al. (2009). Immunological serotype interactions and their effect on the epidemiological pattern of dengue. *Proc. R. Soc. B.*, 276, 2541–2548.

[24] Map of Thailand. Retrieved from http://thethailandlife.com/map-of-thailand

[25] Ionides, E., Breto, C., & King, A. A. (2006). Inference for nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA*, **103**, 18438–18443. (DOI:10.1073.pnas.0603181103.)

[26] He, D., Ionides, E. L., King, A. A. (2010). Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. *J. R. Soc. Interface*, **7**, 271–283. (DOI:10.1098/rsif.2009.0151)

[27] Stollenwerk, N., Aguiar, M., Ballesteros, S., Boto, J., Kooi, W. B., Mateus, L. (2012). Dynamic noise, chaos and parameter estimation in population biology. *Interface Focus*, 2, 156–169.

[28] Aguiar, M., Kooi, W. B., Rocha, F., Ghaffari, P. and Stollenwerk, N. (2012). How much complexity is needed to describe the fluctuations observed in dengue hemorrhagic fever incidence data? *submitted*. Available at arXiv:1111.3844v2 [nlin.CD]

[29] Aguiar, M., Kooi B. W. & Stollenwerk N. (2012). Scaling of stochasticity in DHF epidemics, *accepted for publication in MMNP*.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Dynamics of some Parallel Dynamical Systems over Digraphs

Juan A. Aledo[1], Silvia Martinez[1] and Jose C. Valverde[1]

[1] *Department of Mathematics, University of Castilla-La Mancha*

emails: JuanAngel.Aledo@uclm.es, Silvia.MSanahuja@uclm.es, Jose.Valverde@uclm.es

Abstract

In a previous research, for parallel dynamical systems over digraphs corresponding to the simplest Boolean functions AND and OR, we proved that only fixed or eventually fixed points appear, as occurs over undirected dependency graphs. Nevertheless, for general Boolean functions, it was shown that any period can appear, depending on the Boolean function that infers the global evolution operator of the system and on the structure of the dependency digraph. In this sense, this work analyzes the orbit structure of parallel discrete dynamical systems over some special digraph classes.

Key words: Discrete dynamical systems; periodic orbits; parallel dynamical systems; directed graphs; Boolean functions.
MSC 2000: 37B99; 37E15; 37N99; 68R10; 94C10.

1 Introduction

In the last decade, several works have been dealt with the development of mathematical foundations for a theory of simulation. The first one of a series of these works was [3], where sequentially updated cellular automata (sCA) over arbitrary graphs are employed as a paradigmatic framework. This first step was followed by [4], [5] and [6], where the authors developed this theory, analyzing the asymptotic behavior.

Computer simulations involve generation of dynamics by iterating local mappings. These processes have been usually modeled by cellular automaton. Wolfram [17] analyzed a set of cellular automata and showed that, despite their simple construction, some of them are capable of complex behavior. Later, based on a deeper research [18], he suggested that

many one-dimensional cellular automata fall into four basic behavior classes: three of them exhibiting a similar behavior to fixed points, periodic orbits and chaotic attractors and the fourth one so that asymptotic properties are undecidable. The concept of parallel dynamical system generalizes the one of the cellular automaton.

For convenience, it is common to rename the local mappings as *entities* (cells in the language of cellular automata theory), which are the lowest level of aggregation of the system. In many processes, there are many entities and each entity has a state at a given time (see [3, 4, 5]). The update of states of the entities constitutes an evolution in time of the system, i.e., a discrete dynamical system (see [7, 13]).

The update of the states is determined by dependency relations of the entities and local rules, which together constitute the (global) *evolution operator* of the dynamical system (see [12], [16]). If the states of the entities are updated in a parallel (or synchronous) manner, the system is called a *parallel dynamical system* (PDS), while if they are updated in a sequential order, the system is named *sequential dynamical system* (SDS) [7, 13, 14].

In particular, in [7] parallel and sequential dynamical systems are studied, considering OR (resp. AND) and NOR (resp. NAND) as global functions. Following these ideas, in [1] we extend these results for the parallel case, giving a complete characterization of the orbit structure of any parallel dynamical system with any maxterm (resp. minterm) as a global function. As a result, for the simplest maxterm (resp. minterm), OR (resp. AND), only fixed or eventually fixed points can appear, while for a general maxterm (resp. minterm), uniquely periodic or eventually periodic orbits of period lower than or equal to 2 can be found.

The results in [1] opened new different research directions concerning parallel dynamical systems on Boolean functions. One of them consists in considering non-reciprocal relations between two related entities, because it could occur that an entity influences another one, but not vice versa, as happens in practice [8]. This can be modeled by a *directed dependency graph* or *dependency digraphs* of relations.

These non-reciprocal relations emerge also in other applied models created for the simulation of aspects of the behavior of biological systems. This occurs in [10], where Kauffman constructed molecular automata for modeling a gene as a binary (on-off) device and studied the behavior of large, randomly constructed nets of these binary genes (see also [11]). A Kauffman net of size n and connectivity k consists of n interconnected vertices, each one having k inputs and one output. The update of any gene is determined by the (directed) dependency relations and local rules which are given by random Boolean functions.

In computer simulation, entities are related and they get information from the related ones in their own neighborhood. In order to get a graphical idea of the situation, every entity is usually represented by a vertex of an undirected graph and two vertices are adjacent if their states influence each other in the update of the system. The undirected graph so built is called the *(undirected) dependency graph* of the system (see [7]).

However, as we said before, it could occur that an entity influences another one, but not vice versa. Actually, in practice, the process of information exchange is not bidirectional [8]. This could be represented by an arc whose initial vertex would be the influencing entity and the final vertex would correspond to the influenced entity, so obtaining a directed graph or digraph of relations. The directed graph so built will be called the *directed dependency graph* of the system.

We actually determine a dynamical system over a directed dependency graph (see [15] for a similar approach that considers cellular automata over Cayley graphs) by associating to each vertex i, a state $x_i \in \{0,1\}$ and a local map f_i defined on the states of the influencing vertices and the vertex/entity i, and which returns its new state $y_i \in \{0,1\}$. We shall denote this digraph $D=(V,A)$, where $V=\{1,2,\ldots,n\}$ is the vertex set and A is the arc set.

For every vertex/entity $1 \leq i \leq n$, we shall consider all the vertices that influence it in an update of the system. Thus, we denote

$$I_D(i)=\{j \in V | (j,i) \in A\}$$

The evolution or update of the system is implemented by local functions which are the restrictions of a global one. In this context, for updating the state of an entity i, the corresponding local function acts only on the state of that entity itself and the states of the entities in $I_D(i)$ which influence i.

Actually, it can be stated the following definition.

Definition *Let $D=(V,A)$ be a digraph on $V=\{1,2,\ldots,n\}$. Then a map*

$$F:\{0,1\}^n \to \{0,1\}^n, \qquad F(x_1,x_2,\ldots,x_i,\ldots,x_n)=(y_1,y_2,\ldots,y_i,\ldots,y_n),$$

where y_i is the updated state of the entity/vertex i by applying a local function f_i over the states of the entities in $\{i\} \cup I_D(i)$, constitutes a discrete dynamical system called parallel directed dynamical system *over $\{0,1\}^n$.*

In this work, the global evolution operator F of the system will be induced by *Boolean functions*. A Boolean function describes how to determine a Boolean output from some Boolean inputs. Thus, such functions play a fundamental role in questions as design of circuits or computer processes [9]. In our context, they correspond to components of the evolution operator of the dynamical system.

In a previous paper [2], for parallel dynamical systems over digraphs corresponding to the simplest Boolean functions AND and OR, we proved that only fixed or eventually fixed points appear, as occurs over undirected dependency graphs. Nevertheless, for general Boolean functions, it was shown that any periodic orbit can exist, depending on the Boolean function that infer the global evolution operator of the system and on the structure of the dependency digraph. In this sense, this work analyze the orbit structure of parallel discrete dynamical systems over some special digraph classes, as complete, circle, line, star graphs and arborescences.

 ISBN:978-84-615-5392-1

Acknowledgements

This work has been partially supported by the grants MTM2011-23221, PEII11-0132-7661 and PEII09-0184-7802.

References

[1] J.A. Aledo, S. Martinez and J.C. Valverde, *Parallel Dynamical Systems on Maxterms and Minterms Boolean Functions*, Math. Comput. Model. 35 (2012) 666–671.

[2] J.A. Aledo, S. Martinez and J.C. Valverde, *Parallel Dynamical Systems over directed dependency graphs*, (under review).

[3] C.L. Barret and C.M. Reidys, *Elements of a theory of computer simulation I*, Appl. Math. Comput. 98 (1999) 241–259.

[4] C.L. Barret, H.S. Mortveit and C.M. Reidys, *Elements of a theory of computer simulation II*, Appl. Math. Comput. 107 (2002) 121–136.

[5] C.L. Barret, H.S. Mortveit and C.M. Reidys, *Elements of a theory of computer simulation III*, Appl. Math. Comput. 122 (2002) 325–340.

[6] C.L. Barrett, H.S. Mortveit, C.M. Reidys, *ETS IV: sequential dynamical systems: fixed points, invertibility and equivalence*, Appl. Math. Comput. 134 (2003) 153-171.

[7] C.L. Barret, W.Y.C. Chen and M.J. Zheng, *Discrete dynamical systems on graphs and Boolean functions*, Math. Comput. Simul. 66 (2004) 487–497.

[8] W.Y.C. Chen, X. Li and J. Zheng, *Matrix method for linear sequential dynamical systems on digraphs*, Appl. Math. Comput. 160 (2005) 197212.

[9] O. Colón-Reyes, R. Laubenbacher, B. Pareigis, *Boolean monomial dynamical systems*, Ann. Combin. 8 (2004) 425–439.

[10] S. A. Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets*, J. Theor. Biol. 22 (1969) 437–467.

[11] S. A. Kauffman, Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, Oxford, 1993.

[12] Y.A. Kuznetsov, Elements of Applied Bifurcation Theory, Springer, New York, 2004.

[13] H.S. Mortveit and C.M. Reidys, *Discrete, sequential dynamical systems*, Discrete Math. 226 (2002) 281–295.

[14] H.S. Mortveit and C.M. Reidys, An Introduction to Sequential Dynamical Systems, Springer, New York, 2007.

[15] Z. Róka, Cellular automata on Cayley graphs, Ph.D. Thesis, Ecole Normale Superieure de Lyon, France, 1994.

[16] S. Wiggins, Introduction to Applied Nonlinear Systems and Chaos, Springer, New York, 1990.

[17] S. Wolfram, *Statistical mechanics of cellular automata*, Rev. Mod. Phys. 55 (3) (1983) 601–644.

[18] S. Wolfram, *Universality and Complexity in cellular automata*, Physica D 10 (1984) 1–35.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Parallel Dynamical Systems with Different Local Functions

Juan A. Aledo[1], Silvia Martinez[1] and Jose C. Valverde[1]

[1] *Department of Mathematics, University of Castilla-La Mancha*

emails: JuanAngel.Aledo@uclm.es, Silvia.MSanahuja@uclm.es, Jose.Valverde@uclm.es

Abstract

In this work, we extend the manner of defining the evolution update of discrete dynamical systems on Boolean functions, without limiting the local functions to be dependent restrictions of a global one. We analyze the cases concerned with parallel dynamical systems (PDS) with the OR, AND, $NAND$ and NOR functions as independent local functions. This extension of the update method widely generalizes the traditional one where only a global Boolean function is considered for establishing the evolution operator of the system. Besides, our analysis allows us to show a richer dynamics in these parallel systems on independent local functions.

Keywords: Discrete dynamical systems; parallel dynamical systems; graphs; Boolean functions.

MSC 2000: 37B99; 37E15; 37N99; 68R10; 94C10.

1 Introduction

Computer processes involve generation of dynamics by iterating local mappings. In fact, a computer simulation is a method for the composition of iterated mappings, typically with local dependency regions [5]. It means that the mappings have to be updated in a specific manner, i.e., an *update schedule*. Update scheduling is also a commonly studied aspect in discrete event simulations [3, 8, 9].

For convenience, it is common to rename the local mappings as *entities*, which are the lowest level of aggregation of the system. In computer processes, there are many entities

an each entity has a state at a given time (see [5, 6, 7]). The update of the states of the entities constitutes an evolution in time of the system, i.e., a discrete dynamical system (see [4, 11]).

The update of the states is determined by relations of the entities, which are represented by a dependency graph, and local rules, which together constitute the (global) *evolution operator* of the dynamical system (see [10], [13]). Actually, by means of Boolean functions, local (Boolean) rules to obtain an output from some inputs are obtained. That is, for updating the state of any entity, it is normally considered a Boolean function that acts only on the state of that entity itself and the states of the entities related to it.

If the states of the entities are updated in a parallel manner, the system is called a *parallel dynamical system* (PDS) [1, 2, 4], while if they are updated in a sequential order, the system is named *sequential dynamical system* (SDS) [4, 11, 12]. Thus, the evolution or update of a PDS or SDS is usually implemented by local functions which are dependent restrictions of a given global Boolean function.

However, in practice, the rule for the information exchange among one entity and those related to it in the system can be different from one entity to another. That is, an entity i can exchange information with the entities related to it by means of a rule f_i, while an entity j can do that by means of another rule f_j completely independent or different from the rule f_i.

For this reason, in this work we extend the way of defining the update of discrete dynamical systems on Boolean functions, without limiting the local functions to be dependent restrictions of a global one. This extension of the update method widely generalizes the traditional one, where only a global Boolean function is considered for establishing the evolution operator of the system, and gives as a result a larger variety of discrete dynamical systems on Boolean functions.

We focus on the cases concerned with parallel dynamical systems (PDS) with the OR, AND, $NAND$ and NOR functions as (independent) local functions, determining the orbit structure of this kind of systems. Our analysis allows us to show a richer dynamics in these parallel systems on independent local functions in comparison with the traditional ones (see [1, 4]).

Acknowledgements

This work has been partially supported by the grants MTM2011-23221, PEII11-0132-7661 and PEII09-0184-7802.

References

[1] J.A. Aledo, S. Martinez, F.L. Pelayo and J.C. Valverde, *Parallel Dynamical Systems on Maxterms and Minterms Boolean Functions*, Math. Comput. Model. 35 (2012) 666–671.

[2] J.A. Aledo, S. Martinez and J.C. Valverde, *Parallel dynamical systems over directed dependency graphs*, under review.

[3] A. Bagrodia, K.M. Chandy, W.T. Liao, A unifying framework for distributed simulation, ACM Trans. Modeling Computer Simulation 1 (4) (1991) 348-385.

[4] C.L. Barret, W.Y.C. Chen and M.J. Zheng, *Discrete dynamical systems on graphs and Boolean functions*, Math. Comput. Simul. 66 (2004), pp. 487–497.

[5] C.L. Barret and C.M. Reidys, *Elements of a theory of computer simulation I*, Appl. Math. Comput. 98 (1999), pp. 241–259.

[6] C.L. Barret, H.S. Mortveit and C.M. Reidys, *Elements of a theory of computer simulation II*, Appl. Math. Comput. 107 (2002), pp. 121–136.

[7] C.L. Barret, H.S. Mortveit and C.M. Reidys, *Elements of a theory of computer simulation III*, Appl. Math. Comput. 122 (2002), pp. 325-340.

[8] D. Jefferson. *Virtual time*. ACM Trans. Programming Languages Systems, 7(3) (1985) 404- 425.

[9] R. Kumar, V. Garg, Modeling and Control of Logical Discrete Event Systems, Kluwer Academic Publishers, Dordrecht, 1995.

[10] Y.A. Kuznetsov, *Elements of Applied Bifurcation Theory*, Springer, New York, 2004.

[11] H.S. Mortveit and C.M. Reidys, *Discrete, sequential dynamical systems*, Discrete Math. 226 (2002), pp. 281-295.

[12] H.S. Mortveit and C.M. Reidys, An Introduction to Sequential Dynamical Systems, Springer, New York, 2007.

[13] S. Wiggins, *Introduction to Applied Nonlinear Systems and Chaos*, Springer, New York, 1990.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Modeling power performance for master–slave applications

Francisco Almeida[1], Vicente Blanco[1] and Javier Ruiz[1]

[1] *Dept. Estadística, I.O. y Computación, Universidad de La Laguna*

emails: falmeida@ull.es, vblanco@ull.es, javer.ruiz@gmail.com

Abstract

With energy costs now accounting for nearly 30 percent of a data centre's operating expenses, power consumption has become an important issue when designing and executing a parallel algorithm. This paper analyzes the power consumption of MPI aplications following the master–slave paradigm. The analytical model is derived for this paradigm and is validated over a master–slave matrix–multiplication. This analytical model is parameterized through architectural and algorithmic parameters, and it is capable of predicting the power consumption for a given instance of the problem over a given architecture. We use an external, metered, power distribution unit has been that allows to easily measure the power consumption of computing nodes without the needings of dedicated hardware. .

Key words: Energy–efficient algorithms; Power performance

1 Introduction

In recent years power consumption has become a major concern in the operation of large-scale datacenters and High Performance Computing facilities. As an example, the 10 most powerful supercomputers on the TOP500 List(www.top500.org) each require up to 12 MW of power for the entire system (computing and cooling facilities). As a result, power-aware computing has been recognized as one of critical research issues in HPC systems.

New processor architectures allow power management through a mechanism called *Dynamic Voltage and Frequency Scaling* (DVFS) where applications or operating system has the ability to select the frequency and voltage on the fly. Depending on required resources for the application you can select a combination of voltage and frequency, denoted as processors state or *p-state*. Different *p-states* deal to different power consumption, allowing power management by applications [6].

```
for (proc = 1; proc <= p; proc++)
        send(proc, work[proc]);
for (proc = 1; proc <= p; proc++)
        receive(proc, result[proc]);
}

/* Slave Processor */
receive(master, work);
compute(work, result);
send(master, result);
```

Listing 1: Master–slave paradigm

Power measurement can give us information about the energy consumed by systems, but is not sufficient for challenges such as attributing power consumption to virtual machines, predicting how power consumption scales with the number of nodes, and predicting how changes in utilization affect power consumption [3]. These tasks require accurate models of the relationship between resource usage and power consumption. Models based on architectural parameters has been widely developed [4, 5].

We have developed an instrumentation framework based on metered PDUs (Power Distribution Units), allowing to measure the power consumption of HPC nodes while applications are executed. In a similar way we model application performance (execution time) [7, 8], we propose to model power consumption using architectural parameters (number of cores, cache misses, memory access, network latency and bandwidth) and algorithmic parameters (problem size). The analytical power models obtained can be used by schedulers to save energy when applications are executed on HPC systems.

The rest of this paper is organized as follows: Section 2 introduces master–slave paradigm and Section 3 describes our experimental setup and measurement system. In Section 4 we introduce an analytical power model for the proposed application. We show the obtained models on Section 5. Finally, Section 6 summarizes the paper with some conclusions and future work.

2 The Master Slave Paradigm

Under the Master–slave paradigm it is assumed that the work W, of size m, can be divided into a set p of independent tasks $work_1, ..., work_p$, of arbitrary sizes $m_1, ..., m_p$, $\sum_{i=1}^{p} m_i = m$, that can be processed in parallel by the slave processors $1, ..., p$. We abstract the master–slave paradigm by the code in Listing 1

The total amount of work W is assumed to be initially located at the master processor, processor p_0. The master, according to an assignment policy, divides W into p tasks and

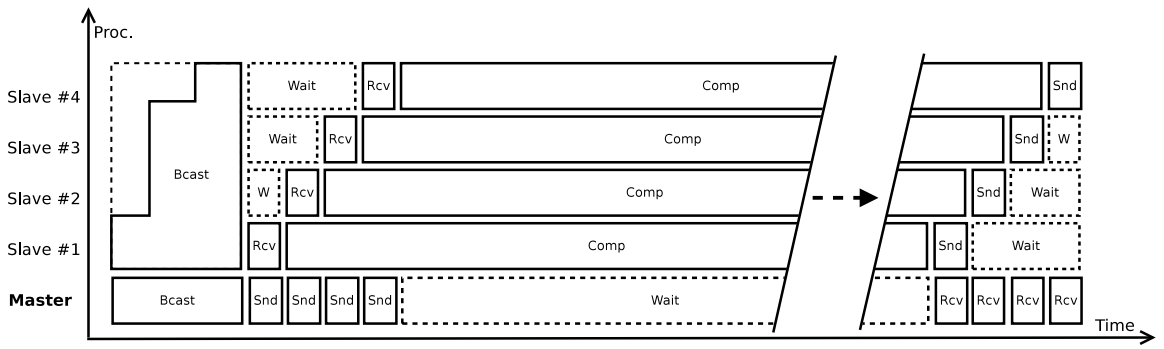


Figure 1: Master-slave diagram of time for matrix multiplication

distributes $work_i$ to processor i. After the distribution phase, the master processor collects the results from the processors. Processor i computes $work_i$ and returns the solution to the master. The master processor and the p slaves are connected through a network, typically a bus, that can be accessed only in exclusive mode. At a given time-step, at most one processor can communicate with the master, either to receive data from the master or to send solutions back to it. The results are returned back from the slaves as a synchronous round robin.

The transmission time from the master processor to the slave processor i will be denoted by $R_i = (\beta_i + \tau_i w_i)$ where w_i stands for the size in bytes associated to $work_i$ and β_i and τ_i represent the latency and per-word transfer time respectively. This communication cost involves the time for the master to send data to a slave and for it to be received. The latency and transfer time may be different on every combination master - slave and they must be calculated separately.

2.1 Master–slave matrix–multiplication

We introduce the master–slave matrix–multiplication application as a case study. First of all, we describe de algorithm and the corresponding performance analytical model.

For the master–slave model that we are using in this paper (Fig. 1), the time gets approximately reduced by a factor of $1/p$, but a small overload is introduced in the process of broadcasting the matrices and to gather the results. The complete process includes four principal segments: (1) Broadcasting of B matrix, (2) Sending of A matrix by blocks, (3) Waiting for slaves to finish computing and (4) Receiving of matrix C. The total execution time is:

$$T_{par} = T_{bcast} + p \cdot T_{snd} + T_{comp} + T_{rcv} \tag{1}$$

 ISBN:978-84-615-5392-1

In order to estimate the time used in the broadcast segment, it is necessary to have determined two parameters related to the network: β_{bcast} and τ_{bcast}, which represent the latency and the bandwidth of the network while performing broadcasting operations. These parameters can be obtained performing a test in C with MPI in the system used [1]. The number of elements to be sent is the size of the matrix squared.

$$T_{bcast} = \beta_{bcast} \cdot \lg(p) + \tau_{bcast} \cdot \lg(p) \cdot n^2 \tag{2}$$

The send an receive operations performed in the master–slave paradigm can also be modeled in a similar way. Again, two parameters are needed, $\beta_{snd/rcv}$ and $\tau_{snd/rcv}$, obtained with a test referenced in [12]. The first parameter is the latency, but it has a negligible effect, because the sub-blocks to be sent are very few. The τ parameter is the bandwidth in node-to-node communication.

$$T_{snd/rcv} = \beta_{snd/rcv} + \tau_{snd/rcv} \cdot \frac{n^2}{p} \tag{3}$$

T_{comp} can be modeled by well-known complexity formula $O(N^3)$ for a sequential matrix–multiplication on each processing element. In Section **??** we give more detail about the model for the computation load and the corresponding energy consumed.

This analytical model to predict the running time of master-slave applications has been widely validated. Our goal now is to obtain a similar analytical model for the power consumption of the master–slave matrix–multiplication algorithm. Section 4 describes de model obtained for this implementation on comodity cluster instrumented with a metered PDU.

3 Experimental Setup

The measuring system used is one manufactured by the company Schleifenbauer [10]. The equipment consists of a power distribution unit (PDU) with one input and nine outlets, and a master device called Gateway. The Gateway serves as a hub for the available PDUs. Each PDU is connected to the gateway with standard UTP cable and a RS-485 transport layer at 100 Kbit/s. The Gateway is initially configured through the USB or RS-232 port, assigning it a management IP address. As the Gateway has an Ethernet interface that connects to the appropriate network, this address allows us to access the Gateway through any browser to change its settings, to consult the measurements of any PDU connected and to turn on and off the outlets for each PDU. Gathering of data coming from each PDU is provided through various suitable interfaces (Perl API, HTTP, MySQL, etc.) The setup for the measurements includes the following steps:

1. Setting the IP Gateway with one accessible from our network.

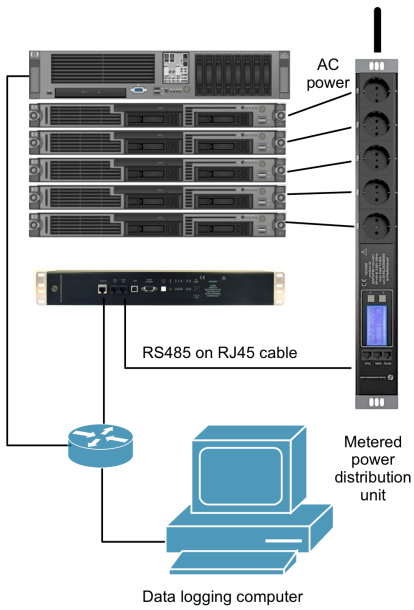


Figure 2: PDU, Gateway and computing nodes connections

2. Mounting the Gateway and the PDU in the data center.
3. Wire UTP cable from the PDU to the Gateway, and back to the PDU, to create a ring in order to provide for additional redundancy.
4. Connect the Gateway to the network.
5. Connect the AC plugs of the nodes of the system under test to the outlets of the PDU. For convenience is advisable to choose consecutive ones.

The equipment installed (Fig. 2) allows the display of consumption data from the PDU by directing a browser to the IP address of the Gateway. The tab "Measurements" allows us to query the data measured in the strip.

In Fig. 3 that shows the web interface, actual current measurements (RMS) for nodes connected to outlets 5 to 9 can be seen. The values correspond to the actual nodes of the cluster which were used in experimentation. Note that the current at idle is not the same for all nodes, although they are identical models. Also it is worth mentioning that the apparent power is the value of current times voltage, expressed in $V \cdot A$. The power factor value tell us how much of the apparent energy is converted into usable energy [2]. Values shown are close to 100%, indicating a good design of the power supplies in the compute nodes.

output	name	energy total (kWh)	energy subtotal (kWh)	power factor (%)	actual current (A)	peak current (A)	actual voltage (V)
			(...)				
4	node19	85	85	97.4	0.99	1.25	219.5
5	node20	659	67	96.3	0.81	1.07	219.6
6	node21	716	73	96.4	0.89	1.13	219.5
7	node22	507	66	96.9	0.79	1.04	220.9
8	node23	662	67	97.0	0.80	1.06	220.8
9	node24	815	72	96.7	0.87	1.13	220.8

Figure 3: Web interface to the Scheleifenbauer Gateway

$$RealPower = V_{RMS} \cdot I_{RMS} \cdot PowerFactor/100$$

The cluster *Tegasaste*, used in the experiment, has 24 nodes, five of which connected to the PDU. The front end node is a 4 x Intel(R) Xeon(TM) @ 3.000 GHz with 1 GB RAM. The computation nodes 20 to 24 have 2 x Intel(R) Xeon(TM) @ 3.200 GHz, 1 GB RAM each. The operating system was Linux 2.6.16.16-papi3.2.1 with gcc version 4.3.2 (Debian 4.3.2-1.1) and MPI 1.2.7.For communication between nodes, an Infiniband(R) switch was used.

3.1 Measurement

To perform the measurements corresponding to a particular experiment, we use an auxiliary computer, connected to the same network as the Gateway. This auxiliary computer allows the collection of data coming from the PDU, while the algorithm of study is being executed in the parallel cluster, and the logging all the events of interest. The company Schleifenbauer provides an API in Perl to access the measurements. The Gateway has a register type of reading, as current, power factor, voltage, etc.. The user only has to choose the corresponding mnemonic and call the ReadRegisters function. It is important to start this Perl script several seconds before the launching of the execution of the algorithm of study, to take account of the initial values of current in the different nodes. Usually a 10 seconds delay was used. Finally another script executes both the Perl script for the PDU and the parallel algorithm in the CPUs.

The execution of this script generates a set of pairs of files, each pair consisting in one file with PDU data and one with CPU data. As already noted, the idle values of current for each node vary, and so the posterior current measurements will be affected by these idle values. This situation is alleviated by offsetting all the measurements, so they are always zero based. The average idle value of current can be added later on to get the real power

consumption. Also, due to the design of the serial protocol connecting the PDU to the Gateway, a measurement takes at least 215 ms. A delay of approximately one second was observed between a measurement and the event that triggers it.
The CPU data file consists in time-stamped events, that can be crossed with the PDU data file to extract the values of current corresponding to each segment of execution. The integration of these current values and the corresponding timestamps generates data of measured power consumption of the parallel algorithm.

4 Power performance model

In this section we propose an analytical power performance model similar to the execution time model introduced in Section 2 for the execution time of a master–slave application. We will use the well–know matrix–multiplication code to ilustrate the proposed model. First, we study the power–aware behavior of three different sequential matrix–multiplication implementation. Following, we will study the communication part of the master–slave application in terms on power consumption.

4.1 Master-slave matrix–multiplication parallel implementation

To implement the transposed variant of the matrix multiplication we used a master-slave schema, with one master and p slaves, where the master process does not compute. Although each node had two processors, only one was used. The main script of experimentation contained sizes from 1024 to 6400, and every multiplication was performed 10 times. To minimize effects related with the order of execution, this was randomized. Figure 4 shows the overall current profile for a 6400 by 6400 multiplication. The study of the resulting 260 pairs of files with PDU and CPU data allowed us to derive some facts.

- Current kept constant during the broadcast, send and receive segments of code, but its level was higher than that of idle state.
- The current of the master process kept constant at roughly the same level while the slaves performed the computation. We did not find any difference between this level and the one related to the communications segments.
- During the computation segments the current reached its maximum, and again kept constant throughout all the execution.

With all this data available we could estimate the total power consumed by the execution. Taking into account that power consumption is closely related to time, it makes sense to begin with the time model for the particular problem we are studying. From the fragment of code corresponding to the core of the algorithm executed, it could be derived that the time depends on the number of floating points operations $2n^3$ but also from the $2n^3$ accesses

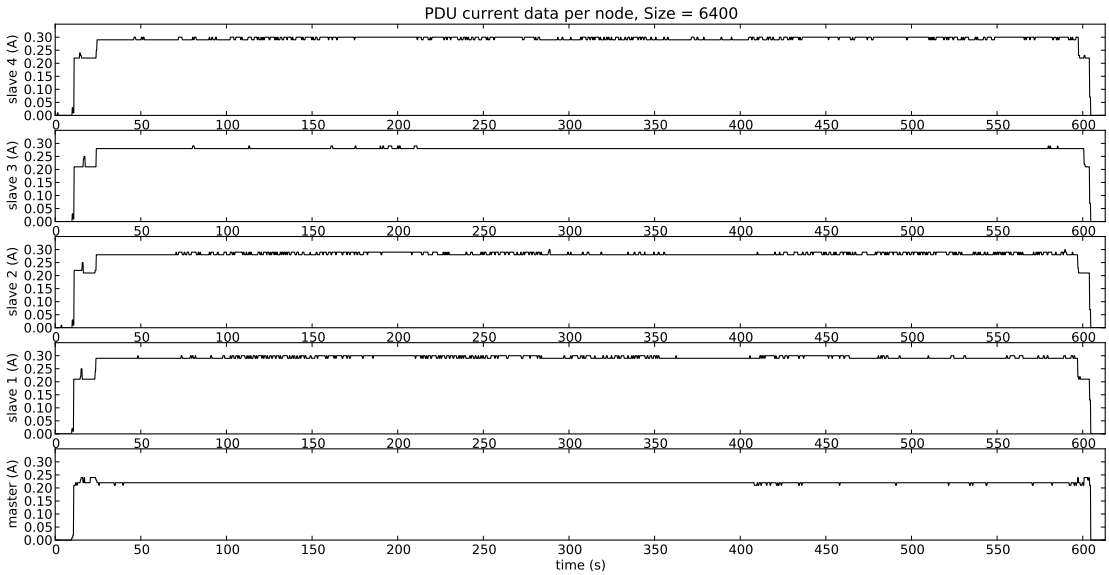


Figure 4: PDU overall current data, matrix–multiplication on 4 processors

to memory (reads) and the n^2 writes to memory.

```
for (i = 0; i < size; ++i){
 for (j = 0; j < size; ++j){
  sum = 0.0;
  for ( k = 0; k < size; ++k){
   sum += A[i * size + k] * B[j * size + k];
  }
  C[i * size + j] = sum;
 }
}
```

To check if the writes to memory affected to the estimated power, a regression with positive coefficients was performed using R [11] and the package *nnls* [9]. It turned out that the write operation did not contribute to the power in this segment. The non zero coefficient obtained, $FlopMem$, includes the contribution of the two floating point operations and the two reads from memory. Thus the estimated time for the sequential case is simply:

$$T_{comp} = 2n^3 \cdot FlopMem \tag{4}$$

 ISBN:978-84-615-5392-1

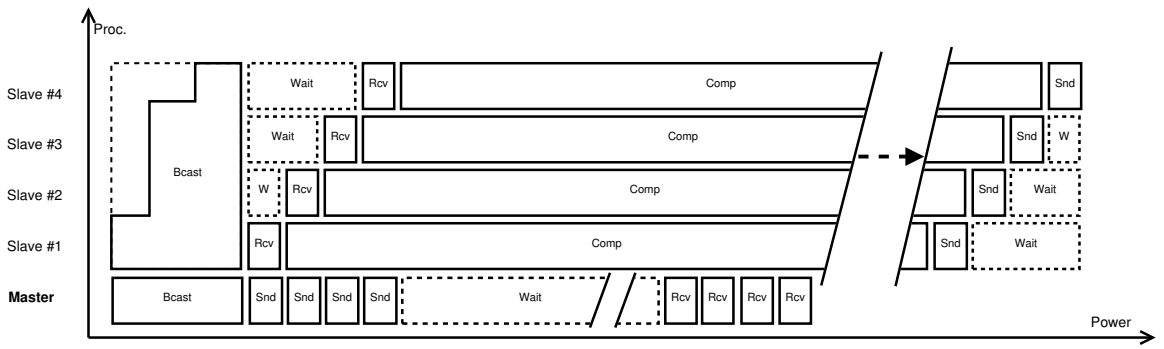


Figure 5: Master-slave diagram of power consumption for matrix multiplication

The total power of the sequential code is obtained using the average current measured during the execution.

$$PW_{comp} = T_{comp} \cdot Curr_{comp} \tag{5}$$

4.2 Model for power consumption

Following the performance model introduced in Section 2, we propose a model power compsumption based on equation 1. Each term of this equation contributes with a fraction of the total power consumption, depending on the average current measured in each segment and its duration. Figure 5 depicts again the master-slave schema, this time showing the power consumption. Note the contribution of the master node, lower than the computing nodes (has a lower *p-state*).

At *bcast* operation on eq. 1, it is necessary an average of the current used during the segment, a value that was obtained with the batch of executions, and that it is the same across the processors involved in the operation. The power consumption associated to the broadcast can be computed using:

$$PW_{bcast} = (p+1) \cdot T_{bcast} \cdot Curr_{bcast} \tag{6}$$

where T_{bcast} is modeled by eq. 2.

The power estimation for the send by blocks segment is similar to the broadcast one. With the time needed to send a sub-block of the matrix A, the estimated power for the whole segment can be calculated. A summation expression is used for the sake of completeness and accuracy, although the total contribution of this part is very small.

$$PW_{snd} = (\sum_{i=1}^{p} i + p) \cdot T_{snd} \cdot Curr_{snd/rcv} \tag{7}$$

where T_{snd} is modeled by eq. 3.

The expression for the computation segment includes the p computing nodes and the contribution of the master process while waiting for the slaves to end the computation work:

$$PW_{comp} = p \cdot T_{comp} \cdot Curr_{comp} + PW_{MasterWait} \tag{8}$$

$$PW_{MasterWait} = [T_{comp} - (p-1) \cdot T_{snd}] \cdot Curr_{wait} \tag{9}$$

The last segment of this master-slave consists in the master receiving the results from the slaves. The expression is identical to the send segment one:

$$PW_{rcv} = (\sum_{i=1}^{p} i + p) \cdot T_{rcv} \cdot Curr_{snd/rcv} \tag{10}$$

Finally the complete expression for the power of the parallel execution is:

$$PW_{total} = PW_{bcast} + PW_{send} + PW_{comp} + PW_{recv} \tag{11}$$

Our analysis of the power consumption model must end with a note on units. We have been using $time \cdot Curr$ as a proxy for energy, but this has to be corrected to be completely right. For convenience we have omitted to multiply the current by the voltage, to get $V \cdot A$. Finally we get to the value of $ApparentPower$ in $W \cdot h$ with the appropriate conversion. We have considered the voltage constant during the executions, averaging the voltage of the outlets of the PDU.

5 Model Validation

The model we have developed depends on architectural and algorithmic parameters that can be measured with the appropriate tests. We show between brackets the values measured in our configuration.

- β_{bcast} [5e-06 s] and τ_{bcast} [4.00641e-09 s], that characterizes the time that the network takes to broadcast a large message.
- $\beta_{snd/rcv}$ [0.0009130886 s] and $\tau_{snd/rcv}$ [1.879013e-08 s], that characterizes the time that takes a node to send a large message to another node.
- $FlopMem$ (1.879013e-08 s] essentially characterizes the computing power of every node, and can be obtained with a simple timed for-loop.
- $Curr_{bcast} = Curr_{snd/rcv} = Curr_{comm} = Curr_{wait}$ [0.2248810 A above idle current] that is the current level at which the communication operations are performed.
- $Curr_{Cmp}$ [0.2921429 A above idle current] characterizes the maximum current per node when one processor is being used at full computation rate.

 ISBN:978-84-615-5392-1

	Measured				Modeled				Error (%)			
Size	$p=4$	$p=5$	$p=6$	$p=7$	$p=4$	$p=5$	$p=6$	$p=7$	$p=4$	$p=5$	$p=6$	$p=7$
2000	25.73	25.452	25.338	24.265	26.14	25.29	24.73	24.33	1.59	-0.62	-2.39	0.27
3000	89.33	85.756	85.833	82.651	88.22	85.37	83.47	82.12	-1.25	-0.45	-2.75	-0.65
4000	212.73	201.985	200.7	196.969	209.11	202.36	197.86	194.64	-1.70	0.18	-1.42	-1.18
5000	412.46	393.303	387.977	383.807	408.41	395.23	386.44	380.16	-0.98	0.49	-0.40	-0.95
6000	714.54	670.201	669.880	656.017	705.73	682.96	667.77	656.92	-1.23	1.90	-0.31	0.14

Table 1: Power consumption for matrix–multiplication with 4 to 7 slaves, in $A \cdot s$

Finally, table 1 shows values for measured and modeled matrix multiplications with 4 to 7 slaves. Column labeled *E*rror shows the relative error made by the prediction. The very low error observed, where highest error made is -1.7, allows to conclude that our analytical model has been validated and predicts the power consuption.

6 Conclusions

We have analyzed the power consumption of the master–slave paradigm over an MPI application. Similar to the performance model in terms of execution time that can be obtained for these kind of implementation, it is possible to obtain an analytical expresion for the energy consumed by these codes while executed on HPC systems. We have implemented a power metered framework based on standard metered PDUs. The experimental infraestructure allows us to monitorize and model any application that can be executed in our cluster. As a case study, we model the matrix–multiplication algorithm by an analytical formula. With this expression we can predict the power consumed by the application on our cluster knowing the problem size and number, the number of slaves used and a set of parameters, architectural–dependent.

Acknowledgements

This work was supported by the Spanish Ministry of Education and Science through TIN2011-24598 and TIN2008-06570-C04-03 projects and through the FPU program. It also was supported by the Canarian Agency for Research, Innovation and Information Society under contract ProID20100222 and has been developed in the framework of the European network COST-ICT-0805 and the Spanish network CAPAP-H2.

References

[1] C.-Y. Chou, H.-Y. Chang, S.-T. Wang, and S.-C. Tcheng. Modeling message-passing overhead on nchc formosa pc cluster. In Y.-C. Chung and J. E. Moreira, editors, *GPC*,

volume 3947 of *Lecture Notes in Computer Science*, pages 299–307. Springer, 2006.

[2] S. P. E. Corporation. Spec power and performance, benchmarl methodology v2.1, Aug. 2011.

[3] J. Davis, S. Rivoire, M. Goldszmidt, and E. Ardestani. Accounting for variability in large-scale cluster power models. In *The Second Exascale Evaluation and Research Techniques Workshop. Held in conjunction with ASPLOS 2011*, 2011.

[4] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. Rountree, and M. E. Femal. Analyzing the energy-time trade-off in high-performance computing applications. *IEEE Trans. Parallel Distrib. Syst.*, 18(6):835–848, 2007.

[5] L. Keys, S. Rivoire, and J. D. Davis. The search for energy-efficient building blocks for the data center. In A. L. Varbanescu, A. M. Molnos, and R. van Nieuwpoort, editors, *ISCA Workshops*, volume 6161 of *Lecture Notes in Computer Science*, pages 172–182. Springer, 2010.

[6] M. Y. Lim, V. W. Freeh, and D. K. Lowenthal. Adaptive, transparent cpu scaling algorithms leveraging inter-node mpi communication regions. *Parallel Computing*, 37(10-11):667–683, 2011.

[7] D. Martínez, J. Albín, T. Pena, J. Cabaleiro, F. Rivera, and V. Blanco. Using accurate AIC–based performance models to improve the scheduling of parallel applications. *Journal of SuperComputing*, 58(3):332–340, 2011. In press: Online `http://dx.doi.org/doi:10.1007/s11227-011-0589-1`.

[8] D. Martinez, J. Cabaleiro, T. Pena, F. Rivera, and V. Blanco. Performance modeling of mpi applications using model selection techniques. In M. Danelutto, J. Bourgeais, and T. Gross, editors, *18th Euromicro Conference on Parallel, Distributed and Network–based Processing. PDP2010*, pages 95–102, Pisa, Italy, Feb. 2010. IEEE Computer Society.

[9] K. M. Mullen and I. H. M. van Stokkum. nnls: The lawson-hanson algorithm for non-negative least squares (nnls), Apr. 2010.

[10] A. Schuermans. Schleifenbauer products bv, Mar. 2012.

[11] Various. R, language and environment for statistical computing and graphics, Mar. 2012.

[12] Z. Xu and K. Hwang. Modeling communication overhead: Mpi and mpl performance on the ibm sp2. *Parallel Distributed Technology: Systems Applications, IEEE*, 4(1):9 –24, spring 1996.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

The solution of Block-Toeplitz linear systems of equations in multicore computers*

Pedro Alonso[1], Daniel Argüelles[2], José Ranilla[2] and Antonio M. Vidal[1]

[1] *Departamento de Sistemas Informáticos y Computación,*
Universitat Politècnica de València, Spain

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

emails: palonso@dsic.upv.es, daniel.arguelles.martino@gmail.com,
ranilla@uniovi.es, avidal@dsic.upv.es

Abstract

There exist algorithms called "fast" which exploit the special structure of Toeplitz matrices so that, e.g., allow to solve a linear system of equations in $O(n^2)$ flops (instead of $O(n^3)$ flops required by classical algorithms). Due to the constantly increasing core count in current computers, it is necessary to parallelize such algorithms in order to get the most of the underlying hardware. In particular, we propose in this paper an efficient implementation of the Generalized Schur Algorithm, a very known algorithm for the solution of Toeplitz systems, to work on a block–Toeplitz matrix. Our algorithm is based on matrix–matrix multiplications with the aim at leveraging *threaded* routines that implement this operation.

Key words: Block-Toeplitz, linear systems, Generalized Schur Algorithm, multicore-computers
MSC 2000: 15B05, 65F05, 68W04, 68W10

1 Introduction

Block-Toeplitz matrices appear in many fields of engineering, e.g., in time series analysis in signal or image processing, and system identification, many times through the solution of a linear system of equations. This paper presents an implementation for the solution of

$$Tx = b\,, \tag{1}$$

*PROMETEO/2009/013, Generalitat Valenciana. Projects TEC2009-13741, TIN2010-14971 and TIN2011-15734-E (CAPAP-H4) of the Ministerio de Ciencia e Innovación. Spain

where the system matrix $T \in \mathbb{R}^{n\times n}$ is a symmetric block–Toeplitz matrix of the form

$$T = \begin{pmatrix} A_0 & A_1^T & A_2^T & \\ A_1 & A_0 & A_1^T & \ddots \\ A_2 & A_1 & A_0 & \ddots \\ & \ddots & \ddots & \ddots \end{pmatrix}, \tag{2}$$

being each block $A_i \in \mathbb{R}^{\nu\times\nu}$, $i = 0, \ldots, N-1$, full dense, and being $b, x \in \mathbb{R}^n$ the right-hand-side and the solution vectors, respectively. In this paper, matrix T (2) is supposed to be positive definite. For simplicity in the exposition we consider hereafter n as an integer multiple of the block size ν, although this restriction can be easily relaxed in the actual algorithms.

We address problem (1) by obtainig the Cholesky factor of T so $T = C^T C$, being C upper triangular, through a well-known algorithm called the Generalized Schur Algorithm (GSA) [2]. The GSA has intrinsic parallelism due to the elements of a given row of C can be computed concurrently. This fact has already been successfully exploited, e.g., in [1], where it was proposed an implementation of the GSA for shared memory. In [3], a version of the GSA based on level 3 operations was proposed. However, the algorithm in [3] is based on complicated operations that are also difficult to implement in parallel.

We propose here a more simple version of the GSA for block-Toeplitz matrices than [3]. The algorithm is also based on level 3 operations, in particular, on matrix-matrix products. Furthermore, to the extent that there are available *threaded* routines which implement a matrix product, our algorithm may draw on the thread level parallelism of multiple cores.

Next section describes our implementation of the GSA algorithm for block-Toeplitz matrices. Section 3 gives some data about results it can be obtained with our implementation. At the end we offer some concluding remarks.

2 The Generalized Schur Algorithm for Block–Toeplitz matrices

For the solution of system (1) we propose to perform the Cholesky factorization $T = C^T C$, where $C \in \mathbb{R}^{n\times n}$ is upper triangular. We use the GSA described in Algorithm 1 to compute this factor.

Algorithm 1 receives matrix G, called *generator*, which has the form

$$G = \begin{pmatrix} U & \bar{A}_1^T & \bar{A}_2^T & \ldots & \bar{A}_{N-1} \\ & A_1^T & A_2^T & \ldots & A_{N-1} \end{pmatrix}, \tag{3}$$

where U is the upper triangular Cholesky factor of A_0 such that $A_0 = U^T U$, and $\bar{A}_i^T = U^{-T} A_i^T$, for $i = 1, \ldots, N-1$.

Algorithm 1 Cholesky decomposition of $T = C^T C$.

Require: Generator G (3), **return** Cholesky factor C.

1: $U = \texttt{chol}(A_0)$ ▷ $(U^T U = A_0)$
2: Solve $U^T G_2 = \begin{pmatrix} A_1^T & A_2^T & \dots & A_{N-1} \end{pmatrix}$
3: $C = \begin{pmatrix} U & G_2 \end{pmatrix}$ ▷ First block-column of C
4: $m = n - \nu$
5: $G_1 = C(:, 1:m)$
6: **for** $i = 1 \to N-1$ **do**
7: $\quad \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \leftarrow \texttt{normalize}\left(\begin{pmatrix} G_1 \\ G_2 \end{pmatrix}\right)$ ▷ Normalization of the generator
8: $\quad C \leftarrow \begin{pmatrix} C & \\ \begin{pmatrix} 0_{\nu \times i \cdot \nu} & G_1 \end{pmatrix} & \end{pmatrix}$ ▷ Adding a new block-row into C
9: $\quad G_1 \leftarrow G_1(:, 1 : m - \nu)$ ▷ Updating G_1 (Shift)
10: $\quad G_2 \leftarrow G_2(:, \nu + 1 : m)$ ▷ Updating G_2 (Shift)
11: $\quad m \leftarrow m - \nu$
12: **end for**

Step shown in line 7, which consists of a call to function `normalize`, performs a *normalization* of the generator G that overwrites it with its own *proper form*. Let the next partition of the generator

$$G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}, \tag{4}$$

where $G_{11}, G_{21} \in \mathbb{R}^{\nu \times \nu}$ and $G_{12}, G_{22} \in \mathbb{R}^{\nu \times m}$, then we say that G is in *proper form* if G_{21} is zero.

Algorithm 2 Routine `normalize`. Normalization of generator G.

Require: $G_{11}, G_{21}, G_{12}, G_{22}$ of partition (4), **return** $G_{11}, G_{21}, G_{12}, G_{22}$ (G_{21} is zero).

1: $[Q, R] = \texttt{qr}(G_{21})$, $G_{21} \leftarrow R$ ▷ $G_{21} = QR$
2: $H = \begin{pmatrix} I & \\ & Q \end{pmatrix}$
3: **for** $j = 1 \to \nu$ **do**
4: $\quad$ **for** $i = 1 \to j$ **do**
5: $\quad\quad \texttt{hyper}(G_{11}(j, j:\nu), G_{21}(i, j:\nu), H(:, j), H(:, i+\nu))$
6: $\quad$ **end for**
7: **end for**
8: $\begin{pmatrix} G_{12} \\ G_{22} \end{pmatrix} \leftarrow H^T \begin{pmatrix} G_{12} \\ G_{22} \end{pmatrix}$

 ISBN:978-84-615-5392-1

At any given iteration of the algorithm, the first ν columns of the generator has the following form

$$\begin{pmatrix} G_{11} \\ G_{21} \end{pmatrix} = \begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix},$$

where x just denote non zero entries. Step 7 of Algorithm 1 consists of zeroing the $\nu \times \nu$ entries of the down square (G_{21}) by calling function `normalize`, (*normalization*). This step can be carried out in different ways. In this paper, we perform a QR decomposition of factor G_{21} inplace, so that G_{21} is replaced by the upper triangular factor R of its QR decomposition (step 1 of Algorithm 2). Below, a succession of hyperbolic transformations allows to nullify the remaining entries of G_{21}, i.e., the entries on the upper triangular part (steps 3 to 7). Algorithm 3 calculates an hyperbolic transformation an applies it to two pair of arrays.

The key of Algorithm 2 is that it works on factors G_{11} and G_{21} to compute the transformations needed to set the generator in proper form. Instead of applying these transformations, a matrix H is built by accumulating those transformations. Consequently, step 8 of Algorithm 2, which is the largest one, has become a matrix-matrix product.

Algorithm 3 Routine `hyper`. Computation and updating of arrays with a hyperbolic rotation.

Require: Arrays u, v, x, y, **return** u, v, x, y.

1: $\alpha = u_1$
2: $\beta = v_1$
3: **if** $\alpha^2 \geq \beta^2$ **then**
4: $\quad \gamma = \sqrt{\alpha^2 - \beta^2}$
5: $\quad \alpha \leftarrow \alpha/\gamma$
6: $\quad \beta \leftarrow \beta/\gamma$
7: $\quad w = (\alpha u - \beta v)$
8: $\quad v \leftarrow (\alpha v - \beta u)$
9: $\quad u = w$
10: $\quad w = (\alpha x - \beta y)$
11: $\quad y \leftarrow (\alpha y - \beta x)$
12: $\quad x = w$
13: **end if**

 ISBN:978-84-615-5392-1

Table 1: Execution time (sec.) with different number of cores and problems sizes (n).

ν	#cores	1536	3584	5632	7680
64	1	0.21	1.10	2.67	5.15
	2	0.19	0.92	2.28	4.27
	4	0.17	0.84	2.02	3.76
	8	0.17	0.81	1.94	3.61
256	1	0.85	3.30	7.47	13.1
	2	0.76	2.68	5.79	9.94
	4	0.72	2.38	5.00	8.43
	8	0.71	2.25	4.69	7.74

For the solution of the two triangular linear systems conducing to the solution of (1) we also implemented a routine that works in parallel. We will denote as GSA to the whole algorithm that solves the linear system (1) by performing the Cholesky factorization of T.

3 Results

The experimental results shown in this section were obtained on a board with two Intel® Xeon® E5420 processors with 4 cores and 6 Mb. of cache memory each, thus resulting in a total of 8 cores for the tests.

In general, "fast" algorithms lack of a large throughput per data. Furthermore, Schur–type algorithms are rather irregular, i.e., the concurrency drops in the last steps of the triangularization process due to the diminishing number of blocks that can be computed in parallel. Nevertheless, as Table 1 shows, the algorithm manages to reduce time when the number of cores increases. We used compilers in Intel® Composer XE 12.1 distribution and routine `dgemm` in MKL 10.3 for the matrix product. This matrix multiplication is threaded so it leverages the available number of cores. The reduction in time achieved with the algorithm depends on the block size ν. As long as the block size is large more efficiency is obtained with the matrix-matrix multiplication with the count of cores. This is what it can be seen if we compare, e.g. for the largest problem size, the time between using 4 or 8 cores.

Although the heaviest step of the algorithm is the triangularization, we have also obtained a certain level of parallelism in solving the two triangular systems.

4 Conclusions

A parallel implementation of a "fast" algorithm has been presented in this paper. We rearranged operations so they could be cast in term of matrix-matrix multiplications. The more efficient the matrix product operation the faster our algorithm will be. The algorithm thus benefits not only of a level 3 operation but also of its multicore implementation.

Under certain conditions (problem size, block size, hardware, ...) our parallel implementation of the GSA well exploits a multicore by always producing the result with a given number of cores in less time than using a fewer number of cores.

References

[1] Pedro Alonso, José M. Badía, and Antonio M. Vidal. An efficient parallel algorithm to solve block–Toeplitz systems. *The Journal of Supercomputing*, 32:251–278, 2005.

[2] J. Chun, T. Kailath, and H. Lev-Ari. Fast parallel algorithms for QR and triangular factorization. *SIAM Journal on Scientific and Statistical Computing*, 8(6):899–913, November 1987.

[3] K. Gallivan, S. Thirumalai, and P. Van Dooren. On solving block toeplitz systems using a block schur algorithm. In Jagdish Chandra, editor, *Proceedings of the 23rd International Conference on Parallel Processing. Volume 3: Algorithms and Applications*, pages 274–281, Boca Raton, FL, USA, August 1994. CRC Press.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

CMB Maps: a Bayesian technique

Pedro Alonso[1], Francisco Argüeso[1], Raquel Cortina[2], José Ranilla[2] and Antonio M. Vidal[3]

[1] *Department of Mathematics, Universidad de Oviedo, Spain*

[2] *Department of Computer Science, Universidad de Oviedo, Spain*

[3] *Department of Computer Systems and Computation, Universidad Politécnica de Valencia, Spain*

emails: palonso@uniovi.es, argueso@uniovi.es, raquel@uniovi.es, ranilla@uniovi.es, avidal@dsic.upv.es

Abstract

In this work we present a new efficient method to detect point sources in Cosmic Microwave Background maps and estimate their fluxes. The method uses previous information about the statistical properties of the sources, so that this knowledge can be incorporated in a Bayesian scheme. The experiments show that the method detects more sources than the results obtained using other strategies. Besides, the technique allows us to fix the number of detected sources in a non-arbitrary way.

Key words: Efficiency, Cosmic Microwave Background, Bayesian
MSC 2000: 65F05, 65Y20, 68W10

1 Introduction

The Cosmic Microwave Background (CMB) is a diffuse radiation which started to propagate freely in the early universe, about 400,000 years after the Big Bang. The CMB is thus a fossil radiation which carries very important information about the fundamental properties of the universe and, in particular, about the chemistry of the early universe. Since its discovery in 1964 ([8]), the CMB has been detected and surveyed by instruments aboard balloons and satellites, such as the NASA satellites COBE in 1992 ([9]) and WMAP in 2003 ([10]).

The ESA satellite Planck was launched in 2009 and has been measuring the CMB fluctuations with unprecedented accuracy, resolution and frequency coverage. The first data obtained by Planck were published in 2011 ([1]). In order to extract all the relevant information from the CMB data, one must remove the contamination produced by extragalactic sources, e.g. galaxies that also emit microwave radiation which has a non-cosmological origin. Several techniques: matched filters, wavelets, etc. have been proposed in the literature (see [6] and [7] for detailed reviews) to deal with the detection problem.

The goal of this paper is to put forward a new efficient method to detect point sources in CMB maps and estimate their fluxes. The method uses previous information about the statistical properties of the sources, so that this knowledge can be incorporated in a Bayesian scheme ([3]). This means a new approach with respect to the tools considered for instance in ([2]), where such an information was not taken into account.

2 Description of the problem

In a region of the sky, we assume to have an unknown number of point-like sources, whose emission is mixed with that of the CMB, $f(x, y)$. A model for the total emission as function of the position (x, y) is given by

$$\tilde{d}(x, y) = f(x, y) + \sum_{\alpha=1}^{n} a_\alpha \delta(x - x_\alpha, y - y_\alpha) \tag{1}$$

where $\delta(x, y)$ is the 2D Dirac delta function, the pairs are the locations of the point sources in our region of the celestial sphere, and a_α are their fluxes. We observe this radiation through an instrument, with beam pattern $b(x, y)$, and a sensor that adds a random noise $n(x, y)$ to the signal measured.

Therefore, the output of our instrument is:

$$d(x, y) = \sum_{\alpha=1}^{n} a_\alpha b(x - x_\alpha, y - y_\alpha) + (f * b)(x, y) + n(x, y) \tag{2}$$

where the point sources and the CMB have been convolved with the beam. In our application, we are interested in extracting the locations and the fluxes of the point sources and consider the rest of the signal as just a disturbance superimposed to the useful signal. If $c(x, y)$ is the signal which does not come from the point sources, model (2) becomes

$$d(x, y) = \sum_{\alpha=1}^{n} a_\alpha b(x - x_\alpha, y - y_\alpha) + c(x, y). \tag{3}$$

If our data set is a discrete map of N pixels, the above equation can easily be rewritten in vector form, by letting d be the lexicographically ordered version of the discrete map

$d(x, y)$, a be the n-vector containing the positive source intensities a_α, c the lexicographically ordered version of the discrete map, $c(x, y)$, and ϕ be an $N \times n$ matrix whose columns are the lexicographically ordered versions of n replicas of the map $b(x, y)$, each shifted on one of the source locations. Equation (3) thus becomes

$$d = \phi a + c. \tag{4}$$

Looking at Eqs. (3) and (4), we see that our unknowns are the number n, the list of locations (x_α, y_α), with $\alpha = 1, \ldots, n$ and the vector a. It is apparent that, once n and (x_α, y_α) are known, matrix ϕ is perfectly determined. Let us then denote the list of source locations by the $n \times 2$ matrix R, containing all their coordinates.

If we want to adopt a Bayesian approach, we must write the posterior probability density of our unknowns. According to Bayes theorem, this posterior probability can be written as

$$p(n, R, a|d) \propto p(d|n, R, a)p(n, R, a) \tag{5}$$

where $p(d|n, R, a)$ is the likelihood function, derived from our data model.

For the CMB plus noise we can assume that $\mathbf{c}$ is a Gaussian random field with zero mean and known covariance ξ. Thus, the likelihood function is

$$p(d|n, R, a) \propto \exp\left(-(d - \phi a)^t \xi^{-1} (d - \phi a)/2\right). \tag{6}$$

To find the prior density $p(n, R, a)$ we need to make a number of assumptions:

1) Both R and a depend on n through the number of their elements. On the other hand, fluxes and positions are, in principle, independent. Thus, we can write

$$p(n, R, a) = p(R, a|n)p(n) = p(R|n)p(a|n)p(n). \tag{7}$$

2) *A priori*, it is reasonable to assume that all the possible combinations of locations occur with the same probability. Therefore

$$p(R|n) = \frac{n!(N - n)!}{N!}, \tag{8}$$

since $N!/(n!(N-n)!)$ is the number of possible distinct lists of n locations in a discrete N-pixel map.

3) It has been checked (see [3]) that the flux distribution can be modeled by a Generalized Cauchy Distribution

$$p(a|n) \propto \prod_{\alpha=1}^{n} \left[1 + \left(\frac{a_\alpha}{a_0}\right)^p\right]^{-\frac{\gamma}{p}}, \tag{9}$$

 ISBN:978-84-615-5392-1

with p a positive number. This distribution obviously assumes that the fluxes of the different sources are mutually independent. In order to work with non-dimensional magnitudes, we define $x_\alpha = a_\alpha/a_0$; we also assume that we will detect point sources above a minimal flux a_i, that leads to the following normalized distribution

$$p(x|n) = \frac{p}{B\left(\frac{1}{1+x_i^p}; \frac{\gamma-1}{p}, \frac{1}{p}\right)} \prod_{\alpha=1}^{n} \left(1+x_\alpha^p\right)^{-\frac{\gamma}{p}}, \tag{10}$$

where B is the incomplete beta function. The values of a_0, p and γ, can be determined by fitting this formula to the point source distribution given by the De Zotti counts model (see [5]).

4) We assume that the number of sources in a given sky patch follows a Poisson distribution, with a known average number of sources λ

$$p(n) = \frac{\lambda^n e^{-\lambda}}{n!}. \tag{11}$$

A detailed account of all these assumptions can be seen in [3].

Finally, by putting together all these prior distributions and the likelihood, we can write the negative log-posterior

$$\begin{aligned} L(n,R,x) &= \frac{1}{2}\left(x^t M x - 2e^t x\right) - \log(N-n)! - n\log(\lambda) \\ &- n\log(p) + n\log B\left(\frac{1}{1+x_i^p}; \frac{\gamma-1}{p}, \frac{1}{p}\right) \\ &+ \frac{\gamma}{p}\sum_{\alpha=1}^{n} \log\left(1+x_\alpha^p\right), \end{aligned} \tag{12}$$

with $M = a_0^2 \phi^t \xi^{-1} \phi$ and $e = a_0 \phi^t \xi^{-1} d$. We assume that we know a_0, p, x_i, γ and λ, so that the unknowns are: the normalized fluxes x, the number of point sources n and the positions of the point sources through the matrix $\phi(R)$.

3 Description of the Algorithm

In [2] an algorithm with a high degree of parallelism that improves, from the computational point of view, the classical approaches for detecting point sources in CMB maps was presented.

In this work, our goal is to find the number of sources and their fluxes and positions by maximizing the posterior probability distribution, or equivalently and more simply, minimizing the negative log-posterior. In conclusion, we search the number of sources, their

 ISBN:978-84-615-5392-1

fluxes and positions, rendered more probable by the data, taking also into account the prior distributions.

Therefore, regarding the flux we minimize the negative log-posterior with respect to x, by taking the derivative and equating to zero, we obtain

$$\sum_{\beta=1}^{n} M_{\alpha\beta}x_\beta - e_\alpha + \frac{\gamma x_\alpha^{p-1}}{1+x_\alpha^p} = 0,\ \alpha = 1, 2, \ldots, n. \tag{13}$$

In order to fix the positions, we assume that the point sources are in the local maxima of e (see [3]). To determine the number of sources, we sort these local peaks from top to bottom and solve (13) successively adding a new source. At the same time, we calculate (12) and select the number of sources which produces its minimum value.

For the sake of simplicity, we redefine $\phi \equiv a_0\,\phi$ and consider $p = 1$, this last assumption is justified by the flux distribution. Besides, taking into account that the thresholding process has been described in [2], in the following we will describe the algorithm without explaining that process again.

The system of non-linear equations we want to solve is

$$\sum_{\beta=1}^{n} M_{\alpha\beta}x_\beta - e_\alpha + \frac{\gamma}{1+x_\alpha} = 0,\ \alpha = 1, 2, \ldots, n, \tag{14}$$

with $M = \phi^t\xi^{-1}\phi$ and $e = \phi^t\xi^{-1}d$. Therefore, the non-linear system can be expressed in matrix form as

$$Mx = e - \gamma v, \tag{15}$$

with $v = (1/(1+x_1), 1/(1+x_2), \ldots, 1/(1+x_n))^t$.

We consider that the matrices M, ϕ and ξ are of order N, while the vectors e and d have N rows. As $\xi \in R^{N\times N}$ is a symmetric positive definite matrix, Cholesky decomposition can be used to obtain a lower triangular matrix such that: $\xi = LL^t$ ([4]). Hence, vector e can be expressed as:

$$e = \phi^t\xi^{-1}d = \phi^tL^{-t}L^{-1}d = \phi^tL^{-t}c_1 = \phi^tc_2 \tag{16}$$

with $c_1 = L^{-1}d$ and $c_2 = L^{-t}c_1$.

Now, in order to construct the matrix M, we calculate

$$M = \phi^t\xi^{-1}\phi = \phi^t(LL^t)^{-1}\phi = (L^{-1}\phi)^t(L^{-1}\phi) = Z^tZ, \tag{17}$$

with $Z = L^{-1}\phi$.

Next, we compute the QR decomposition of $Z = QR$, with $Q \in R^{N\times N}$, orthogonal, and $R \in R^{N\times n}$, upper triangular, and the matrix M can be expressed as

$$M = Z^tZ = (QR)^t(QR) = R^tQ^tQR = R^tR. \tag{18}$$

 ISBN:978-84-615-5392-1

In order to solve (15) we will use the classical Newton-Raphson method and Armijo rule, obtaining a sequence

$$x^{(k+1)} = x^{(k)} - DF(x^{(k)})^{-1}F(x^{(k)}), \tag{19}$$

where

$$F(x) = Mx - e + \gamma v,\ DF(x) = M - \gamma w, \tag{20}$$

with $v = (1/(1+x_1), 1/(1+x_2), \ldots, 1/(1+x_n))^t$ and $w = (1/(1+x_1)^2, 1/(1+x_2)^2, \ldots, 1/(1+x_n)^2)^t$.

We will use the solution of the linear system $Mx = e$ as our initial condition ($x^{(0)}$). Thus, vector $x^{(0)}$ can be computed by solving the triangular linear systems $R^t y = \phi c_2$ and $Rx = y$.

These ideas can be summarized in the following algorithm:

Algorithm CMB Bayesian

Input ϕ, ξ, d, with $\phi, \xi \in R^{N\times N}$, $d \in R^{N\times 1}$

Step 1. Compute ξ: $\xi = L * L^t$

Step 2. Obtain e: $L * c_1 = d$, $L^t * c_2 = c_1$, $e = \phi^t * c_2$

Step 3. Obtain M: $Z = L^{-1}\phi$, $Z = Q * R$, $M = R^t * R$

Step 4. Calculate $F(x)$ and $DF(x)$:

$v = \left(\frac{1}{1+x_1}, \frac{1}{1+x_2}, \ldots, \frac{1}{1+x_n}\right)^t$, $F(x) = (R^t * R) * x - \phi^t * c_2 + \gamma * v$

$w = \left(\frac{1}{(1+x_1)^2}, \frac{1}{(1+x_2)^2}, \ldots, \frac{1}{(1+x_n)^2}\right)^t$, $DF(x) = (R^t * R) - \gamma * w$

Step 5. Solve non-linear system

Step 5.1 Calculate $x^{(0)}$: $R^t * y = \phi * c_2$, $R * x = y$

Step 5.2 Apply Newton-Raphson: $x^{(k+1)} = x^{(k)} - DF(x^{(k)})^{-1} * F(x^{(k)})$

Step 6. Find the number of point sources that minimizes the negative log-posterior

Output x

Preliminary experiments show that the new method detects more sources than the results obtained in [2]. Furthermore, the technique allows us to fix the number of detected sources in a non-arbitrary way.

Acknowledgements

This work was financially supported by the Spanish Ministerio de Ciencia e Innovación and by FEDER (Projects TIN2010-14971, TIN2008-06570-C04-02, TEC2009-13741 and CAPAP-H4 TIN2011-15734-E), Universitat Politècnica de València through Programa de

Apoyo a la Investigación y Desarrollo (PAID-05-11) and Generalitat Valenciana through project PROMETEO/2009/013.

References

[1] Planck collaboration. P.A.R. Ade *et al.*, *Planck early results I. The Planck Mission*, Astron. and Astrophys. **536** (2011) A1.

[2] P. Alonso, F. Argüeso, R. Cortina, J. Ranilla and A. M. Vidal, *Detecting Point Sources in CMB Maps using an Eficient Parallel Algorithm*, J. Math. Chem. **50** (2012) 410–420.

[3] F. Argüeso, E. Salerno, D. Herranz, J. L. Sanz, E. E. Kuruoglu and K. Kayabol, *A Bayesian Technique for the Detection of Point Sources in CMB Maps*, Mon. Not. Roy. Astron. Soc. **414** (2011) 410-417.

[4] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.

[5] G. De Zotti *et al.*, *Predictions for high-frequency radio surveys of extragalactic sources*, Astron. and Astrophys. **431** (2005) 893-903.

[6] D. Herranz, P. Vielva, *Cosmic Microwave Backgound Images*, IEEE Signal Processing Magazine **27** (2010) 67-75.

[7] D. Herranz, F. Argüeso and P. Carvalho, *Compact Source Detection in Multichannel Microwave Surveys: from SZ Clusters to Polarized Sources*, Advances in Astronomy, 2012, in press.

[8] A.A. Penzias, R.W. Wilson, *A Measurement of Excess Antenna Temperature at 4,080 Mc/s*, Astrophys. J. **142** (1965) 419-421.

[9] G. Smoot *et al.*, *Structure in the COBE Differential Microwave Radiometer First-Year Maps*, Astrophys. J. **396** (1992) L1-L5.

[10] D.N. Spergel *et al.*, *First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Determination of Cosmological Parameters*, Astrophys. J. Suppl. **148** (2003) 175-194.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Least squares problem and QR decomposition of Vandermonde matrices

P. Alonso[1], R. Cortina[2], F.J. Martínez-Zaldívar[3] and Antonio M. Vidal[4]

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

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

[3] *Departamento de Comunicaciones, Universitat Politècnica de València*

[4] *Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València*

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

Abstract

Vandermonde matrices appear in many fields of Science and Technology. This paper is motivated by the need to solve least squares problems with complex Vandermonde matrices, using the QR decomposition in Signal Processing applications such as beamforming problems and DOA (Direction of Arrival) analysis, to be executed in devices with low computation power as mobile devices. We present an analysis of existing algorithms that solve the linear least squares problem with this kind of matrices. Some of these algorithms have been extended to explicitly compute the QR decomposition. New algorithms have been developed starting from the existing ones, obtaining also an algorithm for updating of the QR decomposition when a new column is added to the Vandermonde matrix, and an incremental method for computing the QR decomposition starting from the updating algorithm.

Key words: Vandermonde, QR, DOA, Beamforming, Szegö polynomials

1 Introduction

Vandermonde matrices appear in many fields of Science and Technology. As an example, this kind of structured matrices can represent the steering matrix of signal sources in an

array of sensors [7] used in beamforming and DOA (Direction Of Arrival) Signal Processing applications. The targets of these applications are usually to improve the signal-to-noise ratio of the receiving signals, to determine the number of signal sources, to estimate some parameters of them, to track the movement of the sources, etc. For instance, Generalized Sidelobe Canceller (GSC) is an efficient implementation of the Linear Constrained Minimum Variance algorithm for optimum beamforming, where a Vandermonde system must be solved as a part of the total solution [8]. We can find this kind of applications in fields as Seismology, Biomedicine, Astronomy, Radar, Sonar, etc., where these structured data appear.

Several algorithms have been proposed in the literature that solve Vandermonde systems or get their QR factorization, with less computational complexity but with less accurate results than other algorithms for systems with non-structured matrices (see [1] and [3]). Anyway, these algorithms can be suitable in real time applications where execution time may be more important than an extremely precise result, due to either the characteristics of the application or to a lack of high computing power in the system (low performance hardware, energy efficient devices as mobile devices, etc.).

1.1 State of the Art

Since the early nineties, there are not significant contributions in algorithms for solving efficiently Vandermonde systems or getting its QR factorization, taking advantage of the structure of this kind of matrices. In [1] a fast algorithm for computing the QR decomposition of a complex column Vandermonde matrix is shown, with *quadratic* complexity but with poor precision in the results, particularly in the orthogonality of the $\mathbf{Q}$ matrix. In [2], [3] and [4], some other algorithms are presented; they are based on discrete least squares approximation of a real-valued function given at arbitray distinct nodes in $[0, 2\pi)$ by trigonometric polynomials which fits better in some of the signal processing problems presented before. The Stieljes procedure for Szegö polynomials is used to get an intermediate solution, and compared with a better technique based on solving and inverse eigenvalue problem of a Hessenberg matrix with real positive subdiagonal elements. Both methods share the way the solution is obtained from the intermediate solution. Here, the precision of the results are highly dependent on how the nodes are distributed along the interval $[0, 2\pi)$, getting optimal results when they are equispaced in this interval, and worse results when they are randomly distributed in such interval, and even worse when they are concentrated and equispaced in a narrower subinterval.

1.2 Objectives and paper organization

This paper is motivated by the need to solve least squares problems with complex Vandermonde matrices, using the QR decomposition in beamforming problems and DOA analysis. The objective is to provide efficient algorithms that can be executed in devices with low

computation power as mobile devices. In this paper we present an analysis of existing algorithms that solve the aforementioned problems. Some of these algorithms have been extended to compute the QR decompositivon in an explicit way. New algorithms have been developed starting from the existing ones, obtaining the QR decomposition in a way such that its use is easy for solving the least squares problem and for updating the QR decomposition when a new column is added to the Vandermonde matrix. An incremental method for the QR decomposition is developed starting from the ideas of the previous updating algorithm.

The rest of the paper is organized as follows: The section 2 is devoted to the description of the QR decomposition algorithm as well as its updating and the least mean squares problem. In section 3, a precision experimental analysis of the algorithms is done. Finally, in section 4, the conclusions are shown.

2 Algorithms

Let $\{z_1, z_2, \ldots, z_m\}$ be a set of m distinct nodes, let $\{w_1^2, w_2^2, \ldots, w_m^2\}$ be a set of positive weights. For functions g and h defined at the nodes z_k ($\mathbf{g} = \big(g(z_1), g(z_2), \ldots g_(z_m)\big)^T$), denote the inner product on the unit circle:

$$< g, h >= \sum_{k=1}^{m} \overline{g(z_k)} h(z_k) w_k^2.$$

The nodes with a complex exponential formulation $z_l = e^{i\theta_l}, \quad 1 \leq l \leq m$ are specially interesting in signal processing applications such as beamforming or DOA.

The first algorithm shown in [3], known as *Stieljes procedure for Szegö polynomials*, solves the system:

$$\mathbf{DAc} = \mathbf{Dg} \tag{1}$$

where $\mathbf{D} = \mathrm{diag}(w_1, w_2, \ldots, w_m)$. $\mathbf{A}$ is the transposed Vandermonde matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{n-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z_m & z_m^2 & \cdots & z_m^{n-1} \end{pmatrix} \in \mathbb{C}^{m \times n} \tag{2}$$

and $\mathbf{c} = (c_0, c_1, \ldots c_{n-1})^T$ is the set of coefficients of

$$p(z) = \sum_{j=0}^{n-1} c_j z^j$$

such that the discrete least squares error $< g - p, g - p >$ is minimized.

 ISBN:978-84-615-5392-1

If we get the QR decomposition of $\mathbf{A} = \mathbf{QR}$, with $\mathbf{Q} \in \mathbb{C}^{m\times m}$ a unitary matrix and $\mathbf{R} \in \mathbb{C}^{m\times n}$ an upper triangular matrix, then the solution of (1) can be expressed as:

$$\mathbf{c} = \mathbf{R}^{-1}\mathbf{c}', \qquad \text{with} \qquad \mathbf{c}' = \mathbf{Q}^H\mathbf{Dg}$$

where $\mathbf{Q}^H$ denote the conjugate transpose of the matrix $\mathbf{Q}$.

The computation and application of matrices $\mathbf{Q}$ and $\mathbf{R}$ are efficiently done using the Szegö polynomials relation with the problem [6]. Let $\{\phi_j\}_{j=0}^{m-1}$ denote the family of orthogonal Szegö polynmials with respect to some inner product. The grade of ϕ_j is j and its leading coefficient is positive. Matrix $\mathbf{R}$ can be deduced from the relation [3]:

$$z^{k-1} = \sum_{j=1}^{k} r_{jk}\phi_{j-1}(z), \qquad 1 \le k \le n.$$

Hence, the j^{th} column of $\mathbf{R}^{-1}$ is the vector with the coefficients of the polynomial $\phi_{j-1}(z)$. Matrix $\mathbf{Q}$ is determined with:

$$q_{kj} = \phi_j(z)w_k.$$

The algorithms that compute the QR factorization of a matrix without exploiting any structure need $O(mn^2)$ flops, [5]. This method needs only $O(mn)$ floating point operations to obtain the result. Once $\mathbf{c}'$ is obtained, the final solution $\mathbf{c}$ requieres only $O(n^2)$ flops (see algorithm 4.1 of [3]).

The second method is numerically more accurate than the previous one and it is based on an algorithm that constructs a unitary upper Hessenberg matrix from spectral data using elementary unitary similarity transformations [4], [3] (inverse eigenvalue problem), getting $\mathbf{c}' = \mathbf{Q}^H\mathbf{Dg}$ in $O(mn)$ arithmetic operations.

Let $\mathbf{\Lambda} = \text{diag}(z_1, z_2, \ldots, z_m)$, then we can compute a unitary upper Hessenberg matrix, $\mathbf{H}$, with real positive subdiagonal elements:

$$\mathbf{U}^H\mathbf{\Lambda}\mathbf{U} = \mathbf{H}, \tag{3}$$

if the first column of $\mathbf{U}$ is determined. This first column is:

$$\mathbf{u}_1 = \mathbf{U}\mathbf{e}_1 = \sigma_0^{-1}(w_1, w_2, \ldots, w_m)^T \tag{4}$$

where $\sigma_0 = \left(\sum_{k=1}^{m} w_k^2\right)^{1/2}$. Matrix $\mathbf{H}$ in (3) can be obtained using unitary reflectors computed to get zeros sequentially in certain positions. Hence, the matrix $\mathbf{Q}$ is the first n columns of matrix $\mathbf{U}$.

The final solution $\mathbf{c} = \mathbf{R}^{-1}\mathbf{c}'$ is obtained in $O(n^2)$, as in the previous method.

2.1 Gram-Schmidt and modified Gram-Schmidt methods for obtaining Q

The second algorithm presented in the last subsection allows to construct an Arnoldi like procedure [5] for computing easily the factors **Q** and **R** of **A**. Similar ideas are used to get the QR decomposition of a matrix using the Gram-Schmidt method.

Given the first column of **U** as in (4) and expressing (3) as $\mathbf{\Lambda U} = \mathbf{UH}$ we can obtain

$$\mathbf{\Lambda u}_k = h_{1k}\mathbf{u}_1 + h_{2k}\mathbf{u}_2 + \ldots + h_{kk}\mathbf{u}_k + h_{k+1,k}\mathbf{u}_{k+1}, \qquad k = 1, 2, \ldots, n-1.$$

Hence, as **U** is a unitary matrix:

$$\begin{aligned} h_{jk} &= \mathbf{u}_j^H \mathbf{\Lambda u}_k, \qquad j = 1, 2, \ldots, k, \\ \mathbf{v} &= (\mathbf{\Lambda} - h_{kk}\mathbf{I})\mathbf{u}_k - \sum_{j=1}^{k-1} h_{jk}\mathbf{u}_j, \\ h_{k+1,k} &= ||\mathbf{v}||_2, \\ \mathbf{u}_{k+1} &= \mathbf{v}/h_{k+1,k}. \end{aligned}$$

Algorithms based on Gram-Schmidt techniques are usually numerically unstable due to catastrophic cancellation in premature subtractions. That is the reason why they are usually modified in order to avoid a great deal of subtractions before orthogonalizations. This technique is used in the next algorithm. In this case, we compute previously all the possible coefficients of $\mathbf{h}_k$ (column k of **H**) that we can compute. Let us remember that $h_{jk} = \mathbf{u}_j^H \mathbf{\Lambda u_k}$, then we can find the value of h_{kk} from the expression

$$\mathbf{q} \equiv \mathbf{\Lambda u}_k - \sum_{j=1}^{k-1} h_{jk}\mathbf{u}_j = h_k k\mathbf{u}_k + h_{k+1,k}\mathbf{u}_{k+1},$$

hence

$$h_{kk} = \mathbf{u}_k^H \mathbf{q}.$$

Now, we obtain

$$\mathbf{u}_{k+1} = (\mathbf{q} - h_{kk}\mathbf{u}_k)/h_{k+1,k},$$

with

$$h_{k+1,k} = ||\mathbf{q} - h_{kk}\mathbf{u}_k||_2.$$

 ISBN:978-84-615-5392-1

2.2 Updating

Let us suppose we know the QR decomposition of $\mathbf{A} \in \mathbb{C}^{m\times n}$, $m > n$, and we want to compute the QR decomposition of a new matrix $\mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$, where $\mathbf{A}_1 \in \mathbb{C}^{m\times(n+1)}$, $m \geq n+1$, is the original $\mathbf{A}$ matrix with an additional column. Obviously, we can take advantage of this matrix structure to get an easy algorithm that computes $\mathbf{Q}_1$ and $\mathbf{R}_1$. Matrix $\mathbf{Q}_1$ is the matrix made up of $n+1$ columns of $\mathbf{U}$ in the expression (3), so only it is necessary to compute one additional column of $\mathbf{U}$ to get $\mathbf{U}_1$. Besides, $\mathbf{R}_1 = \mathbf{U}_1^H\mathbf{A}_1$, so the first n columns of $\mathbf{U}_1$ match the ones of $\mathbf{U}$ plus zeros to complete the $n+1$ components, and

$$\mathbf{R}_1(:, n+1) = \mathbf{U}_1^H\mathbf{A}_1(:, n+1).$$

2.3 An incremental algorithm based on the updating technique

From the point of view of the applications, it is interesting to provide algorithms that receives information gradually and process it in the same way. An efficient algorithm can be designed starting from the previous updating algorithm for computing the QR decomposition of a matrix which columns are growing increasingly.

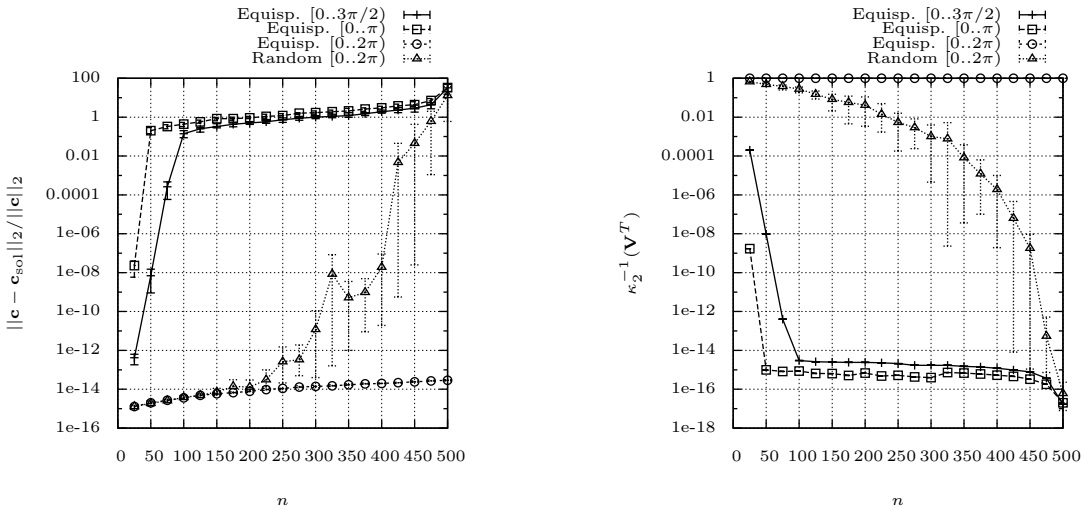
3 Experimental results

Experimental precision results have been obtained using double precision arithmetic, with a constant number of rows of the transposed Vandermonde matrix ($m = 500$), varying the number of columns (n) upto m. The resulting Vandermonde system has been solved and compared with several methods. The experiment results have been averaged ten times. The experiments take into account two kind of nodes: equispaced nodes in certain subinterval, and uniformly distributed random nodes in certain subinterval. All the experiments have been repeated using single precision arithmetic, obtaining qualitatively the same conclusions.

3.1 Matrix conditioning and result precision

Figure 1(a) shows the relative error obtained using the Lapack `GELS` subtroutine solving a Vandermonde least squares problem generated with different node distribution, and Figure 1(b) shows the reciprocal 2-norm condition number of each matrix.

The worst behavior is obtained when the nodes are distributed equispacedly in a subinterval narrower than $[0, 2\pi)$ (the narrower, the worse results). The best results are obtained when the nodes are distributed equispacedly in the interval $[0, 2\pi)$. When the nodes are distributed randomly in the $[0, 2\pi)$ interval, the precision of the results are in the intermediate positions with worse results when the matrix is getting squared ($n \approx m$), following



(a) Lapack `GELS` solution relative error

(b) Reciprocal 2-norm condition number

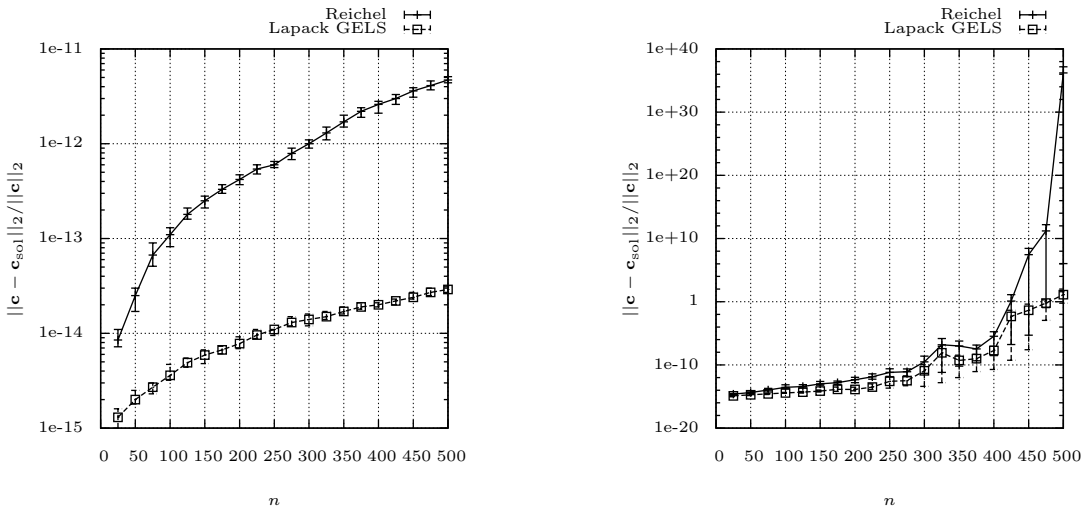
Figure 1: Error in Lapack `GELS` solution and reciprocal condition number relation

the trend of the condition number.

Figures 2(a) and 2(b) show a comparison of the relative error in the solution between the Reichel method and the `GELS` Lapack method using matrices generated with equispaced nodes and random nodes respectively, both in the $[0, 2\pi)$ interval, with better results for the Lapack case.

3.2 Orthogonality results

Figures 3(a) and 3(b) show a comparison of the orthogonality error, $||\mathbf{I} - \mathbf{Q}^H\mathbf{Q}||_2$, among the Reichel, Lapack, Gram-Schmidt and Modified Gram-Schmidt methods, using matrices generated with equispaced nodes and random nodes respectively, both in the $[0, 2\pi)$ interval. For the equispaced nodes case, the worst results are for the Reichel method; the rest of the methods share similar performance (with better results for Gram-Schmidt methods). For the random nodes case, there exist a crosspoint in the behavior: before the crosspoint, all the methods get a similar orthogonality precision, with a subtle better performance for the Gram-Schmidt methods; after the crosspoint, the Gram-Schmidt methods loses this precision.



(a) Equispaced nodes in $[0, 2\pi)$

(b) Random nodes in $[0, 2\pi)$

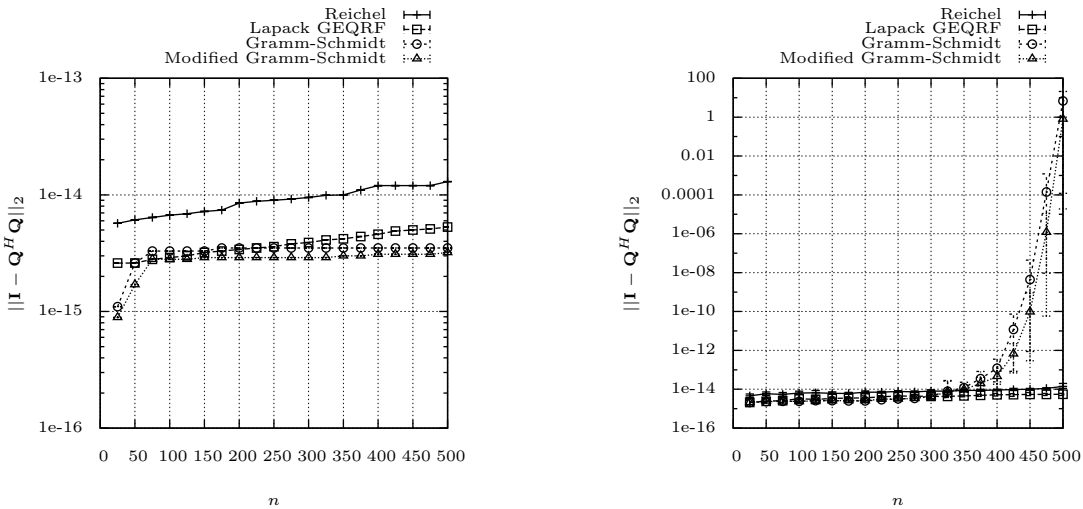
Figure 2: Solution relative error

3.3 Incremental QR algorithm results

Figures 4(a) and 4(b) show the orthogonality error, $||\mathbf{I} - \mathbf{Q}^H\mathbf{Q}||_2$, and the decomposition relative error, $||\mathbf{A} - \mathbf{QR}||_2/||\mathbf{A}||_2$, respectively when the incremental QR algorithm is used, observing that the performance is comparable with the non-incremental counterpart algorithm results.

4 Conclusions

Vandermonde matrices are difficult to work with due to their numerical properties. In this paper we have analyzed the behavior of several algorithms that solve the least square problem and obtain the QR decomposition of a Vandermonde matrix. The obtained peformance is as expected: it depends strongly on the condition number and it get worse when the matrix is becoming square. Our contribution is an algorithm for obtaining the updating of the QR decomposition of a Vandermonde matrix, and a QR incremental algorithm suitable for real time signal processing applications.



(a) Equispaced nodes in $[0, 2\pi)$

(b) Random nodes in $[0, 2\pi)$

Figure 3: Orthogonality error

Acknowledgements

This work was financially supported by the Spanish Ministerio de Ciencia e Innovación projects TEC2009-13741 and TIN2010-14971, the Vicerrectorado de Investigación de la UPV through Programa de Apoyo a la Investigación y Desarrollo (PAID-05-11-2733) and Generalitat Valenciana through projects PROMETEO/2009/013 and ACOMP/2012/076.

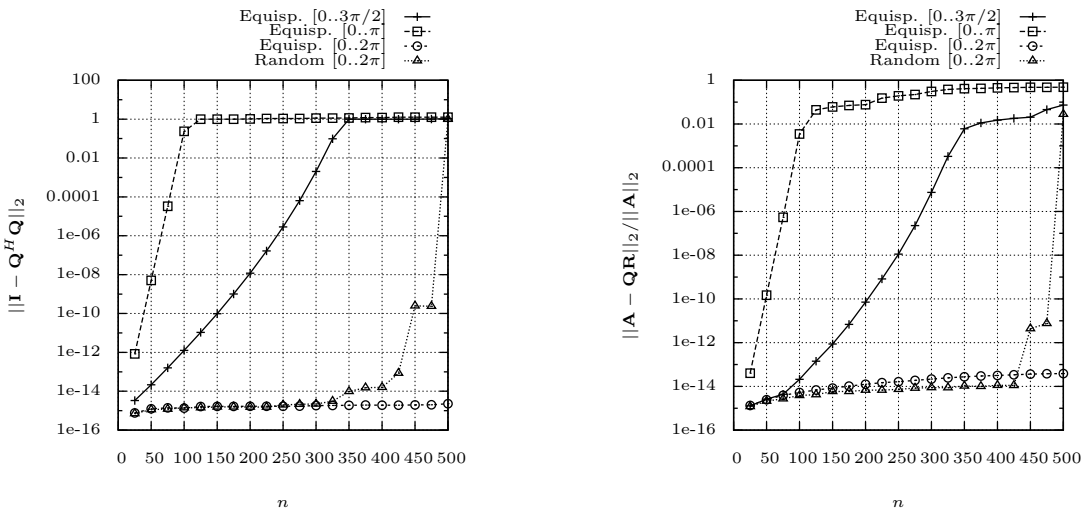
References

[1] C. J. Demeure, *Fast QR Factorization of Vandermonde Matrices*, Linear Algebra and its applications. **124** (1989) 165–194.

[2] L. Reichel, *Fast QR Decomposition of Vandermonde-like matrices and polynomial least squares approximation*, SIAM J. Matrix Anal. Appl. **12(3)** (1991) 552–564.

[3] L. Reichel, G. S. Ammar, and W. B. Gragg, *Discrete least squares approximation by trigonometric polynomials*, Mathematics of Computation. **57(195)** (1991) 273–289.

[4] G. S. Ammar, W. B. Gragg and L. Reichel, *Constructing a Unitary Hessenberg Matrix from Spectral Data*, Numerical Linear Algebra, Digital Signal Processing and



(a) Ortogonality error

(b) Decomposition error

Figure 4: Error in incremental QR algorithm

Parallel Algorithms. G. H. Golub and P. Vand Dooren, eds. NATO ASI Series **F70** (1991) 385–395.

[5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix computations (3rd ed.)*, Johns Hopkins University Press, 1996.

[6] U. GRENANDER AND G. SZEGÖ, *Toeplitz forms and their applications*, Chelsea, New York, 1984.

[7] H. KRIM AND M. VIBERG, *Two decades of Array Signal Processing research. The parametric approach*, IEEE Signal Processing Magazine. **13** (1996) 67–94.

[8] W. LIU AND S. WEISS, *Wideband beamforming: concepts and techniques*, Wiley, 2010.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Collaborative work in mathematics with a wiki

Pedro Alonso[1] **and Rafael Gallego**[1]

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

emails: palonso@uniovi.es, rgallego@uniovi.es

Abstract

In this work we show an educational activity consisting of problem solving tasks in Mathematics on a group work basis with a wiki. To this end, we use the software Mediawiki which is open source and it natively allows to introduce mathematical formulae using LaTeX syntax. On the other hand, it has a sophisticated system to watch pages and its functionality can be increased by means of extensions made by a large user community.

Key words: wikis, mediawiki, problem solving, mathematics, collaborative work, e-learning

1 Introduction

A Wiki is a suitable platform to prepare collaborative works as an alternative to more traditional ways of presenting works in paper or in a electronic file (.doc, .pdf, etc.). A wiki also allows to develop cooperative learning habits. By means of the history associated to every page of the wiki, a teacher can know in detail the contribution of each member of the group to the proposed tasks.

The e-learning platform moodle allows to set up wikis, both for a single user and for groups of users. However, it shows some deficiencies when introducing mathematical formulae, which is of paramount importance in problem solving in Mathematics. Similar problems arose with the known web site wikispaces [1] which allows to set up small wikis for free.

The developer of the first wiki software was Ward Cunningham [2], who in 1994 made *WikiWikiWeb* [3], originally described as "the simplest online database that could possibly work". The most known wiki is doubtlessly the *Wikipedia* [5], based initially on the software

UseMod. Later, the founders of the project made their own software called Mediawiki [4] which is currently used. The biggest wiki nowadays is the Wikipedia in English, that on April 1st, 2012, had near four million articles. The Wikipedia is more used and is larger than the famous Encyclopædia Britannica, and many of its articles are more accurate.

The most important features of a wiki are:

- **It has an hypertextual structure**. This makes collaboration possible. Users can create and edit articles without the necessity of knowing HTML, the language behind web pages. It is not necessary use a web page editor either.
- **Social authorship**. Anyone can write and edit any article. A typical wiki invites all users to participate, but this is not essential. The process of creating and editing articles is very fast. The articles are very dynamic, so they are changing continuously and they are never considered as closed.
- **Change history**. Every page of the wiki has a history page attached where one can see the contributing users and the dates of every single editing. Depending on the software, changes can be undone and it is possible to revert a page to a previous state.

Wikis in Education are also used to develop collaborative works. E-learning platforms like Moodle [6] incorporate the possibility of setting up wikis. These can be monitorized by the teacher in order to follow the detailed evolution of its development. It is also possible to use open source software to create wikis, like Mediawiki, the Wikipedia engine.

Teachers also use wikis as an alternative to upload their notes on the Internet. This has a number of advantages:

- No knowledge of the HTML language neither of web page editors is needed to create pages in the wiki.
- It is easy to keep, update and make new contributions.
- It allows to introduce external resources.
- Depending on how the subject is organized, the wiki could be opened to the participation of the students.

2 Description of the work

The main objective of the work presented in this text is the development through a wiki of problem solving tasks in Mathematics on a group work basis. Furthermore it is intended to obtain and develop a number of competencies within the European higher education system framework. The activity was carried out the second four-month period of the last academic year in a Numerical Analysis course at the University of Oviedo, in Spain.

 ISBN:978-84-615-5392-1

During the academic year 2008-2009 the students of our course were proposed an activity consisting of making periodically some exercises that had to be given to the teachers after a certain period. The volunteers were arranged in groups of three or four people. The resulting numeric mark of the assessment of the activity, up to a maximum of 2 points, was added to the mark of the theory exams as long as the final mark did not exceed 10 points. A problem we noted was that the real contribution of every student to the work could not be determined.

During the academic year 2009-2010 we carried out the educational activity described in this document. The organization was similar to that of the previous year with the important difference that now the exercises had to be solved on a wiki. For every exercise a deadline was fixed. Beyond the deadline the permission to edit the corresponding pages of the wiki was revoked.

Since we are dealing with mathematical problems, it is necessary that all sort of mathematical expressions can be introduced in the wiki. The builtin wiki module of the moodle elearning software allows the inclusion of formulae with LaTeX syntax [8], but we found it too buggy. Instead, we decided to use the free Mediawiki software, on which the famous Wikipedia relies. The features of Mediawiki can be extended by means of a big number of add-ons made by the user community. Moreover, it is possible to insert natively in the articles mathematical expressions using LaTeX syntax.

Although the features of Mediawiki to introduce mathematical formulae are actually satisfactory, we decided to add extra functionality by installing an extension named Wikitex [7]. Wikitex allows to insert in the wiki not only mathematical expressions of arbitrary complexity, but also graphics (made with GNUplot [9]), chess matches, several kinds of diagrams, etc.

We were aware that most of the students had little or no knowledge of LaTeX. This fact did not have to prevent the students from doing the work so we offered two alternatives to make the process easier: an exhaustive LaTeX tutorial was made in the wiki itself and also the students could use the *Mathtype* [10] software that includes an equation editor capable of exporting to the mediawiki format.

In Mediawiki, the main page of an article comes with its corresponding discussion page. This is useful for the group members to organize their work and exchange ideas. Since the wiki can be accessed on the Internet, the students can work in the wiki without being physically in a meeting. They only need to have a computer connected to Internet. This is a clear advantage against other more traditional collaborative work strategies.

The history of the wiki plays an important role. The teacher can know in detail the contribution of each member of the group to the proposed work by looking at the history associated to the relevant pages. This allows to measure the collaboration inside the group. When the students gave a hard copy of their works, this kind of evaluations could not be done objectively. In Mediawiki, the page history allows to compare different versions and

 ISBN:978-84-615-5392-1

undo changes. This way, a page can be reverted to a previous state.

3 Assessment process

The assessment process was based on a rubric. We give the details in this section.

3.1 Rubric

Once finished, the exercises were checked on printed versions that later were given to the students. We also considered the possibility to make the corrections in the wiki itself, but eventually we gave up the idea since we did not have enough time. The task of checking the exercises in the wiki is certainly very interesting from the students learning point of view and it will be taken into account in future activities. In the wiki, the students could easily check both their own exercises and those from other groups.

For each student, the exercises were marked following the next rubric:

Category	4	3	2	$0-1$
Neatness and Organization (15%)	The work is presented in a neat, clear, organized fashion that is easy to read.	The work is presented in a neat and organized fashion that is usually easy to read.	The work is presented in an organized fashion but may be hard to read at times.	The work appears sloppy and unorganized. It is hard to know what information goes together.
	$7.5-10$	$5-7.4$	$2.5-4.9$	$0-2.4$
Explanation and Completion (35%)	The level of explanation and completion is at least 75%	The level of explanation and completion is between 50% and 75%	The level of explanation and completion is between 25% and 50%	The level of explanation and completion is under 25%
	4	3	2	$0-1$
Oral Test (up to 50%)	The student has answered properly to the questions asked by the teacher, showing a good knowledge of the exercises	The student has answered quite correctly to the questions asked by the teacher	The student has shown difficulties to answer the questions asked by the teacher	The student has not answered properly to most of questions asked by the teacher

 ISBN:978-84-615-5392-1

Note the existence of an *oral test*. This was compulsory. Although several students decided not to do the oral test (so they were ruled out in the activity), the percent did not reach the 5% of the total number. Taking advantage of these tests, we made an opinion poll to know the students' opinion about the activity.

3.2 Computation of the final marks of the exercises

We considered two marks for each exercise, namely:

- Neatness and Organization (n_1): $0 \leq n_1 \leq 4$
- Explanation and Completion (n_2): $0 \leq n_2 \leq 10$

On the other hand, the students had to defend their work in an oral test, after which they got a mark n_3, $0 \leq n_3 \leq 4$.

If a group made N exercises of a total number of M, the final mark n_f of every member of the group, computed up to 2 points, is given by:

$$n_f = (p+q)\frac{N}{M}, \qquad 0 \leq n_f \leq 2,$$

where

$$p = \frac{2}{N}\sum_{i=1}^{N}\left(\frac{n_1^{(i)}}{4}0.15 + \frac{n_2^{(i)}}{10}0.35\right),$$

$$g = \begin{cases} -p & \text{if } n_3 = 0, \\ -0.25 & \text{if } n_3 = 1, \\ 0.25 & \text{if } n_3 = 2, \\ 0.75 & \text{if } n_3 = 3, \\ 1 & \text{if } n_3 = 4. \end{cases}$$

Note that a bad mark in the oral test ($n_3 = 0$ or 1) causes the final mark n_f to be decreased. Finally, the mark n_f was added to that of the theory exams as long as the resulting number was not greater than 10 points.

4 Conclusions

We have confirmed the advantages of working with a wiki against more traditional ways of group work. The teacher can track the students' progresses any time and the works can be accessed in a single location. Besides, he can know each individual contribution inside the group by means of the history pages. On the other hand, the students can work with just

a computer with Internet connection. The wiki has discussion pages that students can use to organize their work and exchange ideas.

We think that this experience has been really positive from the point of view of both the teacher and the students. Furthermore, this activity made it possible that many students worked with a wiki for the first time. This may stimulate them to write an article for the Wikipedia in the future.

We plan to extend this activity to other subjects in Mathematics. We also intend to do the revision of the exercises in the wiki itself, both by teachers and students.

References

[1] http://www.wikispaces.com

[2] B. Leuf y W. Cunningham. *The Wiki way: Collaboration and Sharing on the Internet.* Addison-Wesley, 2001.

[3] http://en.wikipedia.org/wiki/WikiWikiWeb

[4] http://www.mediawiki.org

[5] http://www.wikipedia.org

[6] http://www.moodle.org

[7] http://wikisophia.org/wiki/Wikitex

[8] http://www.latex-project.org

[9] http://www.gnuplot.info

[10] http://www.dessci.com/en/products/mathtype

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A self-adjusting algorithm for solitary wave simulations

I. Alonso-Mallo[1] and Nuria Reguera[2]

[1] *Department of Applied Mathematics, University of Valladolid, Spain*

[2] *Department of Mathematics and Computation, University of Burgos, Spain*

emails: isaias@mac.uva.es, nreguera@ubu.es

Abstract

We introduce a practical algorithm to automate the simulations of solitary wave solutions of some nonlinear dispersive wave equations. The full discretization consists of a spatial discretization with a local basis and an invariant preserving time integrator. The algorithm includes a dynamic cleaning of dispersive tails and an automatic detection of the complete separation of the main pulses.

Key words: Solitary waves, cleaning procedures, conservative methods
MSC 2000: 65M20, 65M99, 35Q53, 76B25

1 Introduction

The purpose of this work is to develop a dynamic algorithm for the simulation of solitary waves. This subject requires to pay attention to some numerical problems. In many situations the numerical solution evolves into a main pulse or train of pulses along with some small waves of different nature that sometimes form dispersive tails. This is the case, for example, when managing with small perturbations of waves or when studying the interaction of two or more solitary waves. Then the use of a finite computational window along with periodic boundary conditions forces to 'clean' the solution: isolating the main pulses, eliminating somehow and sometime the tails and turbulences, and leaving the main pulses alone to evolve.

The simplest case to study is the perturbation of an only solitary wave which evolves into one main pulse along with dispersive tails [2].

Another interesting case in which we focus now is the interaction of two solitary waves. In this case, the computation of the cleaning region is more complicated. Taking into

account the periodic boundary conditions, when the waves are 'completely separated' there will be two cleaning intervals, while in other case there will be only one. That is why, it is essential that the algorithm detects automatically when the interaction starts and when the waves that appear after the collision are completely separated. The algorithm that we propose cleans the dispersive tails and turbulences that appear so that they can not alter the new solitary waves that arise after the collision. Moreover, this algorithm allows to carry on the experiment in order to study several consecutive collisions.

Although our algorithm can be used for a wide class of partial differential equations, we here consider the BBM equation

$$u_t + u_x + uu_x - u_{txx} = 0, \tag{1}$$

where $u = u(x,t)$ is a real-valued function of the two real independent variables x, t. This equation appear in certain models about the propagation of small-amplitude, nonlinear, dispersive long waves [4, 5]. Solitary wave solutions of (1) are of the form

$$u(x,t) = A\text{sech}^2(K(x - ct - L_0)), \quad A = 3(c-1), \quad K = \frac{1}{2}\sqrt{1 - \frac{1}{c}}, \tag{2}$$

where the parameter $c > 1$ represents the velocity of the wave.

2 Description of the algorithm

We consider the discretization of the initial boundary value problem for the BBM equation with periodic boundary conditions

$$\begin{aligned} &u_t + u_x + uu_x - u_{txx} = 0, \quad x \in [0, L],\, t \geq 0, \\ &u(0,t) = u(L,t), \\ &u_x(0,t) = u_x(L,t), \\ &u(x,0) = u_0(x), \quad x \in [0, L]. \end{aligned} \tag{3}$$

The cleaning technique that we propose requires a spatial discretization with local character so that the cleaning of small perturbations does not alter the whole computational window and therefore the main pulses. In this work we are going to consider cubic finite elements although other options are also possible.

On the other hand, in order to choose a suitable time integrator it is important to take into account that the semidiscrete system obtained after the spatial discretization retains a Hamiltonian structure and has as conserved quantities the corresponding discrete versions of the invariants of the original problem. The conservation of these invariants quantities through the numerical integration is a convenient property for a time integrator [1, 3]. Taking this fact into account there are still several possibilities. Between them we have chosen the classical implicit midpoint rule. This symplectic method presents a good behavior with respect to the invariants of our problem while at the same time it is quite easy to implement letting us focus on the implementation of the algorithm.

2.1 Cleaning technique

Let us explain now very briefly the cleaning technique. For simplicity, we consider the case when the initial condition in (3) has evolved into an only main pulse along with some turbulences, although in Section 2.2 it will be used for the case of two pulses. Our cleaning technique calculates in a dynamic way a suitable cleaning region for which a relation between the velocity, the amplitude and the 'support ' of the main wave is assumed. In this context the term 'support ' associated to a previously fixed tolerance ε will represent an interval where the profile is greater than ε. This relation could be estimated if it is not known in an exact way (see [2] for details).

In order to clean at time t_n of the computation, the point $x_{max,n}$ where the numerical solution attains its maximum absolute value is calculated with the maximum accuracy that the spatial discretization allows. For this, a first estimation $\widetilde{x}_{max,n}$ is done by means of the numerical velocity c_n of the main wave. Then, the maximum nodal value $x_{jmax,n}$ is calculated reducing the search, for efficiency, to the nearest nodes to $\widetilde{x}_{max,n}$. Finally, $x_{max,n}$ is found by calculating the point where the cubic Hermite piecewise interpolant associated to the numerical solution in the adjacent intervals to $x_{jmax,n}$ reaches its maximum.

Once the point $x_{max,n}$ has been found, the algorithm computes the supports of the solitary wave with velocity c_n associated to two given tolerances $\varepsilon_1 > \varepsilon_2$: $(\beta_{1,n}, \beta_{2,n}) \subset (\gamma_{1,n}, \gamma_{2,n})$, both centered at $x_{max,n}$. Then, the solution is set equal to zero outside $(\gamma_{1,n}, \gamma_{2,n})$, while in the intervals $(\gamma_{1,n}, \beta_{1,n})$ and $(\beta_{2,n}, \gamma_{2,n})$ a cubic interpolation is implemented in order to obtain a smooth enough numerical approximation.

2.2 Algorithm for simulating the collision of two solitary waves

Now we focus on the case of two solitary waves. That is, we assume that the numerical solution consists of two main pulses traveling with different velocities. We propose an algorithm for simulating the interaction of both pulses, cleaning in an automatic way the turbulences that appear due to the collision. Moreover, the cleaning technique will let to carry on the experiment in order to study the successive collisions of the waves formed after each interaction, which is possible due to the periodic boundary conditions.

At each time step t_n the algorithm is going to consider three intervals associated to each of the two main pulses (we use superscripts to refer to each pulse). Two of them are the ones associated to the cleaning procedure: $(\beta_{1,n}^{(j)}, \beta_{2,n}^{(j)})$, $(\gamma_{1,n}^{(j)}, \gamma_{2,n}^{(j)})$, for $j = 1, 2$, following with the notation of previous section. If

$$(\gamma_{1,n}^{(1)}, \gamma_{2,n}^{(1)}) \bigcap (\gamma_{1,n}^{(2)}, \gamma_{2,n}^{(2)}) = \emptyset$$

and therefore the waves are completely separated, there will be two cleaning intervals (we can clean 'between' both waves), while in other case, there will be only one.

 ISBN:978-84-615-5392-1

Notice that in order to apply the cleaning technique in this case, the points $x_{max,n}^{(1)}, x_{max,n}^{(2)}$ where each main pulse attains its amplitude should be calculated. Nevertheless, this can not be done during a short period of time when the collision is taking place, and an estimation should be done instead of computing these points as mentioned in Section 2.1. In order to determine this period of time, two more intervals $(\alpha_{1,n}^{(j)}, \alpha_{2,n}^{(j)})$, centered at $x_{max,n}^{(j)}$ for $j = 1, 2$ and smaller than the previous ones are needed. The interval $(\alpha_{1,n}^{(j)}, \alpha_{2,n}^{(j)})$ is the support of the solitary wave with velocity $c_n^{(j)}$ associated to a given tolerance ε_3 greater that the previous ones ε_1 and ε_2. Then, if

$$(\alpha_{1,n}^{(1)}, \alpha_{2,n}^{(1)}) \bigcap (\alpha_{1,n}^{(2)}, \alpha_{2,n}^{(2)}) = \emptyset,$$

then points $x_{max,n}^{(1)}, x_{max,n}^{(2)}$ will be calculated as explained at Section 2.1, and otherwise an estimation should be done.

Numerical experiments confirm the good properties and usefulness of the proposed algorithm.

Acknowledgements

This research has been supported by MCINN project MTM2011-23417 cofinanced by FEDER funds.

References

[1] I. Alonso-Mallo, A. Durán and N. Reguera, *Simulation of coherent structures in nonlinear Schroedinger-type equations*, J. Comput. Phys. **227** (2010), 8180–8198.

[2] I. Alonso-Mallo, A. Durán and N. Reguera, *A numerical technique of cleaning in solitary-wave simulations*, Proceedings of the 2011 International Conference on Computational and Mathematical Methods in Science and Engineering.

[3] J. Álvarez and A. Durán, *A numerical scheme for periodic travelling-wave simulations in some nonlinear dispersive wave models*, J. Comp. Appl. Math. **235** (2011), 1790–1797.

[4] T. B. Benjamin, J. L. Bona and J. J. Mahony, *Model equations for long waves in nonlinear dispersive systems*, Phil Trans R Soc Lond A **272** (1972), 47–78.

[5] D. H. Peregrine, *Calculations of the development of an undular bore*, J Fluid Mech **25** (1996), 321–330.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Hierarchical approaches for multicast based on Euclid's algorithm

J.A. Alvarez-Bermejo[1], N. Antequera[2] and J.A. Lopez-Ramos[2]

[1] *Department of Computers Architecture and Electronics, University of Almeria*

[2] *Department of Algebra and Analysis, University of Almeria*

emails: jaberme@ual.es, nicolas.antequera@gmail.com, jlopez@ual.es

Abstract

We introduce a hierarchical approach for secure multicast where rekeying of groups of users is made through a method based on Euclid's algorithm for computing GCD. We consider tree arrangements of users and a distributed protocol by groups with group managers that may help both distributing the information and detecting some possible inner attacks.

Key words: Security, Multicast, Euclid's algorithm,
MSC 2000: 94A60, 68P25

1 Introduction

Multicast communications allow a host to simultaneously send information to a set of other hosts, avoiding the establishment of point-to-point connections with all of them. There exist many situations where multicast reveals to be the most suitable way to distribute the information such as pay-per-view IPTV or P2PTV, private multiconferences, or any private service that involves several participants or clients. This has increased the interest in researching on appropriate protocols for secure multicast. Some surveys on this field can be found in [2], [9], or more recently in [12].

In [3] the authors made a computational approach to the problem and introduce a solution based on the Chinese Remainder Theorem, the so-called Secure Lock. However, as shown in [4], computational requirements become quickly huge as the number of user grows. To reduce the number of computations, in [10], a divide-and-conquer extension of Secure Lock is introduced. It combines the well-known Hierarchical Tree Approach, [11] and the

Secure Lock. The authors propose an arrangement of the members as in a HTA, and use Secure Lock to refresh keys on each tree level.

In [7] the authors introduce a new computational method based on Euclid's algorithm for computing the greatest common divisor of two integers that shows to be adequate in an environment where users are constantly joining and/or leaving the system with low communication overcomes and key storage and gets forward and backward secrecy: new members cannot decrypt information multicasted before their arrival and those leaving the multicast group are not able to access the encrypted information after their departure. The protocol has three parts: a key distribution scheme, an alternative key refreshment authentication and a validation protocol between authorized users. This scheme was object of a cryptoanalysis in [8], but positively addressed in [1]. However, as shown in [7], the length of the rekeying messages grows linearly as the number of users.

The aim of this work is to show how properties of the Euclid's approach combined with the hierarchical tree approach gives rise to a powerful method that allows to multicast messages in environments with huge and highly dynamic audiences with very low communication overcomes, including the length of the rekeying messages. We will show also how the use of Euclid's approach becomes natural in some hierarchical tree situations from the properties of the prime numbers, certainly "the core of this method".

The structure of this paper is as follows. Firstly we recall briefly Euclid's protocol for multicast. Then we discussed rekeying messages in a hierarchical tree distribution of users and in situations where users have different attributes for accessing different services or information. These approaches constitute centralized protocols, i.e., a single entity is in charge of creating and distributing the rekeying messages. Finally, in the last section we introduce a distributed situation where our approach is also suitable because of its nature. Now some detached users, namely, the group managers, are in charge of creating rekeying messages from an original one coming from the Key Server for their corresponding controlled groups. This distributed approach is particularly appropriated when trying to avoid certain type of attacks that can be developed by legal users as we will show.

2 Join and leaves in HTA+Euclid approach

As in HTA and the Secure Lock + HTA approaches, our proposal uses the divide and conquer strategy. As in the Secure Lock+HTA case, the number of transmissions is reduced with respect to the HTA case, the computational requirements at the Key Server's side are still very low and the length of rekeying messages is considerably reduced, so we can give service to a much bigger number of users without delaying in rekeying operations. The idea is exactly the same as the one introduced in [10]. However let us assume a more general scenario than that considered in [10, Section 3.4]. Consider a hierarchical tree with a depth of 4, i.e., the number of levels below root is 3, and a degree of n, i.e., the number of children

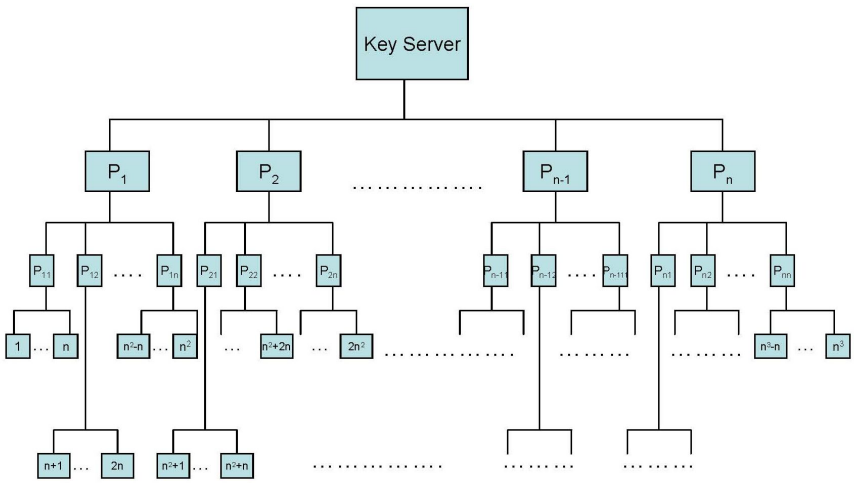


Figure 1: Hierarchical Tree

below each parent node is n (see Figure 1).

Let us assume with out loss of generality that user 1 wants to leave. It can be easily observed that we need the following messages to refresh the key and preserve forward secrecy.

1. Key Server $\rightarrow \{2, \ldots, n\}$: E($P_S$, P_1, $P_{1,1}$), being P_S the session key at the Key Server and using the private information hold by users $\{2, \ldots, n\}$.

2. Key Server $\rightarrow \{n+1, \ldots, n^2\}$: E($P_S$, P_1,) using the private information $P_{1,2}, \cdots, P_{1,n}$.

3. Key Server $\rightarrow \{n^2+1, \ldots, n^3\}$: E($P_S$) using the private information $P_2, \cdots, P_n$.

where P_i, $P_{i,j}$ are prime numbers for every $i = 1, \ldots, n$ and $j = 1, \ldots, n$ and by $E(-)$ we mean the encryption of the corresponding information using the Euclid's approach, i.e., in the first message $E(P_1) = u = P_1^{-1}\ mod\ \prod_{i=2}^{n} P_{1,j}$.

We also detach that if we are dealing with primes of 1024-bit length, then the length of messages 1, 2 and 3 are about $3 \cdot n \cdot 128$, $2 \cdot n \cdot 128$ and $n \cdot 128$ bytes respectively. These means, in case $n = 100$, that we are giving service up to one million users and messages are about 37kb, 25kb and 12kb respectively and the computing time to generate them does not depend on the number of users, since operations are just multiply or divide by a prime a product of primes, say L, select a new value k for $\delta = k + p$ and calculate the greatest common divisor of L and δ in case we are dealing with the session key. In the other cases,

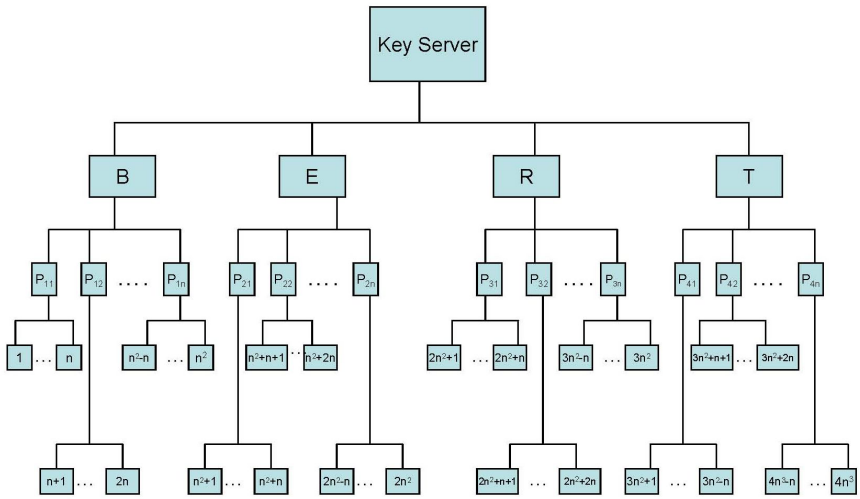


Figure 2: Hierarchical Tree Multilevel Security

computations reduce to the calculation of the inverse of prime module modulo the product of some other different primes.

3 Multilevel security and Euclid's method

As it was show in the previous section the Euclid's algorithm allows to deliver efficiently a secret to a huge plurality of users. But this can be also used to rekey in environments with a key hierarchy as the Secure Lock case (cf. [10, Section 4]). The argument is essentially the same although we will describe it since depending on the situations we could simplify it in some way.

Assume first that the audience is composed by a huge amount of users, for instance in a Pay-Per-View TV broadcasting and that we have four levels of service. In the highest one, T, we get the complete set of services offered, let us say movies, sports, entertainment and general channels. Then we have a reduced version where not all the services are obtained, a packet containing sports, entertainment and general channels, namely R. The third category, E, could be formed by just entertainment and general channels and, finally, the basic packet, B, offering simply general channels. This situation is represented by Figure 2. In this case, the protocol is exactly the same as that introduced [10, Section 4.2] and we do not encounter any problem with computational requirements as outlined in that case, where this method is applicable to just situations where the number of users is reduced due to this fact.

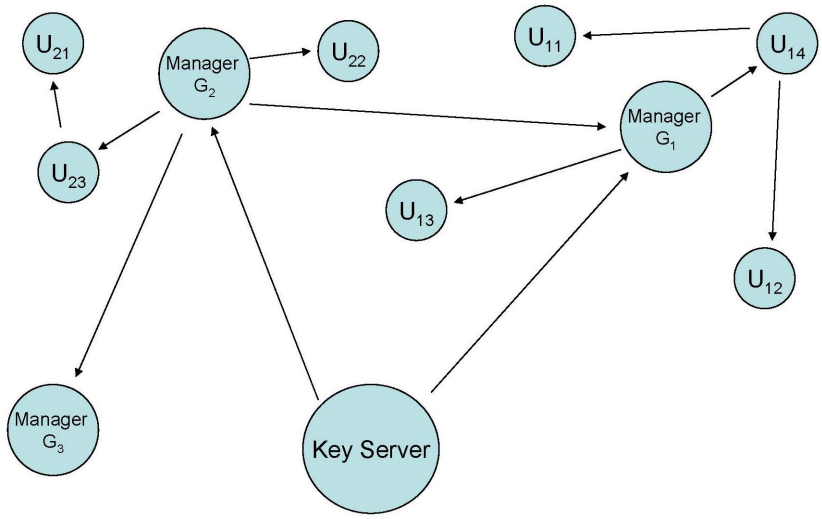


Figure 3: Distributed Protocol Groups

4 Group controllers, a decentralized protocol

Euclid's approach is also applicable for a decentralized protocol. These architectures divide management of large groups among subgroups with a trusted agent in charge of each subgroup. The best known example is probably Iolus ([6]). The situation we are proposing is as follows. Audience is divided into subgroups, each one managed by a trusted agent or group manager and there is a Server in charge of distributing contents, encrypted with a session key and assigning private information to new users and to group managers. Information provided by the Server, besides the above mentioned encrypted contents consists of the following:

Initialization steps

1. Users are distributed by groups $\{G_1, \ldots, G_n\}$, each one managed by a trusted user or group manager u_j of group G_j, $j = 1, \ldots, n$, as shown in Figure 4. The Key Server assigns a prime $p_{j,i}$ for every user $u_{j,i}$ in group G_j in the system, including the corresponding primes for the group managers, that we will denote by p_j.

2. The Server calculates the products $L_G = \prod_{i=1}^{n} p_i$ and $L_j = \prod_{i=1}^{k_j} p_{j,i}$.

3. The Server sends individually $\{L_G, L_j\}$ to u_j, the group manager of G_j.

Distributing the information

 ISBN:978-84-615-5392-1

1. The Server encrypts the information with a session key K_S and sends it to the group managers using L_G.

2. The group manager receives the encrypted key and decrypts it using p_j. Then sends K_S using L_j to users $u_{j,i}$ in group G_j.

3. Each user $u_{j,i}$ gets K_S by using $p_{j,i}$.

Rekeying messages

1. A rekeying message without a user joining or leaving the system just runs as above.

2. If user $u_{j,i}$ leaves or joins the system, then the Server calculates the new L_j.

   (a) The server sends individually L_j to u_j.

   (b) The new session key is distributed as explained above.

4.1 Security on the distributed approach

As pointed out in the introduction, in [8] the authors proposed a "man in the middle attack" against the Euclid's approach that is easily avoided, as for any cryptosystem, by adding some information that provides authentication to distributed messages. In [7] an authentication protocol associated with the key distribution protocol based on the Euclid's approach was introduced. In [8] the authors also show an attack on this authentication protocol using a multiple of the product L_j that can be calculated by any member in G_j. This was addressed in [1] in several manners, but one of them is specially appropriated to the distributed situation. In that case (cf. [1, Section III]), it was shown that the attack, developed by a legal user, can be detected by users whose private information is less than a determined bound, namely some random information that is selected by the attacker. The easiest way to avoid this is, as noted in that case, to assign the group manager a prime p_j that is less than any other prime $p_{j,i}$ in group G_j. In this way, the group manager will detect the attack without making the Server intervene in the authentication process of internal messages in group G_j.

The same attack can be developed at a higher level by one of the group managers. However, there will exist one of them, that one holding the least prime of all p_j, that will detect the attack. In case groups corresponds to groups with a different status in the hierarchy, what is advisable to avoid this attack is that the group manager of the group with highest priorities holds this detached prime and that the other primes are distributed from the least to the highest in the inverse hierarchy of groups, i.e., those managers corresponding to groups with higher priorities will have the least primes.

Acknowledgements

First contributing author is supported by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117), and Junta de Andalucía (P08-TIC-3518), in part financed by the European Regional Development Fund (ERDF). Third contributing author is supported by the Spanish Ministry of Science and Innovation (TEC2009-13763-C02-02) and Junta de Andalucía (FQM 0211).

References

[1] N. Antequera and J.A. Lopez-Ramos, *Remarks and countermeasures on a cryptoanalysis of a secure multicast protocol*, Proceedings of 7th International Conference on Next Generation Web Services Practices, Salamanca 2011, Salamanca (Spain) (2011) 201–205.

[2] K.-C. Chan and S.-H.G. Chan, *Key management approaches to offer data confidentiality for secure multicast*, IEEE Network **17**(5) (2003) 30-39.

[3] G. Chiou and W. Chen, *Secure broadcasting using the secure lock*, IEEE Trans. Softw. Eng. **15**(8) (1989) 929–934.

[4] P.S. Kruus and J. P. Macker, *Techniques and issues in multicast security*, Proceedings of Military Communications Conference, MILCOM (1998) 1028–1032.

[5] B. Liu, W. Zhang and T. Jiang, *A Scalable Key Distribution Scheme for Conditional Access System in Digital Pay-TV System*, IEEE Consumer Electronics **50**(2) (2004) 632–637.

[6] S. Mittra, *Iolus: A framework for scalable secure multicasting*, Proceedings of the ACM SIGCOMM 27(4) (New York, Sept.) ACM, New York (1997) 277288.

[7] J.A.M. Naranjo, N. Antequera, L.G. Casado and J.A. Lopez-Ramos, *A suite of algorithms for key distribution and authentication in centralized secure multicast environments*, J. Comp. Appl. Math. **236**(12) (2012) 3042–3051.

[8] A. Peinado and A. Ortiz, *Cryptanalysis of Multicast Protocols with Key Refreshment Based on the Extended Euclidean Algorithm*, Proceedings of CISIS 2011, Lecture Notes in Computer Sciences, 6694 (2011) 177-182.

[9] S. Rafaeli and D. Hutchison, *A Survey of Key Management for Secure Group Communication*, ACM Computing Surveys **35**(3) (2003) 309–329.

[10] O. Scheikl, J. Lane, R. Boyer and M. Eltoweissy, *Multi-level secure multicast: the rethinking of secure locks*, Parallel Processing Workshops, 2002. Proceedings. International Conference (2000) 17–24.

[11] D. Wallner, E. Harder and R. Agee, *Key management for multicast: Issues and architectures*, RFC 2627 (1999).

[12] S. Zhu and S. Jajodia, *Scalable group key management for secure multicast: A taxonomy and new directions*, Network Security, H. Huang, D. MacCallum and D.-Z. Du (eds.) Springer, United States, (2010) 57–75.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Linking formal and informal ubitiquous learning schemes using m-learning and social networking

J.A. Álvarez Bermejo[1], L.J. Belmonte Ureña[2] and C. Bernal Bravo[1]

[1] *Centro de Investigación de Comunicación y Sociedad (CySOC), Universidad de Almería*

[2] *Departamento de Economía Aplicada, Universidad de Almería*

emails: jaberme@ual.es, lbelmont@ual.es, cbernal@ual.es

Abstract

Recent studies are focusing on how social networks impact the learning process and how students organize themselves to face collaborative tasks via these networks, as well as their impact on the learning outcomes of the students. In a number of these studies, learning social aspects are analyzed, showing, among other issues of interest, that participating in social networks positively affects students' self-esteem. This paper proposes to seize social networking through the usage of conceptual maps designed for university courses. These maps are enriched with with links to related courses, resources, social networks, provided in a way that smartphones and tablets can manage it.

Key words: mobile learning, ubiquitous learning, social networking, conceptual maps

1 Introduction

The human being is inherently a social entity and as such depends on the rest of individuals of the community to develop. The human being needs to be part of a group (family, team, etc.) and the fact of not belonging to one is a handicap for the individual identity. As the person develops and interacts, his/her group is particularized and customized with new contacts. A person then, creates a network of contacts collected from every group that this person encounters during a life cycle (work, university, etc.). The major disadvantage of these groups is that they are disjoint, for it is rare—and time consuming—to find points in common between individuals in order to build an homogeneous net. Therefore, the individual needs the group and the group is built upon the individual.

The appearance of the Web 2.0 as a phenomenon where anyone of us can be a producer and a consumer of content has been the catalyst for on-line communities, which turns the Internet into the necessary glue between the mentioned counterparts. From a social perspective, a social network provides paths to make contacts, to become part of a group, and not only that, to be an outstanding member. There are many studies devoted to social networks' impact on learning processes. Some also address how students organize themselves to face collaborative tasks via these networks, as well as their impact on the learning outcomes [3]. In [6] learning social aspects are analyzed, showing, among other issues of interest, how participating in social networks positively affects students' self-esteem. This paper presents a tool that seizes the interesting learning benefits that social networks offer, together with the enriched conceptual maps designed to be used in smartphones and tablets.

2 Social Networking and informal learning

By using social networks we find an interesting path to avoid role based barriers, which are the most evident handicaps for learning. Two different types of social networks can be defined according to the structures (formal and informal) of information exchange in the organization, formal networks and informal networks. Each network has its drawbacks and strong points. We think that a convenient architecture combines both networks, the proportion and the interaction is still a factor to be considered in each case. For instance, formal networks are composed by individuals aggregated according to their specialization. This leads to structural holes (unconnected clusters) like at the University, where teachers generate contents and students can only consume it. Only those individuals who have access to different clusters in the network are able to build bridges that remove such structural gaps and can generate proposals based on their broad perspectives. In informal networks the aggregations are created through trust relationships that arise between members of the same organization. The longer two individuals are emotionally involved, the more time and effort are both willing to invest for the benefit of each other, whenever collaborative barriers are overcome. This includes the collaborative effort, so poorly valued in higher education. Removing the role-based barrier would favor a learning process that could complement the traditional learning process in the University. The social networks that should be part of the learning process are both formal and informal networks. In any organization and in the University in particular, we can find two different ways to build knowledge, the formal network infrastructure and the informal one. We enhance the informal infrastructure as an useful complement to the formal infrastructure. In our proposal, the courses are informally related and the student does not feel forced to learn. Barriers among courses are removed by integrating concepts from each course. Students are demanding channels through which they can freely speak their minds. Places where they can meet other students and establish

relationships that would be impossible to create otherwise. As a result of this, they are motivated to capture and exploit more information and in a more convenient way, while participation in work groups also improves. Conventional social networks are a constant in the life of the university student. In this sense, social networks are already integrated in campus. It is true that these networks tend to weight up leisure versus intellectual tasks. But if general purpose social networks are properly used, they can motivate the student to enhance his/her cultural perspective. Gaining recognition in this virtual community is an asset that everyone is looking for. The way to reach this resides in the way the student manages and shares knowledge. Here is where another hot topic appears: is it common that a student shares the knowledge that makes him/her unique? It is obvious that sharing knowledge may make a student lose his/her privileged position, but this investment is proved to have a highly valued social counterpart [1]. According to certain theories of social learning, there are three elements that affect the learning of the individual: the individual itself, his/her peers and the situation. We are trying to act on these three, by motivating the individual, connecting him to others and letting him do-it anytime, anywhere. Learning can thus be understood as a social process in which students interact with their peers. Learning, on the other hand, depends on social context, such as the student's observations and interactions with others. Then, the success in achieving learning goals depends on the participation in social settings and on the way the group approaches the situation in which knowledge must be generated. In social networks, students have an extraordinary ability to express themselves, to establish relationships with others and to interact with them (without temporal nor geographical restrictions) in order to cover the learning needs that arise.

2.1 Personal centers for innovation

Social Networking can also be used as a knowledge backup. In [**?**] we can find an illustrative example of an Accenture employee who got an offer for a position at another company. As no access to the Accenture's knowledge platform was available, this worker decided to open a *hub for his personal innovation*. He got connections to peer workers and new paths to discover products and projects were opened. To get this, one need to invest too much time and efforts to get a real payoff. All this came down to a useful digital trade mark for him.This digital identity made him be invited by relevant blogs and on-line firms to get his opinion on diverse points. From scratch, we think that promoting students to build their own digital identity in order to reach a personal trade mark that allow them to gain momentum is a must for educational instructors. In this sense, the work presented here tries to enable any student to access educational content from everywhere at anytime. To provide ubiquitous content, a framework to build conceptual maps linked to content in the Internet (to be downloaded at demand) is required. Upcoming sections are devoted to demonstrate how smartphones and tablets are perfect tools to learn. Section 4 shows the solution that

 ISBN:978-84-615-5392-1

we propose to achieve this goal.

3 Mobile Learning : Smartphones and tablets

We can define it in a very simplistic manner: It is e-learning through mobile computational devices. Another feasible definition, combined with distance education: Mobile learning is learning with a specific device, at any time, any place. The device must be capable of presenting learning content and providing wireless two-way communication between teacher(s) and student(s). Adding Social Networking to the equation, Mobile learning could be understood as a new way of learning using mobile networks and tools,with the aim of expanding digital learning channel to get educational information, resources and services anytime, anywhere.

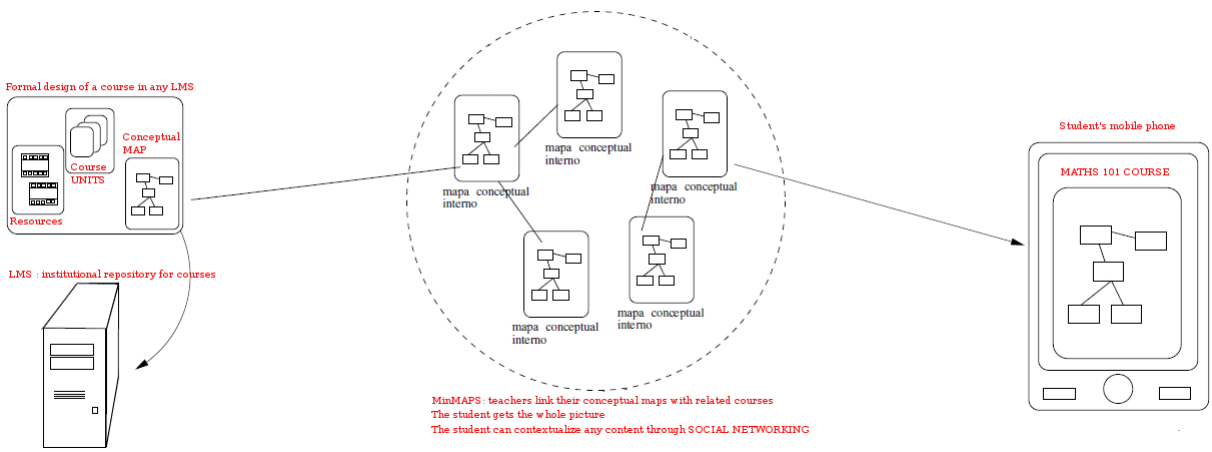
People do not access to social networks unless they do see a clear advantage (utility principle). As smartphones offer more and more facilities such as accessing the Internet through 3G networks, people is accessing social networks through their smartphones to *do useful tasks instead of wasting time doing nothing while waiting the bus, i.e.*. So, two aspects, useful for learning, can be combined here: smartphones (or tablets) and the access to social networking. Studies such as the one conducted by Dough Vogel et al in [5] shows that mobile learning is shown to be useful to learn. Digital natives are, therefore, at the center of their own personal learning environment (e.g., smartphones, tablets, iPods, etc.) and in contrast to this, limited attention has been given to the impact on learning of mobile devices and associated applications in education. This is why MindMaps is presented here. Usage of mLearning and social networking is further correlated with performance as exhibited on exams (as our we can tell from our experiences). Particular attention is given to the pattern of mobile learning application use, e.g., for exploration of alternatives. We can enhance learning motivation by emphasizing the importance and applicability of the material and by trying to connect the material to students' intrinsic motives. They particularly note that learning motivation is likely to be greater if a student feels a particular class is consistent with their interests and with personally satisfying career goals. However, learning motivation is malleable and can change over time. In mobile learning environment, students not only study in the classroom or computer but also in any place thanks to their mobile devices. In this paper, we build a learning model based on a tool to navigate through connected conceptual maps, where items are linked to social networks, to University LMS, and enriched with links to resources. The teacher can create conceptual maps for his courses and connect them to other courses' maps, can provide meta-information for each concept so the student can go for alternative information in social networks, the student can also get only summarized resources into his mobile to study the course while waiting the bus, i.e. We are using both devices, smartphones as the primary device from the students' side and tablets pcs (TPC) for lecturers. Studies like the one done in [4] states that using the tablet

PC as a lecturing device offers the instructor a new set of tools upon which their teaching can be based as it provides the instructor with an extended set of educational tools. From the perspective of the audience, the TPC allows the instructor to maintain a connection with their students. In [2] it is shown how this learning model based in smartphones and tablet pcs, enable a way of knowledge sharing, with the tool that we present in this paper, we intend to extend these characteristics to certain social networks where learning and intelligence sharing can be boosted.

4 MindMaps. Our solution

It has been proven that smartphones and tablets are a new chance for students and teachers to seize a new methodology that seems to be better than many others because of its ubiquitous and how the content is summarized to be distributed to such devices. Also, connectivity in these devices is wisely used to access on-line resources when idle time slots occurs, so time is now better employed than before. The tool we are presenting here is designed to seize the advantages underlined for smartphones and tablets in the learning context, but also to let the students explore and get extra motivation for what they are learning. The application is divided into two sections, one *for faculties* and the other one *for students*.

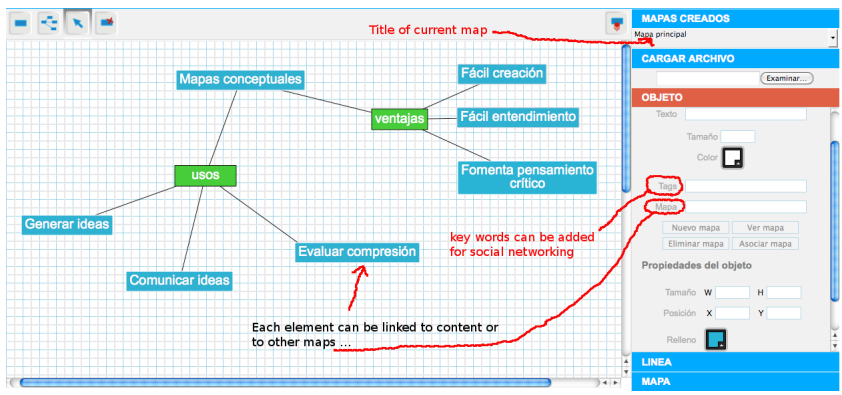
Figure 1: Using MindMaps from smartphones



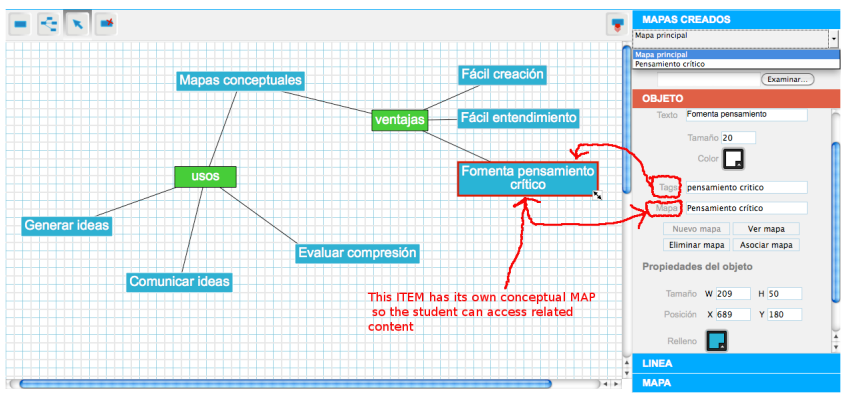
The student uses his smartphone to download maps, and navigate through each concept, in formal or informal networks. The teacher can also leave PDF documents, videos, etc. associated to each concept to provide further information. The teacher creates his own

maps, and leave them in the institutional LMS (learning management system) used by the students. In Fig. 1 the connection to the course content designed by the teacher is shown. Fig. 2 shows the side of MindMaps for teachers. Each concept can be enriched with extra content or links to other maps or social networks (in this case using the tags that the teacher provides).

Figure 2: Creating Conceptual Maps and Linking them to other maps or content (in LMS or in Social Networks)



(a) Creating the Map: the teacher must provide a title and meta-information used to link the current map with other resources

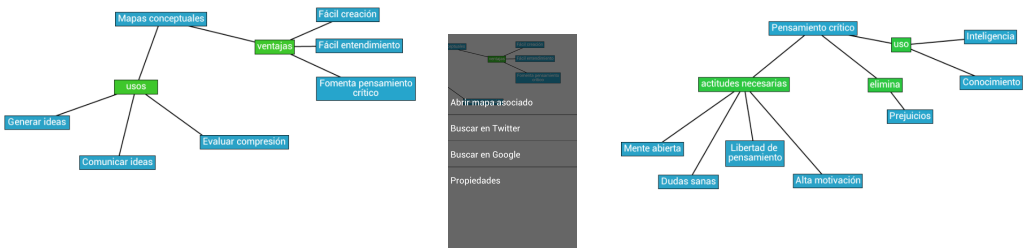


(b) MindMaps allow you to link items to other maps (in the institutional LMS) or to Social Network content. This map is later saved as an exportable file for smartphones or tablets

In Fig. 3 the map is displayed in the smartphone as shown in Fig. 3(a). When the student interacts with the map, on a certain concept, a context menu appears to let him download the resources provided by the teacher, or by explore social networks for that

concept, or link to other maps (from the same course or from other courses that are related). The student is always provided with a track of the maps he followed to be always contextualized.

Figure 3: MindMaps are accessed by students anywhere, anytime by simply accessing or downloading the map file form the LMS server. When an item is touched, a context menu(b) appears allowing the navigation proceed to other maps, social networks or content.



(a) Main Map for a course (b) Menu (c) Submap accessed through the context menu.

5 Conclusions

In general, our results provide some support for our research model, see Fig.1, and associated postulates. Empirically, those students who were motivated to use the mobile applications tended to achieve higher levels of performance as indicated on their proof exam. This is independent of whether the motivation was intrinsic or extrinsic noting that fewer than half of the students could download and use the application as a test group. The final exam included a question to gather information about their real comprehensive knowledge (*Q10. Please explain which is the relationship between units of this course and give reasons why the units are ordered in such way*): those who used MindMaps answered correctly, those who didn't, provided obvious answers. We can conclude that those who received extra knowledge, that contextualized it and were motivated to do so, gained extra skills from this course that is to be useful for the courses related (and they are aware). From the technical side, we could check that the tablet PC offers many advantages over the traditional blackboard approach to improve the overall learning experience of the students. It enabled the instructor to engage students more thoroughly through the use of added multimedia content.

Acknowledgements

Our sincere gratitude to José Luis López López, for his valuable work and advices during the development of MindMaps. José Luis López López is a student that extended and improved MindMaps for his Bachelor's degree.

References

[1] Peter Marks, Peter Polak, Scott McCoy, and Dennis Galletta. Sharing knowledge. *Commun. ACM*, 51(2):60–65, February 2008.

[2] N. Matsuuchi, T. Yamaguchi, H. Shiba, K. Fujiwara, and K. Shimamura. Collaborative learning system providing interactive lesson through tablet pcs on wlan. In *Information and Telecommunication Technologies, 2008. APSITT. 7th Asia-Pacific Symposium on*, pages 47 –51, april 2008.

[3] Zhengzheng Pan. Trust, influence, and convergence of behavior in social networks. *Mathematical Social Sciences*, 60(1):69–78, 2010.

[4] M. Stickel. Impact of lecturing with the tablet pc on students of different learning styles. In *Frontiers in Education Conference, 2009. FIE '09. 39th IEEE*, pages 1 –6, oct. 2009.

[5] Doug Vogel, David M. Kennedy, Kevin Kuan, Ron Kwok, and Jean Lai. Do mobile device applications affect learning? In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, page 4, jan. 2007.

[6] Angela Yan Yu, Stella Wen Tian, Douglas Vogel, and Ron Chi-Wai Kwok. Can learning be virtually boosted? an investigation of online social networking impacts. *Comput. Educ.*, 55(4):1494–1503, December 2010.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Effects Of Diffusion And Transmembrane Potential On Current Through Ionic Channels

Daniele Andreucci[1], Dario Bellaveglia[1], Emilio N.M. Cirillo[1] and Silvia Marconi[1]

[1] *Department of Basic and Applied Sciences For Engineering*
Section of Mathematics,
Sapienza University of Rome Italy

emails: daniele.andreucci@sbai.uniroma1.it,
dario.bellaveglia@sbai.uniroma1.it, emilio.cirillo@uniroma1.it,
silvia.marconi@sbai.uniroma1.it

Abstract

We report on some recent results on the modeling of ion exchange through cell membranes, with special attention to the issue of selection of preferred ionic species in the presence of different transmembrane voltages and different concentrations in the cytosol.

Key words: ionic channel, potassium channel, gating, selectivity

1 Introduction

Biological literature has been dealing with potassium currents across cell membranes for a long time (see, for instance, the reviews [3, 10]). The ubiquitous presence and the importance of ionic channels selecting potassium for transmembrane exchange are by now well established.

In the modeling of the large variety of existing ionic channel types it is generally accepted that they all form selective pores in the cell membrane which are able to switch between an open state and a closed state. The open state is the one, obviously, that allows for permeation of a selected ionic species (potassium in K^+–channels).

Such change of state, which is called *gating*, is stochastical in character. Its relation to *selectivity*, i.e., the ability of the channel to allow the flux of a particular ionic species, is

not yet completely understood. The way in which either one is achieved can be different from channel to channel [6].

Some models, see [5, 7, 8, 9] describe to some extent the dynamics of ion permeation through the selectivity filter of the channel. In this kinetic approach the concentration of the ionic species in the cell is modeled as a constant parameter. In [2], following [11], we introduced a model where the channel is lumped to a two state stochastic point system, but the interaction between the dynamics of the ions inside the cell and that of the selectivity filter itself is taken into account. That is to say, the channel is seen as a part of the cell more than as an isolated structure. In that paper both an analytical and Monte Carlo study showed the possibility to achieve gating via selection. A continuos version of the model has been investigated in [1].

Here we deal with a modification of that model, aimed at taking into account the effect of an external voltage difference through the cell membrane. For simplicity we confine ourselves to a one-dimensional implementation of the model, where exact calculations can be carried out for the stochastical quantities. Our purpose is to predict the behavior of the current–voltage curves. The model is then defined to mimic the three effects that seem to be the most relevant in the process: (i) diffusion of the ions inside the cell; (ii) dynamics of the selectivity filter; (iii) dynamics of the ions inside the channel.

We compare the current–voltage behaviors predicted by our model with those measured in experiments [4] and find them to be in very good agreement.

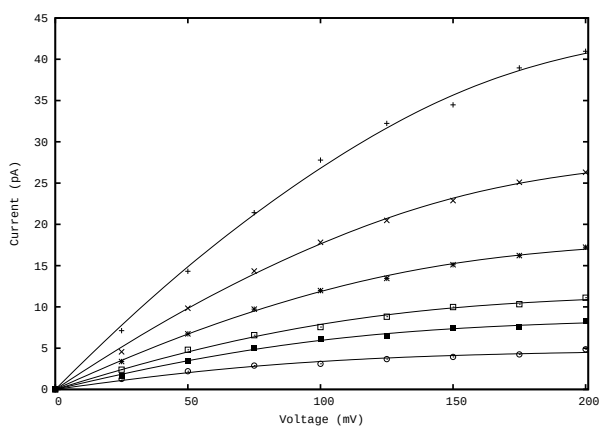


Figure 1: Potassium flux at different concentrations. Experimental measures (symbols) and model predictions (curves) at the concentrations of 20, 50, 100, 200, 400, 800 mM (from bottom to top).

 ISBN:978-84-615-5392-1

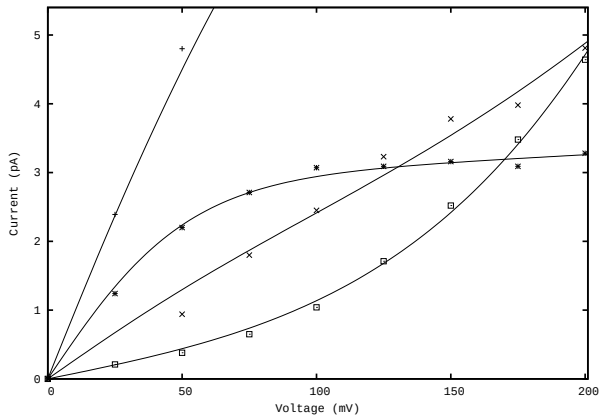


Figure 2: Fluxes of different ionic species. Experimental measures (symbols) and model predictions (curves) at the concentration of 100 mM for the species Rb^+ $\square$, NH_4^+ $\times$, Tl^+ $*$, K^+ $+$.

2 Results

In Figure 1 we compare the experimentally measured current of K^+ ions with the one predicted by our model according to the relation

$$I = S_I f_K \,, \tag{1}$$

where f_K represents the flux (in number) of K^+ ions, and S_I is a parameter of the model connected to the diffusivity of ions in the cytosol; we omit the explicit definition of f_K. In reading the results in Figure 1 one should keep in mind that the different parameters appearing in our model have been tuned to fit the curve corresponding to concentration 800 mM there, and that the other curves have been fitted by using only S_I in (1) and the probability p that the channel is open. Similar results can be obtained by using only either one of S_I, p.

In Figure 2 we report the predictions of the model for currents of other (selected by the channel) ionic species: rubidium Rb^+, thallium Tl^+, ammonium NH_4^+, and again potassium K^+, all at the same concentration of 100 mM.

References

[1] D. Andreucci and D. Bellaveglia, *Permeability of Interfaces with Alternating Pores in Parabolic Problems*, Asymptotic Anal. (in press).

[2] D. Andreucci, D. Bellaveglia, E. N. M. Cirillo and S.Marconi, *Monte Carlo study of gating and selection in potassium channels*, PhysṘeview E **84** (2011) 021920 1–13.

[3] D. Fedida and J. C. Hesketh, *Gating of voltage-dependent potassium channels*, Prog. Bio. Mol. Biology **75** (2001) 165–199.

[4] M. LeMasurier, L. Heginbotham and C. Miller, *KcsA: It's a Potassium Channel*, J. GenṖhysiol. **118** (2001) 303–313.

[5] S. Mafé and J. Pellicer, *Ion conduction in the KcsA potassium channel analyzed with a minimal kinetic model*, PhysṘeview E **71** (2005) 022901 1–4.

[6] C. Miller, *Ionic hopping defended*, J. GenṖhysiol. **113** (1999) 783–787.

[7] P. H. Nelson, *A permeation theory for single-file ion channels: Corresponding occupancy states produce Michaelis–Menten behavior*, J. ChemṖhys. **117** (2002) 11396–11403.

[8] P. H. Nelson, *Modeling the concentration-dependent permeation modes of the KcsA potassium ion channel*, PhysṘeview E **68** (2003) 061908 1–8.

[9] P. H. Nelson, *Modeling the concentration-dependent permeation modes of the KcsA potassium ion channel*, J. ChemṖhys. **134** (2011) 165102 1–13.

[10] M. Recanatini, A. Cavalli and M. Masetti, *Modeling hERG and its Interactions with Drugs: Recent Advances in Light of Current Potassium Channel Simulations*, ChemMedChem **3** (2008) 523–535.

[11] A. M. J. VanDongen, *K channel gating by an affinity-switching selectivity filter*, PNAS **101** (2004) 3248–3252.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A simple meta-epidemic model

Marika Barengo[1], Isabella Iennaco[1] and Ezio Venturino[1]

[1] *Dipartimento di Matematica "Giuseppe Peano", Università di Torino,*
via Carlo Alberto 10, 10123 Torino, Italy

emails: marika1288@libero.it, isabella.iennaco@gmail.com,
ezio.venturino@unito.it[1]

Abstract

In this paper we present and analyse a simple model for disease transmission among two different geographical locations. Our goal is to unveil the role of the migration coefficients on the disease evolution and to understand what happens to the system on the whole if some external disturbances modify the system topology or the individuals habits. The analysis discusses these modifications as possible tools for disease eradication.

Key words: epidemics, disease transmission, migrations
MSC 2000: AMS codes 92D30

1 Introduction

The role of diseases in shaping populations dynamics is widely recognized. Mathematical epidemiology has progressed in the past century to provide the epidemiologists with instruments apt to forecast the disease evolution and take suitable measures agains their propagation. In fact, it is mainly due to mathematical results that in 1980 the WHO has discontinued worldwide the vaccination against smallpox, thereby declaring this disease, which has affected humanity for centuries, eradicated.

In this paper we consider a simple system in which two patches are present. One population occupies them both, and can migrate from one to the other one. We investigate the stable states and discuss how they are modified when communications between patches are interrupted and when only some of the individuals are able to migrate.

[1]This paper was completed and written during a visit of the third author at the Max Planck Institut für Physik Komplexer Systeme in Dresden, Germany. The author expresses his thanks for the facilities provided.

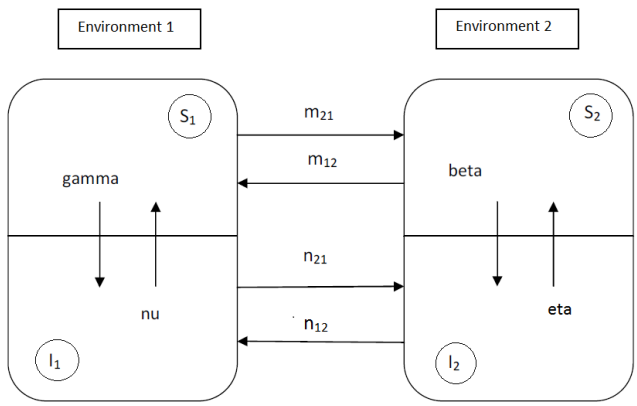


Figure 1: Scheme of the model

2 Model formulation

We consider two compartmentalized environments containing each individuals of a population, subject to a transmissible disease, so that they can be either susceptible or infected, S_i and I_i, $i = 1, 2$, respectively.

We assume that m_{ij} is the susceptibles' migration rate from environment j to environment i, with $i, j = 1, 2$. Similarly, let n_{ij} denote the migration rate of infected from environment j to environment i, with $i, j = 1, 2$. In principle they are different as migrations, entailing in general an effort, are perhaps less possible for infected, that in view of the disease, are weaker individuals. We assume the disease to be recoverable, and that the recovery rate may be influenced by the environment. Thus let ν and η denote these rates. We also assume that ecological conditions in the two environments are in principle different, thereby affecting the disease propagation and its recovery. Thus, the disease transmission rate in the first environment is γ while it is β in the second one. Overall, then, we have two copies of the classical SIS model linked via migrations. A schematic representation of the situation is graphically illustrated in Figure 1.

Using mass action incidence to model disease transmission, the model thus becomes

$$\begin{aligned} \dot{S}_1 &= -\gamma S_1 I_1 + \nu I_1 - m_{21} S_1 + m_{12} S_2 \\ \dot{I}_1 &= \gamma S_1 I_1 - \nu I_1 - n_{21} I_1 + n_{12} I_2 \\ \dot{S}_2 &= -\beta S_2 I_2 + \eta I_2 + m_{21} S_1 - m_{12} S_2 \\ \dot{I}_2 &= \beta S_2 I_2 - \eta I_2 + n_{21} I_1 - n_{12} I_2 \end{aligned} \tag{1}$$

 ISBN:978-84-615-5392-1

Further, like in the classical epidemic models, [1], we make the assumption that the total population in the two enviroments has a constant value N, so that the recruitment rate of newborns matches the mortality rate, natural plus disease-related. Thus, $S_1 + I_1 + S_2 + I_2 = N$, and this is reflected by the fact that on summing the equations in (1), we find $\dot{S}_1 + \dot{I}_1 + \dot{S}_2 + \dot{I}_2 = 0$. We can thus express S_1 as function of the remaining populations, $S_1 = N - I_1 - S_2 - I_2$ and therefore eliminate it from the system. We obtain then the simplified system

$$\begin{aligned} \dot{I}_1 &= \gamma\,(N - I_1 - S_2 - I_2)\,I_1 - \nu I_1 - n_{21}I_1 + n_{12}I_2 \\ \dot{S}_2 &= -\beta S_2 I_2 + \eta I_2 + m_{21}\,(N - I_1 - S_2 - I_2) - m_{12}S_2 \\ \dot{I}_2 &= \beta S_2 I_2 - \eta I_2 + n_{21}I_1 - n_{12}I_2 \end{aligned} \tag{2}$$

2.1 Analysis of the equilibria

For the system (2) there are only two possible equilibria, in addition to the origin $E^{(0)} = (0, 0, 0)$, which is an equilibrium only if the total population vanishes, as consequence of the second equation (2). These points are the disease-free point $E^{(1)} = (0, S_2^{(1)}, 0)$ and the coexistence equilibrium, with endemic disease, $E^{(2)} = (I_1^{(2)}, S_2^{(2)}, I_2^{(2)})$. The former is always feasible, with

$$S_2^{(1)} = N\mathscr{M}_0, \quad \mathscr{M}_0 \equiv \frac{m_{21}}{m_{21} + m_{12}}.$$

To find the components of the coexistence equilibrium, we sum the second and the third equilibrium equations thereby replacing the second one, and solve the third one to get the following algebraic system

$$\begin{cases} 0 = \gamma N I_1 - \gamma I_1^2 - \gamma S_2 I_1 - \gamma I_2 I_1 - \nu I_1 - n_{21}I_1 + n_{12}I_2 \\ 0 = n_{21}I_1 + N m_{21} - m_{21}I_1 - m_{21}S_2 - m_{21}I_2 - m_{12}S_2 - n_{12}I_2 \\ I_1 = \dfrac{1}{n_{21}}(n_{12} + \eta - \beta S_2)I_2 \end{cases} \tag{3}$$

Letting

$$\mathscr{M}_1 \equiv \frac{n_{21}}{m_{12} + m_{21}}, \quad \mathscr{M}_2 \equiv \frac{n_{12}}{m_{12} + m_{21}}, \quad \mathscr{M}_3 \equiv \frac{m_{12}}{m_{12} + m_{21}},$$

from the second equation we find

$$S_2 = \frac{N m_{21} - I_1 m_{21} + n_{21}I_1 - m_{21}I_2 - n_{12}I_2}{m_{12} + m_{21}} = \mathscr{M}_0(N - I_1 - I_2) + \mathscr{M}_1 I_1 - \mathscr{M}_2 I_2,$$

which immediately gives a necessary feasibility condition, namely

$$\mathscr{M}_0 N > (\mathscr{M}_0 - \mathscr{M}_1)I_1 + (\mathscr{M}_2 + \mathscr{M}_0)I_2. \tag{4}$$

 ISBN:978-84-615-5392-1

Back substitution into (3) gives two equations in I_1 e I_2,

$$EI_1^2 + 2FI_1I_2 + 2GI_1 + 2HI_2 = 0, \tag{5}$$
$$AI_2^2 + 2BI_1I_2 + 2CI_2 + 2DI_1 = 0, \tag{6}$$

where we set $E \equiv -\gamma(\mathscr{M}_1 + \mathscr{M}_3)$, $F \equiv \frac{1}{2}\gamma(\mathscr{M}_2 - \mathscr{M}_3)$, $G \equiv \frac{1}{2}(\gamma N\mathscr{M}_3 - n_{21} - \nu)$, $H \equiv \frac{1}{2}n_{12}$, $A \equiv \beta(\mathscr{M}_0 + \mathscr{M}_2)$, $B \equiv \frac{1}{2}\beta(\mathscr{M}_0 - \mathscr{M}_1)$, $C \equiv \frac{1}{2}(n_{12} + \eta - \beta N\mathscr{M}_0)$ and $D \equiv -\frac{1}{2}n_{21}$.

Equations (5) and (6) identify two conic sections, both through the origin and respectively intersecting the coordinate axes at the points $(-2GE^{-1}, 0)$ and $(0, -2CA^{-1})$. The former conic for

$$\frac{-n_{12}(\mathscr{M}_1 + \mathscr{M}_3)}{\mathscr{M}_2 - \mathscr{M}_3} \neq \gamma N\mathscr{M}_3 - n_{21} - \nu, \quad n_{12} \neq m_{12} \tag{7}$$

is a hyperbola with center at $(-HF^{-1}, -(FG - EH)F^{-2})$ and asymptotes

$$x = -\frac{H}{F}, \qquad x + \frac{H}{F} = -\frac{2F}{E}\Big(y + \frac{FG - EH}{F^2}\Big).$$

In case $n_{12} = m_{12}$ it becomes a parabola with axis $x = -\frac{G}{E}$ and vertex $(-GE^{-1}, G^2(2EH)^{-1})$, with negative height there since $EH = -n_{12}\gamma(\mathscr{M}_1 + \mathscr{M}_3) < 0$. The degenerate case

$$\frac{-n_{12}(\mathscr{M}_1 + \mathscr{M}_3)}{\mathscr{M}_2 - \mathscr{M}_3} = \gamma N\mathscr{M}_3 - n_{21} - \nu, \quad n_{12} \neq m_{12} \tag{8}$$

leads instead to the two straight lines

$$x = -\frac{2F}{E}y, \qquad x + \frac{H}{F} = 0. \tag{9}$$

If

$$-\frac{n_{21}(\mathscr{M}_0 + \mathscr{M}_2)}{\mathscr{M}_0 - \mathscr{M}_1} \neq n_{12} + \eta - \beta N\mathscr{M}_0, \quad n_{21} \neq m_{21} \tag{10}$$

also (6) is a hyperbola, with center at $(-(BC - DA)B^{-2}, -DB^{-1})$ and asymptotes

$$x + \frac{BC - DA}{B^2} = -\frac{A}{2B}\Big(y + \frac{D}{B}\Big), \qquad y = -\frac{D}{B}.$$

For $n_{21} = m_{21}$ it becomes a parabola with axis $y = -CA^{-1}$ and vertex $(C^2(2DA)^{-1}, -CA^{-1})$. It also has negative height at the vertex in view of the fact that $2DA = -n_{21}\beta(\mathscr{M}_0 + \mathscr{M}_2) < 0$. The degenerate case is obtained if

$$-\frac{n_{21}(\mathscr{M}_0 + \mathscr{M}_2)}{\mathscr{M}_0 - \mathscr{M}_1} = n_{12} + \eta - \beta N\mathscr{M}_0, \quad n_{21} \neq m_{21}, \tag{11}$$

 ISBN:978-84-615-5392-1

giving thus the two straight lines

$$y = -\frac{D}{B}, \qquad x = -\frac{A}{2B}y. \tag{12}$$

These conic sections will always intersect in the first quadrant, providing a feasible equilibrium if (4) is satisfied, but for the degenerate cases for which $HF > 0$ and $DB > 0$ (or equivalently $FE < 0$ and $AB < 0$). In the degenerate cases, we can explicitly write the equilibrium:

i) $-\frac{H}{F} > 0, -\frac{D}{B} > 0$ imply $E^{(2)} = \left(\frac{-n_{12}}{\gamma(\mathscr{M}_2-\mathscr{M}_3)}, \frac{n_{21}}{\beta(\mathscr{M}_0-\mathscr{M}_1)}, \frac{n_{12}(\mathscr{M}_0-\mathscr{M}_1)}{\gamma(\mathscr{M}_2-\mathscr{M}_3)} + \frac{n_{12}}{\beta} + \frac{\eta}{\beta}\right)$;

ii) $-\frac{H}{F} > 0, -\frac{D}{B} < 0$ imply $E^{(2)} = \left(\frac{-n_{12}}{\gamma(\mathscr{M}_2-\mathscr{M}_3)}, \frac{(\mathscr{M}_0-\mathscr{M}_1)n_{12}}{\gamma(\mathscr{M}_0+\mathscr{M}_2)(\mathscr{M}_2-\mathscr{M}_3)}, N\mathscr{M}_0 + \frac{(\mathscr{M}_0-\mathscr{M}_1)(n_{12}-n_{21})}{\gamma(\mathscr{M}_2-\mathscr{M}_3)}\right)$;

iii) $-\frac{H}{F} < 0, -\frac{D}{B} > 0$ imply $E^{(2)} = \left(\frac{(\mathscr{M}_2-\mathscr{M}_3)n_{21}}{(\mathscr{M}_1+\mathscr{M}_3)(\mathscr{M}_0-\mathscr{M}_1)}, \frac{n_{21}}{\beta(\mathscr{M}_0-\mathscr{M}_1)}, \frac{(\mathscr{M}_2-\mathscr{M}_3)n_{21}}{(\mathscr{M}_1+\mathscr{M}_3)\beta} + \frac{n_{12}}{\beta} + \frac{\eta}{\beta}\right)$.

From the equilibria that we just found, we observe that it is not possible that the disease becomes a pandemic in the environment, i.e. it invades the whole population and the susceptibles totally disappear. This represents a good result from the epidemiological point of view.

2.2 Stability

For the stability analysis we write the Jacobian $J = (J_{ik})$, $i, k = 1, 2, 3$ of the system (2):

$$\begin{bmatrix} -2\gamma I_1^{(i)} + \gamma(N - I_2^{(i)} - S_2^{(i)}) - \nu - n_{21} & -\gamma I_1^{(i)} & -\gamma I_1^{(i)} + n_{12} \\ -m_{21} & -\beta I_2^{(i)} - m_{21} - m_{12} & -\beta S_2^{(i)} + \eta - m_{21} \\ n_{21} & \beta I_2^{(i)} & \beta S_2^{(i)} - \eta - n_{12} \end{bmatrix} \tag{13}$$

For the origin $E^{(0)}$, one eigenvalue can be immediately evaluated, $-(m_{12} + m_{21}) < 0$, the remaining ones are the roots of a quadratic stemming from a suitable 2×2 submatrix $\widehat{J}_0$ of J. The Routh-Hurwitz conditions reduce to

$$-\mathrm{tr}(\widehat{J}_0) = \nu + \eta + n_{21} + n_{12} > 0, \quad \det(\widehat{J}_0) = \eta\nu + \nu n_{12} + \eta n_{21} > 0,$$

so that $E^{(0)}$ is always stable. Recall again, compare the discussion at the beginning of Section 2.1, that this means that the whole population is wiped out, including S_1 which do not explicitly appear in the model. This result is not good from the environmental point of view, since the population disappears completely, but it should be expected. In fact, if the population drops it will never be able to recover as no such specific mechanisms are present in the model.

 ISBN:978-84-615-5392-1

For $E^{(1)}$ the eigenvalues of (13) are $\lambda_1 = -m_{21} - m_{12} < 0$ and the roots of the quadratic

$$\begin{aligned}&\lambda^2 - \lambda(\gamma N - \gamma N \mathscr{M}_0 - \nu - n_{21} + \beta N \mathscr{M}_0 - \eta - n_{12})\\&+(\gamma N^2 \beta \mathscr{M}_0 - \eta\gamma N - n_{12}\gamma N - \gamma N^2 \beta \mathscr{M}_0^2 + \gamma N \mathscr{M}_0 \eta\\&+\gamma N \mathscr{M}_0 n_{12} - \nu\beta N \mathscr{M}_0 + \nu\eta + \nu n_{12} - \beta N \mathscr{M}_0 n_{21} + \eta n_{21}) = 0,\end{aligned}$$

for which the Routh-Hurwitz criterion gives the stability conditions

$$\begin{aligned}&N\mathscr{M}_0(\gamma - \beta) > \gamma N - \nu - n_{21} - \eta - n_{12},\\&N\mathscr{M}_0[\gamma N\beta(1 - \mathscr{M}_0) + \gamma(\eta + n_{12}) - \beta(\nu + n_{21})] > (\eta + n_{12})(\gamma N - \nu) - \eta n_{21}.\end{aligned} \tag{14}$$

A Hopf bifurcation should arise when the value of the parameter $\mathscr{M}_0$ crosses the critical value

$$\mathscr{M}_0^{\dagger} \equiv \frac{\gamma N - \nu - n_{21} - \eta - n_{12}}{N(\gamma - \beta)}. \tag{15}$$

Assuming at first $\gamma > \beta$, a feasible bifurcation arises only if the total population size is large enough, namely for

$$\frac{1}{\beta}(\nu + n_{21} + \eta + n_{12}) > N > \frac{1}{\gamma}(\nu + n_{21} + \eta + n_{12}).$$

Conversely, the result also holds for $\gamma < \beta$ if the above inequalities are all reversed. In spite of this theoretical result, our simulation have not been able to reveal limit cycles. We therefore conjecture that there might be some incompatibility among the above given conditions.

For the coexistence equilibrium $E^{(2)}$, the caracteristic equation is the cubic $\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$, for which the Routh-Hurwitz stability conditions are $a_0 > 0$, $a_2 > 0$ and $a_2 a_1 > a_0$. Explicitly, in terms of the Jacobian components, they become

$$\begin{aligned}&J_{22}^{(2)} J_{33}^{(2)} J_{11}^{(2)} < \beta I_2 J_{11}^{(2)} J_{23}^{(2)} + \gamma I_1 m_{21} J_{33}^{(2)} + n_{21}\gamma I_1 J_{23}^{(2)} + m_{21}\beta I_2 J_{13}^{(2)} + n_{21} J_{22}^{(2)} J_{13}^{(2)},\\&J_{11}^{(2)} + J_{22}^{(2)} + J_{33}^{(2)} < 0,\\&\beta I_2[J_{23}^{(2)} J_{22}^{(2)} + J_{23}^{(2)} J_{33}^{(2)} - J_{13}^{(2)} m_{21}] + \gamma I_1 m_{21}[J_{11}^{(2)} + J_{22}^{(2)}] + n_{21} J_{13}^{(2)}[J_{11}^{(2)} + J_{33}^{(2)}] < n_{21} J_{23}^{(2)}\gamma I_1\\&+2J_{22}^{(2)} J_{33}^{(2)} J_{11}^{(2)} + \left(J_{11}^{(2)}\right)^2 (J_{22}^{(2)} + J_{33}^{(2)}) + \left(J_{22}^{(2)}\right)^2 (J_{11}^{(2)} + J_{33}^{(2)}) + \left(J_{33}^{(2)}\right)^2 (J_{22}^{(2)} + J_{11}^{(2)}).\end{aligned} \tag{16}$$

For the parameters $N = 50$, $\gamma = 2.2$, $\beta = 0.05$, $\nu = 1.6$, $\eta = 3.6$, $n_{21} = 0.6$, $n_{12} = 0.5$, $m_{21} = 0.7$, $m_{12} = 0.8$, we find the coexistence equilibrium indeed at a stable state, see Figure 2.

3 The case of migration only in one direction

We assume now that migration from patch 2 into patch 1 is not possible, $m_{12} = n_{12} = 0$, for which $\mathscr{M}_0 = 1$, $\mathscr{M}_2 = \mathscr{M}_3 = 0$. The equilibria are here denoted by $\widetilde{E^{(k)}}$, $k = 0, \ldots, 3$.

 ISBN:978-84-615-5392-1

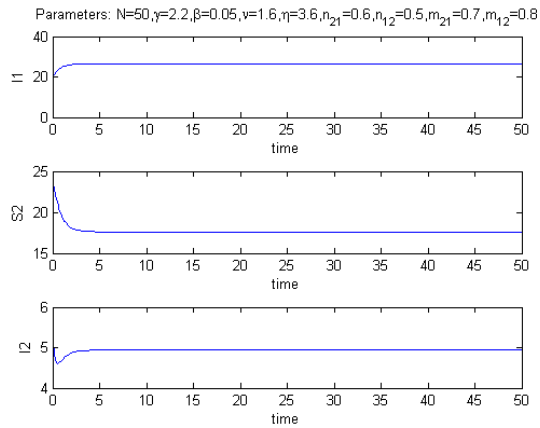


Figure 2: Coexistence at a stable state.

For $\widetilde{E^{(0)}} \equiv E^{(0)} = (0,0,0)$, the results of the stability analysis still hold, together with their ecological consequences.

For equilibrium $\widetilde{E^{(1)}}$, we now have $\widetilde{S_2^{(1)}} = N$, always feasible.

For this model an extra equilibrium arises, namely $\widetilde{E^{(3)}}$ with nonvanishing components

$$\widetilde{S_2^{(3)}} = \frac{\eta}{\beta}, \qquad \widetilde{I_2^{(3)}} = N - \frac{\eta}{\beta}.$$

which is feasible for

$$\eta < \beta N. \tag{17}$$

For the coexistence equilibrium $\widetilde{E^{(2)}}$ in this case recall that $\mathscr{M}_0 = 1$, $\mathscr{M}_2 = \mathscr{M}_3 = 0$. Thus $S_2 = N - I_1 - I_2 + \frac{n_{21}}{m_{21}} I_1 > 0$. The conic (5) becomes now just the pair of straight lines

$$I_1 = 0, \qquad I_1 = -\frac{(\nu + n_{21})m_{21}}{\gamma n_{21}} < 0.$$

Therefore it intersects the second conic (6), which now is the hyperbola

$$\frac{\beta}{n_{21}} I_2^2 + \Big(\frac{\beta}{n_{21}} - \frac{\beta}{m_{21}}\Big) I_1 I_2 + \Big(\frac{\eta - \beta N}{n_{21}}\Big) I_2 - I_1 = 0,$$

only at the origin and at the point $M(0, N - \frac{\eta}{\beta})$, giving back the equilibria $\widetilde{E^{(1)}}$ and $\widetilde{E^{(3)}}$. Thus in this case no coexistence equilibrium is possible.

At $\widetilde{E^{(1)}}$ the eigenvalues are $\lambda_1 = -\nu - n_{21} < 0$, $\lambda_2 = -m_{21} < 0$, $\lambda_3 = \beta N - \eta$, giving the stability condition

$$\eta > \beta N. \tag{18}$$

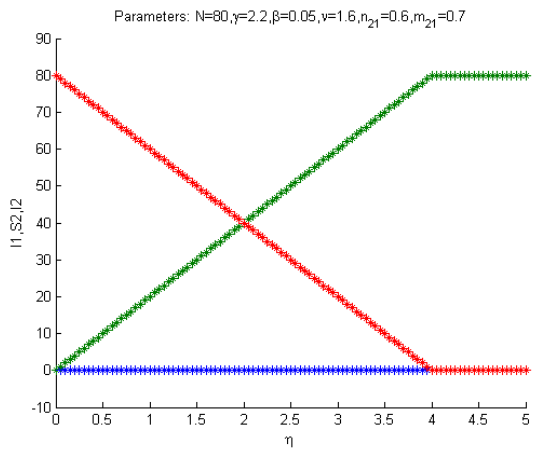


Figure 3: Transcritical bifurcation as function of the parameter η, for the case with the migrations from environment 2 to environment 1 forbidden.

Thus in this case the disease is eradicated and the whole healthy population settles only in patch 2.

For $\widetilde{E^{(3)}}$ the eigenvalues are $\lambda_1 = -m_{21} < 0$, $\lambda_2 = -\nu - n_{21} < 0$, $\lambda_3 = -\beta N + \eta$. Thus, stability follows for $\eta < \beta N$, which is always satisfied in view of the feasibility condition (17). Hence, when feasible, $\widetilde{E^{(3)}}$ is always stable.

These results show that a transcritical bifurcation arises when $\widetilde{E^{(3)}}$ and $\widetilde{E^{(1)}}$ collide, for instance as a result in a change in the parameter η crossing the critical value $\eta^\dagger \equiv \beta N$, or, alternatively, if the total population crosses the critical value $N^\dagger \equiv \eta\beta^{-1}$, see Figure 3, for the parameter values $N = 80$, $\gamma = 2.2$, $\beta = 0.05$, $\gamma = 1.6$, $n_{21} = 0.6$, $m_{21} = 0.7$.

Thus if the ratio of disease recovery over the disease incidence in the second environment is lower than the total population N, see (18), then $\widetilde{E^{(1)}}$ is the only stable equilibrium, so that the disease is eradicated and the whole population settles in the second enviroment. This is shown in Figure 4 for the parameter values $N = 50$, $\gamma = 2.2$, $\beta = 0.05$, $\gamma = 1.6$, $\eta = 3.6$, $n_{21} = 0.6$, $m_{21} = 0.7$.

On the contrary, if the above condition is not satisfied, the disease becomes endemic and the whole population, susceptibles and infected, still settles in the second environment, since in this case the equilibrium $\widetilde{E^{(3)}}$ becomes feasible, (17) and is stable. See Figure 5 for a graphical description with the parameter values $N = 80$, $\gamma = 2.2$, $\beta = 0.05$, $\gamma = 1.6$, $\eta = 3.6$, $n_{21} = 0.6$, $m_{21} = 0.7$.

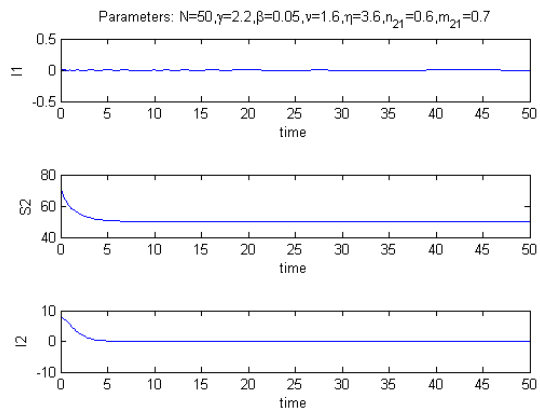


Figure 4: Equilibrium $\widetilde{E^{(1)}}$ for the case with the migrations from environment 2 to environment 1 forbidden.

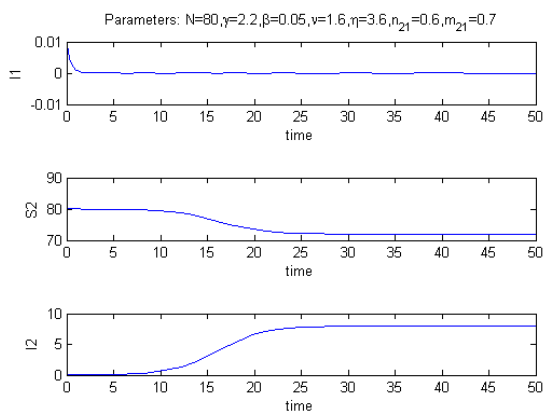


Figure 5: Equilibrium $\widetilde{E^{(3)}}$ for the case with the migrations from environment 2 to environment 1 forbidden.

 ISBN:978-84-615-5392-1

4 Infected do not migrate

In this situation, let us denote the equilibria as $\widehat{E^{(k)}}$, for $k = 0, \ldots, 4$. Here $n_{12} = n_{21} = 0$, so that $\mathscr{M}_1 = \mathscr{M}_2 = 0$. Again at the origin the conclusions of the original system (2) still hold. In particular we then find $\widehat{E^{(1)}} \equiv E^{(1)}$, $\widehat{E^{(3)}}$ with nonzero components

$$\widehat{S_2^{(3)}} = \frac{\eta}{\beta}, \qquad \widehat{I_2^{(3)}} = N - \frac{\eta}{\beta \mathscr{M}_0},$$

and a new equilibrium, $\widehat{E^{(4)}}$, whose nonvanishing components are

$$\widehat{I_1^{(4)}} = \frac{\gamma N \mathscr{M}_3 - \nu}{\gamma \mathscr{M}_3}, \qquad \widehat{S_2^{(4)}} = \frac{\mathscr{M}_0 \nu}{\gamma \mathscr{M}_3}.$$

$\widehat{E^{(3)}}$ is feasible for

$$\beta N \mathscr{M}_0 > \eta, \tag{19}$$

while the feasibility condition for $\widehat{E^{(4)}}$ is

$$\gamma N \mathscr{M}_3 > \nu. \tag{20}$$

For the coexistence equilibrium $\widehat{E^{(2)}}$ we have the following considerations. The two conic sections become now both degenerate: in the first case we have $I_1 = 0$ and $I_2 = -I_1 - (\nu + \gamma \mathscr{M}_0 N)(\gamma \mathscr{M}_3)^{-1}$, for the second one instead $I_2 = 0$ and $I_2 = -N^{-1} I_1 + 1 - \eta(\beta N \mathscr{M}_0)^{-1}$. No intersections of these lines in the interior of the first quadrant are therefore possible, hence no coexistence equilibrium exists in this case.

The eigenvalues of the Jacobian at $\widehat{E^{(1)}}$ are $-m_{12} - m_{21} < 0$ and the pair $\beta N \mathscr{M}_0 - \eta$, $\gamma N \mathscr{M}_3 - \nu$ giving the stability condition

$$N < \min \left\{ \frac{\eta}{\beta \mathscr{M}_0}, \frac{\nu}{\gamma \mathscr{M}_3}, \right\}. \tag{21}$$

At $\widehat{E^{(3)}}$ the eigenvalues are $\gamma \eta \mathscr{M}_3 \Big(\beta \mathscr{M}_0 \Big)^{-1} - \nu$ and the roots of a quadratic, for which the Routh-Hurwitz criterion in view of the feasibility condition (19) reduces to

$$\frac{\mathscr{M}_3}{\mathscr{M}_0} < \frac{\beta \nu}{\gamma \eta}. \tag{22}$$

Finally, at $\widehat{E^{(4)}}$ the eigenvalues are $\beta \nu \mathscr{M}_0 (\gamma \mathscr{M}_3)^{-1} - \eta$ giving

$$\frac{\mathscr{M}_3}{\mathscr{M}_0} > \frac{\beta \nu}{\gamma \eta} \tag{23}$$

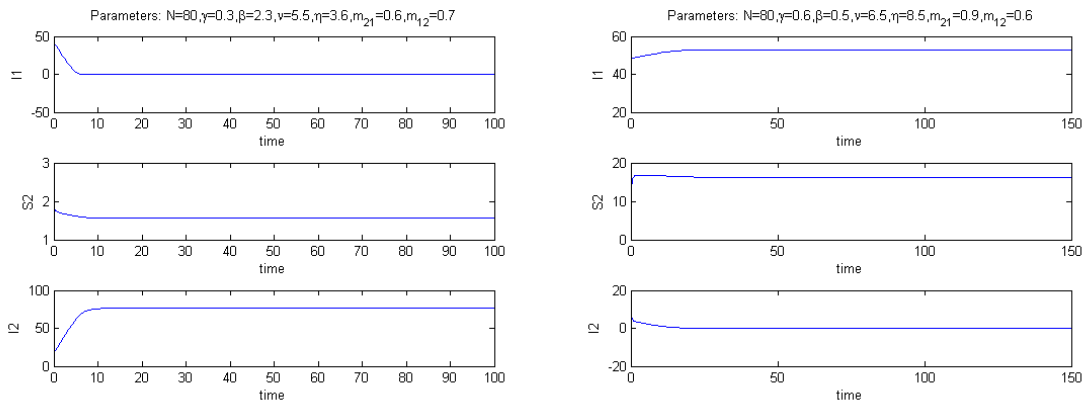


Figure 6: Left: stability of the equilibrium $\widehat{E^{(3)}}$. Right: stability of the equilibrium $\widehat{E^{(4)}}$.

and the roots of the quadratic

$$\lambda^2 - \lambda\Big(- m_{21} - m_{12} - \gamma N + \frac{\nu}{\mathscr{M}_3}\Big) + m_{12}\Big(\gamma N - \frac{\nu}{\mathscr{M}_3}\Big) = 0$$

for which the Routh-Hurwitz conditions are

$$m_{12} + m_{21} + \gamma N - \frac{\nu}{\mathscr{M}_3} > 0 \quad m_{12}\Big(\gamma N - \frac{\nu}{\mathscr{M}_3}\Big) > 0. \tag{24}$$

But the inequalities (24) hold unconditionally when the equilibrium is feasible, in view of (20), so that stability reduces to just condition (23).

Again, there is a transcritical bifurcation when $\widehat{E^{(3)}}$ and $\widehat{E^{(4)}}$ exchange their stability. This is influenced by the value of the parameter

$$\rho \equiv \frac{\beta\nu\mathscr{M}_0}{\gamma\eta\mathscr{M}_3}$$

being smaller or larger than 1. Further, when $\widehat{E^{(1)}}$ is stable, it is the only feasible equilibrium, while when either $\widehat{E^{(3)}}$ or $\widehat{E^{(4)}}$ are feasible, $\widehat{E^{(1)}}$ is always infeasible.

Thus in this case the disease gets eradicated from the ecosystem, if the system settles to equilibrium $\widehat{E^{(1)}}$. Else at $\widehat{E^{(3)}}$ it is eradicated only in environment 1, see Figure 6 left, or finally at $\widehat{E^{(4)}}$ it is eradicated only in environment 2, see Figure 6 right. No other possibilities exist, when the ecosystem thrives.

5 Conclusions

We now discuss briefly some consequences that can be drawn from the above analyses. Recall that all these systems are fragile, in the sense that the whole population can vanish,

but this is due to an intrinsic assumption of the model, that disregards essentially vital dynamics. We will not address this issue any further and concentrate instead on more relevant outcomes.

The original system (2) can settle only to an equilibrium in which the disease is eradicated, and the susceptibles are partitioned among the two patches, or a coexistence equilibrium in which all subpopulations thrive in both enviroments. Further, evidently from what just mentioned, the disease cannot invade the whole environment totally affecting the population, i.e. wiping out all the susceptibles.

By breaking the path from patch 2 into patch 1, the population cannot coexist with the disease in both environments, as it does when communications are allowed. Instead, it would settle to equilibrium $\widetilde{E^{(3)}}$, i.e. no infected remain in patch 1, where in fact the whole population is depleted, but rather the disease remains endemic in patch 2. This occurs if condition (17) holds. But when either the total population N or the disease incidence β decrease suitably, or instead the recovery rate η increases, then the disease gets eradicated also in patch 2 and all the healthy population migrates into patch 2 and remains there, at equilibrium $\widetilde{E^{(1)}}$. The fact that patch 1 gets completely depleted is not surprising, as here migrations back into it are forbidden and only outward migrations are allowed.

As a comparison, in these very same conditions, note that when communications are allowed in both directions, the population distributes according to the proportions $\mathscr{M}_3$ and $\mathscr{M}_0$ respectively in patch 1 and 2. The conditions for which the disease is eradicated are also modified, compare (14) and (18).

When infected are too weak to migrate, there cannot again be a coexistence equilibrium. In this case either the disease is eradicated from the environment, at equilibrium $E^{(1)}$, or it remains endemic solely in patch 2, the system settling at equilibrium $\widehat{E^{(3)}}$, or only in patch 1, with the system attaining equilibrium $\widehat{E^{(4)}}$. In both these last two cases, susceptibles are present in both patches. The discriminating parameter appears to be ρ. When the disease disappears from the whole ecosystem, at equilibrium $\widehat{E^{(1)}}$, the susceptibles distribute among the two environments according respectively to the proportions $\mathscr{M}_3$ and $\mathscr{M}_0$ for patch 1 and 2 as for (2) when also infected can migrate. In case infected are segregated to their own environment, therefore, it might becomes easier to fight the disease in each patch, since the system's outcome among the two equilibria is regulated by the parameter ρ. Instead, it is necessary to violate both feasibility conditions (19) and (20) in order to have disease eradication in the whole environment.

Thus, breaking communications in one direction or forbidding infected to migrate, are both possible means of disease eradication, either everywhere or just in one selected patch.

References

[1] J.C. Frauenthal, *Mathematical Modeling in Epidemiology*, Springer Verlag, 1980.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Tracing the Power and Energy Consumption of the QR Factorization on Multicore Processors

María Barreda[1], Sandra Catalán[1], Manuel F. Dolz[1], Rafael Mayo[1] and Enrique S. Quintana-Ortí[1]

[1] *Dpto. Ingeniería y Ciencia de Computadores, Univ. Jaime I, 12.071–Castellón (Spain)*

emails: mvaya@guest.uji.es, al106631@uji.es, dolzm@icc.uji.es, mayo@icc.uji.es, quintana@icc.uji.es

Abstract

In this paper we analyze the interaction between computational performance, power dissipation and energy consumption of several high-performance implementations of the QR factorization, a crucial matrix operation for the solution of linear systems of equations and linear least squares problems. Our experimental results on a multiprocessor platform equipped with recent multicore technology from AMD show the interaction between these three factors.

Key words: Power, energy, QR factorization, high performance, multicore processors.

1 Introduction

For many decades, a considerable effort has been spent in the optimization of dense linear algebra routines (as well as underlying numerical kernels like the BLAS) from the point of view of computational performance, for an ample range of target computer architectures, from vector processors, to superscalar/VLIW architectures and, in recent years, hardware accelerators (e.g., graphics processors, FPGAs, DSPs, etc.). All this labor has produced a number of high quality libraries which, nowadays, are widely utilized by scientific and engineering applications. Examples of these libraries include LAPACK and `libflame` for desktop servers [2, 20], and distributed-memory (message-passing) versions as ScaLAPACK and PLAPACK for clusters of computers [7, 19]. The stock of numerical packages is still growing, together with the evolution of hardware architectures, and ongoing developments comprise PLASMA, MAGMA or `libflame+SuperMatrix` [15, 14, 11].

On the other hand, *power* (dissipation) is currently recognized as a crucial factor that will exert strong influence on the design of future computer systems, from basic components (processors, memory, NIC, etc.), to large-scale HPC facilities and data processing centers [8, 9, 10]. However, the evaluation and tuning of the power and/or energy consumption of the linear algebra codes in the above-mentioned libraries are still in their beginnings, in spite of the considerable benefits that, in general, energy-aware software can yield [1].

In [5], we performed an initial study of the (computational) performance and power-energy balance for several high-performance implementations of two common matrix operations for the solution of linear systems, the LU factorization and the Cholesky decomposition [12], on a multicore platform. In this paper we extend this study to cover a third matrix decomposition, the QR factorization, key for the solution of (overdetermined) linear systems and linear-least squares problems [12]. Our evaluation of the traditional blocked "slab-based" implementations of this matrix operation in the LAPACK and MKL libraries, as well as an implementation of the incremental QR factorization [13] reveal the trade-off of the factors in the performance-power-energy triangle.

The rest of the paper is structured as follows. In Section 2 we revisit the two blocked algorithms for the QR factorization. In Section 3 we evaluate the performance, power dissipation and energy consumption of these algorithms on 12 cores of a recent AMD-based multiprocessor. Finally, in Section 4 we offer some concluding remarks.

2 The QR Factorization

The QR factorization decomposes a matrix $A \in \mathbb{R}^{m \times n}$ into the product $A = QR$ where $Q \in \mathbb{R}^{m \times m}$ is orthogonal and $R \in \mathbb{R}^{m \times n}$ is upper triangular. For simplicity, hereafter we will assume that the matrix is square (i.e., $m = n$). In the following, we will also consider that A is partitioned into blocks of size $b \times b$, denote the (i, j) block in this partitioning as A_{ij}, and assume that n is an integer multiple of the block size b, i.e., there exists an integer s such that $n = s \cdot b$.

We next review two right-looking blocked algorithms, for the traditional (slab-based) QR factorization and the incremental QR factorization, which represent the state-of-the-art to attain high performance in the solution of linear systems and linear least-squares problems on current multicore processors [16].

2.1 The traditional algorithm for the QR factorization

The traditional QR factorization of a matrix $A \in \mathbb{R}^{n \times n}$ proceeds in panels of b columns (or slabs), as illustrated by Algorithm 1. In practice, the factor R overwrites the upper triangular part of A while the orthogonal matrix is not built but implicitly stored as a collection of Householder vectors using the annihilated entries in the strictly lower triangle of A and an auxiliary vector of order n. Besides, the algorithm requires $4n^3/3$ floating-point

arithmetic operations (flops). Provided $n \gg b$ and b is chosen to be in the range $[128, ..., 512]$, the bulk of the computations in this algorithm is in the application of orthogonal transforms to the trailing submatrix $A_{k:s,j}$. A large fraction of these computations can be cast in terms of matrix-matrix products and, therefore, notable performance and a considerable level of concurrency can be expected from tuned implementations of this operation for multicore platforms.

Algorithm 1 Right-looking blocked algorithm for the QR factorization.

1: **for** $k = 1, 2, \ldots, s$ **do**
2: $A_{k:s,k} = Q_{k:s,k:s} \cdot R_{kk}$ QR factorization
3: **for** $j = k + 1, k + 2, \ldots, s$ **do**
4: $A_{k:s,j} \leftarrow Q^T_{k:s,k:s} A_{k:s,j}$ Apply orthogonal transforms
5: **end for**
6: **end for**

2.2 The incremental QR factorization

The incremental QR factorization, initially proposed to solve the updating problem and/or as an out-of-core algorithm, has attracted considerable attention in recent years due to its high degree of concurrency [4, 6, 17]. Algorithm 2 presents a blocked procedure to compute the incremental QR factorization of a matrix A. By carefully exploiting the special structure of the blocks involved in the procedures for the "2×1" QR factorization and the "2×1" application of orthogonal transforms, the practical cost of this algorithm is reduced to $4n^3/3$ flops; see [13] for details. Under the same conditions as above (i.e., $n \gg b$ and b and $b \in [128, ..., 512]$), high performance can be expected from an implementation of this algorithm that employs highly optimized version of the four numerical kernels that are involved.

3 Experimental Results

The following experiments were obtained using IEEE double-precision arithmetic on a platform with four AMD Opteron 6172 processors, operating at 2.1 GHz, and 256 GB of RAM. The implementation of BLAS was that provided in Intel MKL (v10.3.9). Performance traces were obtained using `Extrae` (v2.2.0) and `Paraver` (v4.1.0); power traces were obtained using our own software module compatible with `Paraver` and a microcontroller-based internal powermeter that measures the power dissipated internally by the main board with a sampling frequency of 25 Hz [3]. The problem size was set to n=10,240 and one single processor (12 cores) were employed in the evaluation. Similar results were found for other problem dimensions and number of processors/cores.

 ISBN:978-84-615-5392-1

Algorithm 2 Right-looking blocked algorithm for the incremental QR factorization.

1: **for** $k = 1, 2, \ldots, s$ **do**
2: $A_{kk} = Q_{kk} R_{kk}^T$ — QR FACTORIZATION
3: **for** $j = k+1, k+2, \ldots, s$ **do**
4: $A_{kj} \leftarrow Q_{kk}^T A_{kj}$ — APPLY ORTHOGONAL TRANSFORMS
5: **end for**
6: **for** $i = k+1, k+2, \ldots, s$ **do**
7: $\begin{pmatrix} A_{kk} \\ A_{ik} \end{pmatrix} = \begin{pmatrix} Q_{kk} \\ Q_{ik} \end{pmatrix} R_{ik}$ — 2×1 QR FACTORIZATION
8: **for** $j = k+1, k+2, \ldots, s$ **do**
9: $\begin{pmatrix} A_{kj} \\ A_{ij} \end{pmatrix} \leftarrow \begin{pmatrix} Q_{kk} & 0 \\ Q_{ik} & I \end{pmatrix}^T \begin{pmatrix} A_{kj} \\ A_{ij} \end{pmatrix}$ — 2×1 APPLY ORTHOGONAL TRANSFORMS
10: **end for**
11: **end for**
12: **end for**

Three implementations were evaluated for the QR factorization:

- LAPACK: The legacy codes for this factorization from `http://www.netlib.org` (routine `dgeqrf`), with parallelism exploited within the invocations to Intel (multi-threaded) MKL BLAS. This routine mimics the numerical procedure in Algorithm 1. The block size was b=128 as for the problem dimensions (n), architecture and BLAS employed in our experiments, this value was close to the optimal.

- MKL: The code from the Intel library for the QR factorization. While, in principle, this routine also responds to the procedure in Algorithm 1, it contains important features (e.g., look-ahead) to optimize performance. Unfortunately, the source code is not available.

- SMPSs: C codes for the incremental QR factorization, linked to the sequential MKL BLAS, with task-level parallelism extracted by the SMPSs runtime system [4]. In the experiments, we set b=256 and the inner block size to 64.

Figure 1 displays the invocations of kernels and the associated power consumption obtained for the execution of the LAPACK routine `dgeqrf`, which computes the QR factorization following procedure sketched in Algorithm 1. From top to bottom, the first two plots correspond to the trace of the complete factorization (kernels and power, respectively), while next two zoom into the first two iterations of the routine (third and fourth plots, for kernels and power respectively). These results illustrate that, on this platform, the LAPACK routine linked to the multi-threaded implementation of BLAS from Intel MKL interleaves

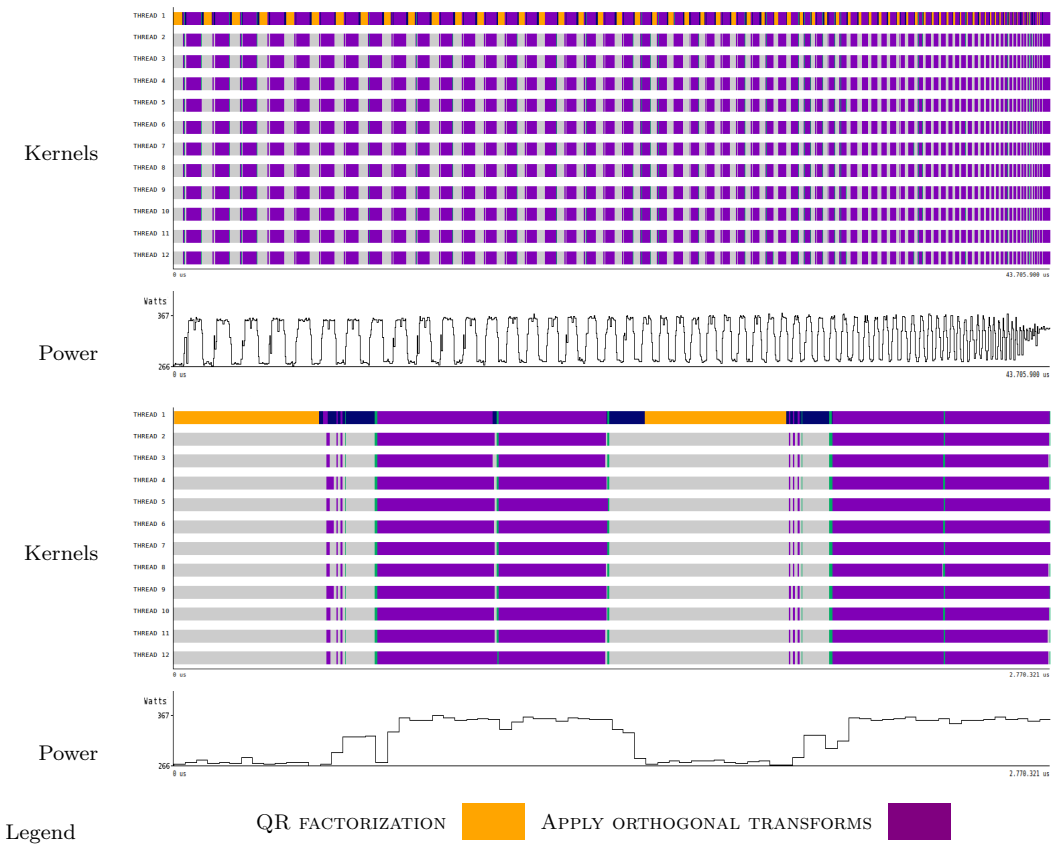


Figure 1: Trace of LAPACK `dgeqrf`. From top to bottom: kernels and power for the complete factorization; kernels and power for the first two iterations of loop k (see Algorithm 1).

sequential and concurrent phases, with the former ones corresponding to the QR factorization of the "current" slab and the parallel ones to the application of orthogonal transforms. From the power consumption perspective, the interlaced sequential and concurrent activity leads to periods of low and high power, respectively, which vary between 266.9 and 376.6 Watts.

Figure 2 evaluates the implementation of the multi-threaded implementation of routine `dgetrf` for the QR factorization in MKL. In this case, we sample the hardware counters for the L2 cache misses and the MFLOPS (millions of flops per second) (`PAPI_L2_DCM` and `PAPI_FP_INS` respectively). These plots show that the MKL routine attains a high utilization of the architectures cores, except for the initial and final stages of the factorization, due to the lack of concurrency at this points. Also, during the execution, there appear some performance drops (captured by the yellow areas in the middle of the MFLOPS plot) which

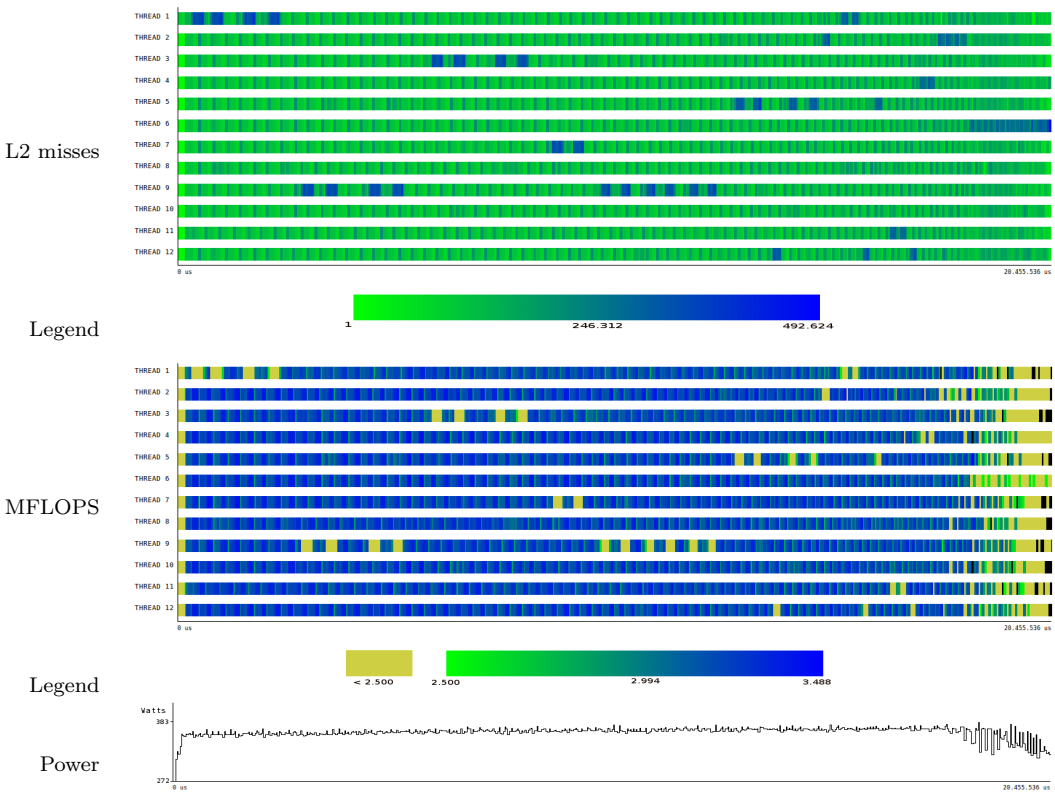


Figure 2: Trace of MKL `dgeqrf`. From top to bottom: L2 cache misses rate, MFLOPS, and power.

may correspond to an unexpectedly high L2 cache miss rate (see the first plot). It is very likely that the MKL routine applies some sort of of look-ahead [18] to overlap the factorization of the "current" panel with the application of orthogonal transforms from previous factorizations to the panels to its right. We can expect that this explains the lack of periods of serial execution and, therefore, the flat pattern of the power line, which now has a minimum at 326.6 Watts and a maximum at 366.0 Watts.

Figure 3 reports the kernel execution and power dissipation of the SMPSs task-parallel C implementation of the incremental QR factorization; see Algorithm 2. The kernel for the 2×1 application of orthogonal transforms dominates the theoretical cost and, as the figure clearly exposes, the execution time of the implementation. There are little synchronization points, due to the higher concurrency of this particular algorithm, which is leveraged by SMPSs to maintain all cores/threads executing tasks (kernels) most of the time, and accounts for the flat profile of the power line for this routine.

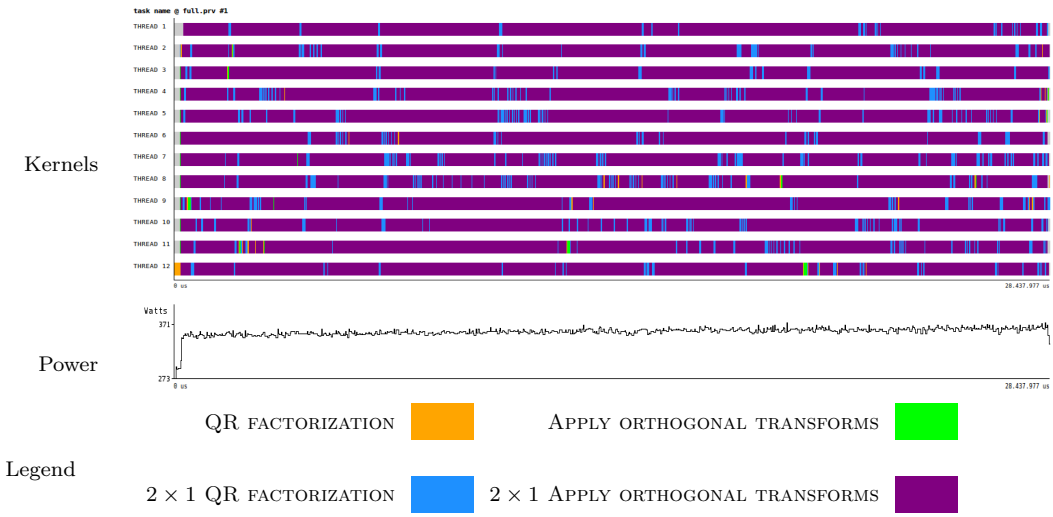


Figure 3: Trace of the C implementation of the QR factorization parallelized with SMPSs. From top to bottom: Kernels and power.

Finally, Table 1 compares the three implementations for the QR factorization from the viewpoint of Time (T, in seconds); GFLOPS (billions of flops per second); minimum, average and maximum power ($P_{\min}$, P_{avg} and $P_{\max}$, respectively, all in Watts); and total energy (E_{tot}, in Joules).

4 Conclusions

We have analyzed the execution of three different, high-quality implementations of numerical procedures for the QR factorization on a multicore platform. The LAPACK code linked

	QR factorization		
	LAPACK	MKL	SMPSs
T (s)	43.70	20.45	28.43
GFLOPS	32.76	69.99	50.35
$P_{\min}$ (W)	266.96	272.11	249.38
P_{avg} (W)	326.63	366.08	357.23
$P_{\max}$ (W)	376.65	383.44	371.72
E_{tot} (J)	14,276	7,488.3	10,159

Table 1: Performance, power and energy of the different implementations for the QR factorization.

with a multi-threaded implementation of BLAS clearly obtains the worst results, from the point of view of both execution time and energy consumption. Indeed, the higher energy consumption of this case is a direct consequence of the longer execution time since on average, the power dissipated by this implementation is inferior to those of the MKL and SMPSs alternatives. Compared with the MKL code, the SMPSs variant yields longer execution time (39.02%) and energy consumption (35.66%) in spite of its slightly inferior average power dissipation (0.97%).

These results demonstrate the direct relation between execution time and energy for compute-intensive dense linear algebra codes. A consequence is that, as a general principle, for this kind of numerical algorithms, one direct manner of optimizing energy consumption is to tune the routine for maximum computational performance.

Acknowledgments

This research was supported by project TIN2011-23283 of the *Ministerio de Economía y Competitividad* and FEDER.

References

[1] Susanne Albers. Energy-efficient algorithms. *Commun. ACM*, 53:86–96, May 2010.

[2] E. Anderson, Z. Bai, J. Demmel, J. E. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. E. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1992.

[3] R. Badia, J. Labarta, M. Barreda, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí, and R. Reyes. Tools for power and energy analysis of parallel scientific applications. In *The 41st International Conference on Parallel Processing – ICPP*, 2012. Submitted.

[4] R. M. Badia, J. R. Herrero, J. Labarta, J. M. Pérez, E. S. Quintana-Ortí, and G. Quintana-Ortí. Parallelizing dense and banded linear algebra libraries using SMPSs. *Concurrency and Computation: Practice and Experience*, 21(18):2438–2456, 2009.

[5] M. Barreda, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí, and R. Reyes. Binding performance and power of dense linear algebra operations. In *Proceedings of the 10th IEEE International Symposium on Parallel and Distributed Processing with Applications – ISPA*, 2012. To appear.

[6] Alfredo Buttari, Julien Langou, Jakub Kurzak, and Jack Dongarra. A class of parallel tiled linear algebra algorithms for multicore architectures. *Parallel Comput.*, 35(1):38–53, 2009.

[7] J. Choi, J. J. Dongarra, R. Pozo, and D. W. Walker. Scalapack: A scalable linear algebra library for distributed memory concurrent computers. In *Proceedings of the Fourth Symposium on the Frontiers of Massively Parallel Computation*, pages 120–127. IEEE Comput. Soc. Press, 1992.

[8] J. Dongarra et al. The international ExaScale software project roadmap. *Int. J. of High Performance Computing & Applications*, 25(1), 2011.

[9] M. Duranton et al. The HiPEAC vision, 2010. Available from `http://www.hipeac.net/roadmap`.

[10] Wu-chun Feng, Xizhou Feng, and Rong Ge. Green supercomputing comes of age. *IT Professional*, 10(1):17 –23, jan.-feb. 2008.

[11] FLAME project home page. `http://www.cs.utexas.edu/users/flame/`.

[12] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[13] Brian C. Gunter and Robert A. van de Geijn. Parallel out-of-core computation and updating the QR factorization. *ACM Transactions on Mathematical Software*, 31(1):60–78, March 2005.

[14] MAGMA project home page. `http://icl.cs.utk.edu/magma/`.

[15] PLASMA project home page. `http://icl.cs.utk.edu/plasma/`.

[16] G. Quintana-Ortí, E.S. Quintana-Ortí, R.A. van de Geijn, F.G. Van Zee, and E. Chan. Programming matrix algorithms-by-blocks for thread-level parallelism. *ACM Trans. Math. Softw.*, 36(3):14:1–14:26, 2009.

[17] Gregorio Quintana-Ortí, Enrique S. Quintana-Ortí, Robert van de Geijn, Field Van Zee, and Ernie Chan. Programming matrix algorithms-by-blocks for thread-level parallelism. *ACM Transactions on Mathematical Software*, 36(3):14:1–14:26, 2009.

[18] Peter Strazdins. A comparison of lookahead and algorithmic blocking techniques for parallel matrix factorization. Technical Report TR-CS-98-07, Department of Computer Science, The Australian National University, Canberra 0200 ACT, Australia, 1998.

[19] Robert A. van de Geijn. *Using PLAPACK: Parallel Linear Algebra Package*. The MIT Press, 1997.

[20] F. G. Van Zee. `libflame`. the complete reference, 2008. In preparation. `http://www.cs.utexas.edu/users/flame`.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Increasing the exactness of spline quasi-interpolants

D. Barrera[1], A. Guessab[2], M. J. Ibáñez[1] and O. Nouisser[3]

[1] *Departamento de Matemática Aplicada, Universidad de Granada*

[2] *Laboratoire de Mathématiques Appliquées, Université de Pau et des Pays de l'Adour*

[3] *Département de Mathématiques et Informatique, Université Cadi-Ayyad*

emails: dbarrera@ugr.es, allal.guessab@univ-pau.fr, mibanez@ugr.es, otheman.nouisser@yahoo.fr

Abstract

Given a spline discrete quasi-interpolation operator Q_d, which is exact on the space $\mathbb{P}_m$ of polynomials of total degree at most m, we propose a general method to determine a new differential quasi-interpolation operator Q_r^D which is exact on $\mathbb{P}_{m+r}$. Q_r^D uses the values of the function to be approximated at the points involved in the linear functional defining Q_d as well as the partial derivatives up to the order r at the same points. From this result, we then construct and study a first order differential quasi-interpolant based on the C^1 B-spline on the equilateral triangulation with an hexagonal support.

Key words: B-splines, Box splines, Differential Quasi-interpolants, Discrete quasi-interpolants, Optimal approximation order

1 Introduction

Quasi-interpolation based on a B-spline is a general approach for efficiently constructing approximants, with low computational cost. Its effectiveness is particularly due to its small support, to achieve local control via suitable spline coefficients in the space spanned by the translates of the B-spline.

In the recent paper [2], it is shown how to modify a given linear operator such that the resulting operator reproduces polynomials to the highest possible degree, and such that the approximation order is the best possible.

As a main application of this tool, new spline quasi-interpolation operators are derived, based on a uniform type-1 triangulation τ approximating regularly distributed data.

2 Notations

Let τ be a uniform triangulation of the plane with grid points $(A_i)_{i\in\mathbb{Z}^2}$. Let us denote by $\mathbb{P}_k$ the space of bivariate polynomials of total degree at most k, and by $S_k^l(\tau)$ the space of piecewise polynomial functions in $C^l\left(\mathbb{R}^2\right)$ of total degree at most k, defined on τ. If $M \in S_k^l(\tau)$ is a B-spline, we denote by $\mathbb{P}(M)$ the space of polynomials of maximal total degree included in the space $\mathcal{S}(M)$ spanned by translates of M. We will assume that $\mathbb{P}(M) = \mathbb{P}_m$ for some positive integer m.

For a real valued function f and $k \in \mathbb{N}$, we say $f \in C^k\left(\mathbb{R}^2\right)$ if f is k times continuously differentiable in the following sense: the directional derivatives of order l, $l = 0, \ldots, k$, at $x \in \mathbb{R}^2$ along the direction $y \in \mathbb{R}^2$ defined as

$$D_y^l f(x) = \frac{d^l}{dt^l} f(x+ty)|_{t=0}$$

exist and depend continuously on x. When the directional derivative exists for y, it may be extended to multiples by defining

$$D_{\alpha y}^l f(x) = \alpha^l D_y^l f(x), \ \alpha \in \mathbb{R}.$$

For $f \in C^k\left(\mathbb{R}^2\right)$, we introduce

$$\left|D^k f\right| = \sup_{x\in\mathbb{R}^2} \sup\left\{\left|D_y^k f(x)\right| : y \in \mathbb{R}^2, \|y\| = 1\right\},$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$. It follows that for any $x, y \in \mathbb{R}^2$, we have

$$\left|D_y^k f(x)\right| \leq \left|D^k f\right| \|y\|.$$

3 Modified differential bivariate spline quasi-interpolants

We are interested in the dQIO Q_d based on the B-spline $M \in S_k^l(\tau)$ given by the expression

$$Q_d[f](x) := \sum_{i\in\mathbb{Z}^2} \lambda f(\cdot + A_i) M(x - A_i), \tag{1}$$

where λ is the linear functional defined as $\lambda f := \sum_{j\in J} c_j f(-A_j)$, for a finite subset $J \subset \mathbb{Z}^2$ and $c := (c_j)_{j\in J} \in \mathbb{R}^{\#J}$, with $\#J$ denoting the cardinality of J (cf. [1, p. 63]). Q_d is a linear map into $\mathcal{S}(M)$ which is local and bounded, and we shall construct Q_d to reproduce $\mathbb{P}_m$.

We will assume that the free parameters c_j, which define the functional λ, are given in such a way that Q_d is exact on $\mathbb{P}_m$. So, Q_d achieves the order of approximation $m + 1$.

 ISBN:978-84-615-5392-1

The dQIO Q_d can be expressed as

$$Q_d[f](x) = \sum_{i\in\mathbb{Z}^2} f(A_i)\, L(x - A_i), \tag{2}$$

where

$$L(x) := \lambda M(x + \cdot) = \sum_{j\in J} c_j M(x - A_j). \tag{3}$$

If in (2) we replace $f(A_i)$ by its Taylor polynomial approximation of degree r at A_i, then the resulting operator

$$T_r^D[f](x) := \sum_{i\in\mathbb{Z}^2} \left(\sum_{l=0}^{r} \frac{1}{l!} D^l_{x-A_i} f(A_i) \right) L(x - A_i) \tag{4}$$

reproduces polynomials up to degree $\max\{m, r\}$. But the new modified operator

$$Q_r^D[f](x) := \sum_{i\in\mathbb{Z}^2} \left(\sum_{l=0}^{r} \frac{(m + r - l)!r!}{l!\,(m + r)!\,(r - l)!} D^l_{x-A_i} f(A_i) \right) L(\cdot - A_i), \tag{5}$$

reproduces polynomials up to degree $m + r$. This result is a consequence of the following integral representation of the error for the quasi-interpolants Q_r^D.

Theorem 1 *Let $f \in C^{m+r+1}(\mathbb{R}^2)$. Then, for all $x \in \mathbb{R}^2$, we have*

$$f(x) - Q_r^D[f](x) = \sum_{i\in\mathbb{Z}^2} \left(\int_0^1 K_{mr}(t)\, D^{m+r+1}_{x-A_i} f(A_i + t(x - A_i))\, dt \right) L(x - A_i), \tag{6}$$

where $K_{mr}(t) := (-1)^m \frac{t^m(1-t)^r}{(m+r)!}$.

As another immediate corollary of Theorem 1, we obtain the following error estimate.

Corollary 2 *Suppose that $f \in C^{m+r+1}(\mathbb{R}^2)$. Then, for all $x \in \mathbb{R}^2$, we have*

$$\left| f(x) - Q_r^D[f](x) \right| \le \left| D^{r+m+1} f \right| R(x), \tag{7}$$

where

$$R(x) := \frac{m!r!}{(m + r)!\,(m + r + 1)!} \sum_{i\in\mathbb{Z}^2} \|x - A_i\|^{m+r+1} |L(x - A_i)|. \tag{8}$$

 ISBN:978-84-615-5392-1

We may estimate the approximation error as a function of the free parameters c_j, when the function to be approximate is sufficiently regular. Indeed, by (3), the sum in the function $R(x)$ involved in the error estimate established in equations (7)-(8) may be written in terms of the coefficients of the functional λ as follows:

$$\begin{aligned}\sum_{i\in\mathbb{Z}^2}\|x-A_i\|^{m+r+1}\left|L\left(x-A_i\right)\right| &\leq \sum_{j\in J}|c_j|\sum_{i\in\mathbb{Z}^2}\|x-A_i\|^{m+r+1}M\left(x-A_i-A_j\right)\\ &=\sum_{j\in J}|c_j|\sum_{i\in\mathbb{Z}^2}\|x-A_i+A_j\|^{m+r+1}M\left(x-A_i\right) \qquad (9)\\ &=: F\left(c,x\right)\end{aligned}$$

One possible strategy would be to select the free parameters c_j, in the starting operator Q_d, in such a way to minimize the upper bound function $F(c,x)$ subject to equality constraints on c_j, yielding the exactness of Q_d on $\mathbb{P}_m$. However, in order to obtain a much simpler minimization problem, we suggest a formulation with an upper bound $F(c)$ of $F(c,x)$, depending only on c.

Due to the interesting properties of C^1 quartic splines on type-1 triangulations, we will solve this kind of problem starting from the C^1 cubic B-spline defined on the equilateral triangulation of the real plane, under specific imposed conditions on the sequence c. We will give a detailed treatment including an explicit error bound estimation for the corresponding operator, which are especially useful in practice.

References

[1] C. de Boor, K. Höllig and S. Riemenschneider, Box splines, Springer-Verlag, New York.

[2] A. Guessab, O. Nouisser and G. Schmeisser, Multivariate approximation by a combination of modified Taylor polynomials, J. Comput. Appl. Math. **196** (2006), 162–179.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A new more consistent Reynolds model for piezoviscous hydrodynamic lubrication problems in line contact devices

Guy Bayada[1], Begoña Cid[2], Guillermo García[2] and Carlos Vázquez[3]

[1] *ICJ UMR CNRS 5208, INSA de Lyon*

[2] *Department of Applied Mathematics II, University of Vigo*

[3] *Department of Applied Mathematics, University of A Coruña*

emails: Guy.Bayada@insa-lyon.fr, bego@dma.uvigo.es, guille@dma.uvigo.es, carlosv@udc.es

Abstract

Hydrodynamic lubrication problems in piezoviscous regime are usually modeled by the classical Reynolds equation combined with a suitable law for the pressure–dependence of viscosity. By taking into account the pressure–viscosity dependence in the Stokes equation and to derive the Reynolds equation in the thin film limit, a new model has been proposed by Rajagopal & Szeri [5]. However, in [5] some additional simplifications are assumed. In the present work, avoiding these simplifications, from a Stokes equation in piezoviscous regime we deduce a new Reynolds model for line contact lubrication problems, in which the cavitation phenomenon is also taken into account. Thus, the new complete model consists of a nonlinear free boundary problem associated to the proposed new Reynolds equation.

Moreover, the classical model, Szeri's one and the here proposed one are simulated through the development of some numerical algorithms involving upwind schemes, finite elements method, duality type numerical strategies and fixed point techniques. Finally, several numerical tests are performed to carry out a comparative analysis among the different models.

Key words: Hydrodynamic lubrication, Reynolds equation, piezoviscosity, cavitation phenomenon, free boundary

1 Introduction

In the mechanical and mathematical literature concerning the models for piezoviscous hydrodynamic lubrication problems, different classical devices have been considered, such as journal–bearings, rolling–bearings, rolling–ball–bearings,... In all of these situations, the behavior of the lubricant pressure in the thin film setting has been classically modeled by the Reynolds equation, in which the pressure–dependence of viscosity is usually introduced *a posteriori* by some expression, such as, for example, the Barus law:

$$\mu = \mu_0\, e^{\alpha\, p}, \tag{1}$$

where μ, μ_0, p and α denote the viscosity, the zero pressure viscosity, the pressure and the piezoviscosity coefficient. In this procedure, the thin film limit from Stokes equation to Reynolds one is obtained regardless of the pressure–viscosity dependence. Moreover, the cavitation phenomenon can be incorporated through Reynolds model or Elrod–Adams one (see [1], for instance). However, by assuming that the viscosity depends on pressure in Stokes equation according to Barus law, in [5] a more careful derivation of the limit Reynolds equation is carried out. More precisely, after some simplifying assumptions the following set of equations is obtained:

$$\frac{d}{d\,x}\left[\left(\frac{h^3}{\mu} - 12\,\alpha\,\int_0^h y(h-y)\,\frac{\partial\,u}{\partial\,x}\,dy\right)\frac{d\,p}{d\,x}\right] = 6\,s\,\frac{d\,h}{d\,x} \tag{2}$$

$$\frac{\partial\,u}{\partial\,x} + \frac{\partial\,v}{\partial\,y} = 0\,, \tag{3}$$

where h, (u, v) and $(s, 0)$ denote the gap between surfaces, the velocity field in the thin film and the velocity field at the lower surface, assuming that the upper surface is fixed. Next, by approximating $\partial\,u/\partial\,x$ by $d\,u_{av}/d\,x$ where u_{av} is an average velocity, they introduce the flow rate $Q = h(x)\,u_{av}(x)$ to deduce the following modified Reynolds equation

$$\frac{d}{d\,x}\left[\left(\frac{h^3}{\mu} + \alpha\,Q\,\frac{d\,h^2}{d\,x}\right)\frac{d\,p}{d\,x}\right] = 6\,s\,\frac{d\,h}{d\,x}\,. \tag{4}$$

2 An alternative model for piezoviscous lubrication

In this work, mainly a more rigorous model avoiding the simplification considered in [5] is proposed. Indeed, a new equation for the case of a line contact with Reynolds model for cavitation is obtained. The proposed methodology to deduce a new family of models is based on a fixed point technique to solve the equations (2)-(3), starting from an initial velocity $u^0(x, y)$ which vanishes at the upper surface and is equal to $(s, 0)$ at the lower one, satisfying the flux condition

$$\int_0^h u^0(x, y)\,dy = Q. \tag{5}$$

 ISBN:978-84-615-5392-1

Some easy computations lead to the expression

$$u^0(x,y) = (\frac{s\,h}{2} - Q)\frac{6}{h^3}(y^2 - y\,h) + s\,(1 - \frac{y}{h})\ . \tag{6}$$

Next, replacing u by u^0 in (2), the first alternative model can be obtained

$$\frac{d}{d\,x}\left[\left(\frac{h^3}{\mu} - 12\,\alpha\,\frac{d\,h}{d\,x}\left(\frac{s\,h^2}{30} - \frac{Q\,h}{10}\right)\right)\frac{d\,p}{d\,x}\right] = 6\,s\,\frac{d\,h}{d\,x}\ . \tag{7}$$

It is possible to use a fixed point technique just by replacing u^0 in the second equation to obtain v^0, next obtaining a new u^1 in order to build a new Reynolds equation, and so on. In this way, each fixed point iteration provides a new model, however it does not seem possible to obtain a generic procedure and the complexity of the expressions involved is increasing. So the *first iteration equation* (7) is the model we consider here.

In order to compare the results with those of [5], we introduce the angular coordinate, t, by the change $x = R\,\sin(t),\ t \in [-\pi/2, \pi/2]$, where R is the radius of the cylinder. We define the film thickness by

$$h(t) = -\frac{R}{n}(1 + n\,\cos(t)), \quad n = -\frac{R}{h_0 + R}, \quad h_0 = h(0)$$

and we introduce the following non-dimensional terms:

$$\overline{h} = -\frac{n\,h}{R},\ \ \overline{\mu} = \frac{\mu}{\mu_0},\ \ \overline{p} = \frac{p\,h_0}{\mu_0\,s},\ \ \overline{\alpha} = \frac{\alpha\,\mu_0\,s}{h_0},\ \ \overline{\beta} = \overline{\alpha}\frac{h_0}{R}.$$

Furthermore, we consider the Reynolds model for cavitation (see [1], for example) and we obtain the following dimensionless formulation of the problem: find $\overline{p}$, such that

$$\frac{d}{d\,t}\left[G(t)\,\frac{d\,\overline{p}}{d\,t}\right] = 6\,\frac{d\,\overline{h}}{d\,t}\,,\ \ \overline{p} > 0 \quad \text{ in } \Omega^+ \tag{8}$$

$$\overline{p} = 0 \quad \text{ in } \Omega_0 \tag{9}$$

$$\overline{p}(t_2) = \frac{d\,\overline{p}}{d\,t}(t_2) = 0 \tag{10}$$

$$\overline{p}(-\pi/2) = \overline{p}(\pi/2) = 0\ , \tag{11}$$

being $\Omega^+ = \{t \in \Omega / \overline{p}(t) > 0\}$ and $\Omega_0 = \{t \in \Omega / \overline{p}(t) = 0\}$. Moreover,

$$G(t) = \left(\overline{h}^3\,e^{-\overline{\alpha p}} - \frac{12\,\overline{\beta}}{\cos(t)}\,\frac{d\,\overline{h}}{d\,t}\left(\frac{\overline{h}^2}{30} - \frac{\overline{h}\,\overline{h}(t_2)}{20}\right)\right)\,\frac{1}{-n\,\overline{h}_0\,\cos(t)}\,,$$

where t_2 is the unknown free boundary and $\overline{h}_0$ denotes the minimum of the dimensionless gap. This model includes the isoviscous case ($\overline{\alpha} = \overline{\beta} = 0$) and the classical piezoviscous

 ISBN:978-84-615-5392-1

model ($\overline{\alpha} \neq 0, \overline{\beta} = 0$). The Rajagopal & Szeri model may be written in a similar way but with a slightly different expression of $G(t)$.

Note that problem (8)–(11) admits a formulation in terms of variational inequalities and their numerical solutions may be obtained by the combination of finite element techniques with classical projection methods or the more complex duality type algorithm ([2]). The application of these techniques to classical piezoviscous formulations can be found in [3] and [4], for example.

Another alternative is to restart from the modified Reynolds equation, to integrate it and interpret the result as the conservation of the flow

$$Q = \frac{s\,h}{2} - \frac{h^3}{12\,\mu}\,\frac{d\,p}{d\,x} + \alpha\,\frac{d\,h}{d\,x}\left(\frac{s\,h^2}{30} - \frac{Q\,h}{10}\right)\,\frac{d\,p}{d\,x}\,.$$

In this case, we can pose the following first order ordinary differential equation (ode):

$$\frac{d\,p}{d\,x} = \frac{\dfrac{s\,h}{2} - Q}{\dfrac{h^3}{12\,\mu} - \alpha\,\dfrac{d\,h}{d\,x}\left(\dfrac{s\,h^2}{30} - \dfrac{Q\,h}{10}\right)}\,. \tag{12}$$

Next, taking into account that the unknown free boundary point x_2 satisfies $p(x_2) = p'(x_2) = 0$ we get from (12) that $Q = \frac{s}{2}\,h(x_2)$, so that we can write the dimensionless initial value ode problem in the form:

$$\frac{d\,\overline{p}}{d\,t} = \frac{6\,(-n)\,\overline{h}_0\,\cos^2(t)\,\left[\overline{h}(t) - \overline{h}(t_2)\right]}{\overline{h}^{\,3}(t)\,\mathrm{e}^{-\overline{\alpha}\,\overline{p}}\,\cos(t) + \dfrac{6}{5}\,\overline{\beta}\,n\,\sin(t)\,\overline{h}(t)\left[\dfrac{\overline{h}(t)}{3} - \dfrac{\overline{h}(t_2)}{2}\right]} \tag{13}$$

$$\overline{p}(-\pi/2) = 0\,. \tag{14}$$

In order to compute $\overline{p}$ and t_2 we propose an iterative numerical scheme combining the `ode15s` integrator of MATLAB to solve (13)–(14) in $[-\pi/2, t_2]$, for each value of t_2, with a *regula falsi* algorithm to search the final value of $t_2 \in (0, \pi/2)$, such that $\overline{p}(t_2) = 0$.

Note that the corresponding *first order ODE version* for isoviscous, classical piezoviscous and Rajagopal–Szeri models can be obtained in an analogous way and solved with the same numerical methods.

So far, we propose two alternatives for the numerical simulation: the use of a first order ode solver and the use of a characteristics finite elements method combined with a duality method to solve the free boundary problem associated to the second order elliptic equation.

3 Numerical tests

In order to assess the relevance of both new more rigorous models for the piezoviscous case, several numerical tests have been carried out. The first impression is that the maximum

values of pressure are slightly different but the variation of the maximum values of viscosity are higher. In general, the solution of our alternative model appears to be closer to the classical solution than the one proposed in [5], as illustrated by Figure 1 and Table 1.

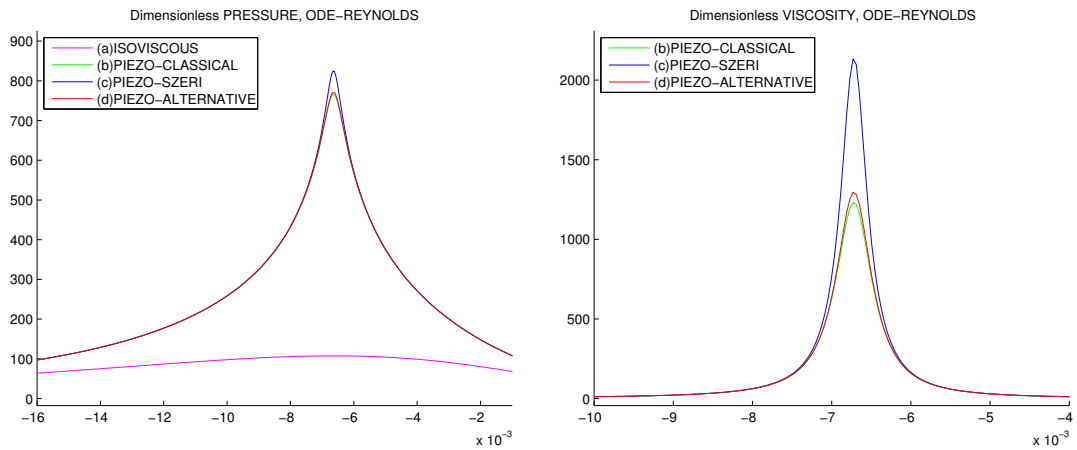


Figure 1: Dimensionless pressure and dimensionless viscosity

	ODE-R	IV-R-BM
a) Isoviscous	107.5200	107.5203
b) Piezo–Classical	765.7870	723.8485
c) Piezo–Szeri	824.8810	881.8840
d) Piezo–Alternative	771.1110	722.7223

Table 1: Maximum dimensionless pressure for different models with first order EDO solver (ODE-R) and second order variational inequality solver (IV-R-BM)

A first conclusion is that the problem has a large gradient near the maximum pressure region so that is difficult to capture the almost "spike" qualitative behavior of the solution in all models. The here proposed one seems to be closer to the classical piezoviscous solution that the one proposed by Rajagopal and Szeri. In this sense, we have proposed a rigorous methodology to include piezoviscosity previously to the limit procedures followed for obtaining Reynolds model from Stokes one that overcomes the simplifications in [5]. Furthermore, it seems that the numerical results obtained for this new model in the regime we have considered result to be close to the ones obtained by the classical model mainly used in mechanical and mathematical literature.

Now we are trying to extend the use of the alternative model to the case in which Elrod–Adams model for cavitation is used, as it results more realistic when cavitation phenomenon

appears in the convergent region (starvation phenomenon).

Acknowledgements

This work has been partially supported by the MEC Research Project (MTM2010-21135-C02).

References

[1] G. Bayada and M. Chambat, *Sur quelques modélisations de la zone de cavitation en lubrification hydrodynamique*, J. of Theor. and Appl. Mech. **5(5)** (1986) 703–729.

[2] A. Bermúdez and C. Moreno, *Duality methods for solving variational inequalities*, Comp. Math. with Appl. **7** (1981) 43–58.

[3] N. Calvo, J. Durany and C. Vázquez, *Comparación de algoritmos numéricos en problemas de lubricación hidrodinámica con cavitación en dimensión uno*, Rev. Int. Met. Num. Calc. Dis. Ing. **13, 2** (1997) 185–209.

[4] J. Durany, G. García and C. Vázquez, *Numerical computation of free boundary problems in elastohydrodynamic lubrication*, Appl. Math. Modelling **20** (1996) 104–113.

[5] K. R. Rajagopal and A. Z. Szeri, *On an inconsistency in the derivation of the equations of elastohydrodynamic lubrication*, Proc. R. Soc. Lond. **A 459** (2003) 2771–2786.

[6] O. Reynolds, *On the theory of lubrication and its application to the Tower's experiments*, Phil Trans. R. Soc. Lond. **177** (1886) 159–209.

[7] A. Z. Szeri, *Fluid film lubrication:theory and design*, Cambridge University Press, 1998.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Taking Care of the Singularities in the Probabilistic Evolutionary Quantum Expectation Value Dynamics

N.A. BAYKARA[1] and METİN DEMİRALP[2]

[1] *Department of Mathematics, Marmara University*

[2] *Informatics Institute, Istanbul Technical University*

emails: nabaykara@gmail.com, metin.demiralp@gmail.com

Abstract

This work is somehow the extension of the work to be presented by Metin Demiralp in this conference. Purpose is the same as before, to get an infinite set of ODEs over the expectation values of the state vector's outer powers. Almost completely same strategy is followed here. The basic difference requesting extension is the singularity in the commutator of the state vector with the system Hamiltonian. The analyticity in this commutator is missing and the related problems are bypassed by defining inverse outer powers of the state vector.

Key words: Probabilistic Evolution Equations, Quantum Expected Values, Singular Hamiltonians.

1 Introduction

Probabilistic Evolution Equations [1] and their solution is a quite new approach to solve ODEs, and also PDEs via expectation values as long as they can be defined. This approach extends the space to an infinite one by using the integer outer powers of the state vector. Then an infinite set of ordinary differential equations (ODEs) is constructed such that it is linear and has an infinite constant coefficient matrix. This facilitates the theory however at the expense of dealing with infinitely many items. Curious readers can refer certain new resources [2, 3] on this topic.

We start with the definition of the expected value of a given operator $\widehat{O}$ as follows

$$\frac{d\left\langle \widehat{O}\right\rangle (t)}{dt}=\int_{\mathcal{V}}d\mathcal{V}\psi\left(\mathbf{x},t\right)^{*}\left\{ \frac{i}{\hbar}\left[\,\widehat{H}\widehat{O}-\widehat{O}\widehat{H}\,\right]\right\} \psi\left(\mathbf{x},t\right)=\left\langle \frac{i}{\hbar}\left[\,\widehat{H}\widehat{O}-\widehat{O}\widehat{H}\,\right]\right\rangle \tag{1}$$

where $\widehat{H}$ and $\psi\left(\mathbf{x},t\right)$ stand for the system Hamiltonian and the wave function while $\mathcal{V}$ and $d\mathcal{V}$ denote the spatial volume of the integration and the infinitesimal volume element respectively. This equality's dependence on the operator under consideration disables universality. Hence, it better to deal with the state vector whose elements are operators like positions and momenta, instead of this operator. We define the state vector denoted by $\mathbf{s}$ as follows

$$\mathbf{s}\equiv\left[\,\widehat{s}_{1}\,\,...\,\,\widehat{s}_{n}\,\right]^{T} \tag{2}$$

where n denotes the "System's dimension. The state vector's outer square (outer or Kronecekr product with itself) is given explicitly below

$$\mathbf{s}^{\otimes2}\equiv\mathbf{s}\otimes\mathbf{s}\equiv\left[\,s_{1}\mathbf{s}^{T}\,\,...\,\,s_{n}\mathbf{s}^{T}\,\right]^{T}. \tag{3}$$

This can be extended to the following general formula

$$\mathbf{s}^{\otimes m}\;\;\equiv\;\;\mathbf{s}\otimes\mathbf{s}^{\otimes(m-1)}\equiv\left[\,s_{1}\mathbf{s}^{\otimes(m-1)^{T}}\,\,...\,\,s_{n}\mathbf{s}^{\otimes(m-1)^{T}}\,\right],\qquad m=0,1,2,3,... \tag{4}$$

where the mth outer power of the state vector has n^{m} number of elements. The zeroth outer power is defined as the universal scalar, just 1 (that is, it is a single element vector).

The state vector's expected value satisfies the following equation

$$\frac{d\left\langle \mathbf{s}\right\rangle (t)}{dt}=\left\langle \frac{i}{\hbar}\left[\,\widehat{H}\widehat{\mathbf{s}}-\widehat{\mathbf{s}}\widehat{H}\,\right]\right\rangle \tag{5}$$

We assume

$$\frac{i}{\hbar}\left[\,\widehat{H}\widehat{\mathbf{s}}-\widehat{\mathbf{s}}\widehat{H}\,\right]\equiv\sum_{j=0}^{\infty}\mathbf{H}_{j}\mathbf{s}^{\otimes j} \tag{6}$$

where $\mathbf{H}_{j}$ is a rectangular matrix of $n\times n^{j}$ type. (5) can be extended to the outer powers by using certain properties of the outer product together with the matrix product to get

$$\frac{d\left\langle \mathbf{s}^{\otimes j}\right\rangle (t)}{dt}=\sum_{\ell=0}^{\infty}\mathbf{E}_{j,\ell}\left\langle \mathbf{s}^{\otimes(j-1+\ell)}\right\rangle (t),\qquad j=0,1,2,... \tag{7}$$

where

$$\mathbf{E}_{j,\ell}\equiv\sum_{k=0}^{j-1}\mathbf{I}^{\otimes k}\otimes\mathbf{H}_{\ell}\otimes\mathbf{I}^{\otimes(j-1-k)}. \tag{8}$$

If we define

$$\boldsymbol{\xi}(t) \equiv \left[\left\langle \mathbf{s}^{\otimes 0} \right\rangle (t)^T \left\langle \mathbf{s}^{\otimes 1} \right\rangle (t)^T \ ... \ \right]^T, \qquad \mathbf{E} \equiv \begin{bmatrix} \mathbf{E}_{0,0} & \cdots & \mathbf{E}_{0,m} & \cdots \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{E}_{m,0} & \cdots & \mathbf{E}_{m,m} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{9}$$

then we obtain

$$\frac{d\boldsymbol{\xi}(t)}{dt} = \mathbf{E}\boldsymbol{\xi}(t) \tag{10}$$

which is an infinite set of ODEs whose coefficient matrix $\mathbf{E}$ is composed of constant elements. The second block element of its solution gives the sought expected value of the state vector. The solution can be formally written as

$$\boldsymbol{\xi}(t) = \mathrm{e}^{t\mathbf{E}}\boldsymbol{\xi}(0) \tag{11}$$

where

$$\boldsymbol{\xi}(0) \equiv \left[\left\langle \mathbf{s}^{\otimes 0} \right\rangle (0)^T \left\langle \mathbf{s}^{\otimes 1} \right\rangle (0)^T \ ... \ \right]^T \tag{12}$$

and

$$\left\langle \mathbf{s}^{\otimes m} \right\rangle (0) \equiv \int_{\mathcal{V}} d\mathcal{V} \psi_0 (\mathbf{x})^* \mathbf{s}^{\otimes m} \psi_0 (\mathbf{x}), \qquad m = 0, 1, 2, ... \tag{13}$$

We find this information sufficient for our purposes here. Further details can be found in Metin Demiralp's paper[Ref] in this conference.

2 Singularities in the Hamiltonian

The Probabilistic Evolution Philosophy is based on the inspiration from the analyticity and therefore Taylor series. However this inspiration remains applicable only when the system's Hamiltonian has no singularities. Since the Hamiltonian's dependence on the momenta are rather polynomial its position dependent part, that is, the potential gains a lot of importance for the singularities. If the potential function has singularities somewhere in the complex plane of the spatial variables then it affects the solution of the Schrödinger equation and therefore the expected values. This means that (6) remains no longer valid. It must be replaced by something different. In the case of polar singularities Taylor series are replaced by Laurent series which has inverse powers of the independent variable or its deviation from a fixed reference point, together with the nonnegative powers. This inspires us to introduce the inverse outer powers of the state operators. We define

$$\mathbf{s}^{\otimes(-1)} \equiv \frac{1}{n} \left[\widehat{s_1^{-1}} \ ... \ \widehat{s_n^{-1}} \right] \tag{14}$$

 ISBN:978-84-615-5392-1

hence the transposition operation helps us to realize the outer inversion. We have

$$\mathbf{s}^{\otimes(-1)} \otimes \mathbf{s}^{\otimes j} = \mathbf{s}^{\otimes(j-1)}, \qquad j = 0, \pm 1, \pm 2, ... \tag{15}$$

which urges us to assume

$$\frac{i}{\hbar}\left[\,\widehat{H}\widehat{\mathbf{s}} - \widehat{\mathbf{s}}\widehat{H}\,\right] \equiv \sum_{j=-\infty}^{\infty} \mathbf{H}_j \mathbf{s}^{\otimes j} \tag{16}$$

where the coefficients are in the abstract operators mapping related outer power to a vector space where $\mathbf{s}$ lies.

This assumption and the abovementioned extensive definitions permits us to construct an infinite set of equations as follows

$$\frac{d\boldsymbol{\xi}(t)}{dt} = \mathbf{E}\boldsymbol{\xi}(t) \tag{17}$$

where

$$\mathbf{E} \equiv \begin{bmatrix} \mathbf{E}^{(1,1)} & \mathbf{E}^{(1,2)} \\ \mathbf{E}^{(2,1)} & \mathbf{E}^{(2,2)} \end{bmatrix}, \qquad \boldsymbol{\xi}(t) \equiv \left[\left\langle \mathbf{s}^{(1)} \right\rangle (t)^{T} \left\langle \mathbf{s}^{(2)} \right\rangle (t)^{T} \right]^{T}. \tag{18}$$

3 Conclusion

In the last three equalities the entities superscripted by integers between parantheses are all infinite blocks whose types are compatible to what we have said above. (17) again corresponds to an infinite set of ODEs. However, this time, the indexing of the infinite entities are not only on the nonnegative integers but over all integers. This, of course, complicates the issue a little bit more even though it is still possible to construct truncation approximant. However, this time truncation is not only downward, it is both upward and downward. More information will be given in the presentation and also in the relevant paper.

References

[1] M. Demiralp, *Quantum Expected Value Dynamics in Probabilistic Evolution Perspective*, CMMSE2012 Proceedings (this conference)

[2] M. Demiralp, E. Demiralp, L. Hernandez-Garcia, *A probabilistic foundation for dynamical systems: theoretical background and mathematical formulation*, J. Math. Chem. **58** (2012) 850-869. 2012.

[3] E. Demiralp, D. Demiralp, L. Hernandez-Garcia, *A probabilistic foundation for dynamical systems: phenomenological reasoning and principal characteristics of probabilistic evolution*, J. Math. Chem. **58** (2012) 870-880.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Real-time optimization of wind farms and fixed-head pumped-storage hydro-plants

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

In this paper we analyze whether real-time compensation of wind power plant deviation penalties is profitable by means of the coordinated optimization of the wind power plant with a pumped-storage hydro-plant. We shall make use of optimal control techniques to carry out the optimization. We shall also analyze another possible solution based on compensation carried out *a posteriori*, instead of in real time.

Key words: Optimal Control, Pumped-Storage Plant, Wind Farm
MSC 2000: 49J52, 49M05

1 Introduction

The new regulations allow wind farms to go to the market to sell the energy generated by their facilities. If wind farms offer in the pool, they will prepare their offers and schedule their power production. However, a major problem exists: the unpredictability of wind farm production. Forecasting errors lead to the wind farm incurring financial losses, known as deviation penalties. Diverse methods have also been proposed to store this energy [1]. In this paper we focus on combined use of a wind farm with pumped-storage plants.

Some authors ([2], [3]) have researched the operation of a wind farm cooperating with a micro-hydroelectric power plant and a pumped-storage hydro-plant. Previous studies exclusively employ the storage ability to compensate for wind power imbalances. However, this approach is not representative for large pumped-storage plants in power systems. One of the techniques used for large pumped-storage plants ([4], [5]) is to calculate the optimal amount of spinning reserve that the system operator should provide so as to be able to

respond to errors in forecasts. The combined operation of wind farms and a pumped-storage hydro-plant is also analyzed in [6].

The present paper aims to calculate the optimal operation of the pumped-storage plant, simultaneously pursuing two goals: to maximize revenue in conventional operations in the day-ahead market and to coordinate with the wind power producer with the aim of partially compensating for wind power imbalances. In this paper we shall consider a large capacity pumped-storage working jointly with a wind farm adjacent to its facilities. We shall consider them to be a single unit (a *wind-hydro power plant*). Two different joint configurations for the resulting joint-unit formed by the pumped-storage plant and the wind farm are considered. In the first (uncoordinated operation), the pumped-storage plant does not compensate for the errors due to forecasting wind power. In the second (coordinated operation), we shall attempt to compensate for these errors in real time. We shall see in this paper that the fact that the pumped-storage plant is a fixed-head plant will mean that the optimal solution is of a very special type: bang-singular-bang. This will have crucial consequences in coordinated operation and we shall present a qualitative study of the real-time compensation of forecasting errors. In view of the result obtained in this study, we shall propose a second solution: to employ the over-generation deviations of the wind power plant *a posteriori* to pump water into the upper reservoir of the pumped-storage plant, thus increasing profits. Finally, we present a realistic example.

2 Problem description and model overview

The day-ahead market in the Spanish wholesale electricity market is organized as a set of twenty-four simultaneous hourly auctions. The simple bid format consists of a pair of (hourly) values: quantity $q\,(MWh)$ and price $p\,(euro/MWh)$. The problem we shall solve is the one faced by a wind-hydro power plant when preparing its offers for the day-ahead market. This basic scheduling, with plants working independently, is based on the volume of water $b\left(m^3\right)$ that must be used and on the best forecast of wind power generation available each hour $W^f(t)(MW)$. Unfortunately, wind power forecasts within a $14-38$ hour time horizon are usually highly inaccurate and hence incur deviation penalties.

As regards the pumped-storage plant, we shall model it in great detail without any additional simplifications. For a large capacity reservoir, the effective head is constant over the optimization interval and here the fixed-head hydro-plant model is defined. In plants of this type, the active power generated, $P(MW)$, is represented by the linear equation: $P(z'(t)) = Az'(t)$, where A represents the efficiency and diverse parameters related to the geometry of the hydro-plant (see [7]) and $z'(m^3/s)$ is the rate of water discharge. Taking into account the conversion losses of the pumping process, we must therefore introduce the efficiency, η, in the model.

We consider $z'(t)$ to be bounded by technical constraints: $q_{\min} \leq z'(t) \leq q_{\max}, \forall t \in [0, T]$

 ISBN:978-84-615-5392-1

and we assume that b is the volume of water that must be discharged over the entire optimization interval $[0,T]$, so: $z(0)=0$, $z(T)=b$. The function P is thus defined piecewise as:

$$P(z') := \begin{cases} A \cdot z' & \text{if} \quad z' \geq 0 \\ \eta \cdot A \cdot z' & \text{if} \quad z' < 0 \end{cases} \tag{1}$$

3 Optimization of a fixed-head pumped-storage plant

In a previous paper [8] by the authors, we presented an algorithm that allows the optimal solution of a fixed-head pumped-storage plant to be obtained. The objective function is given by hydraulic profit over the optimization interval, $[0,T]$. Profit is obtained by multiplying the hydraulic production of the pumped-storage hydro-plant by the clearing price, $\pi(t)$, at each hour, t. An Optimal Control problem can thus be mathematically formulated as follows:

$$\begin{aligned} &\max_{(u,z)} \int_0^T L(t,z(t),u(t))dt = \max_{(u,z)} \int_0^T \pi(t)P(u)dt \\ &z' = u; \quad z(0)=0, z(T)=b; \quad u_{\min} \leq u(t) \leq u_{\max} \end{aligned} \tag{2}$$

For the Optimal Control problem (2), we define the Hamiltonian in normal form:

$$H(t,z,u,\lambda) := L(t,z,u) + \lambda u = \pi(t)P(u) + \lambda u \tag{3}$$

and the resulting Hamiltonian, H, is linear in the control variable, u. It is well known [9] that when the Hamiltonian is linear in u, the optimality condition leads to the optimal u^* being undetermined if the switching function $\Phi(x,\lambda) \equiv H_u = 0$. An added complication arises in our problem: the Hamiltonian is defined piecewisely and the derivative of H with respect to u (H_u) presents discontinuity at $u=0$. When non-differentiable objective functions arise in optimization problems, the generalized (or Clarke's) gradient (see [9]) must be considered. Based on the above theoretical results, in [8] we determined the *bang-singular-bang (b-s-b) optimal solution*:

$$u^*(t) = \begin{cases} u_{\max} & \text{if} \quad A \cdot \pi(t) > -\lambda_0 \\ u_{\text{sing}} = 0 & \text{if} \quad -\lambda_0 \in [A \cdot \pi(t), \eta \cdot A \cdot \pi(t)] \\ u_{\min} & \text{if} \quad \eta \cdot A \cdot \pi(t) < -\lambda_0 \end{cases} \tag{4}$$

The previous algorithm interpolates $\pi(t)$ and works with a continuous function. Thus, by adjusting the switching times, it is capable of achieving the final volume b to discharge with the desired precision. However, generating companies must in fact present offers in the day-ahead market for each of the 24 hours of the following day. That is, we need to convert a continuous variable into a discrete variable. We shall lose an essential feature in this conversion: we shall no longer be able to achieve any final volume of water precisely.

 ISBN:978-84-615-5392-1

In fact, the volume discharged in the b-s-b solution must belong to the set of M possible values: $\Omega = \{b_1, b_2, ..., b_M\}$. The plant operator therefore only needs choose in $\Omega = \{b_i\}_{i=1}^{M}$ the nearest value, without exceeding the available volume, b ($b_{sol} < b < b_{sol+1}$). In this case, b_{sol} is the discharged volume corresponding to the optimal b-s-b solution.

4 Qualitative analysis of real-time optimization

In view of the above results, we shall conduct a qualitative study on the b-s-b solution. Let us assume we have obtained the solution for a certain λ_{sol} (calculated by aiming at a certain final volume, b_{sol}). We can know the price, π_{turb}, above which it is of interest to discharge water, and we can know the price, π_{pump}, below which it is of interest to pump water. It is shown that, between the instants of pumping (t_{pump}), stoppage (t_{stop}) and discharging water (t_{turb}), the following relations exist between the prices:

$$\pi(t_{pump}) < \pi(t_{stop}); \quad \pi(t_{stop}) < \pi(t_{turb}); \ \pi(t_{turb}) > \eta \cdot \pi(t_{pump}) \tag{5}$$

Furthermore, between two instants of stoppage, it is verified that:

$$\pi(t^1_{stop}), \pi(t^2_{stop}) \in \left[\frac{\lambda_{sol}}{\eta \cdot A}, \frac{\lambda_{sol}}{A}\right] \tag{6}$$

When the plant operator prepares its offer for the day-ahead market for day D, this solution obtained for the pumped-storage plant, assuming the market prices and available water to be known, is the one that it will offer, seeing as it maximizes profits. The wind power plant will offer according to the best forecast for wind power production available at 10 hours the day before, $D-1$. However, when day D arrives, deviations will almost certainly be produced between the actual wind power production, $W^r(t)$, and the forecasted production, $W^f(t)$. In this context, we shall pose the following question: when faced with a deviation in wind power generation at the instant t, might it be of interest to the pumped-storage plant to modify its behavior in real time (i.e. at t) so as to compensate for the deviation penalties of the wind farm and thus achieve a greater joint profit?

Let us call $d(t) = W^r(t) - W^f(t)$ the deviation of the wind farm at the instant t, $p^+(t)$ the price the market pays the over-generation deviation (which will be a certain fraction s of the market price) and $p^-(t)$ the price we must pay for the under-generation penalty (which will be a certain fraction l of the market price). Let us assume in all cases that the deviations are against the system. We shall analyze in detail the two possibilities for the deviations, $d(t)$, of the wind power plant: 1) the over-generation deviation, and 2) the under-generation deviation.

Let us now consider the first case. 1) If the wind farm presents an over-generation deviation, the hydro-plant will be able to act at t in only two cases: 1a) If it was stopped, it will use the over-generation from the wind power plant to pump water; 1b) If it was

discharging water, it will produce less power to compensate for the over-generation of the wind farm. If it was already pumping, as the solution is of the b-s-b type, it will not be able to act. Let us analyze sub-case 1a). At instant t, the hydro-plant was stopped and pumped $d(t)(MW)$ at zero cost. The amount of water pumped at t which will then be used is: $d(t)/\eta.A$. With this modification, the deviation in wind power generation does not produce any profit at t, and we must find an instant t^* at which it is of interest to the pumped-storage plant to discharge this water. At t^*, the hydro-plant may be stopped (sub-case 1a1) or pumping (sub-case 1a2), seeing that, as the solution is of the b-s-b type, if it was discharging water, the turbines cannot be put to greater use. It should be borne in mind that this action will mean a change in its scheduling and will hence result in a penalty; in this case, for over-generation.

We shall analyze all the other cases in a similar way to this case and shall see in a detailed manner that the conditions that must be fulfilled for the real-time modification to be of interest can never be given by the conditions (5) and (6). Conclusion: *no real-time modification is of interest.*

5 *A posteriori* optimization of a wind-hydro power plant

Subsequent to the above study, we posed the question as to whether it is possible to model the functioning of the wind-hydro power plant so as to operate in a coordinated manner *a posteriori* and thus improve profits. We shall not make real-time compensations for under-generation deviations in wind power. We shall however compensate for over-generation deviations in wind power. We shall attempt to use the surplus wind power generated on day D to pump water, thereby avoiding penalties for over-generation on day D and subsequently use this water in the hydro-plant by discharging it on the following day $D+1$. Furthermore, as we are working for the day-ahead market, we shall eliminate all the uncertainty associated with the process.

$$B = \int_0^T \left(\pi^{D+1}(t)P^{D+1}(t) + \pi^D(t)W^D(t) - C^D(t)\right) dt \tag{7}$$

The total profit over the optimization interval $[0, T]$ is revenue minus cost. Revenue is obtained by multiplying the hydraulic production, $P(t)$, and the wind power production, $W(t)$, by the clearing price, $\pi(t)$, at each hour, t. The unique cost in our system is the cost of deviation penalties, $C(t)$. Accordingly, and in order for the comparison to be rigorous, the wind power production is considered to be sold to the market on day D and that of the hydro-plant on day $D + 1$. We shall use superscripts to denote the day under consideration. In uncoordinated operation, we shall have that $z(T) = b_{sol}$. In the coordinated configuration, the profit obtained shall have to take into account the reduction in deviation penalties, $C(t)$, and the increase in the volume of water available: $z(T) = b_{sol} + b^*$. To illustrate

 ISBN:978-84-615-5392-1

the behavior of this solution, we shall consider an example of a wind-hydro power plant and compare the uncoordinated and the coordinated configurations. We shall see that it is possible to obtain profit in the latter case.

Acknowledgements

This work was supported by the Spanish Government (MEYC, project: MTM2012-32961).

References

[1] E. Spahic, G. Balzer, B. Hellmich, W. Munch, *Wind Energy Storages - Possibilities*, Proc. Power Tech. 2007 IEEE Lausanne (2007), 615–620.

[2] E. D. Castronuovo, J. A. P. Lopes, *On the optimization of the daily operation of a wind-hydro power plant*, IEEE Trans. Power Syst. **19** (2004), 1599–1606.

[3] J. S. Anagnostopoulos, D. E. Papantonis, *Pumping station design for a pumped-storage wind-hydro power plant*, Energy Conversion and Management, **48** (2007), 3009–3017.

[4] M.A. Ortega-Vazquez, D.S. Kirschen, *Estimating the spinning reserve requirements in systems with significant wind power generation penetration*, IEEE Trans. Power Syst. **24** (2009), 114–124.

[5] A. Jaramillo, E. D. Castronuovo, I. Sanchez, J. Usaola, *Optimal operation of a pumped-storage hydro plant that compensates the imbalances of a wind power producer*, Electr. Pow. Syst. Res. **81** (2011), 1767– 1777.

[6] J. Garcia-Gonzalez, R. de la Muela, L. Santos, A. Gonzalez, *Stochastic joint optimization of wind generation and Pumped-Storage units in an electricity market*, IEEE Trans. Power Syst. **23** (2008), 460–468.

[7] M. E. El-Hawary, G. S. Christensen, *Optimal Economic Operation of Electrical Power Systems*, Academic Press, New York, 1979.

[8] L. Bayon, J. M. Grau, M. M Ruiz, P. M. Suarez, *Optimization of a pumped-storage fixed-head hydro-plant: the bang-singular-bang solution*, Math. Prob. Eng. (2011), 1-11.

[9] F.H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A metapopulation model of competion type

Davide Belocchio[1], Giacomo Gimmelli[1], Alessandro Marchino[1] and Ezio Venturino[1]

[1] *Dipartimento di Matematica "Giuseppe Peano", Università di Torino, via Carlo Alberto 10, 10123 Torino, Italy*

emails: davide.belocchio@alice.it, giacomo.gimmelli@virgilio.it, marchino.alessandro@gmail.com, ezio.venturino@unito.it[1]

Abstract

In this paper we present and analyse a simple two populations model for migrations among two different environments. The interactions among populations are of competing type for resources. Further, an external agent acts on the populations by maintaining their levels at constant value. Equilibria are investigated. A sufficient condition for the coexistence equilibrium is provided.

Key words: populations, competition, migrations
MSC 2000: AMS codes (92D25, 92D40)

1 Model formulation

We consider two environments among which two competing populations can migrate, denoted by P and Q. Let P_i, Q_i, $i = 1, 2$ be their sizes in the two environments. Here the subscripts denote the environments in which they live. Let each population thrive in each environment according to logistic growth, with possibly differing reproduction rates, respectively r_i for P_i and s_i for Q_i, and carrying capacities, respectively again K_i for P_i and H_i for Q_i. We take them to be different, since they may be influenced by the environment. Further let a_i denote the interspecific competition rate for P_i due to the presence of the popolation Q_i and b_i denote conversely the interspecific competition rate for Q_i due to the presence of the popolation P_i.

[1]This paper was completed and written during a visit of the fourth author at the Max Planck Institut für Physik Komplexer Systeme in Dresden, Germany. The author expresses his thanks for the facilities provided.

Let m_{ij} the migration rate from environment j to environment i for the popolation P_j and let n_{ij} be the migration rate from j to i for the Q_j's.

The resulting model has the following form:

$$\begin{aligned}
\dot{P}_1 &= r_1P_1(1-\frac{P_1}{K_1}) - a_1P_1Q_1 - m_{21}P_1 + m_{12}P_2 \equiv A(P_1,P_2,Q_1,Q_2), \\
\dot{Q}_1 &= s_1Q_1(1-\frac{Q_1}{H_1}) - b_1Q_1P_1 - n_{21}Q_1 + n_{12}Q_2 \equiv C(P_1,P_2,Q_1,Q_2), \\
\dot{P}_2 &= r_2P_2(1-\frac{P_2}{K_2}) - a_2P_2Q_2 - m_{12}P_2 + m_{21}P_1 \equiv B(P_1,P_2,Q_1,Q_2), \\
\dot{Q}_2 &= s_2Q_2(1-\frac{Q_2}{H_2}) - b_2Q_2P_2 - n_{12}Q_2 + n_{21}Q_1 \equiv D(P_1,P_2,Q_1,Q_2).
\end{aligned} \tag{1}$$

At this point we make a strong assumption. We suppose that there is an external agent that keeps the populations in check, by removing individuals of the two populations at rates u for the P's and v for the Q's. These control activities are performed in the same way in both environments. Thus (1) gets modified as follows:

$$\begin{aligned}
\dot{P}_1 &= r_1P_1(1-\frac{P_1}{K_1}) - a_1P_1Q_1 - m_{21}P_1 + m_{12}P_2 - \frac{1}{2}u(t), \\
\dot{Q}_1 &= s_1Q_1(1-\frac{Q_1}{H_1}) - b_1Q_1P_1 - n_{21}Q_1 + n_{12}Q_2 - \frac{1}{2}v(t), \\
\dot{P}_2 &= r_2P_2(1-\frac{P_2}{K_2}) - a_2P_2Q_2 - m_{12}P_2 + m_{21}P_1 - \frac{1}{2}u(t), \\
\dot{Q}_2 &= s_2Q_2(1-\frac{Q_2}{H_2}) - b_2Q_2P_2 - n_{12}Q_2 + n_{21}Q_1 - \frac{1}{2}v(t).
\end{aligned} \tag{2}$$

Note that in fact the removal functions $u(t)$ and v are functions of time through all the population sizes P_i and Q_i. These controls are unknown, but we can get by, since we know their aim, which is to keep both populations at the constant fixed levels P and Q. From this it follows that $P_2 = P - P_1$, $Q_2 = Q - Q_1$. Further, we must have $\dot{P}_1 + \dot{P}_2 = 0$ and $\dot{Q}_1 + \dot{Q}_2 = 0$, i.e.

$$0 = A + B - u, \quad 0 = C + D - v.$$

Substituting back into the system (2) and eliminating the variables P_2 and Q_2, we find

$$\dot{P}_1 = \frac{1}{2}[A - B], \quad \dot{Q}_1 = \frac{1}{2}[C - D].$$

We can now expand these expressions to obtain the final form of the model

$$\begin{aligned}\dot{P}_1 &= \frac{1}{2}\left[P\left(2m_{12}-r_2+a_2Q+\frac{r_2}{K_2}P\right)+\left(\frac{r_2}{K_2}-\frac{r_1}{K_1}\right)P_1^2\right. \\ &\left.+P_1\left(r_1+r_2-2m_{12}-2m_{21}-a_2Q-2\frac{r_2}{K_2}P\right)-a_2Q_1P+(a_2-a_1)P_1Q_1\right], \\ \dot{Q}_1 &= \frac{1}{2}\left[Q\left(2n_{12}-s_2+b_2P+\frac{s_2}{H_2}Q\right)+\left(\frac{s_2}{H_2}-\frac{s_1}{H_1}\right)Q_1^2\right. \\ &\left.+Q_1\left(s_1+s_2-2n_{12}-2n_{21}-b_2P-2\frac{s_2}{H_2}Q\right)-b_2QP_1+(b_2-b_1)Q_1P_1\right]\end{aligned} \tag{3}$$

2 Equilibria

Let us consider the system (3). For ease of computation, we shall make the following re-parametrizations:

$$\begin{aligned}&\alpha = P\left(2m_{21}-r_2+a_2Q+\frac{r_2}{K_2}P\right), && \beta=\frac{r_2}{K_2}-\frac{r_1}{K_1}, && \gamma=-a_2P<0,\\ &\delta = r_1+r_2-2m_{12}-2m_{21}-a_2Q-2\frac{r_2}{K_2}P, && \epsilon=a_2-a_1, &&\\ &\zeta = Q\left(2n_{21}-s_2+b_2P+\frac{s_2}{H_2}Q\right). && \eta=\frac{s_2}{H_2}-\frac{s_1}{H_1}, && \theta=-b_2Q<0,\\ &\iota = s_1+s_2-2n_{12}-2n_{21}-b_2P-2\frac{s_2}{H_2}Q, && \kappa=b_2-b_1; &&\end{aligned}$$

by means of which the system may be written in the form

$$\begin{aligned}\dot{P}_1 &= \frac{1}{2}\left(\alpha+\beta P_1^2+\delta P_1+\gamma Q_1+\epsilon P_1Q_1\right), \\ \dot{Q}_1 &= \frac{1}{2}\left(\zeta+\eta Q_1^2+\iota Q_1+\theta P_1+\kappa Q_1P_1\right).\end{aligned} \tag{4}$$

Seeking equilibria, it is easily seen that the origin in the P_1-Q_1 phase plane is a feasible equilibrium only for special values of the total populations P and Q stemming from the condition

$$\alpha=\zeta=0, \tag{5}$$

which give either the condition $P=Q=0$, i.e. the ecosystem disappears, or alternatively the following values for the total population values

$$\begin{aligned}P &= \frac{\left(2K_2m_{21}-K_2r_2\right)s_2-a_2K_2\left(2H_2n_{21}-H_2s_2\right)}{a_2b_2H_2K_2-r_2s_2}, \\ Q &= \frac{\left(2H_2n_{21}-H_2s_2\right)r_2-b_2H_2\left(2K_2m_{21}-K_2r_2\right)}{a_2b_2H_2K_2-r_2s_2}.\end{aligned} \tag{6}$$

 ISBN:978-84-615-5392-1

Observe that a nonzero value of P in this case means that the whole population P is contained in the second environment only, as $P_1 = 0$, and similarly for Q. The Jacobian of (3) in the neighborhood of the origin is

$$\frac{1}{2}\begin{pmatrix}\delta & \gamma\\ \theta & \iota\end{pmatrix}$$

and by means of the Routh-Hurwitz Criterion we can state that the stability of the origin is ensured by the conditions

$$\frac{1}{a_2b_2H_2K_2 - r_2s_2}\left(a_2H_2\left(b_2K_2\left(-2m_{12} - 2n_{12} + r_1 + s_1\right) + r_2\left(2n_{21} - s_2\right)\right)\right.$$

$$\left. + s_2\left(b_2K_2\left(2m_{21} - r_2\right) + r_2\left(2m_{12} - 2m_{21} + 2n_{12} - 2n_{21} - r_1 + r_2 - s_1 + s_2\right)\right)\right) < 0$$

$$\frac{1}{a_2b_2H_2K_2 - r_2s_2}\left(a_2H_2\left(b_2K_2\left(\left(2m_{12} - r_1\right)\left(2n_{12} - s_1\right) - \left(2m_{21} - r_2\right)\left(2n_{21} - s_2\right)\right)\right.\right.$$

$$\left. + r_2\left(2n_{21} - s_2\right)\left(-2n_{12} + 2n_{21} + s_1 - s_2\right)\right)$$

$$\left. - \left(2m_{12} - 2m_{21} - r_1 + r_2\right)s_2 \times \left(b_2K_2\left(2m_{21} - r_2\right) + r_2\left(2n_{12} - 2n_{21} - s_1 + s_2\right)\right)\right) > 0.$$

In Figure 1 we report a situation leading to this equilibrium.

There are then the two boundary equilibria with only one population in environment 1, P_1 or Q_1, with population levels that can be calculated from

$$0 = \frac{1}{2}\left(\zeta + \theta P_1\right)$$

for $Q_1 = 0$ and for $P_1 = 0$ from

$$0 = \frac{1}{2}\left(\alpha + \gamma Q_1\right),$$

each subject to the condition that the other equation in (3) must be satisfied. Thus substituting these values into the remaining equation in (3), a relationship between the parameters is obtained, which thus leads to the equilibria

$$\begin{aligned} P_1 &= \frac{Pb_2H_2 + 2H_2n_{21} + \left(Q - H_2\right)s_2}{b_2H_2}, \quad Q_1 = 0, \\ P_1 &= 0, \quad Q_1 = \frac{Qa_2K_2 + 2K_2m_{21} + \left(P - K_2\right)r_2}{a_2K_2}, \end{aligned} \tag{7}$$

together with the respective conditions

$$\begin{aligned} \alpha &= \frac{\zeta(\delta\theta - \beta\zeta)}{\theta^2}, \\ \zeta &= \frac{\alpha(\gamma\iota - \alpha\eta)}{\gamma^2}. \end{aligned} \tag{8}$$

We will not expand the latter in terms of the original parameters due to the length of the resulting equations. We just note that the denominators in general do not vanish but for degenerate cases. In addition to (8), feasibility conditions are respectively

$$Pb_2H_2 + 2H_2n_{21} + Qs_2 \geq H_2s_2, \quad Qa_2K_2 + 2K_2m_{21} + Pr_2 \geq K_2r_2. \tag{9}$$

The Jacobian matrices in the neighborhood of these boundary equilibria are respectively

$$\frac{1}{2\theta}\begin{pmatrix} \delta\theta - 2\beta\zeta & \gamma\theta - \epsilon\zeta \\ \theta^2 & \iota\theta - \kappa\zeta \end{pmatrix}, \quad \frac{1}{2\gamma}\begin{pmatrix} \delta\gamma - \alpha\epsilon & \gamma^2 \\ \theta\gamma - \alpha\kappa & \iota\gamma - 2\alpha\eta \end{pmatrix}.$$

Again by means of the Routh-Hurwitz criterion we can obtain the stability conditions for these equilibria, that for the point $(P_1, 0)$ have the form

$$\theta(\delta + \iota) > \zeta(2\beta + \kappa), \quad \theta^2\epsilon\zeta + \delta\theta^2\iota + 2\beta\kappa\zeta^2 > \theta\left(\gamma\theta^2 + 2\beta\zeta\iota + \delta\kappa\zeta\right)$$

and for the equilibrium $(0, Q_1)$ take the form

$$\gamma(\delta + \iota) > \alpha(2\eta + \epsilon), \quad \gamma^2\delta\iota + \alpha\gamma^2\kappa + 2\epsilon\eta\alpha^2 > \gamma\left(\theta\gamma^2 + 2\alpha\delta\eta + +\alpha\epsilon\iota\right),$$

having used the fact that $\gamma < 0$ and $\theta < 0$.

Then, there is the coexistence equilibrium. From (4) we can see that this point may be found by intersecting the two nullclines. An analysis of these curves reveals that they are hyperbolas, apart from degenerate cases. In fact, we can write them e.g. in matrix notation as follows

$$(P_1, Q_1, 1)\begin{pmatrix} \beta & \frac{1}{2}\epsilon & \frac{1}{2}\delta \\ \frac{1}{2}\epsilon & 0 & \frac{1}{2}\gamma \\ \frac{1}{2}\delta & \frac{1}{2}\gamma & \alpha \end{pmatrix}\begin{pmatrix} P_1 \\ Q_1 \\ 1 \end{pmatrix} = 0,$$

and

$$(P_1, Q_1, 1)\begin{pmatrix} 0 & \frac{1}{2}\kappa & \frac{1}{2}\theta \\ \frac{1}{2}\kappa & \eta & \frac{1}{2}\iota \\ \frac{1}{2}\theta & \frac{1}{2}\iota & \zeta \end{pmatrix}\begin{pmatrix} P_1 \\ Q_1 \\ 1 \end{pmatrix} = 0,$$

from which it is immediately seen that the determinants of the matrices in general do not vanish and

$$\Gamma_1 = \det\begin{vmatrix} \beta & \frac{1}{2}\epsilon \\ \frac{1}{2}\epsilon & 0 \end{vmatrix} = -\frac{1}{4}\epsilon^2 < 0,$$

and

$$\Gamma_2 = \det\begin{vmatrix} 0 & \frac{1}{2}\kappa \\ \frac{1}{2}\kappa & \eta \end{vmatrix} = -\frac{1}{4}\kappa^2 < 0,$$

thereby justifying the above claim. But in view of the fact that apart from $\gamma < 0$ and $\theta < 0$ all the coefficients of the two hyperbolas do not have a definite sign, conditions ensuring that at least one intersection must be present are not easy to assess. But the set

 ISBN:978-84-615-5392-1

$\Omega = \{(P_1, Q_1)\colon 0 \leqslant P_1 \leqslant P \wedge 0 \leqslant Q_1 \leqslant Q\}$ is invariant for the flow generated by (3). This is a 2-dimensional compact invariant subset, therefore the Poincaré-Bendixson Theorem holds and we have that the only three possible ω-limit sets for orbits having their initial conditions in Ω must be

- a critical point,
- a limit cycle,
- a finite number of critical points $c_1 \ldots, c_k$ and a countable number of limit orbits (heteroclinic and/or homoclinic) whose α- and ω-limit sets belong to $\{c_1 \ldots, c_k\}$.

Furthermore, by means of the Sard-Smale Theorem we can easily infer that the subset of the parameter space in which we can have the non-coexistence equilibria must have measure zero. Thus for almost all values of the parameters we do not have non-coexistence equilibria.

Following these results, the problem of finding the coexistence equilibrium reduces to finding a Dulac function for (3). Up to now, we have the following sufficient condition for the existence of such a function: if

$$r_2 < \min\{2m_{21}, r_1\}, \quad s_2 < \min\{2n_{21}, s_1\} \tag{10}$$

then the function

$$D(P_1, Q_1) = \frac{1}{P_1 Q_1}$$

is a Dulac function for the system (3). This allows us to ensure that whenever the sufficient condition holds there cannot be any limit cycle, and therefore the system must either tend to a critical point or to orbits connecting the critical points. One such instance is empirically shown in Figure 2.

We have also run simulations attempting to obtain persistent oscillations around the coexistence equilibrium, but were not successful.

3 Conclusion

A metapopulation model with two patches and two competing populations migrating among them has been considered, subject to the constraint that whole populations are held at a constant value by an external agent. In particularly unfavorable circumstances the system is shown to disappear. This can occur only for a special set of parameters, satisfying condition (5) and thereby giving the total populations at specific levels, (6).

Equilibria containing only one population in just one environment are also obtained for very specific parameter values, (8) and attain the values given in (7). One population is

thus surviving in patch 1 and the other one in patch 2. This is clearly possible because in the assumptions of the model, stating that each one of them grows logistically in absence of the competing one, we are saying that sufficient resources are available for them to thrive in each environment.

The existence and stability of the coexistence equilibrium has been shown numerically, as the analysis proves to be too complicated, in view of the fact that it originates from the intersection of conic sections, which can be shown to be hyperbolas, but for which the precise determination of their position in the phase plane it too complicated. Thus even to establish sufficient conditions for their feasibility is not possible. We provide only (10), which is rather based on the use of the Dulac function and the Poincaré-Bendixson theorem.

In situations as the one presented here it is important to address the question to what is the system's outcome if some of the connecting paths between patches are cut out. Here unfortunately we are not able to state anything precisely, since all that can be established is that by annihilating the pairs m_{12}, m_{21} and n_{12}, n_{21}, thereby stating that one of the two populations does not migrate, or even just m_{12} and n_{12} to mean that the path from patch 2 into patch 1 is blocked, simplifications in the conditions (5), (8) and the corresponding population levels (6) and (7) are obtained. Note however that not even one of the coefficients of the conic sections obtained by setting to zero the system (4) attains a definite sign, also in these special cases. Therefore in this situation not much more can be analytically stated about the consequences of breaking interpatch communications.

References

[1] J. T. Cronin, *Movement and spatial population structure of a prairie planthopper*, Ecology, **84** (2003) 1179–1188.

[2] I. Hanski, *Single-species spatial dynamics may contribute to long-term rarity and commonness*, Ecology **66** (1985) 335–343.

[3] I. Hanski, M. Gilpin (Ed.s), *Metapopulation biology: ecology, genetics and evolution*, London: Academic Press, 1997.

[4] I. Hanski, A. Moilanen, T. Pakkala, M. Kuussaari, *Metapopulation persistence of an endangered butterfly: a test of the quantitative incidence function model*, Conservation Biology **10** (1996) 578–590.

[5] G. Lei, I. Hanski, *Metapopulation structure of Cotesia melitaearum, a parasitoid of the butterfly Melitaea cinxia*, Oikos **78** (1997) 91–100.

[6] R. Levins, Some demographic and genetic consequences of environmental heterogeneity for biological control, Bulletin of the Entomological Society America **15** (1969) 237–240.

 ISBN:978-84-615-5392-1

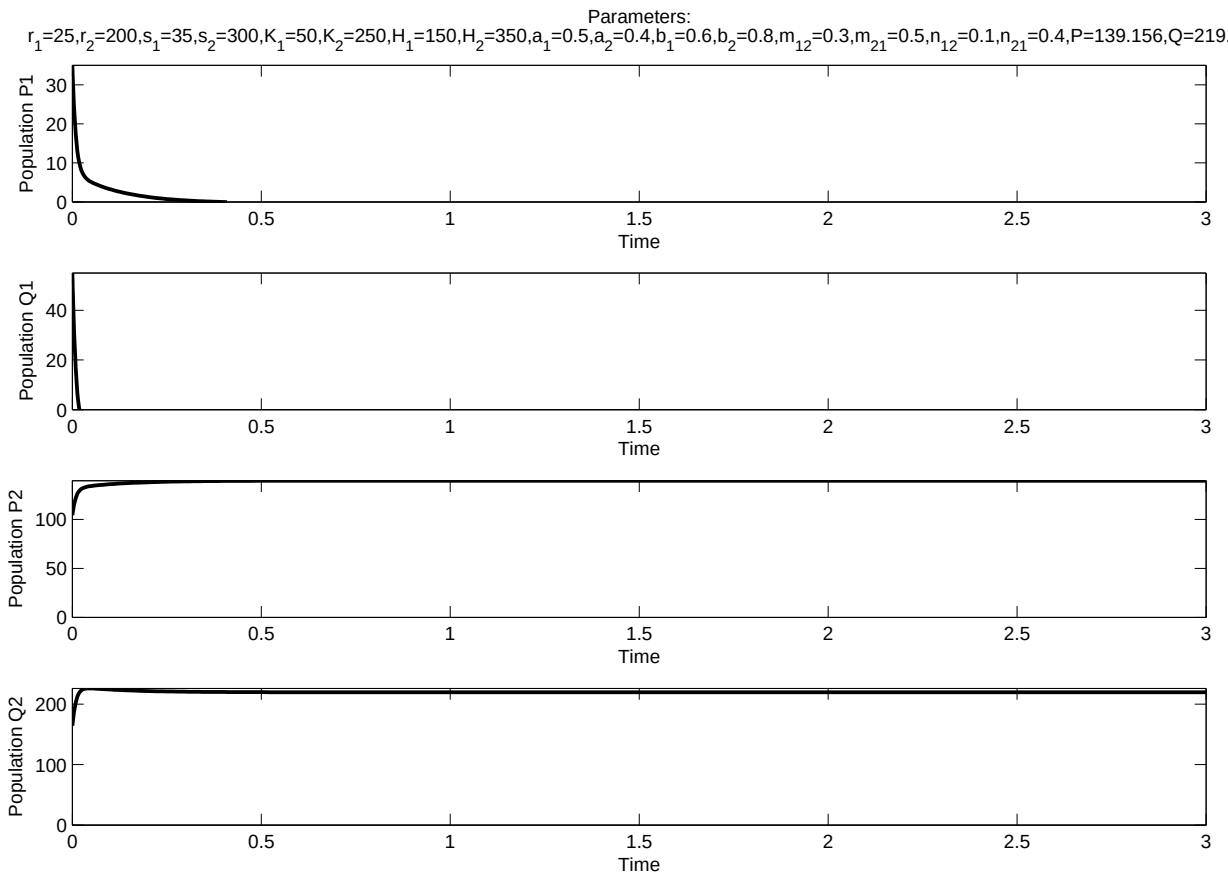


Figure 1: The origin of the P_1—Q_1 phase plane is asymptotically stable for the initial condition $(P_1, Q_1) = (35, 55)$ and parameter values $r_1 = 25$, $r_2 = 200$, $s_1 = 35$, $s_2 = 300$, $K_1 = 50$, $K_2 = 250$, $H_1 = 150$, $H_2 = 350$, $a_1 = 0.5$, $a_2 = 0.4$, $b_1 = 0.6$, $b_2 = 0.8$, $m_{12} = 0.3$, $m_{21} = 0.5$, $n_{12} = 0.1$, $n_{21} = 0.4$ with $P = 139.156$ and $Q = 219.188$.

 ISBN:978-84-615-5392-1

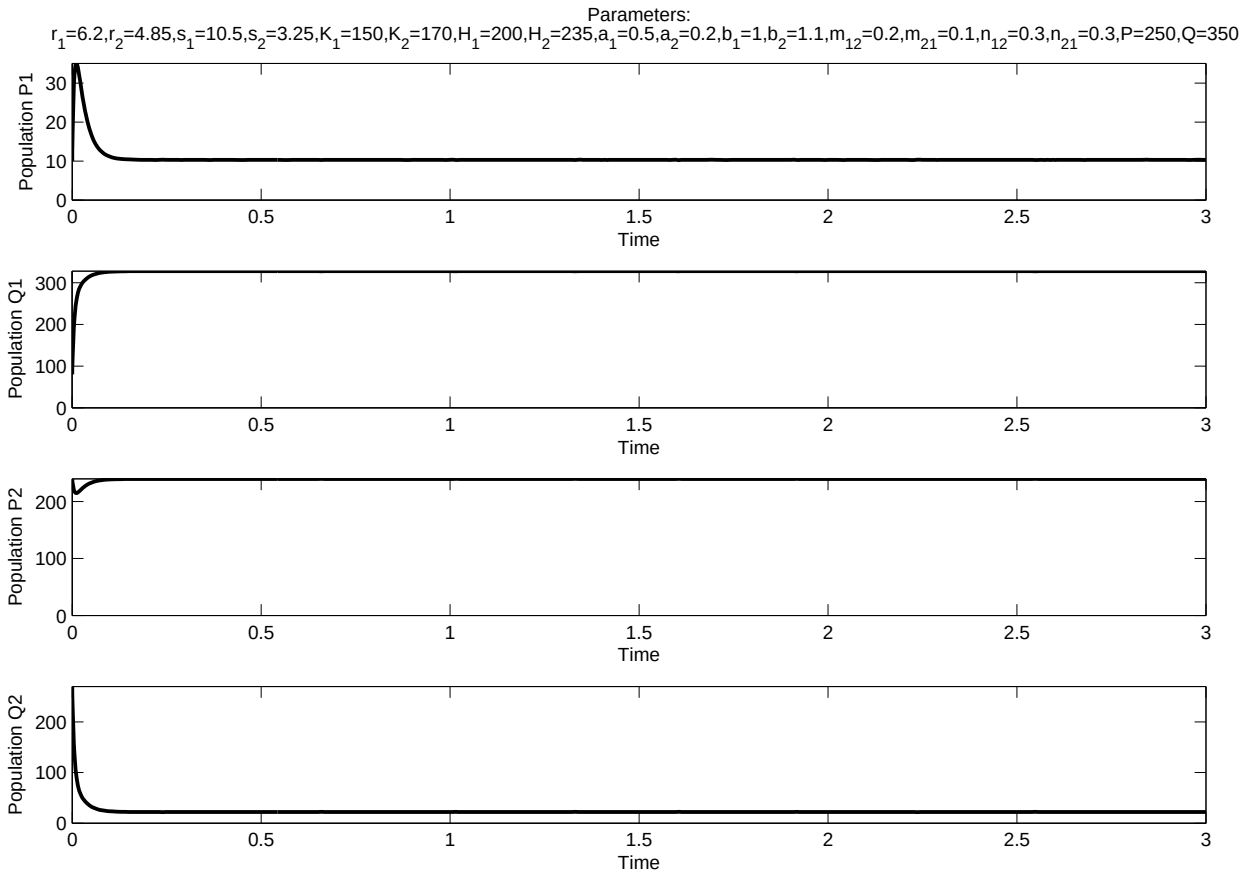


Figure 2: The coexistence equilibrium $(P_1, Q_1) = (9.85, 328.4)$ is stable for the initial condition $(P_1, Q_1) = (10, 80)$ and $r_1 = 6.2$, $r_2 = 4.85$, $s_1 = 10.5$, $s_2 = 3.25$, $K_1 = 150$, $K_2 = 170$, $H_1 = 200$, $H_2 = 235$, $a_1 = 0.5$, $a_2 = 0.2$, $b_1 = 1$, $b_2 = 1.1$, $m_{12} = 0.2$, $m_{21} = 0.1$, $n_{12} = 0.3$, $n_{21} = 0.3$ with $P = 250$ and $Q = 350$. Note that using these parameters the sufficient condition (10) for the existence of a Dulac function does not hold.

 ISBN:978-84-615-5392-1

[7] H. Malchow, S. Petrovskii, E. Venturino, *Spatiotemporal patterns in Ecology and Epidemiology*, Boca Raton: CRC, 2008.

[8] A. Moilanen, I. Hanski, *Habitat destruction and competitive coexistence in a spatially realistic metapopulation model*, Journal of Animal Ecology **64** (1995) 141-144.

[9] A. Moilanen, A. Smith, I. Hanski, *Long-term dynamics in a metapopulation of the American pika*, American Naturalist **152** (1998) 530–542.

[10] R. L., Schooley, L. C. Branch, *Spatial heterogeneity in habitat quality and cross-scale interactions in metapopulations*, Ecosystems **10** (2007) 846-853.

[11] J. A. Wiens, *Wildlife in patchy environments: metapopulations, mosaics, and management*, in D. R. McCullough (Ed.) Metapopulations and Wildlife Conservation, Washington: Island Press, 53–84, 1996.

[12] J. A. Wiens, *Metapopulation dynamics and landscape ecology*, in I. A. Hanski, M. E. Gilpin (Ed.s) San Diego: Academic Press, 43–62, 2007.

[13] J. Wu, *Modeling dynamics of patchy landscapes: linking metapopulation theory, landscape ecology and conservation biology*, in Yearbook in Systems Ecology (English edition) Beijing: Chinese Academy of Sciences, 1994.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

On Optimal Allocation of Redundant Components for Systems of Dependent Components

Félix Belzunce Torregrosa[1], Helena Martínez Puertas[2] and José María Ruíz Gómez[1]

[1] *Department of Statistics and Operations Research, University of Murcia (Spain).*

[2] *Department of Statistics and Applied Mathematics, University of Almería (Spain).*

emails: belzunce@um.es, hmartinez@ual.es, jmruizgo@um.es

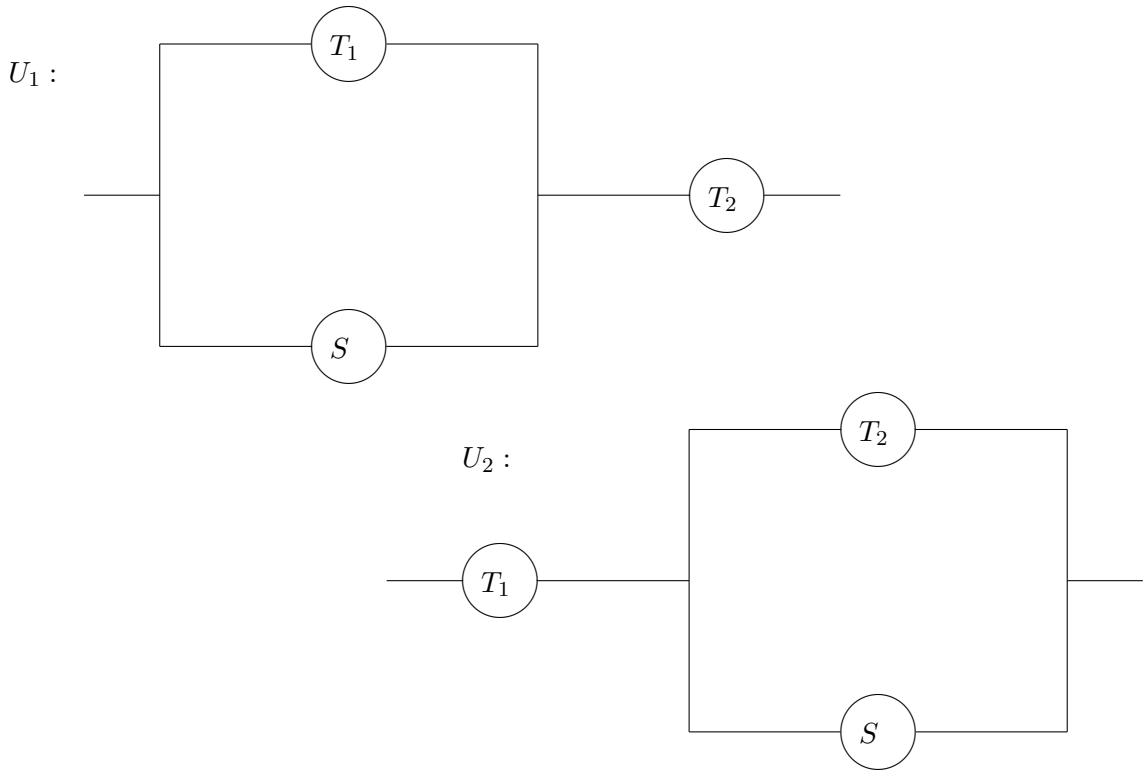
Abstract

We consider the problem of optimal allocation of a redundant component for series, parallel and k- out of- n systems of more than two components, when all the components are dependent. We show that for this problem is naturally to consider multivariates extensions of the joint bivariates stochastic orders. However, these extensions have not been defined or explicitly studied in the literature, except the joint likelihood ratio order, which was introduced by Shanthikumar and Yao (1991).

Key words: Standby and active redundancy, multivariate extensions of the joint stochastic order.
MSC 2000: 60E15, 60K10.

1 Introduction

The problem of where to allocate a redundant component in order to increase the reliability of a system is one of the important problems in reliability. Two types of redundancy, active and standby, are the most common types of redundancy. An active redundant is put in parallel with a component at the same time and a standby redundant component is put to use upon the failure of the original component. As an example consider a series system of two components with random lifetimes $T1$ and $T2$ and let us consider an additional component, with random lifetime S, in parallel redundancy with any of the two components. Then we have two systems $U_1 = \min\{\max(T_1, S), T_2\}$ and $U_2 = \min\{T_1, \max(T_2, S)\}$,



and the problem is which of the two system has a larger lifetime in some probabilistic sense.

This problem has been studied along the 90's, and more recently in Valdés and Zequeira (2003), Romera, Valdés and Zequeira (2004) and Valdés and Zequeira (2006). All these papers consider the case where the components are independent. In the case of dependent components not too much work has been done, and as far as we know, only the papers by Kotz, Lai and Xie (2003), da Costa Buneo (2005) and da Costa Bueno and do Carmo (2007) deals with this problem.

The purpose is to consider this idea when all the components are dependent. Thus, let us consider a system with $n > 2$ components with random lifetimes $T_1, T_2, ..., T_n$ and an additional component, with random lifetime S, that can be put in parallel or standby redundancy with any of the n components. Then, we can consider n diferent systems and the problem is which of the n system has a larger lifetime in some probabilistic sense. First, we discuss the case of series and parallel systems and later we extend the results to the case of k-out-of-n systems. For obtaining the results, previously has been necessary to consider multivariates extensions of the joint bivariates orders introduced by Shanthikumar and Yao (1991) and Shanthikumar, Yamazaki and Sakasegawa (1991).

 ISBN:978-84-615-5392-1

Acknowledgements

Supported by Ministerio de Educación y Ciencia under Grant MTM2009-08311 and by Fundación Senéca (CARM 08811/PI/08).

References

[1] Boland, P.J., El-Neweihi, E. and Proschan, F. (1992). Stochastic order for redundancy allocations in series and parallel systems. *Advances in Applied Probability*. **24**, 161–171.

[2] El-Neweihi, E. and Sethuraman, J. (1993). Optimal allocation under partial ordering of lifetimes of components. *Journal of Applied Probability*. **25**, 914–925.

[3] Mi, J. (1999). Optimal active redundancy allocation in k-out-of-n system. *Journal of Applied Probability*. **36**, 927–933.

[4] Romera, R., Valdés, J.E. and Zequeira, R.I. (2004). Active-redundancy allocations in systems. *IEEE Transactions on Reliability*. **53**, 313–318.

[5] Shanthikumar, J.G., Yamazaki, G. and Sakasegawa, H. (1991). Characterization of optimal order of servers in a tandem queue with blocking. *Operations Research Letters*. **10**, 17–22.

[6] Shanthikumar, J.G. and Yao, D.D. (1991). Bivariate characterization of some stochastic order relations. *Advances in Applied Probability* . **23**, 642–659.

[7] Singh, H. and Misra, N. (1994). On redundancy allocations in systems. *Journal of Applied Probability*. **31**, 1004–1014.

[8] Singh, H. and Singh, R.S. (1997). Note: Optimal allocation of resources to nodes of series systems with respect to failure rate ordering. *Naval Research Logistics*. **44**, 147–152.

[9] Valdés, J.E. and Zequeira, R.I. (2003). On the optimal allocation of an active redundancy in a two-component series system. *Statistics and Probability Letters*. **63**, 325–332.

[10] Valdés, J.E. and Zequeira, R.I. (2006). On the optimal allocation of two active redundancies in a two-component series system. *Operations Research Letters*. **34**, 49–52.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Fixed point techniques and Schauder bases to approximate the solution of the nonlinear Fredholm–Volterra integro–differential equation

M. I. Berenguer[1], D. Gámez[1] and A. J. López Linares[1]

[1] *Department of Applied Mathematics, University of Granada*

emails: maribel@ugr.es, domingo@ugr.es, alopezl@ugr.es

Abstract

With the aid of fixed–point theorem and biorthogonal systems in adequate Banach spaces, the problem of approximating the solution of a nonlinear Fredholm–Volterra integro–differential equation is turned into a numerical algorithm, so that it can be solved numerically.

Key words: Biorthogonal systems, fixed–point, nonlinear Volterra–Fredholm integro-differential equation, numerical methods.
MSC 2000: AMS codes 65R05, 45J05, 45L05, 45N05.

1 Preliminaries

Denoting by $\mathcal{C}([0,1]\times\mathbb{R})$ and $\mathcal{C}([0,1]^2\times\mathbb{R})$ the Banach space of all continuous and real–valued functions defined on $[0,1]\times\mathbb{R}$ and $[0,1]^2\times\mathbb{R}$ respectively, equipped with their usual sup–sup norm, let us consider the following problem associated to the Fredholm–Volterra integro–differential equation: given $\rho\in\mathbb{R}$, $k_1,k_2\in\mathcal{C}([0,1]^2\times\mathbb{R})$, and $f\in\mathcal{C}([0,1]\times\mathbb{R})$, find $x\in\mathcal{C}^1([0,1])$ such that

$$\begin{cases} x'(t)=f(t,x(t))+\displaystyle\int_0^1 k_1(t,s,x(s))ds+\int_0^t k_2(t,s,x(s))ds \qquad ((t,s)\in[0,1]^2) \\ x(0)=\rho. \end{cases} \tag{1}$$

Frequently the mathematical modelling of problems arising from de real world (see [17] and the references there are in) deal with problem (1). These are usually difficult

to solve analytically and in many cases the solution must be approximated (see [9], [10], [14]-[16]). The use of fixed–point techniques in the numerical study of linear and nonlinear differential, integral and integro–differential equations has also proven successful in some works, as [2]–[8], [11] and [12]. The purpose of this work is to develop an effective method for approximating the solution of (1) using Schauder basis and another classical tool in Analysis: the fixed–point theorem. This algorithm generalizes the developed ones in [2], [6], [3] and [12] for linear Fredholm–Volterra integro–differential, nonlinear Volterra integro–differential, nonlinear Fredholm integro–differential and nonlinear differential equation respectively.

To establish our numerical method, we first need to review some results of a theoretical nature in section 2. We arrive at a numerical method for approximating the solution of (1) in section 3, and in order to state the results about convergence and to study the error of the proposed algorithm, we will assume that k_1, k_2 and f satisfying a Lipschitz condition with respect to the last variables: there exist $L_f, L_{k_1}, L_{k_2} \geq 0$ such that

$$|f(t,y) - f(t,z)| \leq L_f|y-z|$$

$$|k_1(t,s,y) - k_1(t,s,z)| \leq L_{k_1}|y-z|$$

$$|k_2(t,s,y) - k_2(t,s,z)| \leq L_{k_2}|y-z|$$

for $t, s \in [0,1]$ and for $y, z \in \mathbb{R}$, with $M := L_f + L_{k_1} + \frac{L_{k_2}}{2} < 1$. Finally, in section 4 we illustrate the theoretical results with a example.

2 Two tools of a theoretical nature.

Two fundamental tools will be used to establish the algorithm needed to solve the problem (1). The first is the Banach fixed–point theorem (see [1]):

Banach fixed–point theorem. *Let $(X, \|\cdot\|)$ be a Banach space, let $F : X \longrightarrow X$ and let $\{\gamma_n\}_{n\geq 1}$ be a sequence of nonnegative real numbers such that the series $\sum_{n\geq 1}\gamma_n$ is convergent and for all $y, z \in X$ and for all $n \geq 1$, $\|F^n y - F^n z\| \leq \gamma_n \|y - z\|$. Then F has unique fixed point $x \in X$. Moreover, if $\overline{x}$ is an element in X, then we have that for all $n \geq 1$,*

$$\|F^n\overline{x} - x\| \leq (\sum_{i=n}^{\infty}\gamma_i)\|F\overline{x} - \overline{x}\|.$$

In particular, $x = \lim_n F^n(\overline{x})$.

The second tool applied consists of biorthogonal systems in Banach spaces $\mathcal{C}([0,1])$ and $\mathcal{C}([0,1]^2)$ (see [13] and [18]). We will make use of the usual Schauder basis for simplicity in the exposition, although the numerical method given works equally well by replacing it with any complete biorthogonal system in $\mathcal{C}([0,1]^2)$. Let us consider the usual Schauder

bases $\{b_n\}_{n\geq 1}$ in the space $\mathcal{C}([0,1])$ and $\{B_n\}_{n\geq 1}$ in $\mathcal{C}([0,1]^2)$ and $\{P_n\}_{n\geq 1}$ and $\{Q_n\}_{n\geq 1}$ the sequences of (continuous and linear) *projections* in $\mathcal{C}([0,1])$ and $\mathcal{C}([0,1]^2)$ respectively. Let it be $\{b_n^*\}_{n\geq 1}$ and $\{B_n^*\}_{n\geq 1}$ the associated sequence of (continuous and linear) *coordinate functionals* in $\mathcal{C}([0,1])$ and $\mathcal{C}([0,1]^2)$ respectively.

3 The numerical method. Study of convergence and error.

Our starting point is the formulation of (1) in terms of a certain operator L as follows: let $T:\mathcal{C}([0,1])\longrightarrow\mathcal{C}([0,1])$ be the linear and continuous operator defined by

$$Tx(t):=\rho+\int_0^t f(u,x(u))\,du+\int_0^t\int_0^1 k_1(u,s,x(s))\,ds\,du+\int_0^t\int_0^u k_2(u,s,x(s))\,ds\,du$$
$$(0\leq t\leq 1,\ x\in C([0,1])).$$

It is a simple matter to check that a function $x\in\mathcal{C}^1([0,1])$ is the solution of (1) if and only if x is a fixed point of the operator T.

The condition assumed on M and the Banach fixed point theorem allow us to establish the existence of one and only one solution x of (1), which is

$$x=\lim_m T^m(\overline{x}).$$

On the other hand, using Schauder's bases introduced in the section 2, let us consider the functions $\varphi\in\mathcal{C}([0,1])$ and $\phi_1,\phi_2\in\mathcal{C}([0,1]^2)$, defined respectively by

$$\varphi(t)=f(t,x(t)),$$
$$\phi_1(t,s)=k_1(t,s,x(s)),$$
$$\phi_2(t,s)=k_2(t,s,x(s)).$$

Let $\{\delta_n\}_{n\geq 1},\{\mu_n\}_{n\geq 1},\{\nu_n\}_{n\geq 1}$ be the sequences of scalars satisfying $\varphi=\sum_{n\geq 1}\delta_n b_n,\ \phi_1=\sum_{n\geq 1}\mu_n B_n,\ \phi_2=\sum_{n\geq 1}\nu_n B_n$. Then for all $t\in[0,1]$ we have that

$$(Tx)(t)=\rho+\sum_{n\geq 1}\delta_n\int_0^t b_n(u)\,du+\sum_{n\geq 1}\mu_n\int_0^t\int_0^1 B_n(u,s)\,ds\,du+\sum_{n\geq 1}\nu_n\int_0^t\int_0^u B_n(u,s)\,ds\,du \tag{2}$$

where

$$\delta_1=\varphi(t_1);\quad\text{and for}\quad n\geq 2,\quad \delta_n=\varphi(t_n)-\sum_{k=1}^{n-1}b_k^*(\varphi)b_k(t_n)$$

and

$$\mu_1=\phi_1(t_1,t_1),\quad \nu_1=\phi_2(t_1,t_1)$$

 ISBN:978-84-615-5392-1

and for $n \geq 2$,

$$\mu_n = \phi_1(t_i, t_j) - \sum_{k=1}^{n-1} B_k^*(\phi_1) B_k(t_i, t_j),$$

$$\nu_n = \phi_2(t_i, t_j) - \sum_{k=1}^{n-1} B_k^*(\phi_2) B_k(t_i, t_j), \quad \text{with} \quad \tau(n) = (i, j),$$

where

$$\tau(n) := \begin{cases} (\sqrt{n}, \sqrt{n}), & \text{if} \quad [\sqrt{n}] = \sqrt{n} \\ (n - [\sqrt{n}]^2, [\sqrt{n}] + 1), & \text{if} \quad 0 < n - [\sqrt{n}]^2 \leq [\sqrt{n}] \\ ([\sqrt{n}] + 1, n - [\sqrt{n}]^2 - [\sqrt{n}]), & \text{if} \quad [\sqrt{n}] < n - [\sqrt{n}]^2 \end{cases}$$

and for a real number p, $[p]$ will denote its integer part.

We can then calculate iteratively using (2), at least in a theoretical way, the solution of (1). From a practical viewpoint, in general these calculations are not possible explicitly. The idea of our numerical method is to use an appropriates Schauder basis in the spaces $\mathcal{C}([0,1])$, $\mathcal{C}([0,1]^2)$ truncating the functions of such spaces by means of the projections of the Schauder bases. Specifically, in view of (2), we consider the sequence $\{x_r\}_{r\geq 0}$ defined as follows:

Let $x_0(t) := \overline{x}(t) \in C^1([0,1])$ and $m \in \mathbb{N}$. Define inductively, for $r \in \{1, \ldots, m\}$ and $0 \leq t, s \leq 1$ the functions:

$$\varphi_{r-1}(t) := f(t, x_{r-1}(t))$$

$$\sigma_{r-1}(t, s) := k_1(t, s, x_{r-1}(s))$$

$$\psi_{r-1}(t, s) := k_2(t, s, x_{r-1}(s))$$

$$x_r(t) := \rho + \int_0^t P_{n_r}(\varphi_{r-1}(u))\, du + \int_0^t \int_0^1 Q_{n_r^2}(\sigma_{r-1}(u, s))\, ds\, du + \int_0^t \int_0^u Q_{n_r^2}(\psi_{r-1}(u, s))\, ds\, du,$$

where $n_r \in \mathbb{N}$ with $n_r \geq 2$.

The following result show that the sequence $\{x_r\}$ approximates the solution of (1) and give the error. For this, let us assume that $k_1,\ k_2 \in \mathcal{C}^1([0,1]^2 \times \mathbb{R})$, $f \in \mathcal{C}^1([0,1] \times \mathbb{R})$ such that for each $i \in \{1, 2\}$, $k_i,\ \frac{\partial k_i}{\partial t},\ \frac{\partial k_i}{\partial s},\ \frac{\partial k_i}{\partial x},\ f,\ \frac{\partial f}{\partial t},\ \frac{\partial f}{\partial x}$ satisfy a global Lipschitz condition in the last variable.

Theorem. *Let $m \in \mathbb{N}, n_r \in \mathbb{N},\ n_r \geq 2$ and, $\{\varepsilon_1, \ldots, \varepsilon_m\}$ a set of positive numbers such that for all $r \in \{1, \ldots, m\}$*

$$\Delta T_{n_r} \leq \frac{\varepsilon_r}{2\Lambda_{r-1}(\beta + 3\beta^2)}$$

where

$$\Lambda_{r-1} := \max\left\{\|\varphi'_{r-1}\|,\ \left\|\frac{\partial\sigma_{r-1}}{\partial t}\right\|,\ \left\|\frac{\partial\sigma_{r-1}}{\partial s}\right\|,\ \left\|\frac{\partial\psi_{r-1}}{\partial t}\right\|,\ \left\|\frac{\partial\psi_{r-1}}{\partial s}\right\|\right\}$$

Then

$$\|Tx_{r-1} - x_r\| \leq \varepsilon_r.$$

Moreover, if x is the exact solution of the integro–differential equation (1), then the error $\|x - x_m\|$ is given by

$$\|x - x_m\| \leq \frac{M^m}{1-M}\|T\overline{x} - \overline{x}\| + \sum_{r=1}^{m} M^{m-r}\varepsilon_r.$$

4 A numerical example

We consider the following Fredholm-Volterra integro-differential equation with the exact solution $x(t) = t^2$. Its numerical results are given in following table.

$$\begin{cases} x'(t) = \dfrac{1}{120}(10 + 180t - 15t^3 - 6t^5)t + \dfrac{1}{8}tx(t) \\ \qquad\qquad + \displaystyle\int_0^1 (2 - 3(s+t) + 6st)x(s)ds + \int_0^1 \frac{st}{5}x(s)ds \quad ((t,s) \in [0,1]^2) \\ x(0) = 0. \end{cases}$$

We have fixed the subset $\{t_i\}_{i\geq 1}$ chosen for constructing the Schauder basis $\{b_n\}_{n\geq 1}$ in $\mathcal{C}([0,1])$ and $\{B_n\}_{n\geq 1}$ in $\mathcal{C}([0,1]^2)$, specifically, $t_1 = 0$, $t_2 = 1$; and for $n \in \mathbb{N}\cup\{0\}$, $t_{i+1} = \frac{2k+1}{2^{n+1}}$ if $i = 2^n + k + 1$ where $0 \leq k < 2^n$ are integers. To define the sequence $\{x_r\}$, we take $n_j = i$ (for all $j \geq 1$). In addition we include, a table exhibiting, for $i = 9, 17$ and 33, the absolute errors committed for certain representative points of $[0,1]$ when we approximate the exact solution $x(t)$ by means of the iteration $x_r(t)$.

Table. Absolute errors for the example

	$i = 9$	$i = 17$	$i = 33$
t	$\lvert x_4(t) - x(t)\rvert$	$\lvert x_4(t) - x(t)\rvert$	$\lvert x_4(t) - x(t)\rvert$
0.125	7.63E−4	1.84E−4	4.04E−5
0.250	1.25E−3	3.02E−4	6.42E−5
0.375	1.47E−3	3.52E−4	7.12E−5
0.5	1.42E−3	3.35E−4	6.11E−5
0.625	1.10E−3	2.47E−4	3.36E−5
0.750	4.84E−4	8.88E−5	1.15E−5
0.875	4.30E−4	1.44E−4	7.47E−5
1	1.65E−3	4.54E−4	1.56E−4

 ISBN:978-84-615-5392-1

Acknowledgements

Research partially supported by Junta de Andalucía Grant FQM359 and the E.T.S.I.E. of the University of Granada.

References

[1] K. Atkinson and W. Han, *Theoretical Numerical Analysis*, 2nd Ed., Springer-Verlag, New York, 2005.

[2] M. I. Berenguer, D. Gámez, A. J. López Linares, *Fixed point iterative algorithm for the linear Fredholm-Volterra integro-differential equation*, Journal of Applied Mathematics (article in press).

[3] M.I. Berenguer, M.V. Fernández Muñoz, A.I. Garralda-Guillem and M. Ruiz Galn, *A sequential approach for solving the Fredholm integro-differential equation*, App. Numer. Math. 62 (2012), pp. 297-304.

[4] M. I. Berenguer, A. I. Garralda-Guillem, M. Ruiz Galán, *An approximation method for solving systems of Volterra integro-differential equations*, Appl. Numer. Math. (2011), doi: 10.1016/j.apnum.2011.03.007 (article in press).

[5] M.I. Berenguer, D. Gámez, A.I. Garralda Guillem, M. Ruiz Galán and M. C. Serrano Pérez, *Biorthogonal systems for solving Volterra integral equation system of the second kind*, J. Comput. Appl. Math. 235 (2011), pp. 1875–1883.

[6] M.I. Berenguer, A.I. Garralda Guillem and M. Ruiz Galán, *Biorthogonal systems approximating the solution of the nonlinear Volterra integro–differential equation*, Fixed Point Theory A. 2010 (2010), doi:10.1155/2010/470149. Article ID 470149, 9 pages.

[7] M.I. Berenguer, D. Gámez, A.I. Garralda Guillem, and M. C. Serrano Pérez, *Nonlinear Volterra integral equation of the second kind and biorthogonal systems*, Abstr. Appl. Anal. 2010 (2010) doi:10.1155/2010/135216. Article ID 135216, 11 pages.

[8] M.I. Berenguer, M. V. Fernández Muñoz, A.I. Garralda Guillem and M. Ruiz Galán, *Numerical treatment of fixed point applied to the nonlinear Fredholm integral equation*, Fixed Point Theory A. 2009 (2009), doi:10.1155/2009/735638. Article ID 735638, 8 pages.

[9] M. Dehghan and R. Salehi, *The numerical solution of the non–linear integro–differential equations based on the meshless method*, J. Comput. Appl. Math. 236 (2012), pp. 2367–2377.

[10] Fariborzi Araghi, M. A. and Sadigh Behzadi Sh. *Solving nonlinear Volterra–Fredholm integro–differential equations using He's variational iteration method*, Int. J. Comput. Math. 88 (2011), pp. 829–838.

[11] D. Gámez, A.I. Garralda Guillem and M. Ruiz Galán, *High order nonlinear initial–value problems countably determined*, J. Comput. Appl. Math. 228 (2009), pp. 77–82.

[12] D. Gámez, A.I. Garralda Guillem and M. Ruiz Galán, *Nonlinear initial–value problems and Schauder bases*, Nonlinear Anal.-Theor. 63 (2005), pp. 97–105.

[13] B. Gelbaum and J. Gil de Lamadrid, *Bases on tensor products of Banach spaces*, Pacific J. Math. 11 (1961), pp. 1281–1286.

[14] K. Malekejad, B. Basirat and E. Hashemizadeh, *Hybrid Legendre polynomials and Block–Pulse functions approach for nonlinear Volterra-Fredholm integro–differential equations*, Comput. Math. Appl. 61 (2011), pp. 2821–2828.

[15] K. Maleknejad, E. Hashemizadeh and B. Basirat, *Computational method based on Berstein operational matrices for nonlinear Volterra–Fredholm–Hammerstein integral equations*, Commun. Nonlinear Sci. 17 (2012), pp. 52–61.

[16] K. Parand and J. A. Rad, *Numerical solution of nonlinear Volterra–Fredholm–Hammerstein integral equations via collocation method based on radial basis functions*, Appl. Math. Comput. 218 (2012), pp. 5292–5309.

[17] M. Rahman, Z. Jackiewicz and B. D. Welfert, *Stocastic approximations of perturbed Fredholm Volterra integro-differential equation arising in mathematical neurosciences*, Appl. Math. Comp. 186 (2007), 1173-1182.

[18] Z. Semadeni, *Product Schauder bases and approximation with nodes in spaces of continuous functions*, Bull. Acad. Polon. Sci. 11 (1963), pp. 387–391.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Job Scheduling in Hadoop Non-dedicated Shared Clusters

Aprigio Bezerra[1], Porfidio Hernández[1], Antonio Espinosa[1] and Juan Carlos Moure[1]

[1] *Department of Computer Architecture and Operating Systems, University Autonoma de Barcelona*

emails: abezerra@caos.uab.es, porfidio.hernandez@uab.es, antoniomiguel.espinosa@uab.es, juancarlos.moure@uab.es

Abstract

We describe the analysis of a Hadoop map-reduce programming platform running on a non-dedicated computer cluster. In the work presented, we have a group of local applications sharing the computational resources with additional Hadoop applications. Our proposal describes the use of shared resources to execute read-mapping bioinformatics Hadoop applications. To do so, we must ensure that local applications performance is not affected. Also, the effective use of a Hadoop platform on a non-dedicated system needs a special analysis and configuration of a list of parameters. One of the most relevant aspects to consider is the job scheduling policy to allow the co-scheduling of several instances of Hadoop applications.

Key words: map-reduce, job scheduling, bioinformatics, non-dedicated cluster

1 Introduction

Recent advances in different scientific disciplines are generating vast amounts of data that must be stored and analysed efficiently. In this sense, it is interesting to consider the impact of Next Generation Sequencing technologies that provide whole genomes as the result of each experimentation.

Large data repositories are usually built on top of computer clusters. In this way, we can use the added computing and storage capacity of all the individual computers. This kind of systems have become the most common source of computing found in the majority of scientific laboratories.

In many institutions, specially universities and research centers, we can find computer workstation networks that are shared by different users as non-dedicated environments. That is, systems run well-known local tasks usually with a tight schedule. This is the case of computer laboratories devoted to teaching activities in university schools. Such non-dedicated resources are also used to run scientific applications in idle times using tools like Condor [1] or by the implementation of a social contract that does not compromise the resources needed by the local applications.

Our main goal is to evaluate the usage of such non-dedicated cluster to execute data intensive parallel applications without disturbing the system response to local applications. We are going to evaluate the performance of a system when running a well defined set of local load together with an additional set of data intensive Hadoop [2] applications. We are using control groups [3] to make reservations of needed computational resources. A new Hadoop job scheduler will be presented to select Hadoop tasks to be executed from a queue of pending jobs. The policy implemented by the scheduler will analyze the available resources of the non-dedicated cluster and the properties of the launched additional Hadoop tasks to improve resource usage and execution times.

In section 2, we are describing the background of the work, namely mapReduce environments like Hadoop and bioinformatics applications that run on Hadoop. Then, in section 3, we describe the applications considered as local and additional load. Finally, in section 4 we are showing the experiments done and, in section 5 some conclusions obtained.

2 Bioinformatics Hadoop applications and related work

During the last years, programming paradigms and computing platforms needed by web and High Performance Computing (HPC) applications are converging. The usage of highly distributed NoSQL databases like Cassandra [5] or Hbase [4] for many large dataset operations in web applications with millions of users has required the use of computer clusters to store the data. These web applications also have promoted the use of new programming environments that facilitate the task of processing large amounts of user data in parallel.

As a consequence, new cloud scientific applications, specially in the bioinformatics field are adopting the same tools and environments to solve large data set processing problems like genome sequencing and alignment [6].

From this point of view, if we evaluate the traditional parallel programming environments, application designers have a list of responsibilities to face:

- Decide problem granularity by decomposing the problem in parts
- Task allocation in each node of the computer system
- Coordinate synchronization of different processes in execution

- Fault tolerance policies

Some of these tasks need a certain detailed knowledge of the hardware aspects of the computing system in use. Therefore, parallel applications are usually closely coupled to the hardware where they run on and make it difficult to port applications to other platforms. Programming models like MPI and OpenMP provide a certain level of abstraction over the communication and synchronization operations. Unfortunately, programmers still need to confront many of the design decisions mentioned above. Additionally, programming models are focused on CPU-intensive problems and do not provide a good support for large input data processing.

As a solution to these needs, Google proposed using the map-reduce paradigm [7] focused on processing large input data problems. Map-reduce programs are automatically parallelized and executed in a computer cluster. At runtime, the system provides solutions to each of the parallel application responsibilities. For instance, it applies a predefined input data partitioning policy, a scheduling of the program tasks into the different computer nodes, a fault tolerance management policy, and a transparent communication pattern between the different tasks of the application. With this model, programmers without much experience in distributed environments can effectively use a potentially large system.

Map-reduce is a programming paradigm specially well suited for data parallel applications. Programmers must specify the flow needed for processing input data using two functions: map and reduce. Input data are first processed in blocks to generate intermediate key-value pairs with the use of map tasks. Then, all keys are automatically sorted and merged to allow each reduce task to receive all values generated for a certain sorted key set. Then, values are processed and outputs are finally generated.

Hadoop [2] is a highly configurable map-reduce framework developed as an Apache project. It is implemented in Java and is composed of two main subsystems: HDFS distributed file system and the Hadoop task running system. Both are designed using a master-worker architecture where a master coordinates the local tasks performed by workers.

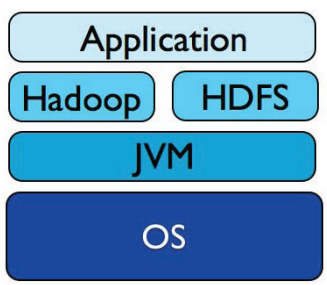


Figure 1: Hadoop main components overview.

 ISBN:978-84-615-5392-1

The specific behavior of Hadoop is managed with the help of a set of more than a hundred configuration parameters. We can adapt the execution conditions of the applications to the actual system in use by providing specific values. In fig. 1 we show the stack of software in the system when a Hadoop application is running. In this case, when considering the configuration of the system we need to address a different set of variables in each of the levels of the stack. Then, we need to identify which variables are relevant for our applications and which values provide the best performance. We will follow a top-down approach with the objective of generalizing relevant aspects to the inner layers of the stack.

An example of a parameter to analyse is the scheduling policy. It should be adapted to co-schedule tasks for many instances of the same application at the Hadoop level. We also will define a set of non-dedicated computational resources to be used by the Java Virtual Machine and the Hadoop applications that will change during the day. User defined job scheduling policies can be used in Hadoop just by configuring a module of the system. By default, job scheduling is done dynamically. Jobs are divided into sets of map and reduce tasks which are placed in a task queue. Hadoop scheduler, managed by master Job Tracker, picks available nodes to launch as many local tasks as possible.

Hadoop job scheduling starts when a cluster node is ready to execute a task, Job Tracker receives a heartbeat signal with a status report. Then it looks for a task in the queue that needs a data block located in the available node. If such a task is found, it is sent to the available node. Else, it selects the first task available to be executed. For each user task to be executed, local Task Tracker creates a new Java Virtual Machine to run it. When the task is completed, Task Tracker eliminates the JVM and reports new idle state to the Job Tracker. Hadoop selects jobs by default applying a FIFO policy. For shared clusters, it also provides some alternatives like Capacity [8] and Fair Share [9] policies to ensure that all tasks receive the same amount of resources over time.

Hadoop framework is aimed for clusters of hundreds of nodes where datasets are distributed. Many of the default parameters and policies are not suitable for non-dedicated systems like ours. For example, data replication policies ensure a reliable fault tolerance behavior, as a node failure is a probable event in large systems, rather than pursuing a better performance by good data locality policies.

In summary, we need to adapt policies and parameters of Hadoop when running static workloads on top of smaller sized non-dedicated clusters. We will analyse the impact of the framework parameter adjustments on the performance of the applications to look for efficiency without affecting local applications.

Other existing projects are adapting map-reduce frameworks to use shared computing resources. Projects like Moon [10] have carried out a similar work of adjusting a map-reduce framework to an opportunistic environment. Also, Adapt project [11] proposes the use of placement strategies in order to improve the performance of map-reduce applications where computing nodes enter and leave the system at any time. Purlieus [12] provides a Hadoop

cluster made of virtual machines as a cloud data center.

Many projects have studied the tuning of the Hadoop parameters to improve its performance. Work of Jiong et al [13] tries to generate a good allocation of resources in heterogeneous computing environments by considering relative capacities of the different nodes. Techniques like delay scheduling [14] improves map-reduce locality delaying the assignation of a task to a node waiting for a free slot with a requested data block.

We share with Zhenhua et al [15] the need of a global decision mechanism to allocate data blocks. They propose an integer linear programming technique while we are creating workgroups of similar jobs in terms of their data access.

Finally, Hadoop resource usage studies have also used c-groups for computing resource reservation [16].

For our work, we are considering a constant set of homogeneous non-dedicated computer nodes running a static workload.

3 Running Hadoop jobs in a non-dedicated cluster

Our experimentation analyzes the sharing of computing resources among two different kind of applications. First of all, we describe the characterization of the local applications. Then, we describe the parallel applications we are going to execute concurrently. Finally, we show the effect of the Hadoop adaptation to the non-dedicated system in the execution of a group of Hadoop applications.

First, we need to ensure that local user applications are not affected during their execution. To do that, we are reserving the resources they need with the use of resource pools named containers.

We propose the use of Linux container implementation. The control groups, cgroups, allow us to reserve resources on the cluster nodes. These resources of each node are divided into a subsystems hierarchy and each cgroup defines a specific percentage of use of a subsystem. From there, we can assign applications, users and processes to each cgroup defining how any combination of cluster resources will be shared.

3.1 Local user activity

Local user activity is modeled by a parametrized benchmark suite. In this way, each program defines a percentage of the actual resource consumption: CPU, memory and network bandwidth. We use this parameters to represent typical Best-Effort applications that run locally in our system. We have profiled our teaching laboratories during some weeks to have a list of realistic values for comparing the benchmark execution.

Table 1 shows a characterization of local applications we considered for our work. In this table we define five different profiles of use of local resources, being a selection of NAS

Profile	NAS	%CPU	%MEM	DISK read	DISK write	NET recv	NET sent
A	EP.A (serial)	99	0.2	26	7	1.32	2.06
B	MG.B (serial)	99	42	37	23	3.84	3.42
C	IS.C (serial)	20	95	23000	12000	0.77	0.72
D	CG.A (parallel)	40	55	54	21	1890	1845
E	FT.A (parallel)	63	61	67	17	1025	965

Table 1: Characterization of the local load using NAS applications. Disk metrics are in blocks/sec and network in Bytes/sec

serial and parallel applications. We describe application needs in terms of CPU, Memory, disk and network usage.

3.2 Hadoop bioinformatics applications

In this work, we have used an implementation of MAQ [17] designed to align short read DNA sequences to a reference genome.

Map-reduce applications can be classified depending on the volume of the data processed in the different stages of the execution. MrMAQ [18] is a Reduce input heavy application programmed in Java using Hadoop open source map-reduce implementation. The basic principles of the application are the same as MAQ: to find matches of a list of short reads with a provided genome reference by using a seed and extend algorithm.

The design of MrMAQ is based on previous map-reduce bioinformatics application ideas [6]. In our particular implementation, Map tasks read both reference and read files to generate 28 base pair seeds from both. Reducers receive keys with matching read and reference sequences, known as hits. Then, reducers extend the alignment for all hits found and calculate their alignment qualities. The application returns a list of all short reads, their alignment positions in the reference, and their alignment qualities.

4 Results

To evaluate the performance of local and Hadoop workload processing, we have used a computer cluster consisting of 9 nodes interconnected by a 100 Mbit network. Nodes are dual processor Intel Pentium 3,4 GHz, with 1 GB of memory and 44 GB of disk for each node. Hadoop version used was 0.20.2 with Oracle Java VM 1.6.0-16. All nodes where running Linux version 2.6.31.

Our non-dedicated cluster runs well-known controlled local load. This load varies over time, allocating a few hours to each local application. We are using several cgroups to define different resource usage configurations to be used by each node of the system during the

normal day activities.

To evaluate the effect of this resource reservation in the local load, we have done a series of experiments. We define a mix of local and additional Hadoop load that executes concurrently with the objective of assuring that the local load is not affected when cgroup resources suit the applications needs. Fig. 2 shows results of the resource consumption when both applications are sharing the resources under different reservation values. We describe the applications execution time in percentage of the time of running alone. Each experiment runs on decrementally local reservation conditions. From there, we select those levels that maximize the resource usage for the rest of the analysis.

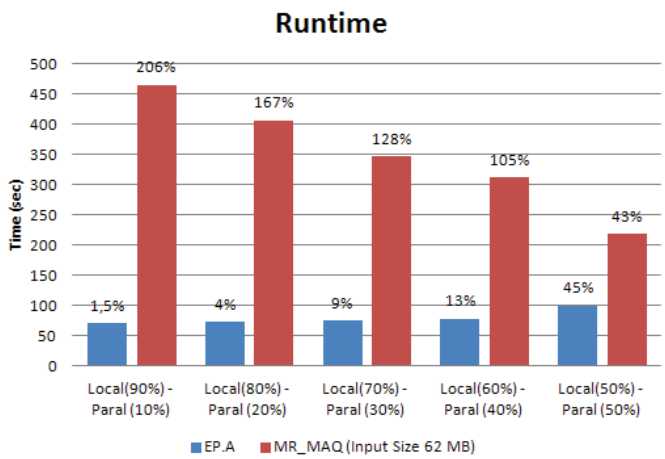


Figure 2: Execution of local and parallel load using cgroup resource reservation

We also analyze the Hadoop application to find the number of map and reduce tasks that minimizes the execution time. In fig 3 we show the execution times of the Hadoop application with 1 Gigabyte input size. From those graphs, we use the number of map and reduce tasks that minimize execution time. In our case, 32 maps and 8 reducers.

Another objective of the work is to evaluate the resources needed by the Hadoop map-reduce applications. For that, we studied the data consumption pattern of mrMAQ, shown in fig 4, in terms of amount of data read and written during the execution. The amount of map output data generated became seven times larger than original input. We call this behavior heavy reduce input. For this kind of data pattern we try to provide as much memory as possible to store intermediate data in Hadoop buffers. Hadoop will save intermediate data to disk every time the buffers fill up, so having larger buffers will avoid spill disk accesses.

Multiple instances of Hadoop applications contend for access to a common set of input files. As those applications are launched, all their map and reduce tasks are available

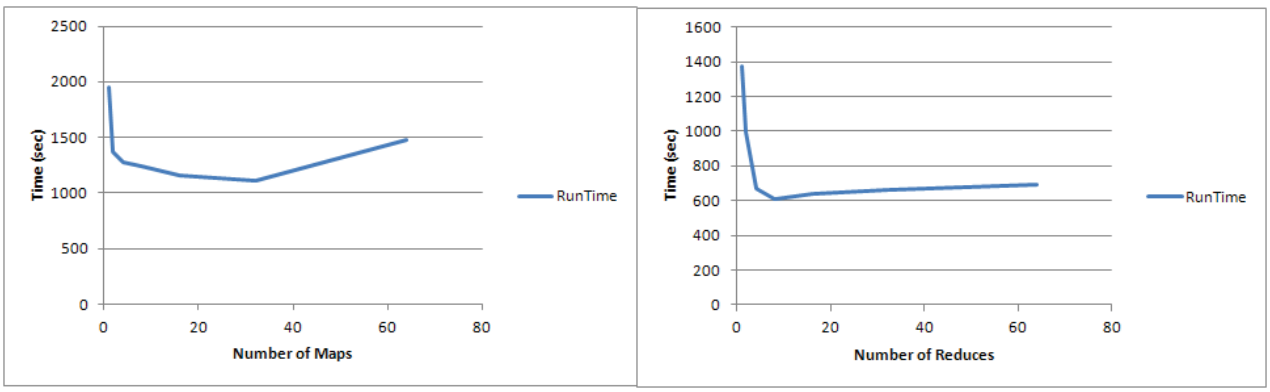


Figure 3: Evaluation of map and reduce tasks of the Hadoop application

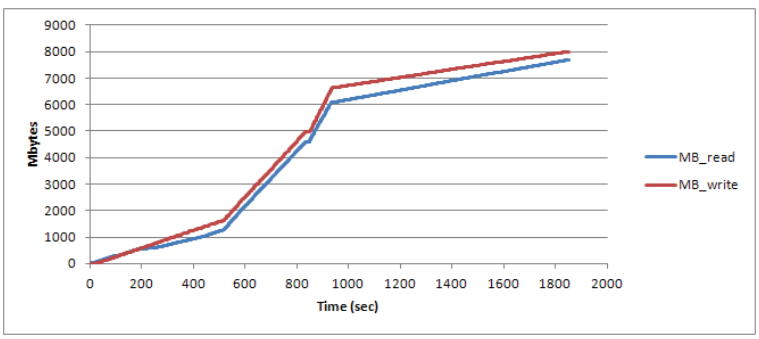


Figure 4: Data read and write consumption profile during the execution

in the Hadoop task queue. Those tasks that request the same input data blocks can be co-scheduled to be executed together. In our case, Hadoop jobs are instances of read-mapping mrMAQ application. All instances that share the same genome reference file form a workgroup. We have developed a new scheduling policy to seek affinity between those tasks in a workgroup that request access to the same input blocks. By this, we want to improve the co-scheduling of tasks that share resources like common input files or need to access the same resource like CPU. A workgroup is treated as a new job entity to be executed by Hadoop.

The scheduling policy implemented follows a simple mechanism. First, a worker Task Tracker sends a heartbeat signal to the master Job Tracker to inform that the node is ready to execute a new task. Then, Job Tracker looks through the tasks of the first job in queue to find a task that is requesting the access to the blocks located in the available node. Then, for all free task slots in the node, the policy also evaluates the jobs queued in search of tasks accessing the same data block. Those tasks will also be assigned to execute in the node.

Our work environment is based on the principle that all applications that share files already form a group before their execution. This assumption is not realistic in most cases. In our cluster, the different instances of the read mapping application arrive at known times and, therefore, can be launched in specific times.

Scheduling system also needs to consider a limited job queue and a job admission module that evaluates the number and size of workgroups that are allowed to be in the queue and launched concurrently. In this way, we can avoid long waiting times for tasks in the execution queue.

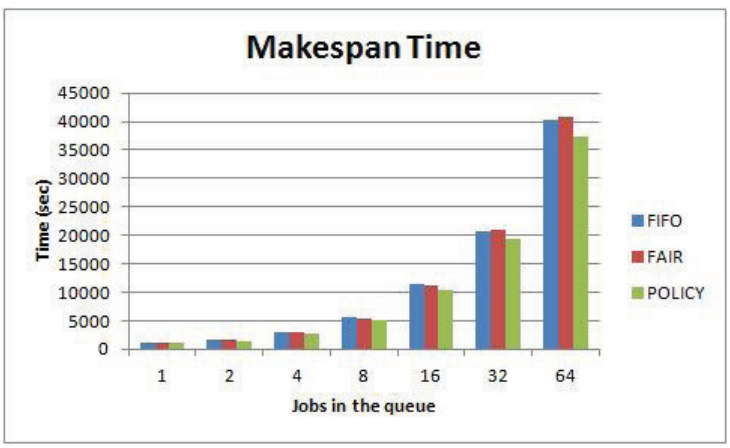


Figure 5: Makespan times of a workgroup execution

Each job submitted to the cluster has an input size of 3.5 GB. Input files were divided into 62 blocks of 64 MB that are distributed between the nodes running HDFS. To evaluate the effects of the block distribution in data locality we did not use block replication. All jobs submitted to the cluster were composed of 62 maps and 14 reducers.

Figure 5 shows the execution times of a workgroup with each job sharing the same genome reference file. We evaluate the execution time increasing the amount of jobs queued from 1 to 64. Makespan time was used as a metric for evaluating the scheduling policy. That is, the total time since the first job was submitted to the cluster until the end of the last execution of job queued.

Scheduling policies compared were FIFO, Fair scheduler and the proposed policy. Fair policy uses a pool of jobs, while the rest use a job queue. In figure 5, we show makespan times obtained by the policies when running an increasing amount of jobs. Results show that the policy proposed improves the average makespan time in 7.8% when compared to FIFO and 8.3% against Fair Scheduler.

In the second set of experiments we used four different reference files. That is, we are executing four workgroups concurrently to compare the execution times using FIFO,

Capacity, and the proposed policy. For the case of Capacity scheduler, we are defining four queues, one for each workgroup. Then, jobs submitted to the cluster were assigned to the corresponding workgroup queue. Cluster resources like available slots were distributed evenly among all queues. The other policies are using a single job queue.

Figure 6 shows makespan times for an increasing amount of jobs belonging to the four workgroups. The number of jobs belonging to each workgroup is divided equally among the number of jobs issued. Results show that the policy proposed improves the average makespan time in 7.3% when compared to FIFO and 3.2% when compared to Capacity scheduler.

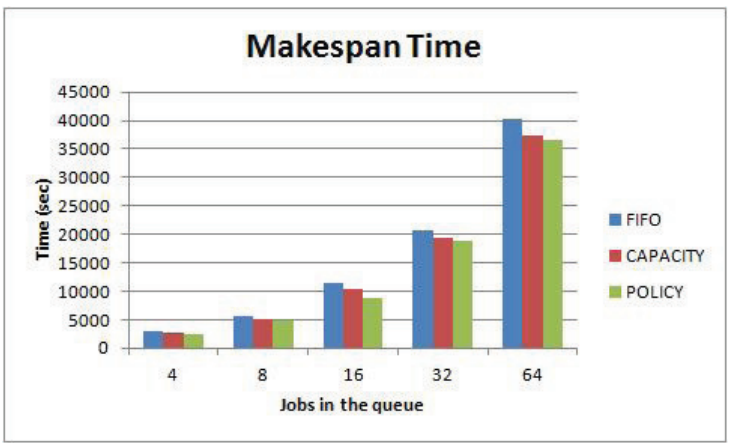


Figure 6: Makespan times of 4 workgroup execution

Both makespan figures show the benefits of applying our framework adaptations to our system. As the Hadoop applications are very dependant on the reduce phase, we need to focus on further improvements on the reducers to get more relevant results.

5 Conclusions and future work

We have analysed the adaptation of Hadoop map-reduce framework to run various instances of bioinformatic applications in a non-dedicated computer system. For that, we have described the application data consumption patterns and presented a new scheduling policy that defines workgroups of applications so that they are co-scheduled together. The execution of these Hadoop bioinformatic applications must be adapted to use existing computing resources so that local applications are not affected. We propose the use of cgroups to make adequate resource reservations during the daily use of our non-dedicated computer system.

Next steps in the research will consider the dynamic definition of local resources and the impact of local resource occupation to the Hadoop application workgroups. Then, the

scheduler will modify its choice of applications to consider tasks that fit better the available resources.

Acknowledgements

This work has been supported by projects number TIN2007-64974 and TIN2011-28689 of Spanish Ministerio de Ciencia y Tecnologia (MICINN).

References

[1] D. Thain, T. Tanembaum, M. Livny, *Distributed Computing in Practice: The Condor Experience*, Concurrency and Computation: Practice and Experience **17** (2005) 323–356.

[2] T. White, *Hadoop. The Definitive Guide*, Second edition. O'Reilly, Sebastopol, 2011.

[3] G. Banga, P. Druschel, J.C. Mogul, *Resource containers: A new facility for resource management in server systems*, Proceedings of OSDI 1999 (1999) 45–58.

[4] L. George, *Hbase. The Definitive Guide*, First edition. O'Reilly, Sebastopol, 2011.

[5] E. Hewitt, *Cassandra. The Definitive Guide*, First edition. O'Reilly, Sebastopol, 2011.

[6] M. Schatz, B. Langmead and S. L. Salzberg, *Cloud Computing and DNA data race*, Nature Biotechnology **28** (2011) 691–693.

[7] J. Dean, S. Gemawat and J. A. Wheeler, *map-reduce: simplified data processing on large clusters.*, ACM Communications **51** (2008) 107–113.

[8] Capacity Scheduler, Tech. rep., Retrieved: February, 2012. `http://hadoop.apache.org/common/docs/r0.20.2/capacity_scheduler.html`

[9] Fair Scheduler, Tech. rep., Retrieved: February, 2012. `http://hadoop.apache.org/common/docs/r0.20.2/fair_scheduler.html`

[10] H. Lin, J. Archuleta, W. Feng, M. Gardner, Z. Zhang, *MOON: map-reduce On Opportunistic eNvironments*, Proc. of the 19th ACM HPCD 2010.

[11] H. Jin, X. Yang, X. Sun, I. Raicu, *ADAPT: Availability-aware map-reduce Data Placement for Non-Dedicated Distributed Computing*, Proc. of ICDCS 2012.

[12] B. Palanisamy, A. Singh, L. Liu, B. Jain, *Purleius: Locality-aware Resource Allocation for map-reduce in a Cloud*, Proc. of ACM/IEEE Conf. on Supercomputing 2011 2011.

[13] J. Xie, S. Yin, X. Ruan, Z. Ding, Y. Tian, J. Majors, A. Manzanares, X. Qin, *Improving map-reduce Performance through Data Placement in Heterogeneous Hadoop Clusters* , Proc. of IEEE IPDPSW 2010.

[14] M. Zaharia, D. Borthakur, J.S. Sarma, K. Elmeleegy, S. Shenker, I. Stoica, *Delay scheduling: a simple technique for achieving fairness in cluster locality and scheduling*, Proc. of the 5th ECCS 2010.

[15] Z. Guo, G. Fox, M. Zhou, *Investigation of Data Locality in mapReduce*, Tech. Report, Indiana University 11/27/2011.

[16] A. Qin, D. Tu, C. Shu, C. Gao, *Xconveyer: Guarantee Hadoop throughput via lightweight OS-level virtualization*, Proc. of IEEE 8th Conf. on Grid and Cooperative Computing 2009.

[17] H. Li, J. Ruan, R. Durbin, *Mapping short DNA sequencing reads and calling variants using mapping quality scores*, Genome Research **18** (2008) 1851–1858.

[18] A. Espinosa, P. Hernandez, J.C. Moure, J. Protasio, A. Ripoll, *Analysis and improvement of map reduce data distribution in read mapping applications*, Journal of Supercomputing. To appear. DOI 10.1007/S11227-012-0792-8

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Ordering and Allocating Parallel Jobs on Multi-Cluster Systems

Héctor Blanco[1], Jordi Lladós[1], Fernando Guirado[1] and Josep Lluís Lérida[1]

[1] *Computer Science Department, Universitat de Lleida*

emails: hectorblanco@diei.udl.cat, jordi.llados@udl.cat,
f.guirado@diei.udl.cat, jlerida@diei.udl.cat

Abstract

The scheduling of jobs in a multi-cluster heterogeneous environment is known as a NP-hard problem, not only for the resource heterogeneity, but also for the possibility of applying co-allocation to take advantage of the greater amount of resources. Previous works in the literature have usually dealt with the co-allocation problem by acting on each jobs, present in the waiting queue, individually. In a previous work, the authors overcome these works by presenting a strategy based on Mixed Integer Programming, which was able to simultaneously allocate all those jobs that fitted into the available resources.

In this paper, the authors present a new algorithm with the power to treat all jobs in the waiting queue as a complete set. The algorithm deals with the job execution order to obtain the fairness allocation, or co-allocation when necessary, that can provide better execution times for all of them.

Key words: Job Scheduling, Multi-Cluster Heterogeneity and Performance, Co-Allocation

1 Introduction

Multi-cluster environments are made up of several clusters of computers, using a dedicated interconnection network with a more predictable performance than in grid environments [1]. In these environments, co-allocation strategies allow jobs to be allocated across different clusters, permitting to execution of those jobs with more requirements than available in each single cluster, thus reducing the internal fragmentation taking advantage of available resources from different clusters, and thus, increasing the job throughput by reducing the

waiting times in the system queue [2]. However, allocating jobs across different clusters can reduce the overall performance when co-allocated jobs contend for the inter-cluster network bandwidth. Moreover, the heterogeneity of the processing and communicating resources notably increases the complexity of the scheduling [2][3][4][5][6] .

A common issue in those previous works is that jobs are treated individually. This means that allocating a job without taking the rest of jobs into account can reduce the performance of future allocations, and could decrease overall system performance [7]. To extend those previous approaches, the authors developed a new scheduling strategy, named *PAS* [8], which selects the best suitable resources by means of a Mixed Integer Programming for a set of jobs that fits the available resources, but without changing the jobs order in the system queue.

The main constraint of the *PAS* strategy is its limitation to act on a set of jobs that fit the available resources without disturbing the arrival order. In the present work, the authors proposed a new strategy called *METL*, for *Minimum Execution Time Loss*, which is able to overcome this limitation considering not only the best allocation, but the order for all the jobs in the waiting queue. This strategy has been tested experimentally and compared with the most common techniques of the literature. The results show that ordering and allocating jobs considering the available resources and their processing- and communicating-requirements, provides best job execution time results than the classic policies.

The rest of the paper is organized as follows. In Section 2, the authors present the strategy for multiple job co-allocation in a multi-cluster environment. Section 3 shows the experimental results. Finally, the conclusions are presented in Section 4.

2 METL Scheduling Policy

In this paper, we consider the parallel jobs following the Bulk-Synchronous Parallel model (BSP) [9] where jobs are made by a fixed number of tasks with similar processing and communication requirements. Under these assumptions the execution time can be expressed as

$$Te_j = Tb_j \cdot [\sigma_j \cdot SP_j + (1 - \sigma_j) \cdot SC_j] \tag{1}$$

where Tb_j denotes the base time of the job j obtained by its execution in dedicated resources, and σ_j denotes the relevance of the processing time with respect to the communication. SC_j and SP_j are the slowdown due to the inter-cluster links and the allocated processing resources respectively. While there is no inter-cluster saturation on the communication links used, $SC_j = 1$. Otherwise, SC_j takes its value from the degree of saturation of the most saturated inter-cluster link used by j, calculated as explained in [10].

A good job co-allocation reduces the network saturation. Previous results [10] had

shown that the allocation of sets of jobs considering both their processing and communication requirements, can be beneficial for the global job's performance.

On the other hand, SP_j determines the effect of the allocated resources on the job execution time. Let R be the set of resources allocated to job j, SP_j is defined as

$$SP_j = \max_{r \in R} \{(\Gamma_r)^{-1}\} \tag{2}$$

where Γ_r be the effective power for each resource $r \in R$. This normalized metric defined in [10] relates the processing power of each resource with its availability, being $\Gamma_r = 1$ when resource $r \in R$ has capacity to run tasks at full speed, and otherwise $\Gamma_r < 1$.

The *METL* policy is able to treat a set of jobs, obtaining their allocation and also their execution order. The way in which the policy is called is an important issue. By calling it every time there is a single job on the waiting queue, it will be impossible to obtain advantage of its ordering capability. On the other hand, trying to allocate a big amount of jobs together could produce bigger waiting times for the jobs, and also the possibility to have unnecessary idle computing resources. By this *METL* is called under the next two assumptions:

1. If there is a single job on the system queue and enough resources, it will be scheduled alone in the most powerful resources but reducing the number of used inter-cluster boundaries.

2. If there are not enough resources to allocate the job, it must wait in the system queue. Before the job will be allocated, it could be possible that other jobs enter to the system queue. *METL* will be called when any resources became free, then all jobs waiting in the system queue become the set of jobs to be treated.

On the next subsections we elaborate the way in which *METL* obtains the job resource allocation and determines their execution order.

2.1 Job Resource Allocation

The main aim of this step is to determine the allocation that can reduce the execution time of a set of jobs, and also to reduce the number of used computational nodes with higher effective power used. There are two steps to do this:

1. Calculating the best allocation: Taking into account the available resources when the policy is applied, the allocation that obtains its minimum execution time is calculated for each individual job. This allocation defines the lower bound execution time.

2. Reducing under-utilised resources. For each of the previous obtained allocations, it is determined those task assignments that do not contribute to reduce the job

execution time. This situation comes from the fact that in the co-allocation process computational nodes r with different effective power (Γ_r) could be allocated, then the processing time for the tasks allocated in the powerful resources will be reduced but not their global execution time, due to communication synchronizations. Thus, those tasks assigned to nodes with higher Γ_r will be moved to other nodes with equal Γ_r than the slowest allocated resources. This re-allocation aids to reduce the inter-cluster links usage and to release the under-utilised resources providing better future allocation opportunities.

Figure 1 shows an example of the job resource allocation procedure. We assume an environment made by two clusters C1={N1,N2} with an effective power $\Gamma_{N1} = \Gamma_{N2} = 0.75$ and C2={N3,N4} with $\Gamma_{N3} = \Gamma_{N4} = 0.5$, which means that cluster C1 is more powerful than C2. For the example, a job made by three tasks ready to be allocated is also supposed. In the figure, the x-axis represents the execution time, and the y-axis the computational nodes, with its respective effective power values. The two steps of the allocation procedure are shown side-by-side.

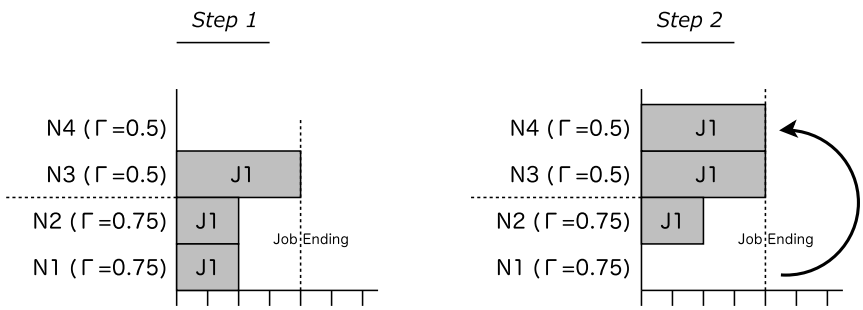


Figure 1: Job Resource Allocation

.

On the left, the first allocating solution obtains the minimum execution time for the job, which are {N1,N2,N3}. However, the final execution time for the job is bounded by the slowest allocated node, which is N3. Thus, the use of the most powerful nodes does not imply the reduction of the job execution time. In this situation, the second step redefines the allocation, as it is shown in the right side, by moving a task from N1 to N4, without penalizing the job execution time and providing better future opportunities.

2.2 Job Allocation Ordering

The main aim of this step is to determine the best order for the set of jobs in the system queue in order to minimize the global execution time. To reach this global optimization with

a fair scheduling for all the jobs, our proposal is to select in each algorithm step the job with the least loss in its execution time considering the status of resources and its availability.

1. Execution time loss calculation. For each job, the difference in execution time with the current resources status, with respect to the obtained in the environment, in dedicated mode, is calculated.

2. Job Selection. The job with the lowest loss in the execution time is selected. If there are no available resources to be allocated, the algorithm estimates the next job to be finished, releases its resources and re-evaluates the jobs waiting in the system queue.

This process is repeated until all jobs in the system queue have been processes, providing for all of them the execution order and allocation. As the set of jobs is treated as a whole, resource starvation is avoided.

2.3 Policy Implementation

The main algorithm was implemented as shown in Algorithm 1, and the final result is the scheduling for all the jobs, consisting on a list with the order in which each job must be executed and also their allocated resources.

The algorithm starts finding the ideal allocation for all the jobs, assuming a dedicated multi-cluster environment (lines 2-4). These allocations will determine the lower execution time bound for each job.

Next, using the function $CalculateAllocation(J, SR)$ (line 9), the allocation for each job considering the current resources availability is calculated. This function is detailed in Algorithm 2, and returns the best possible job allocation with the maximum under-utilised resources. When there are not enough available resources to allocate any job. the algorithm estimates the next job to be finished (line 17), releases its corresponding resources (lines 18-19), and tries to find the most suitable job to be executed under the new conditions.

In order to find the best suitable resources with the minimum underutilization we implemented the Algorithm 2. This algorithm has a list of the set of resources ordered by its effective power. Then, the number of tasks n required by the job is determined (line 2). The first n resources from the set of resources are allocated to the job (line 3). Then, if the job must be co-allocated (line 4), i.e. the number of used clusters is greater than 1, the tasks from the most powerful resources are re-allocated to the slowest cluster (line 5), and the final allocation for the job is returned (line 7).

In order to illustrate how the policy works we show in the rest of this sections an example of the algorithms execution. In this example, we assume a single cluster made up by 5 heterogenous nodes, being their effective powers $\Gamma_{N1}, \Gamma_{N2} = 0.75$, $\Gamma_{N3} = 0.5$, $\Gamma_4 = 0.25$, Γ_{N5}=0.15. The set of jobs waiting to be allocated in the system queue are detailed in Table 1. The example is constructed as iterations over the main algorithm.

Algorithm 1 METL algorithm implementation

```
1: function MainAlgorithm(SJ: Set of jobs, SR: Set of resources)
2:     for all J in SJ do          //Calculate ideal allocations
3:         Ideal_Allocation[J] ← CalculateAllocation(J, SR)
4:     end for
5:     while SJ ≠ ∅ do          //While there are jobs to allocate
6:         min_exec ← ∞
7:         Selected_Job ← NULL
8:         for all J in SJ do          //Calculate real allocations
9:             Allocation[J] ← CalculateAllocation(J, SR)
10:            if Allocation[J] ≠ NULL then          //If the job can be allocated
11:                if min_exec < (Allocation[J] − Ideal_Allocation[J]) then
12:                    min_exec ← (Allocation[J] − Ideal_Allocation[J])
13:                    Selected_Job ← J
14:                end if
15:            end if
16:        end for
17:        if Selected_Job = NULL then          //If no job found that can be allocated
18:            Locate J' in Scheduling_List that finalizes earlier
19:            SR ← SR + Allocation[J']          //Release resources used by J'
20:        else
21:            Scheduling_List ← (Selected_Job, Allocation[Selected_Job])
22:            SJ ← SJ − Selected_Job
23:            SR ← SR − Allocation[Selected_Job]          //Update resources availability
24:        end if
25:    end while
26:    return Scheduling_List
27: end function
```

Algorithm 2 Resource allocation implementation

```
1: function CalculateAllocation(J: Job to treat, SR: Set of resources, ordered by
   effective power)
2:     n ← number of tasks of J
3:     Allocation[J] ← first n nodes from SR
4:     if #Clusters in Allocation[J] > 1 then          //Co-allocation
5:         Move tasks from faster resources to the slowest used cluster
6:     end if
7:     return Allocation[J]
8: end function
```

Iteration 1: First, the ideal and real allocations are calculated. Table 1 shows the Ideal allocation for all jobs, and their estimated execution time. Initially all resources are free, and then, the estimated execution time for each job is the same than the ideal, the differences being 0 in all cases. Thus, the jobs are evaluated in the arrival order to reduce their waiting time, so in this example J1 is the first selected job to be allocated. Next, the resources status is updated, N1 and N2 being unavailable for next allocations.

Job	τ_j	σ_j	Tb_j	Ideal.Exec.Time	Ideal alloc.
J1	2	0.5	100	116.7s	$N1, N2$
J2	3	0.7	50	100s	$N1, N2, N3$
J3	2	0.5	75	87.5s	$N1, N2$
J4	2	0.7	100	123.33s	$N1, N2$

Table 1: Set of jobs in the system queue with the best execution time and allocation in dedicated resources.

Iteration 2: Now, only N3, N4 and N5 are available and jobs J2, J3 and J4 are waiting to be allocated. The best allocation for each waiting job is re-calculated taking into account the available resources. The results are shown in Table 2. As can be seen, J3, which is allocated to N3 and N4, is the job with the lowest loss of time compared to its ideal, so it is the next job to be allocated and their allocated resources status updated.

Job	Execution time	Difference with ideal	Allocation
J2	258.3s	158.3s	$N3, N4, N5$
J3	**187.5s**	**100s**	**N3, N4**
J4	330s	206.7s	$N3, N4$

Table 2: Results for the second iteration. Difference between the ideal and the estimated allocation based on the recent resources status.

Iteration 3: Now, only N5 is available. Neither J2 nor J4 fit the available resources, so the algorithm calculates the first job in execution to be finished. In our case, the first job to finish is J1, releasing the resources N1 and N2. Thus, both J2 and J4 could be allocated and evaluated as in *Iteration 1.* The process will continue until all jobs are finally allocated. Figure 2 shows the resulting scheduling for all the jobs.

As could be observed, the order in which the jobs were executed is different from the original queue. Jobs J3 and J4 were advanced and J2 delayed.

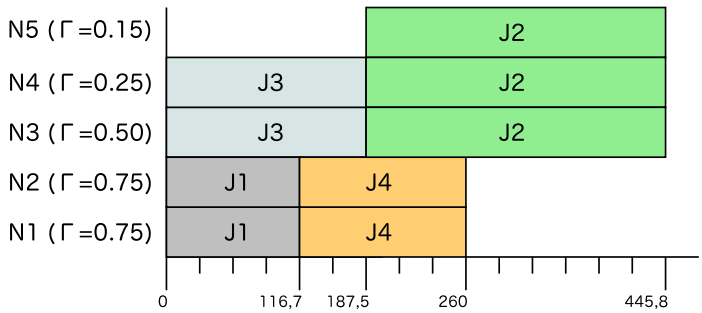


Figure 2: Resulting scheduling for the example

3 Experimentation

The aim of our experimental study was to determine the effectiveness of the scheduling solutions provided by *METL* compared with common scheduling policies from the literature: *First Come First Served* (FCFS), *Short Jobs First* (SJF), *Big Jobs First* (BJF), *Fit Processors First Served* (FPFS), *Short Processing Time* (SPT) and *Long Processing Time* (LPT). The experimentation was carried out by simulation, using the GridSim framework [11], and by characterizing a heterogeneous multi-cluster system as shown in Table 3.

	Num.Nodes	Effective power
Cluster 1	60	1.0
Cluster 2	60	1.5
Cluster 3	60	2.0
Cluster 4	60	1.0

Table 3: Characterization of the experimental environment

In this experimental study the authors used real traces from the HPC2N logs in [12]. A set of three workloads each made up of 15,000 jobs was selected. The majority of the jobs on the there workloads are computational intensive ($\sigma_j = 0.7$), with an average base time of 15,520 seconds and 8.4 tasks per job.

The main aim of the *METL* is to reduce the execution time of the jobs. Accordingly, the average job execution time was used as the comparison metric for the three workloads. The results are shown in Figure 3. The x-axis represents each of the three experimental workloads, and the y-axis represents the average execution time expressed in seconds.

As can be seen, *METL* produces lower average execution times, while the techniques from the literature performed very similarly, producing allocations with higher values.

In order to better analyze the performance of the different techniques, and reveal the

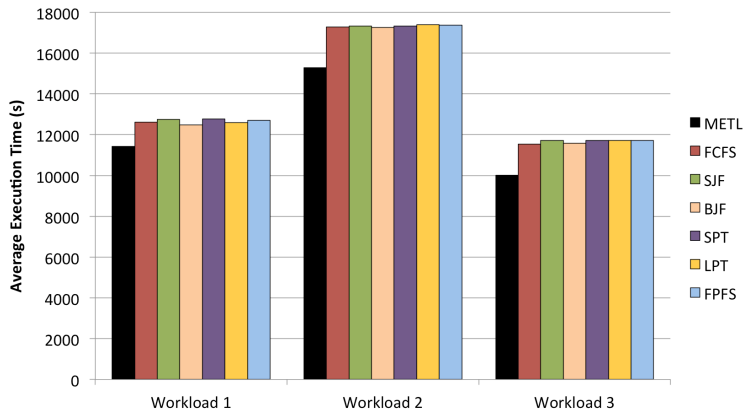


Figure 3: Average Execution Time comparison of three workloads

real effect of the scheduling decisions taken, we also evaluate the differences between the estimated execution time and the base time for all the jobs in the workload. As the results in the previous analysis are similar irrespectively of the workload and the behavior for the classical policies is practically the same, the Workload 3 was taken as a representative sample, and the the *METL* was compared with the well known *SJF* policy.

Figure 4 shown the obtained results. In the x-axys we have the job id for all jobs in the workload. The jobs are ordered from less to higher base time. On the primary y-axys (left side) is represented the base time for each job measured in seconds. As can be seen, the majority of jobs have a low base time, and just only a minor part are big jobs. On the secondary y-axys (right side), is denoted in percentage the difference between the execution time and its base time. Negative values on the right y-axis means that the job was executed faster than in the reference dedicated environment. So the lower the values are, better decisions are taken for the specific technique.

As can be observed, *SJF* produced solutions in which an important number of jobs have a longer execution time than in the dedicated environment. This is represented by the green points on the figure. On the other hand, *METL* was able to produce solutions in which the majority of the parallel jobs executed faster than in the dedicated environment. This is due to its ability to determine the job execution order and a fairness allocation in which the loss of the execution time is optimized and the underutilization of resources is avoided. By this, just only a few number of jobs has increased their execution time, however these jobs are those from the workload with the slowest base time, so the final execution time do not affect significantly on the results.

The degree of complexity of the proposed policy was determined to be $O(n^2)$, where n denotes the number of jobs waiting in the system queue when the *METL* policy is applied. In this experimental study our policy has been treated in average 16 jobs each time it was

executed, obtaining the scheduling solution in $60ms$ in average.

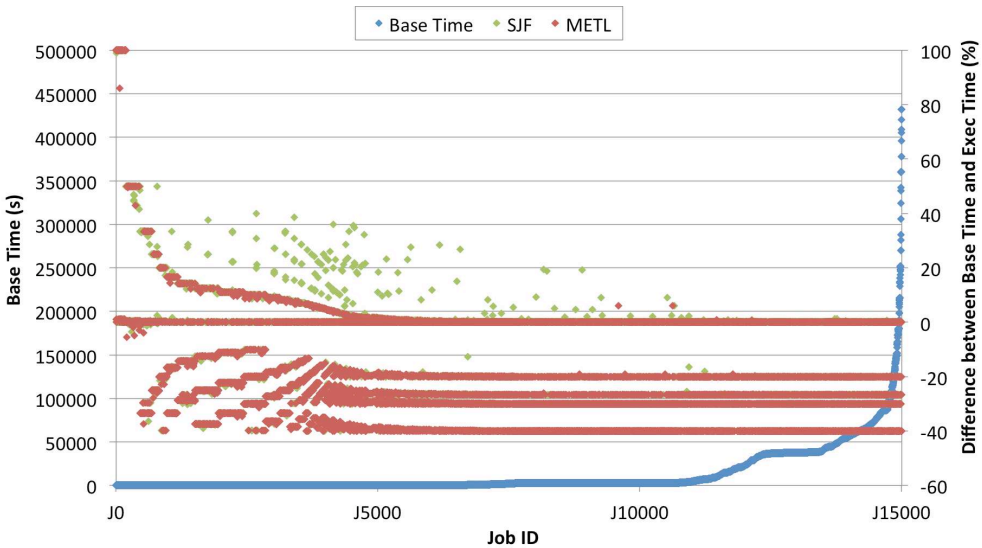


Figure 4: Relation of the Execution Time with the Base Time

4 Conclusions

The present work focuses on multiple job scheduling on heterogeneous multi-cluster environments with co-allocation. The authors proposed a new policy that determines the job execution order and its resource allocation trying to minimize the job execution time, preventing resource underutilization and avoiding starvation. The results, compared with common scheduling techniques from the literature, shown that our proposal was able to produce more fairness scheduling decision reducing the jobs execution time for different types of real workloads traces.

Acknowledgment
This work was supported by the Ministry of Education and Science of Spain under contract TIN2011-28689-C02, TIN2010-12011-E and the CUR of DIUE of GENCAT and the European Social Fund.

References

[1] Javadi, B., Akbari, M.K., Abawajy J.H., *A performance Model for Analysis of Heterogeneous Multi-Cluster Systems*, Parallel Computing, 32(11-12), 831-851, 2006.

[2] Bucur, A.I.D., Epema, D.H.J., *Schedulling Policies for Processor Coallocation in Multicluster Systems*, IEEE TPDS, 18(7), 958-972, 2007.

[3] Jones, W., Ligon, W., Pang, L., Stanzione, D., *Characterization of Bandwidth-Aware Meta-Schedulers for Co-Allocating Jobs Across Multiple Clusters*, Journal of Supercomputing, 2005, 34(2), 135-163, 2005.

[4] Heien, E.M., Fujimoto, N., Hagihara, K., *Static Load Distribution for Communicative Intensive Parallel Computing in Multiclusters*, IEEE PDP'08, 321-328, 2008.

[5] Naik, V.K., Liu, C., Yang, L., Wagner, J., *Online Resource Matching for Heterogeneous Grid Environments*, IEEE/ACM Int. Conf. CCGRID'05, 2, 607-614, 2005.

[6] Lérida, J.L., Solsona, F., Giné, F., García, J.R., Hernández, P., *Resource Matching in Non-dedicated Multicluster Environments*, VECPAR'08, 2008, 160-173, 2008.

[7] Shmueli E., Feitelson D.G., *Backfilling with Lookahead to Optimize the Packing of Parallel Jobs*, J. J. Parallel Distrib. Comput., vol.65(9), pp.1090-1107, 2005.

[8] Blanco, H., Lérida, J.L., Cores F., Guirado, F., *Multiple Job Co-Allocation Strategy for Heterogeneous Multi-Cluster Systems Based on Linear Programming*, Journal of Supercomputing, vol.58(3), pp:394-402, 2011.

[9] Skillicorn, D.B., Hill, J. M.D., McColl, W.F., *Questions and Answers about BSP*, Oxford University Computing Laboratory, 1997.

[10] Blanco, H., Montañola, A., Guirado F., Lérida J.L, *Fairness Scheduling for Multi-Cluster Systems Based on Linear Programming*, CMMSE'10, vol.1, pp:227-239, 2010

[11] *GridSim simulation framework*, http://www.buyya.com/gridsim

[12] *Parallel Workloads Archive*, http://www.cs.huji.ac.il/labs/parallel/workload

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Load Balancing Algorithm for Heterogeneous Systems

Jose Luis Bosque[1], Oscar D. Robles[2], Pablo Toharia[2] and Luis Pastor[2]

[1] *Department of Electrónica y Computadores, Universidad de Cantabria*

[2] *Department of ATC and CCIA, Universidad Rey Juan Carlos*

emails: joseluis.bosque@unican.es, oscardavid.robles@urjc.es,
pablo.toharia@urjc.es, luis.pastor@urjc.es

Abstract

This paper presents a load balancing algorithm for heterogeneous clusters. The heterogeneity is pointed out mainly in assessing the computing capacity of each node. To improve the accuracy of this parameter, we propose a new load index that takes into account two levels of heterogeneity of the current processors: the number of cores per node and the computing power of each core. The experimental results show the impact of this factor on the distribution of workload.

Key words: Heterogeneous computing, load balancing, load index.

1 Introduction

High performance computing (HPC) has signicantly evolved during the last decades making possible to integrate up to hundreds of thousands cores into the currently available petaflop machines [1]. It seems clear that the roadmap to the next generation of exascale computers passes through the integration of large distributed and heterogeneous systems, in which the computational nodes are in turn composed of multi-core processors with a large number of cores and hardware accelerators.

The design and implementation of efficient parallel applications for heterogeneous systems is still a very important challenge. The heterogeneity of these systems introduces a number of important factors that cause the models and algorithms used in homogeneous parallel systems to be outdated, so they do not produce adequate and reliable results [2]. One of the problems that has a deep impact in the performance of the parallel applications, and particularly in heterogeneous systems is workload balancing.

ISBN:978-84-615-5392-1

One of the keys to provide a balanced workload is to properly characterize the computing capabilities of each node of the system. This characterization relies on some static factors (such as the number of cores in a node and their computational power) as well as on some dynamic ones (such as the number of tasks being executed and their requirements). All of them can be considered in one single parameter that will be named *Load Index*, that determines the aforementioned computing capabilities of a node at every moment.

This paper presents this new work load index that characterizes the computing capabilities of the nodes in a parallel system. This proposed index considers the two levels of heterogeneity that can be found in a multi-core system: number of cores and computing power of each node. It can be seen that the first factor is discrete while the second is continuous. In order to test this new parameter, a distibuted, global, emitter-initiated load balancing algorithm has been implemented, that can also turn itself off if the whole system is overloaded.

There are other approaches that can be found in the bibliography. [5] presents EMAS, an Evolutionary Mobile Agent System. It proposes a load index, *Server_Utilization_Status*, based on a 4 different parameters that are merged, although these parameters have different nature as well as different units. On the other hand, [6] uses the *Current real load* of each node. This index is based on parameters such as CPU occupancy rate, memory usage, system I/O usage and network bandwidth occupancy rate, whose influence is altered depending on the different services offered by the cluster. Another similar approach that merges parameters of different nature in a single load index can be found in [8]. Also, a very different approach is proposed in [3]. This paper attempts to improve the accuracy of host load predictions by applying a neural network predictor. A couple of specific algorithms for balancing the workload of iterative algorithms on heterogeneous multiprocessors have been proposed [4, 7]. The main goal consists of designing a strategy that will allow to dynamically analyze the computational power of the processors involved in the heterogeneous system and to determine the computational burden that must be located at each processor.

2 Design of the Load Balancing Algorithm

The approach presented in this paper is a dynamic, distributed, global and non preemptive load balancing algorithm. It is a dynamic algorithm because the assignment of a task to an specific node is performed in run time. Then the task is completely executed in the assigned node, without any kind of task migration. It is a distributed algorithm since every node in the cluster takes its own decisions based on local stored information. Once a node decides to perform a load balancing operation, the partner is selected among all the available nodes in the cluster. Finally, the algorithm does not present any overhead if the nodes are naturally balanced, therefore it can automatically turn itself off on global under-loading or over-loading situations.

2.1 Measuring the state of a node

During this phase the local information needed to determine the workload of the local node is collected, to calculate then the load index, which determines the state of the node. The decision about the local execution of a new task or launching a new load balancing operation is taken based on this state for each node. The load index is periodically computed and the it is discretized in a set of states.

Load Index. The load index is calculated based on two static parameters, the number of cores and the computational power as well as a dynamic parameter, the number of tasks being currently executed in the node. A node has two sources of heterogeneity, the number of cores and their computational power. Thus, it is necessary to specify two different possibilities:

- The number of tasks is lower than the number of cores in the node. Therefore, there are some free cores and this node can accept more tasks, so it will be a *recipient* node.
- The number of tasks is larger than the number of cores. In this case, the load index is calculated based on the following expression:

$$Load_{Index} = \frac{Bogomips_{local\_average}}{Bogomips_{cluster\_average}} \cdot \frac{\#Cores}{\#Tasks} \tag{1}$$

This expression takes into consideration the two sources of heterogeneity in the computational power of a node. In this way the maximum value achievable by the load index of a node depends not only on the number of cores but also on the ratio of the average computational power of its cores with respect to the average computational power of the whole cluster. The load index is calculated periodically each predefined period of time. This period of time should be high enough to minimize the overhead that doing this measure introduces in the system, but also short enough to keep the load index value updated.

States of a Node. The states come from a discretization of the load index in order to minimize the exchange of global information, as well as to simplify load balancing decisions. These states determine the behaviour of a node, and three different ones have been defined:

- *Recipient State*: The state of a node is *recipient* when the number of its running tasks is less than the number of cores it has, or when its load index value is bigger than the threshold *neutral-recipient*. This means that the node has free cores and then it can assume at least one more task, either local or remote.
- *Neutral State*: Its load index has a medium value. In this case all the node's cores are executing at least one task. In this state the node can assume new local tasks but it rejects all remote requests.

 ISBN:978-84-615-5392-1

- *Emitter State*: A node is Emitter when its load index has a value under the Neutral-Emitter threshold. This means that the node has many more tasks than the number of cores it has, and so it can not accept any more tasks.

Change of state. A node will change its state whenever a new measurement of the load index crosses a status threshold. Since this parameter is very volatile, it is possible that a particular node might be continuously changing its state due to small changes in its external load. To prevent this problem, some degree of hysteresis has been added to each threshold. This way, the number of messages due to state changes is reduced.

2.2 Global Information

In order to take decisions about load balancing it is necessary to exchange the state information among the nodes of the cluster. Our approach is a global algorithm, so all the nodes keep updated information about the global system state. A on-state-change driven policy has been implemented where nodes broadcast their workload information when they suffer a change of state. Load balancing operations can only take place between *Recipient* and *Emitter* nodes. Hence, only changes to or from *Recipient* state are significant enough to be communicated and thus, the number of messages is significantly reduced. Each node maintains a state-queue with the information received from other nodes. Only a *Recipient-queue* is needed, because only *Recipient* nodes can accept remote tasks. When a node becomes a *Recipient*, it broadcasts a message to all the nodes of the cluster so each of them will place it at the end of its queue. On the other hand, when a *Recipient* node changes its state to *Neutral* or *Emitter*, it broadcasts a message too and all the cluster nodes will discard it from their state-queues.

2.3 Initiation of Load Balancing Operations

It determines when a new load balancing operation has to begin. Since a load balancing approach that does not interrupt any execution has been proposed, the decision about which node is finally going to execute a task can only be taken in the exact moment of beginning the execution. Therefore, there must be a *emitter-iniciated* rule to take this decision, and the decision about initiating a load balancing operation is *completely local* to the emitter. It must be noticed that on the decision of not doing any load balancing operation, no messages have to be exchanged, so the communications overhead is reduced. The initiation rule must be evaluated every time a new task has to be launched. At this moment, the state of the node is checked, in order to know if it can accept the local execution of the new task or if the search of a better candidate for the execution of the task should be initiated. A node can only accept the local execution of a new task if it is in *recipient* or *neutral* state. Therefore, only if the state of the node is *emitter*, a load balancing operation is initiated.

 ISBN:978-84-615-5392-1

2.4 Partner Localization and Load Distribution

Once the decision of doing a load balancing operation is taken, two important steps must be performed: to locate a partner to send it the exceeding load and to decide how many tasks should be sent to that partner.

Partner localization. This is a completely local operation, since an *emitter* node looks for a partner in its own *recipients* queue. The selection of the partner can be done in many different ways. The approach adopted is to do a first random selection of a few nodes, that are then sorted based on their load index, so the less loaded node is requested. If the request is rejected, the next node in the list will be requested and so on. This strategy has some advantages. It reduces the communications overhead since the load indexes do not have to be constantly updated. Also, it avoids that one candidate is simultaneously chosen by different nodes, because it is in a prominent position in the queue.

Load Distribution. The last step to perform the load balancing operation is to decide how much workload should be sent to the *recipient* node. A good distribution is that in which each node gets an amount of workload that is proportional to its computational power. Therefore, the more similar load indexes are after the load balancing operation, the better the load distribution is. In this approach, first of all the *recipients* are sorted based on their load indexes. Then the *emitter* node sends to the first node in the sorted list as much workload as needed to force its load index to change from *recipient* to *neutral* state. Then, the second *recipient* from the list is selected and the same operation is performed. This process finishes when there is no more workload to send or there are no more *recipients*.

3 Implementation Issues

***Load* process.** It is in charge of the state measure. For this purpose this process checks periodically (using a time interval which can be defined for each execution) the load of the node and computes a value so called *load index*. Based on this index the new state is determined, and if a change of the state happens, then the new state is sent to both *Global* and *Balance* processes. In order to compute the *load index* both static and dynamic information is collected, according to Equation 1. On one hand static data regarding the number of CPUs or cores and their bogomips-based computational power is used. On the other hand dynamic information regarding the load of the system is also fetched. In the proposed approach all the data is collected from the Linux kernel.

- $Bogomips_{local\_avg}$ is the average of the bogomips-based computational power of the cores available.
- $Bogomips_{cluster\_avg}$ is the average of the $Bogomips_{local\_avg}$ values for all the nodes of the system. This value allows to compare the computational power of a specific node against the computational power of the whole system.

 ISBN:978-84-615-5392-1

- $\#Cores$ is the number of cores of the node for which the load index is been computed.
- $\#Tasks$ is the number of tasks being executed in the node. This is a dynamic value which has to be refreshed using a fixed interval.

The load index determines the state of the node, therefore it has to be discretized. This is not a simple task because of the different nature of the values used to compute the load index. Two levels of heterogeneity are comprised in the index and both continuous (computational power) and discrete (number of cores) values have been used. The main goal pursued in this work is that all the nodes have the same computational power versus load ratio. To determine the thresholds, the approach taken is to assign a number of tasks proportional to the average number of cores per node. This also allows limiting the number of tasks per core, so as to ensure a minimum capacity of CPU for each task. Furthermore a hysteresis range is introduced, to avoid unnecessary multiple state changes. With these criteria the selected thresholds are: *Recipient to Neutral* at 0.80; *Neutral to Recipient* at 1.0; *Neutral to Emitter* at 0.667 and *Emitter to Neutral* at 0.727.

Finally, the process *Load* determines if a state change has occurred. If so, it should communicate the new state to the local processes *Global* and *Balance*. Furthermore, if the new state is *Recipient*, the *Load* process calculates and sends to the *Balance* process the maximum number of tasks it can accept, in case of participating in a balancing operation.

***Global* process.** It is in charge of both keeping all the global information updated and finding the list of candidates for a load balancing operation. For the former this process has to be able to receive state changing messages coming from the local *Load* process. Then, it has to propagate this change of state to the instances of the *Global* process on the rest of the nodes. Besides, the list of candidates is generated when the local *Balance* requests it and it has to be delivered to the same process. This way, the implementation of this process is based on a loop that waits for messages from the other processes. Depending on the type of message, different operations are carried out:

- *LOCAL_CHANGE*: this message comes from the local *Load* process. It informs that there has been a change of state in the local node which has to be broadcasted to the rest of the nodes.
- *LOCAL_RESQUEST*: this message is received from the local *Balance* process. It requests to the *Global* process to generate a randomized list of candidates using the recipient list which it keeps updated. The size of the list is a parameter of the algorithm. Finally, this list is sent to the local *Balance* process.
- *REMOTE_CHANGE*: this message comes from a remote *Global* process and informs about the change of state of that remote node. Then the recipients queue and the number of recipient nodes available on the whole system are updated. This update has to be sent to the local *Balance*.

 ISBN:978-84-615-5392-1

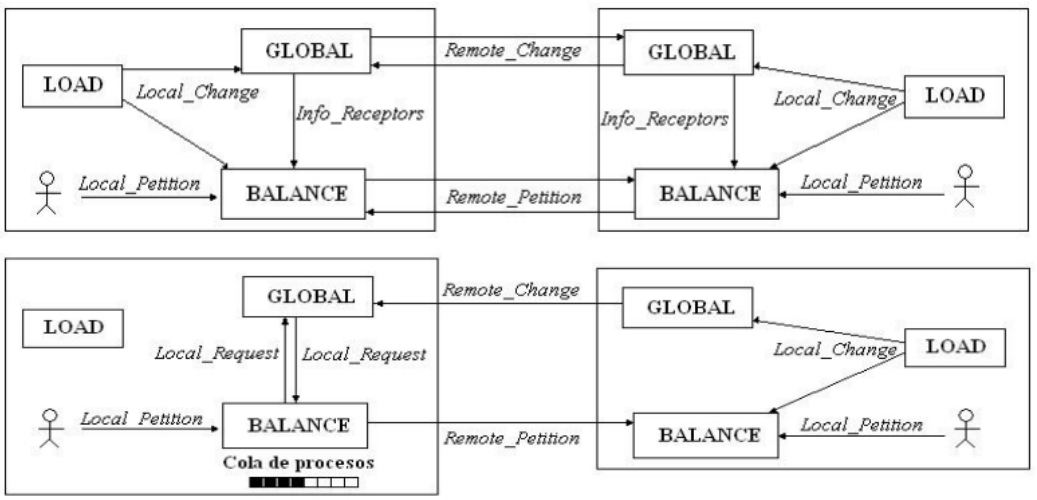


Figure 1: Structure of Processes and communication

***Balance* process.** It decides the operations needed when new execution requests are demanded. This involves localizing the recipients and solving how load balancing operations have to be done. Again, this process is based on an infinite loop which probes for new messages. When a message is received it has to perform the following operations:

- *LOCAL_CHANGE*: The local *Load* process informs of the state change of the local node each time a change occurs, and the amount of tasks that this process can accept.
- *INFO_RECIPIENTS*: this message comes from the local *Global* process representing the number of nodes that are in a *recipient* state at that moment. After receiving this message load balancing operations can be done if there are tasks in the process queue.
- *LOCAL_PETITION*: these messages are local execution requests coming from the users or processes using the node. When a message of this type is received a load balancing operation can be needed In this case, firstly the list of candidates has to be requested to the local *Balance* process using a message tagged with *LOCAL_REQUEST*. Once the list is received remote execution requests are sent as it is explained in the following.
- *REMOTE_PETITION*: they is a request coming from a remote *Balance* process, to execute a remote task, as a part of a load balancing operation. This message starts a protocol to reach an agreement to accept the execution of remote tasks, depending on the current load of the node and the number of remote tasks to execute.

 ISBN:978-84-615-5392-1

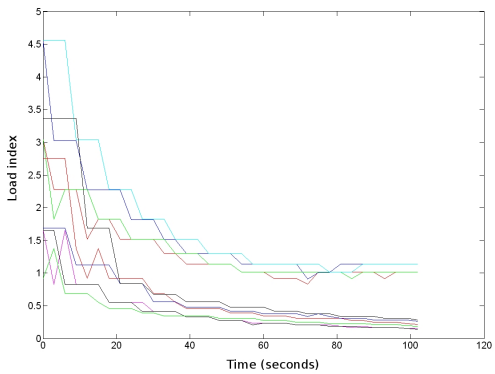


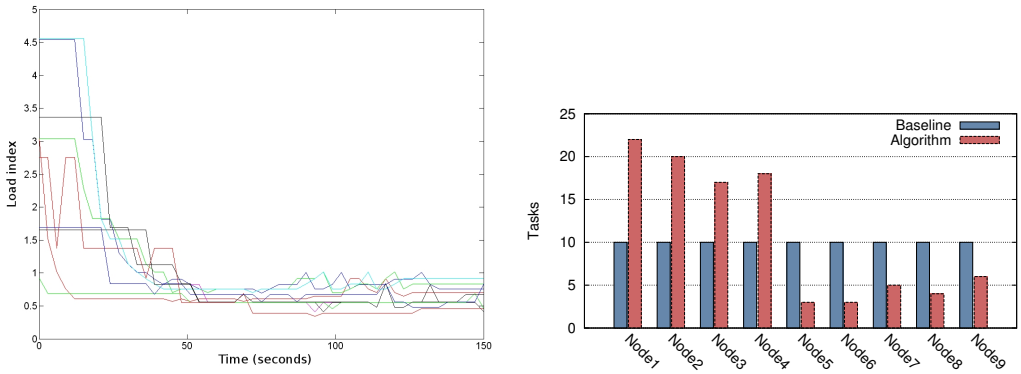
Figure 2: Homogeneous distribution: evolution of the nodes along the time.

Finally figure 1 shows the algorithm architecture, with the number of processes and the communication protocols, both to keep the global information updated and to perform a load balancing operation. Also, it can be seem how the arrival of requests increases the waiting process queue until there is a change of state.

4 Experimental Results

The experiments have been run on a heterogeneous cluster composed of 10 nodes. Four nodes with 8 cores and around 6000 bogomips; 2 nodes with 4 cores and 4422 bogomips; 2 nodes with 4 cores and 3618 bogomips, and 2 nodes with 2 cores and 4341 bogomips each. As it can be noticed there are tow levels of heterogeneity in the nodes: the number of cores and the computing power. The system will be always loaded with 100 identical tasks, that perform a matrix multiplication. The dimension of the matrices has been selected so that the execution takes one minute in the most powerful node. Additionally, the 100 tasks will be loaded one per second to the same node.

In the first experiment the tasks are evenly distributed among all the nodes in the cluster. The results of this experiment will be the baseline to compare with the results obtained with the load balancing algorithm. Figure 2 shows the evolution of the load indexes at each node along the time. The larger the value of the index, the more unloaded a node is and then it can accept more new tasks. Figure shows that there are two very different groups of nodes: the few upper graphs are from the most powerful nodes, while the lower ones are from the least powerful ones. The least powerful nodes have a load index lower than 0.5, meaning that the number of tasks they have double their number of cores. On the other hand, the most powerful nodes have less than one task per core, i.e. their load index is higher than 1. In this case it took the system 8 minutes and 14 seconds to perform



(a) Evolution of the nodes along the time. (b) Number of tasks executed in each node.

Figure 3: Results with load balancing algorithm.

all the tasks.

Now the same experiment is performed but with the load balancing algorithm. All the tasks will be run on the same node so it will dispatch them to the remainder nodes as a result of the load balancing algorithm execution. The communication overhead should be overcome by the expected reduction of times due to the balance of the load indexes. Table 1 shows the values selected for the different parameters of the load balancing algorithm.

Figure 3(a) shows the load indexes of all nodes grouped, suggesting a distribution of tasks proportional to the computational power of the nodes, with the lowest load index value around 0.5. Regarding the load distribution time, it can be seen that now is higher (around 150 seconds) than in the baseline case. This is an expected result since when all the nodes are busy, the algorithm enqueues the tasks and waits for the queues to be empty to avoid a saturation of the system. These both aspects lower down to 4 minutes and 13 seconds the total time the system needs to complete all the tasks, as can be seen in table 2. Figure 3(b) shows the number of tasks each node executes. The differences between the baseline and the algorithm are quite remarkable. It becomes a prove of how fair it is the

Table 1: Parameters used in the experiment.

Parameter	**Value**
Time interval to measure state	3 seconds
Location of Recipients	Sorted list of 3 candidates
Workload	100 tasks, 1 minute long each
Thresholds for changing state	Neutral threshold: To Recipient at 1.00 Recipient threshold: To neutral at 0.80 Transmitter threshold: To neutral at 0.727 Neutral threshold: To transmitter at 0.67

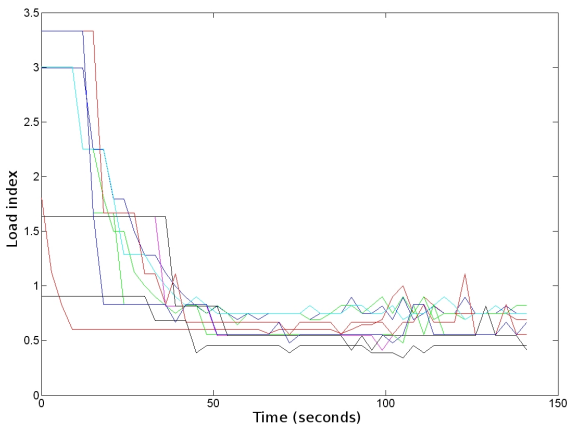


Figure 4: Evolution of the nodes along the time with the new thresholds.

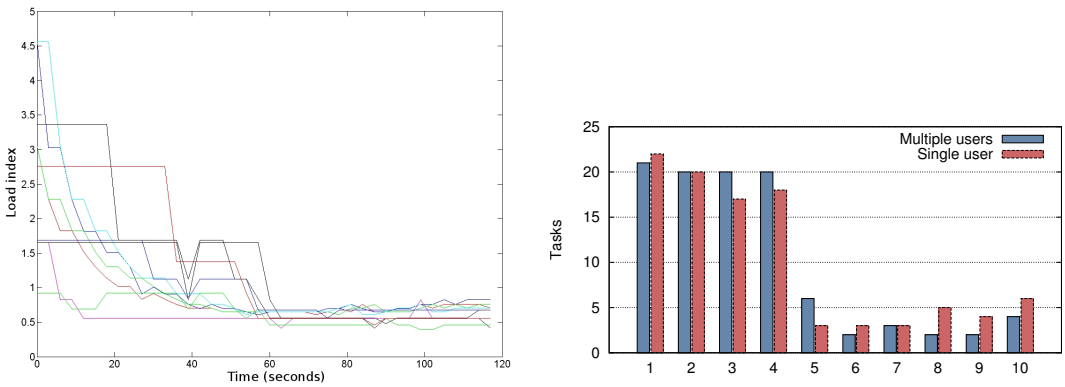
distribution of tasks the algorithm performs. It can be said that the improvement of the total execution time is around 95%.

It is interesting to check the effect of changing the thresholds. The graphs obtained show several nodes with high load indexes that probably can accept more tasks, since they are the nodes that do not change to Recipient state. Then, in order to try to group more the load indexes of all the nodes, a new experiment has been done with the following thresholds: *Recipient to Neutral* at 0.75; *Neutral to Recipient* at 0.8; *Neutral to Emitter* at 0.7 and *Emitter to Neutral* at 0.75. The results obtained can be seen in Figure 4. It can be noticed that the load indexes are more grouped than in the previous experiments. Only the load index of the node that distributes the tasks remains in the same values. This is a problem of the ratio between the refresh interval of the load index and the arrival rate of the tasks to the system. Anyway, the load distribution time is lowered down from 153 to 144 seconds and the total time to perform all the tasks is reduced to 4 minutes and 4 seconds.

Finally, the behavior of the system with concurrent users will be tested. For this purpose the 100 tasks will be launched in a distributed way in 5 different nodes (20 tasks each). Figure 5(a) show that the load indexes of the nodes are more grouped. This initial

Table 2: Comparison between the baseline and the load balancing algorithm.

Parameter measured	**Value in Baseline**	**Value with load balancing algorithm**
Load index of least loaded node	1.14	0.9
Load index of most loaded node	0'.14	0.41
Max. difference between load indexes	1	0.49
Load distribution time	105 seconds	153 seconds
Total execution time	8 mins. 14 seconds	4 mins. 13 seconds
Speedup	—	1.95



(a) Evolution of the nodes along the time. (b) Number of tasks executed in each node.

Figure 5: Cluster with multiple users.

distribution helps the load balancing algorithm so that the total time the system needs to complete all the tasks decreases to 3 minutes and 57 seconds. The reason for this is that the machine that receives the tasks for their distribution always runs more tasks than the remaining ones. If the number of those machines is increased, the overload is more distributed, and therefore the total execution time is reduced. It can be seen in Figure 5(b) that nodes 2, 3, 4 and 5 are more loaded that in the Figure 3(b). This is because the algorithm does not change one node's state to emitter until it is overloaded, forcing the node to accept more tasks than the others.

5 Conclusions and Future Work

This paper presents a workload balancing algorithm that considers the heterogeneity of the nodes available in the system. It is a dynamic, distributed, global, emitter-initiated and non-preemptive algorithm. Also, it is able to turn itself down when all the nodes are overloaded or underloaded, so there is not any improvement achievable by doing load balancing operations. Therefore, it minimizes the overhead of the global system.

Heterogeneity comes mainly from the different computing capabilities of the available nodes. Therefore, it is basic that the algorithm is able to evaluate dynamically those capabilities as accuracy as possible, while the load index can be comparable among all the nodes. This is the reason while the proposed load index considers two levels of heterogeneity: the number of cores per node and the individual computing power of each core.

The experiments done show clearly how the influence of the load index is essential to achieve a right distribution of tasks. This distribution should be in proportion to the computing power of each node. Then it can be seen that the execution time of the experiments

was reduced to a half. It can be also said that the change in the thresholds also affects the execution time.

One future work will be to provide the algorithm with self-learning mechanisms, that will consider the situation of the global load of the system. It would give the algorithm the ability to change itself all the parameters based on the dynamism of the system, being then quite adapted to the work environment.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Education and Science (grants TIN2010-21289, TIN2010-21291-C02-02, Consolider CSD2007-00050 and Cajal Blue Brain project) as well as by the HiPEAC European Network of Excellence.

References

[1] The top500 project. November 2010. http://www.top500.org.

[2] J. Dongarra and A.L. Lastovetsky. *High Performance Heterogeneous Computing*. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, 2009.

[3] Truong Vinh Truong Duy, Y. Sato, and Y. Inoguchi. Improving accuracy of host load predictions on computational grids by artificial neural networks. In *Parallel Distributed Processing. IPDPS 2009. IEEE International Symposium on*, pages 1 –8, may 2009.

[4] Ismael Galindo, Francisco Almeida, and José Manuel Badía-Contelles. Dynamic load balancing on dedicated heterogeneous systems. In *PVM/MPI*, pages 64–74, 2008.

[5] N.K. Gondhi and D. Pant. An evolutionary approach for scalable load balancing in cluster computing. In *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 1259 –1264, march 2009.

[6] Wenzheng Li and Hongyan Shi. Dynamic load balancing algorithm based on fcfs. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*, pages 1528 –1531, dec. 2009.

[7] J. Martnez, F. Almeida, E. Garzn, A. Acosta, and V. Blanco. Adaptive load balancing of iterative computation onheterogeneous nondedicated systems. *The Journal of Supercomputing*, 58:385–393, 2011. 10.1007/s11227-011-0595-3.

[8] Xiaonian Tong and Wanneng Shu. An efficient dynamic load balancing scheme for heterogenous processing system. In *Computational Intelligence and Natural Computing, 2009. CINC '09. International Conference on*, volume 2, pages 319 –322, june 2009.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Freezing in Gold Nanoclusters

Richard K. Bowles[1] and Cletus C. Asuquo[1]

[1] *Department of Chemistry, University of Saskatchewan, Saskatoon, SK, S7N 5C9, Canada*

emails: richard.bowles@usask.ca, cla121@mail.usask.ca

Abstract

Molecular dynamics simulations are used to study freezing in gold nanoclusters. We find that 600 atom gold clusters freeze to four different solid structures, an icoshahedron (*Ih*), a true decahedron (*Dh*), an off-center decahedron (Dh_2) and a face-centered-cubic cluster (*Fcc*), that can be distinguished on the basis of order parameters that measure the degree of surface and core order in the cluster. The icosahedron remains the most common structure formed, even though the *Fcc* is the most stable structure at this cluster size, and connections between the probability of observing a given frozen structure from an ensemble of molecular dynamics trajectories and the nucleation rate and free energy barrier are discussed.

Key words: nanoclusters, freezing, molecular simulation

1 Introduction

It is well known that nanoscale clusters containing only hundred atoms exhibit a rich variety of structural properties that are very different from their bulk materials [1]. Most of the atoms in a large thermodynamically sized system are buried in the "core " of the material where they all share the similar local environments, while comparatively very few particles are located on the surface. Surface effects can then be ignored and the lowest energy structures are usually crystalline solids, such as the face-centred-cubic (*Fcc*) or body-centred cubic crystals (*Bcc*), with long range periodic ordering. However, as the system size decreases a greater fraction of the atoms are at the surface and the nature of the most stable structure results from a balance between volume and surface effects which leads to the appearance of a variety of non-crystalline structures such as icosahedra and decahedra.

While there has been a considerable amount of work focused on understanding and identifying which cluster types have the lowest energy as a function of the number of atoms [2], much less is known about the kinetic processes, such as nucleation, that control which structures are formed. Molecular dynamics simulations of freezing show that clusters, under the same conditions, will freeze to different solid structures [3]. For example, Bartell et al. [4] found that gold clusters containing many thousands of atoms still froze predominantly to *Ih* clusters, even though the thermodynamic $Ih \rightarrow Dh$ transition occurs at $N \approx 500$. This suggests freezing in nanoclusters occurs in a competitive process where a single liquid droplet can freeze to any one of the many accessible solid structures. As a result, in a series of freezing trajectories or events, we would expect to see a distribution of solid cluster types appearing with a given probability that is determined by kinetic factors, such as the nucleation barrier and growth dynamics, instead of their global stabilities. The goal of this work is to explore the nature of the competitive freezing process in gold nanoclusters and to determine the probability of observing a given solid cluster type from an ensemble of freezing trajectories.

2 Method

We perform molecular dynamics (MD) simulations of the freezing of gold clusters with $N = 600$ atoms, modelled using the embedded atom method potential with parameters appropriate for gold. The simulations were carried out in the canonical (N, V, T) ensemble, in a cubic cell with volume $V = 1 \times 10^6$ Å^3 and periodic boundaries, with $T = 700$ K. The equations of motion were integrated using the velocity Verlet algorithm, with a time step, $\Delta t = 2.8$ fs, coupled to a Noose-Hoover thermostat. To generate independent initial starting configurations for each trajectory, a gold cluster was melted at $T = 1500$ K for 2.5×10^5 time steps to ensure all memory of the starting configuration was lost. The cluster was then equilibriated for a further 2.5×10^5 time steps at $T = 1000$ K before saving 500 starting configurations, each separated by 140 ps. These 500 clusters were then instantaneously cooled to 700 K by assigning the atoms a new random velocity chosen from a uniform distribution appropriate for the new T. The MD simulations were followed for 4.8×10^5 time steps.

To study the structure of the clusters formed, the final configuration of the trajectory was subjected to a conjugate gradient quench to its local energy minimum to remove thermal noise from the structure. We then measured a variant of the Steinhardt bond order parameters [5], based on Q_6, that gives us information regarding the order at the surface and the core of the nanoparticle. These are defined as,

$$Q_{b,s} = \sqrt{\frac{4\pi}{13} \sum_{m=-6}^{6} |\frac{1}{N_{b,s}} \sum_{i=1}^{N_{b,s}} q_{6m}(i)|^2}, \tag{1}$$

 ISBN:978-84-615-5392-1

where

$$q_{6m}(i) = \frac{1}{N_{nb}} \sum_{j=1}^{N_{nb}(i)} Y_{6m}(\mathbf{r}_{ij}). \tag{2}$$

The subscripts s and b denote the surface and bulk atoms respectively, which are distinguished using the "cone" algorithm [6]. The summation in Eq. 2 is over the number of neighbours, (N_{nb}), for atom i. Two atoms are considered neighbours if the distance between them is less than or equal to 3.5 Å, which corresponds to the distance to the first minimum of the radial distribution function for gold. $Y_{lm}(\mathbf{r}_{ij}) = Y_{lm}(\theta_{ij}, \phi_{ij})$ are spherical harmonic functions, where θ and ϕ are the polar and azimuthal angles of the vector $\mathbf{r}_{ij}$, respectively. We also use common neighbour analysis [7] (CNA) to identify the local structure of the individual atoms in the clusters.

3 Results and Discussion

Upon cooling, the energy of a typical trajectory settles down to a value consistent with the metastable fluid state which lasts between 200-1400 ps. At some point along the trajectory, the energy drops rapidly, taking 50-100 ps to establish a new lower level, signifying the cluster has frozen to its solid state. Figure 1 shows representatives of the structures observed in our simulations and highlights some of the important structural features of the different solid types using the CNA. The Ih clusters have several five-fold symmetric caps and a central Ih atom, formed from the packing of tetrahedral subunits of locally Fcc atoms, but none of the structures are perfect with small regions of the cluster remaining amorphous. The Dh structures have at least one of their five-fold symmetric caps and a single line of five-fold symmetric atoms running through the core of the cluster. However, while the Dh_2 structure also has a single line five-fold symmetric atoms running through the cluster, this is offset from the center of the cluster and there is no cap. Since the cap structure is usually the first element to be formed in the freezing of the decahedral structures, it appears that the Dh and Dh_2 clusters have distinctly different freezing pathways despite sharing similar structural features. The Fcc structures have no five-fold symmetric atoms and are usually formed from stacked layers of Fcc and, or Hcp atoms.

Figure 2 shows that Q_s and Q_b can be used to clearly distinguish between the different solid cluster types. We also see that the icosahedral clusters are still the most common structure (92%), even though the Fcc structure (4%) is more stable at this cluster size. The Dh and Dh_2 structures appear 3% and 1% of the time, respectively. Nam et al. [8] showed that liquid gold droplets exhibit a degree of ordering at the surface consistent with $\langle 111 \rangle$ facet, which is the lowest energy surface construction, and this may be sufficient to ensure the nucleation barrier to Ih remains low. The low surface tension also ensures the fluid phase partially wets the crystal so that freezing begins near the surface [9]. While the

 ISBN:978-84-615-5392-1

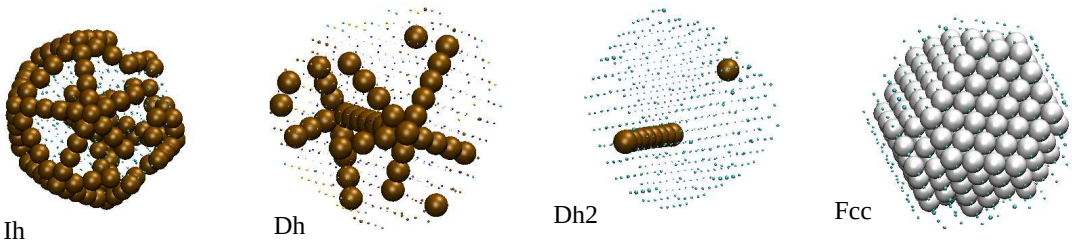


Figure 1: Solid Cluster types formed in freezing trajectories. The dark atoms denote atoms with a local five-fold symmetry identified using CNA. The grey atoms are Fcc atoms. The remaining atoms have been reduced to points.

Dh cluster also freezes through the formation of a five-fold symmetric cap with $\langle 111 \rangle$ facets, it must eventually create a number of $\langle 100 \rangle$ facets, which have a higher surface tension.

Sanders et al. [10] showed, in a competitive nucleation process, that the rate of forming phase i is given by

$$J_i = P_i J \ , \tag{3}$$

where $J = \sum J_i$ is the rate the liquid phase nucleates to any structure and P_i is the probability of seeing the i^{th} structure in an ensemble of nucleation events. According to classical nucleation theory, $J_i = A_I \exp(-\Delta\beta G_i^*)$, where $\Delta\beta G_i^*$ is the height of the free energy barrier for nucleating structure i and A_i is the kinetic prefactor. If we take the ratio of rates between two competing nucleation processes and assume the prefactors for the processes are the same, then the probabilities of observing the structures can be related to the difference in free energy barriers as,

$$\Delta G_{nm}^* = \beta(\Delta G_n^* - \Delta G_m^*) = \ln\left(\frac{P_m}{P_n}\right) \ . \tag{4}$$

Using the probabilities obtained from our ensemble of runs in Eq. 4 gives the difference in free energy barrier heights between the *Fcc* and *Ih* structures as $\Delta G_{Fcc,Ih}^* \approx 3.1kT$. However, while P_i is fundamentally connected to J_i through Eq. 3, the connection to the free energy in Eq. 4 assumes that P_i reflects the probability of finding the critical embryo and that the thermodynamics and kinetic factors are totally decoupled. This may not be the case here, where the free energy barrier is low and the presence of mesoscopic structural motifs growing within the cluster may prevent the system from sampling phase space on the time scale of the freezing event. It may be possible that the ensemble of trajectories provide a mechanism that bypasses the kinetic traps since each trajectory can follow a different path but, in general, it remains a considerable challenge to understand how growth kinetics may

 ISBN:978-84-615-5392-1

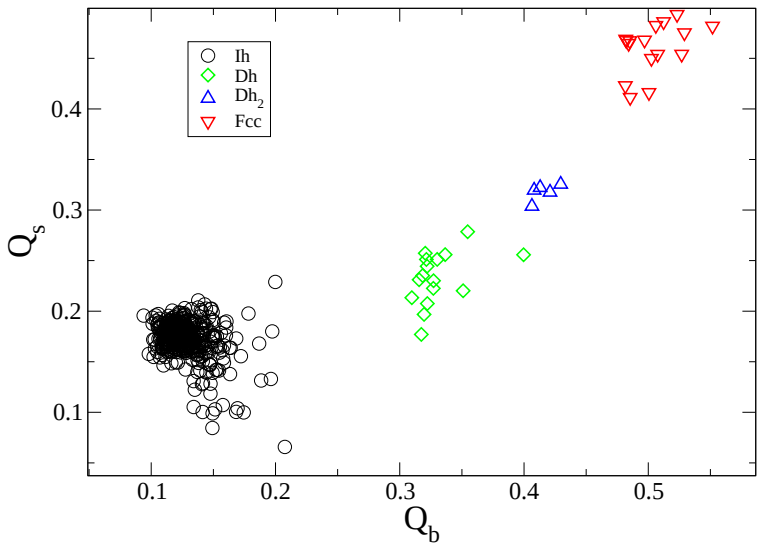


Figure 2: Q_s as a function of Q_b for the different solid structures.

influence the probabilities of seeing the different structures in condensed phases, particularly in the low barrier regime.

Acknowledgements

We thank NSERC for financial support.

References

[1] D. J. Wales, *Energy Landscapes, With Applications to Clusters, Biomolecules and Glasses*, Cambridge University Press, Cambridge, 2003.

[2] Y. H. Xiang, L. J. Cheng, W. S. Cai and X. G. Shao, *Structural distribution of Lennard-Jones clusters containing 562 to 1000 atoms*, J. Chem. Phys. **108** (2004) 9516-9520.

[3] I. Saika-Voivod, L. Poon and R. K. Bowles, *The role of fcc tetrahedral subunits in the phase behavior of medium sized Lennard-Jones clusters*, J. Chem. Phys. **133** (2010) 074503.

[4] Y. G. Chushak and L. S. Bartell, *Melting and freezing of gold nanoclusters*, J. Phys. Chem. B (2001) **105** 11605-11614.

[5] P. J. Steinhardt, D. R. Nelson and M. Ronchentti, *Bond-orientational order in liquids and glasses*, Phys. Rev. B (1983) **28** 784-805.

[6] Y. Wang, S. Teitel and C. Dellago, *Melting of icosahedral gold nanoclusters from molecular dynamics simulations*, J. Chem. Phys. (2005) **122**, 214722.

[7] S. C. Hendy and J. P. K. Doye, *Surface-reconstructed icosahedral structures for lead clusters* Phys. Rev. B. (2002) **66** 235402.

[8] H.-S. Nam, N. M. Hwang, B. D. Yu and J.-K. Yoon, *Formation of an icosahedral structure during the freezing of gold nanoclusters: Surface-induced mechanism*, Phys. Rev. Lett. (2002) **89** 275502.

[9] E. Mendez-Villuendas and R. K. Bowles, *Surface nucleation in the freezing of gold nanoparticles*, Phys. Rev. Lett. (2007) **98** 185503.

[10] D. P. Sanders, H. Larralde and F. Leyvraz, *Competitive nucleation and the Ostwald rule in a generalized Potts model with multiple metastable phases*, Phys. Rev. B (2002) **75** 132101.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Cluster Model of Total - Connected Flow with Local Information

Alexander P. Buslaev[1] **and Marina V. Yashina**[2]

[1] *Department of Mathematics, Moscow State Automobile and Road Tech. University*

[2] *Department of Mathematical Cybernetics and IT , Moscow Technical University of Communications and Informatics*

emails: apal2006@yandex.ru, yash-marina@yandex.ru

Abstract

We introduce some model of a flow. The flow is represented as a sequence of synchronized packets of particles, which move with the velocity depended on the flow density and interact by given laws.

Some quality properties of corresponded system of ordinary differential equations are developed.

Key words:
System of nonlinear differential equations, theory of traffic flow, following-the-leader model, cluster model of particles movement
MSC 2000: AMS codes 74H05, 91F99

1 Introduction

One of the basic models of traffic flow, follow-the-leader model, [1] - [5], [12], can be reduced to the study of differential equations of the following type

$$x_{n+1} - x_n = f(\dot{x}_n), \tag{1}$$

where $x_n(t)$ is a coordinate of the $n-$th vehicle, f increases monotonically, $f(0) > 0$,

$$x_n(t) < x_{n+1}(t), n = 1, 2, ... \ . \tag{2}$$

We call the flow satisfying the conditions (1)-(2) as *total - connected flow.*

Steady state, $\dot{x}_n(t) \equiv C_n, n = 1, ...,$ is equivalent to $\dot{x}_{n+1}(t) \equiv \dot{x}_n(t)$, i.e. $x_{n+1} - x_n \equiv C$ are uniformly located elements of the flow.

Hence, the steady state of the chain $x_1 < x_2 < ... < x_n$ is equivalent to a uniformly moving cluster with the velocity $v = f^{-1}(C)$ and the density $\rho = C^{-1}$.

2 Connected flow of stationary clusters

We assume the flow support consists of clusters. The cluster moves along a straight line at velocity

$$v = f^{-1}(\rho^{-1}) = g(\rho). \tag{3}$$

The function $g(\rho)$ is called *a state function.* The simplest type of a state function is linear function $0 \leq \rho \leq \rho_{max}$, $g(0) = v_{max}$, $g(\rho_{max}) = 0$. For simplicity, we assume $v_{max} = \rho_{max} = 1$.

Strategy of a *total-connected flow* needs the velocity regime of the follower to be adjusted to the velocity regime of the leader. That restricts the separation of flow to independent parts in the sense discussed below.

Neighboring clusters, i.e. *leader and follower,* interact with each other by *information transfer* inside the follower.

If only the leading edge of the follow cluster has the information on the approaching contact with leader, then this part of follower begins to transform itself to adapt to speed mode of the leader.

We refer to such a model as *a cluster model of flow with local information.*

3 Interation of clusters local information

We consider two available scenarios.

3.1 The slow cluster follows the fast leader

As the leading cluster has great velocity, the rising edge of the follower is transforming in the tail of leader with preservation of total mass of particles.

We derive differential equations for interaction between these two clusters.

Suppose that at time t the base of the left (following) cluster, i.e. cluster, which is behind, is the segment (x_1, x_2), and its height, i.e. flow density on the segment (x_1, x_2), equals to y_1, Fig.1.

The base of the cluster, which is moving ahead, is interval (x_2, x_3), having a height of y_2.

The left boundary cluster, which is in front, moving v_1 and, thus, at the moment time $t + \Delta t$ s on the horizontal axis corresponds to the point $x_1 + v_1\Delta t$.

The right boundary of the cluster, which is front moves at a speed of v_2 and at time $t + \Delta t$ s on the x-axis corresponds to the point $x_3 + v_2\Delta t$.

The heights of the clusters remain constant. The right boundary of the cluster, which coincides with the left edge of the right cluster moves with a speed that satisfies the condition that sum of the areas of rectangles remains constant.

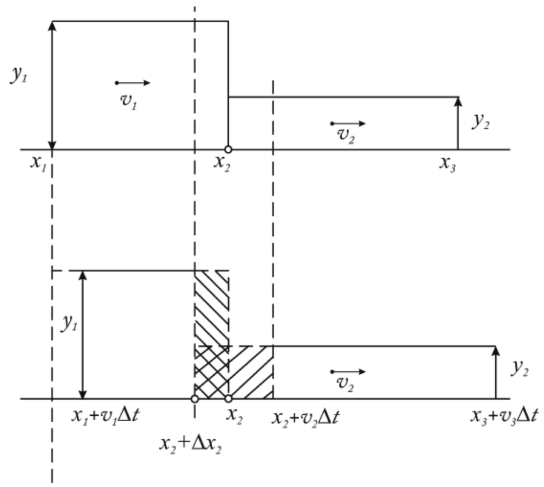


Figure 1: The slow cluster follows the fast cluster

Let $x_2 + \Delta x_2$ be a coordinate of the point on the x-axis corresponding to the boundary at time $t + \Delta t$.

We have for the case of movement of the slow cluster behind the fast cluster (Fig. 1):

$$(x_2 + v_2\Delta t - x_2 - \Delta x_2)\, y_2 = ((x_2 - x_1) - (x_2 + \Delta x_2 - x_1 - v_1\Delta t))y_1, \Longleftrightarrow$$

$$\Longleftrightarrow (v_2\Delta t - \Delta x_2)y_2 = (-\Delta x_2 + v_1\Delta t)y_1, \Longleftrightarrow$$

$$\Longleftrightarrow (v_2y_2 - v_1y_1)\Delta t = \Delta x_2(y_2 - y_1), \Longleftrightarrow$$

$$\Longleftrightarrow \dot{x}_2 = \frac{v_2y_2 - v_1y_1}{y_2 - y_1} = \frac{q_2 - q_1}{y_2 - y_1},$$

$q_i = \rho_i v_i$, $i = 1, 2$.

Thus

$$\begin{cases} \dot{x_1} = v_1 = f(y_1), \\ \dot{x_2} = \frac{v_2y_2 - v_1y_1}{y_2 - y_1} = \frac{q_2 - q_1}{y_2 - y_1}, \\ \dot{x_3} = v_2 = f(y_2). \end{cases} \tag{4}$$

3.2 The fast cluster follows the slow leader

We now assume that the slow cluster is moving ahead of the fast cluster (Fig. 2). Rising edge of the faster follower cluster is transformed into the stern of a slow cluster. Then the boundary of the clusters contact varies taking into account the conservation law of particles.

Hence (Fig.2)

$$(x_2 + v_2\Delta t - x_2 - \Delta x_2)y_2 = ((x_2 - x_1) - (x_2 + \Delta x_2 - x_1 - v_1\Delta t))y_1, \Longleftrightarrow$$

 ISBN:978-84-615-5392-1

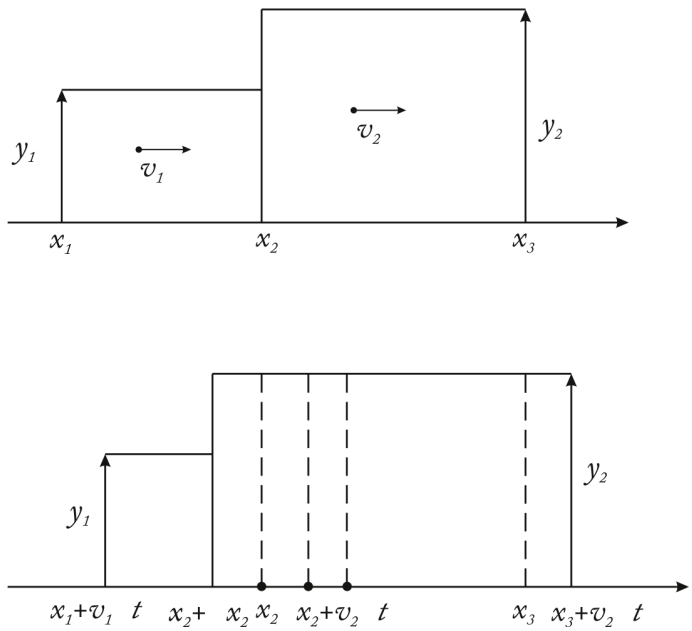


Figure 2: The fast cluster follows the slow cluster

$$\Longleftrightarrow (v_2\Delta t - \Delta x_2)y_2 = (-\Delta x_2 + v_1\Delta t)y_1, \Longleftrightarrow$$

$$\Longleftrightarrow (v_2y_2 - v_1y_1)\Delta t = (y_2 - y_1)\Delta x_2, \Longleftrightarrow$$

$$\Longleftrightarrow \dot{x_2} = \frac{v_2y_2 - v_1y_1}{y_2 - y_1} = \frac{q_2 - q_1}{y_2 - y_1}.$$

As in the case of (4) we obtain

$$\begin{cases} \dot{x_1} = v_1, \\ \dot{x_2} = \frac{v_2y_2 - v_1y_1}{y_2 - y_1} = \frac{q_2 - q_1}{y_2 - y_1}, \\ \dot{x_3} = v_2. \end{cases} \tag{5}$$

4 Cluster model on a ring

Assume a circle is divided into n segments

$$0 \le x_1^0 < x_2^0 < \ldots < x_n^0 < 1 < x_{n+1}^0 = x_1 + 1$$

At each segment $[x_i^0,\ x_{i+1}^0]$ there is defined a density y_i, $1 \le i \le n$. Velocity of $i-$th cluster is determined by the function $v_i = g(y_i)$. In accordance with Section 3, we have the following dynamical system

$$\dot{x}_i = \frac{q_{i+1} - q_i}{y_{i+1} - y_i} = \frac{v_{i+1}y_{i+1} - v_i y_i}{y_{i+1} - y_i},\ 1 \le i \le n. \tag{6}$$

Research of solutions of the system (6) allows to determine quantitative properties of the model. For simplicity, we consider

$$g(y) = 1 - y, 0 \le y \le 1. \tag{7}$$

Then the system (6) takes the form $1 \le i \le n$

$$\dot{x}_i = \frac{q_{i+1} - q_i}{y_{i+1} - y_i} = \frac{(1 - y_{i+1})y_{i+1} - (1 - y_i)y_i}{y_{i+1} - y_i} = 1 - y_{i+1} - y_i. \tag{8}$$

Lemma 1. For each i, $1 \le i \le n$, and sufficiently small t

$$(x_{i+1} - x_i)(t) = (x_{i+1}(0) - x_i(0)) - t(y_{i+1} - y_{i-1}). \tag{9}$$

Proof. Indeed, from (8) we obtain

$$(\dot{x}_{i+1} - \dot{x}_i) = (x_{i+1} - x_i)^{\cdot} = -(y_{i+1} - y_{i-1}).$$

Lemma 2. Suppose that $max_i(y_{i+1} - y_i) > 0$, in particular, when number n is even, $y_i \ne Const$. Then for $1 \le i \le n$,

$$t \ge T_1^* = \min_{i, y_{i+1} - y_i > 0} \frac{x_{i+1}(0) - x_i(0)}{y_{i+1} - y_{i-1}}$$

the model is described by the system (6) up to the numbering and the number of variables $k = k(n) < n$.

Proof. It follows from (9).

We call *a stationary* k-th orbit any set $\{y_1, ...y_n\}$, such that the system (6) has a stationary solution at $n = k$.

Lemma 3. Let $\vec{y} = (y_1, ...y_n)$ be stationary $n-$th orbit

$\Longleftrightarrow$

a) n is even, $y_{i-1} = y_{i+1} = ... = y$, $y_i = y_{i+2} = ... = 1 - y$, $y \ne 0.5$;

b) n is arbitrary, $\vec{y} = 0.5(1, ...1)$.

Proof.

If $y_{i+1} + y_i = 1$, $i = 1, ...n$, then all y_i with even indexes and all y_i with odd indexes are the same.

 ISBN:978-84-615-5392-1

We call *a dynamic $k-th$ orbit* any set $\{y_1, ...y_k\}$, such that $(x_{i+1} - x_i)(t) \equiv C_i$, $i = 1, ..., k$.

It is clear in this case that

$$\begin{cases} y_{i-1} = y_{i+1} = ..., \\ y_i = y_{i+2} = ... \end{cases} \tag{10}$$

In particular, when $k = 2$ equalities (10) are trivial.

Theorem. If $\{y_i\}$ are different, then system (6) , in general position, is reduced to a dynamic 2 - orbit at finite time.

5 Cluster model of total connected multilane movement

For simplicity we suppose the lane number is $m = 2$. If the fast cluster is catching up to the slow cluster, then, under the condition of a free lane, fast cluster flows into the adjacent lane at the contact level, keeping its velocity and density.

If the adjacent lane is busy, the change of the fast cluster into a slow performs by the scenario described in the preceding paragraphs to the moment when the adjacent lane would be free.

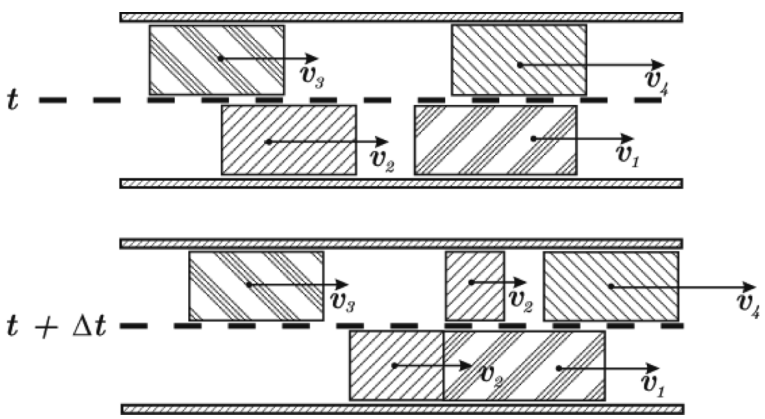


Figure 3: Two lanes cluster flow

The main problem is to describe the dynamics of two-lane cluster model and research the existence problem for a stationary state. Obviously, the number of values of densities $\{y_i\}$ can not increase. However, at the lane changing, the division of a cluster can occur, so the total number of clusters can increase.

6 Conclusions

The study approach combines the collective and individual characteristics of the behavior of elementary flow components, as result of which the movement reduces to interaction of clusters, that are conditionally - stationary in the sense of hydrodynamic models, [6]. On the other, clusters can be considered as particals with behaviour described by probabilistic approach, [7 - 9].

The resulting mathematical problems are independent interest, in spite of the fact that computational algorithms are created and tested, which results also can be discussed.

Acknowledgements

This work has been supported by by Ministry of Education and Science of the Russian Federation, project No.14.740.11.0397, and grant of RFBR No.11-01-12140-ofi.

References

[1] R. B. Morisson *The Traffic Flow Analogy to Compressible Fluid Flow,* Advanced Res. Eng. Bull., 1964.

[2] Hiroshi Inose, Takashi Hamada. *Road Traffic Control.* University of Tokyo Press. 1975

[3] R. W. Rothery *Car Following Models in Traffic Flow Theory*, Transportation research board, ed. Gartner N , Special report, **165** (1992) 4.1 – 4.42.

[4] Pipes L.A. *An operational Analysis of Traffic Dynamics,* Journal of Applied Physics, **24** (1953) 271–281.

[5] Buslaev A.P., Gasnikov A.V., Yashina M.V. *Mathematical Problems of Traffic Flow Theory.* Proceed. of the 2010 International Conference on Computational and Mathematical Methods in Science and Engineering, ed J.Vigo Aguar, Almeria, Spain, 26-30.06.2010, v.1, (2010) p.307-313

[6] Lighthill M.L., Whitham G.B. *On kinematic waves. A theory of traffic flow on long crowed roads.* Proceedings of the Royal Society of London, Piccadilly, London, (1955) A229 (1170) 317–345.

[7] Nagel K., Schreckenberg M. *A cellurar automation model for freeway traffic.* J. Physique. France, **2** (1992), 2221–2229.

[8] C.F. Daganzo. *The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory.* Transp.Res. B. V.28, **4** (1994) 269–287.

[9] C.F. Daganzo. *The cell transmission model, Part II: Network traffic.* Transp.Res. B. V.29, **2** (1995) 79–93.

[10] Nazarov A.I. *On stability of stationary modes in a system of nonlinear ODE arising in modeling of traffic flows.* Vestnik SPbGU, S.1, **3** (2006) (In Russian)

[11] A.P. Buslaev, A.V. Provorov, M.V. Yashina. *Current approaches to the study of connected flow of particles with motivation.* T-Comm. J. Telecommunications and Transport, **2** (2011) 61–62 (In Russian)

[12] A.P. Buslaev, A.V. Gasnikov, M.V.Yashina. *Selected Mathematical Problems of Traffic Flow Theory.* International Journal of Computer Mathematics. Volume 89, **3** , Special Issue: Topics of Contemporary Computational Mathematics. Section B. (2012) 409–432 DOI:10.1080/00207160.2011.611241

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

On algebraic properties of residuated multilattices and the adequate definition of filter

I. P. Cabrera[1], P. Cordero[1], G. Gutiérrez[1], J. Martínez[1] and M. Ojeda-Aciego[1]

[1] *Department of Applied Mathematics, University of Málaga, Spain*

emails: ipcabrera@uma.es, pcordero@uma.es, ggutierrez@uma.es, jmartinezd@uma.es, aciego@uma.es

Abstract

We continue the exploration of the residuated operations in the framework of multilattices. New algebraic properties of residuated multilattices are obtained, together with a study of the different possible approaches to the notion of filter of a residuated multilattice, and propose an adequate definition which combines the requirements associated to the underlying structures of pocrim and multilattice.

Key words: hyperstructures, pocrim, multilattices, filter, residuation.

1 Introduction and preliminary definitions

We continue our study of the algebraic structure of residuated multilattice initiated in [3]. In this paper, we will introduce new algebraic results that allow us to generalize the main properties of residuated lattices, as those introduced in [4]. Moreover, due to the fact that any residuated multilattice combines the structures of multilattice and pocrim, it is possible to use the notion of filter on the multilattice, filter on the pocrim, or give a new definition that combines both. These new properties play an important role so that we can obtain a suitable generalization of the concept of filter.

In order to make this paper as self-contained as possible, we will recall a commonly considered algebraic structure, that is, a partially ordered commutative residuated integral monoid (*pocrim*) [2].

Definition 1 *A tuple* $\mathcal{A} = (A, \leq, \odot, \to, \top)$ *is said to be a* partially ordered commutative residuated integral monoid*, briefly a* ***pocrim****, if, for every* $a, b, c \in A$*, the following properties hold:*

- $(A, \odot, \top)$ *is a commutative monoid with neutral element* $\top$
- $(A, \leq)$ *is a partially ordered set with maximum* $\top$*.*
- *the operations* $\odot$ *and* $\to$ *satisfy the adjointness condition, that is* $a \odot c \leq b$ *if and only if* $c \leq a \to b$*.*

A pocrim $\mathcal{A}$ *is said to be a* ***residuated lattice*** *if* $(A, \leq)$ *is a lattice. Moreover, a residuated lattice in which* $\odot$ *coincides with the meet operation is said to be a* ***Heyting algebra****.*

Monotonicity of the ordering relation $\leq$ wrt $\odot$ follows as a consequence of the definition. Specifically, it holds that $x \leq y$ implies $x \odot z \leq y \odot z$.

It is well-known that residuated lattices are considered to be the algebraic structures of substructural logics [6, 8, 12], which are logics without some of the structural rules of logic: weakening, contraction, or associativity.

We focus here on some extensions of the previously defined notions, by considering a partially-ordered set together with two non-deterministic operations which generalize the supremum and the infimum by weakening the restrictions imposed on a (complete) lattice, namely, the "existence of least upper bounds and greatest lower bounds" is relaxed to the "existence of *minimal* upper bounds and *maximal* lower bounds". Specifically, a *multisupremum of* a and b is defined as a minimal element of the set of upper bounds of a and b, we write $a \sqcup b$ to refer to *the set of all the multi-suprema of* a *and* b; the notion of multiinfimum $a \sqcap b$ is introduced similarly. Now, we can proceed with the formal definition of multilattice and related structures.

Definition 2

- *A poset* $(M, \leq)$ *is said to be a* ***multilattice*** *if for all* $a, b, x \in M$ *with* $a \leq x$ *and* $b \leq x$*, there exists*[1] $z \in a \sqcup b$*, such that* $z \leq x$*; and, similarly, for all* $a, b, x \in M$ *with* $a \geq x$ *and* $b \geq x$*, there exists* $z \in a \sqcap b$*, such that* $z \geq x$*.*
- *A multilattice is said to be* ***full*** *if* $a \sqcup b \neq \varnothing$ *and* $a \sqcap b \neq \varnothing$ *for all* $a, b \in M$*.*

The notion of multilattice was introduced originally by Benado [1], and further studied by Hansen [7], who proposed an algebraic equivalent definition of multilattice. More recently, another algebraic formalisation of the notion of multilattice was introduced in [9, 10]

[1] Note that the definition is consistent with the existence of two incomparable elements *without* any multisupremum. In other words, $a \sqcup b$, and also $a \sqcap b$, can be empty.

as a theoretical tool to deal with some problems in the theory of mechanised deduction in temporal logics. Multilattices arise as well in other research areas, such as fuzzy extensions of logic programming [11]: for instance, one of the hypotheses of the main termination result for sorted multi-adjoint logic programs [5] can be weakened only when the underlying set of truth-values is a multilattice (the question of providing a counter-example on a lattice remains open).

Definition 3 *A **residuated multilattice** is a pocrim whose underlying poset is a multilattice. If, in addition, there exists a bottom element, we say that the residuated multilattice is **bounded**.*

Notice that every residuated multilattice is full: for all $a, b \in M$ we have that $a, b \leq \top$ and, therefore, $a \sqcup b \neq \varnothing$. Furthermore, $a \odot b \leq a$, and $a \odot b \leq b$, hence $a \sqcap b \neq \varnothing$.

It is convenient to remark that any finite poset is actually a multilattice, hence the only proper examples of pocrims which are not multilattices have to be infinite.

2 Algebraic properties of residuated multilattices

Let us recall that if $(A, \leq)$ is a poset, we will denote by $\uparrow$ and $\downarrow$ the upper and lower closure operators respectively. That is, for all $B \subseteq A$

$$B\uparrow = \bigcup_{b\in B}[b) = \bigcup_{b\in B}\{x \in A \mid x \geq b\} \qquad B\downarrow = \bigcup_{b\in B}(b] = \bigcup_{b\in B}\{x \in A \mid x \leq b\}$$

So, we can generalize in terms of residuated multilattice the properties of residuated lattices presented in [4]:

Lemma 1 *Let M be a residuated multilattice, then the following items hold:*

1. $(x \odot y) \sqcup (x \odot z) = \text{minimals}\{x \odot (y \sqcup z)\}$ *for all $x, y, z \in M$.*
2. $x \odot y \in (x \sqcap y)\downarrow$ *for all $x, y \in M$.*
3. $x \odot (x \to y) \in (x \sqcap y)\downarrow$ *for all $x, y \in M$.*
4. $x \to y \in [x \to (x \sqcap y)]$ *for all $x, y \in M$.*

The following result relates $\odot$ to the operators $\sqcup$ and $\sqcap$.

Proposition 1 *Let M a residuated multilattice, then the following holds for all $x, y, z \in M$:*

1. $z \odot (x \sqcap y) \subseteq [(z \odot x) \sqcap (z \odot y)]\downarrow$

 ISBN:978-84-615-5392-1

2. $[(x \odot y) \sqcup (x \odot z)] \subseteq x \odot (y \sqcup z) \subseteq [(x \odot y) \sqcup (x \odot z)]\uparrow$

The result below relates $\rightarrow$ to the operators $\sqcup$ and $\sqcap$.

Proposition 2 *Let M a residuated multilattice, the following holds for all $x, y, z \in M$:*

1. $[(x \rightarrow z) \sqcap (y \rightarrow z)] \subseteq (x \sqcup y) \rightarrow z \subseteq [(x \rightarrow z) \sqcap (y \rightarrow z)]\downarrow$
2. $[(z \rightarrow x) \sqcap (z \rightarrow y)] \subseteq z \rightarrow (x \sqcap y) \subseteq [(z \rightarrow x) \sqcap (z \rightarrow y)]\downarrow$
3. $[(x \rightarrow y) \sqcap (y \rightarrow x)] \subseteq (x \sqcup y) \rightarrow (x \sqcap y) \subseteq [(x \rightarrow y) \sqcap (y \rightarrow x)]\downarrow$

The last result in this section has no counterpart in the case of residuated lattices.

Proposition 3 *Let M a residuated multilattice, then the following holds for all $x, y, z \in M$:*

1. $(x \sqcap y) \rightarrow z \subseteq [(x \rightarrow z) \sqcup (y \rightarrow z)]\uparrow$
2. $z \rightarrow (x \sqcup y) \subseteq [(z \rightarrow x) \sqcup (z \rightarrow y)]\uparrow$

3 The structure of filter in a residuated multilattice

Concerning applications in logic and artificial intelligence, the notions of filter and deductive system [13], closely related to *modus ponens*, deserve to be studied in depth.

Definition 4 *Given $\mathcal{A} = (A, \leq, \odot, \rightarrow, \top)$ a pocrim, a non-empty subset $F \subseteq A$ is said to be a **filter** if the following conditions hold:*

i) if $a, b \in F$, then $a \odot b \in F$

ii) if $a \leq b$ and $a \in F$, then $b \in F$.

Definition 5 *Given $\mathcal{A} = (A, \leq, \odot, \rightarrow, \top)$ a pocrim, a non-empty subset $F \subseteq A$ is said to be a **deductive system** if*

i) $\top \in F$ and

ii) $a \rightarrow b \in F$ and $a \in F$ imply $b \in F$.

Proposition 4 *The definitions of filter and deductive system are equivalent.*

Proof:

 ISBN:978-84-615-5392-1

1. Firstly, let us assume that F is a filter. Let $x \in F$. As $x \leq \top$, then $\top \in F$.

   Moreover, if $a \in F$ and $a \to b \in F$, then $a \odot (a \to b) \in F$. As $a \odot (a \to b) \leq b$, then $b \in F$.

2. Now let us assume that F is a deductive system. $a \leq b \iff a \to b = \top$. As $\top \in F$, $a \to b \in F$ and $a \in F$ we have that $b \in F$.

   Moreover let $a, b \in F$. As $a \odot b \leq a \odot b \iff b \leq a \to (a \odot b)$ and $b \in F$ then $a \to (a \odot b) \in F$. Now, as $a \in F$ we have that $a \odot b \in F$.

□

There exist several ways to give a definition for the notion of filter of a multilattice. In this section, we introduce the one which is more suitable for extending the classical results about congruences and homomorphisms.

Definition 6 *Let $(M, \sqcup, \sqcap)$ be a multilattice. A non-empty set $F \subseteq M$ is said to be a **filter** if the following conditions hold:*

1. *$a, b \in F$ implies $\varnothing \neq a \sqcap b \subseteq F$.*
2. *$a \in F$ implies $a \sqcup b \subseteq F$ for all $b \in M$.*
3. *For all $a, b \in M$, if $(a \sqcup b) \cap F \neq \varnothing$ then $a \sqcup b \subseteq F$.*

Due to the fact that any residuated multilattice combines the structures of multilattice and pocrim, it is possible to use the notion of filter on the multilattice, filter on the pocrim, or give a new definition that combines both. These three notions are not equivalent. To distinguish them, we will write **p-filter** to denote a filter of the pocrim and **m-filter**, a filter of the multilattice. We introduce now the notion of filter in a residuated multilattice.

Definition 7 *Let M be a residuated multilattice. A non-empty subset $F \subseteq M$ is said to be a **filter** if it is a deductive system and the following condition hold: $a \to b \in F$ implies $a \sqcup b \to b \subseteq F$ and $a \to a \sqcap b \subseteq F$.*

Theorem 1 *Let M be a residuated multilattice and F a deductive system. Then, F is a filter if and only if F is an m-filter and the following conditions hold:*

i) for all $x, y \in a \sqcup b$, if $x \to y \in F$ then $y \to x \in F$.

ii) for all $x, y \in a \sqcap b$, if $x \to y \in F$ then $y \to x \in F$.

 ISBN:978-84-615-5392-1

Proof: Suppose that F is a filter and let $a, b \in F$. As $a \leq b \to a$, then $b \to a \in F$. Therefore, $b \to a \sqcap b \subseteq F$. So, given $x \in a \sqcap b$, as $b \to x \in F$ and $b \in F$, then $x \in F$. On the other hand, suppose that there exists $x \in (a \sqcup b) \cap F$. If $a \sqcup b$ is a singleton, then, trivially, $a \sqcup b \subseteq F$. Otherwise, let $y \in a \sqcup b$ such that $x \neq y$. As $a, b \leq x, y$, there exist two different elements $a', b' \in x \sqcap y$ such that $a \leq a'$ and $b \leq b'$. Observe that $\top = a' \to x = a' \to y \in F$. As $x \in F$ and $x \leq y \to x$, then $y \to x \in F$. Thus, $y \to x \sqcup y \subseteq F$ which implies that $y \to a', y \to b' \in F$. From $y \geq a'$, we obtain $y \to b' \leq a' \to b'$ and so, $a' \to b' \in F$. Therefore, $a' \sqcup b' \to b' \subseteq F$, which leads to $x \to b' \in F$. As also $\top = b' \to y \in F$, then $x \to y \in F$. Finally, as $x \in F$, so $y \in F$.

Suppose now that F is an m-filter in which both conditions i) and ii) hold and let $a, b \in M$ such that $a \to b \in F$. By 2, item $[i)]$, it holds

$$[(a \to b) \sqcap (b \to b)] \subseteq (a \sqcup b) \to b$$

thus, there exists $x_1 \in a \sqcup b$ such that $a \to b = x_1 \to b$. If $a \sqcup b$ is a singleton, the proof is over. Otherwise, given $x_2 \in a \sqcup b$, since $\top = b \to x_2 \in F$ and $x_1 \to b \in F$, we have that $x_1 \to x_2 \in F$. Using hypothesis, it implies that also $x_2 \to x_1 \in F$ and again with $x_1 \to b \in F$, we obtain that $x_2 \to b \in F$.

□

4 Conclusions and future work

We have introduced new properties that allow us to obtain a suitable generalization of the concept of filter. As future work, on the one hand, we will focus on the study of homomorphism and congruence in order to guarantee that the classical relationship between these three concepts still holds in the framework of residuated multilattices. On the other hand, the specific form of the new properties introduced in Section 2 strongly suggests a possible interpretation in terms of rough sets, which will be studied later.

Acknowledgements

Partially supported by projects TIN2009-14562-C05-01 (Science Ministry of Spain), and P09-FQM-5233 (Junta de Andalucía).

References

[1] M. Benado. Les ensembles partiellement ordonnés et le théorème de raffinement de Schreier. I. *Čehoslovack. Mat. Ž.*, 4(79):105–129, 1954.

[2] W. J. Blok and J. G. Raftery. Varieties of commutative residuated integral pomonoids and their residuation subreducts. *Journal of Algebra*, 190:280–328, 1997.

[3] I. P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, and M. Ojeda-Aciego. Residuated operations in hyperstructures: residuated multilattices. In J. Vigo-Aguiar, editor, *Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering.*, volume 1, pages 259–266, 2011.

[4] L. C. Ciungu. On the lattice of congruence filters of a residuated lattice. *Annals of University of Craiova*, 33:189–207, 2006.

[5] C. Damásio, J. Medina, and M. Ojeda-Aciego. Termination of logic programs with imperfect information: applications and query procedure. *Journal of Applied Logic*, 5(3):435–458, 2007.

[6] N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated lattices: an algebraic glimpse at substructural logics*, volume 151 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2007.

[7] D. J. Hansen. An axiomatic characterization of multilattices. *Discrete Math.*, 33(1):99–101, 1981.

[8] T. Kowalski and H. Ono. Fuzzy logics from substructural perspective. *Fuzzy Sets and Systems*, 161(3):301–310, 2010.

[9] J. Martínez, G. Gutiérrez, I. P. de Guzmán, and P. Cordero. Generalizations of lattices via non-deterministic operators. *Discrete Math.*, 295(1-3):107–141, 2005.

[10] J. Martínez, G. Gutiérrez, I. P. de Guzmán, and P. Cordero. Multilattices via multi-semilattices. In *Topics in applied and theoretical mathematics and computer science*, Math. Comput. Sci. Eng., pages 238–248. WSEAS, Athens, 2001.

[11] J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. Fuzzy logic programming via multilattices. *Fuzzy Sets and Systems*, 158(6):674–688, 2007.

[12] H. Ono. Substructural logics and residuated lattices—an introduction in *trends in logic: 50 years of studia logica*. *Studia Logica*, 50:177–212, 2003.

[13] J. Rachůnek and D. Šalounová. Filter theory of bounded residuated lattice ordered monoids. *Journal of Multiple-Valued Logic and Soft Computing*, 16(3-5):449–465, 2010.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Graph operations and Lie algebras

José Cáceres[1], Manuel Ceballos[2], Juan Núñez[2], María Luz Puertas[1] and Ángel F. Tenorio[3]

[1] *Departamento de Estadística y Matemática Aplicada, Universidad de Almería.*

[2] *Departamento de Geometría y Topología, Facultad de Matemáticas. Universidad de Sevilla.*

[3] *Dpto. de Economía, Métodos Cuantitativos e Historia Económica, Escuela Politécnica Superior. Universidad Pablo de Olavide.*

emails: jcaceres@ual.es, mceballos@us.es, jnvaldes@us.es, mpuertas@ual.es, aftenorio@upo.es

Abstract

In this paper, we deal with vertex amalgamation on graphs and combinatorial structures in order to obtain some criteria to determine when there exists a Lie algebra associated with a combinatorial structure arising from this operation. Moreover, we show an algorithmic method to implement it.

Key words: Digraph, Combinatorial structure, Lie algebra, Combinatorial operations, Algorithm.
MSC 2000: 17B60, 05C25, 05C20, 05C90, 68W30, 68R10, 05C85.

1 Introduction

Finding relations between different fields of Mathematics is an important goal in mathematical research. Both Lie Theory and Graph Theory are running in a high level due to their several applications in Engineering, Physics and Applied Mathematics, in addition to their theoretical study. There exists a close relation between both theories. For example, graphs have been used to study semisimple Lie algebras, since trees perform an important

role to determine the Dynkin diagrams associated to such algebras [8]. Graph Theory is also applied to study the representation of finite-dimensional algebras [7].

Our main goal is to make progress in the link between Lie algebras and combinatorial structures. Hence, we are proceeding with previous works [1, 2, 3, 4, 6] in the literature opening this research line. This time, we study the translation of vertex amalgamation on graphs and combinatorial structures into the language of Lie algebras.

The structure of this paper is the following: after reviewing some well-known results on Lie and Graph Theory in Section 2, Section 3 recalls the mapping introduced in [1] to associate combinatorial structures with Lie algebras. Next, in Section 4 we study the vertex amalgamation studying some criteria under which structures obtained from this operation are associated with Lie algebras. Finally, Section 5 presents an algorithmic method related to this operation checking if the graph obtained is associated with a Lie algebra.

2 Preliminaries

For a general overview on Lie algebras and graph theory, the reader can consult [9, 5].

Definition 1 *A Lie algebra* $\mathfrak{g}$ *is a vector space with a second bilinear inner composition law* $([\cdot,\cdot])$ *called the* bracket product *or Lie bracket, which satisfies* $[X,X]=0,\ \forall X\in\mathfrak{g}$ and $[[X,Y],Z]+[[Y,Z],X]+[[Z,X],Y]=0,\ \forall X,Y,Z\in\mathfrak{g}$. *The last expression is called the* Jacobi identity.

Given a basis $\{e_i\}_{i=1}^n$ *of* $\mathfrak{g}$, *its* structure (or Maurer-Cartan) constants *are defined by* $[e_i,e_j]=\sum c_{i,j}^h e_h$, *for* $1\leq i<j\leq n$.

Definition 2 *Given a Lie algebra* $\mathfrak{g}$, *its center is* $Z(\mathfrak{g})=\{X\in\mathfrak{g}\mid [X,Y]=0,\ \forall Y\in\mathfrak{g}\}$.

Definition 3 *A* graph *is a ordered pair* $G=(V,E)$, *where* V *is a non-empty set of vertices and* E *is a set of unordered pairs (edges) of two vertices. If the edges are ordered pairs of vertices, then the graph is named* digraph.

Definition 4 *Let* $G=(V,E)$ *be a graph. For a vertex* $v\in V$, *the* (open) neighbourhood *of* v *in* G *is the vertex subset* $N(v)=\{w\in V\mid (v,w)\in E\}$. *Two vertices* $u,v\in V$ *are* twins *if they have the same neighbourhoods; i.e.* $N(u)=N(v)$.

Definition 5 *Given a digraph* $G=(V,E)$, *a vertex* $v\in V$ *is a* sink *(resp. a* source*) if all the edges incident with* v *are oriented towards* v *(resp. oriented from* v*). This definition is illustrated in Figure 1.*

 ISBN:978-84-615-5392-1

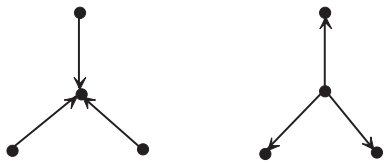


Figure 1: Example of sinks and sources, respectively.

Definition 6 *Given $n \in \mathbb{N}$, P_n is a weighted digraph of n vertices alternating sources with sinks.*

3 Associating combinatorial structures with Lie algebras

Let $\mathfrak{g}$ be an n-dimensional Lie algebra with basis $\mathcal{B} = \{e_i\}_{i=1}^n$. The structure constants are given by $[e_i, e_j] = \sum_{k=1}^n c_{i,j}^k e_k$ and, hence, the pair $(\mathfrak{g}, \mathcal{B})$ is associated with a combinatorial structure built according to the following steps in the method introduced in [1]

a) Draw vertex i for each $e_i \in \mathcal{B}$.

b) Given three vertices $i < j < k$, draw the full triangle ijk if and only if $(c_{i,j}^k, c_{j,k}^i, c_{i,k}^j) \neq (0,0,0)$. Then, the edges ij, jk and ik have weights $c_{i,j}^k$, $c_{j,k}^i$ and $c_{i,k}^j$, respectively.

   b1) Use a discontinuous line (named *ghost edge*) for edges with weight zero.

   b2) If two triangles ijk and ijl with $1 \leq i < j < k < l \leq n$ satisfy $c_{i,j}^k = c_{i,j}^l$, draw only one edge between vertices i and j shared by both triangles (see Figure 2).

c) Given two vertices i and j with $1 \leq i < j \leq n$ and such that $c_{i,j}^i \neq 0$ (resp. $c_{i,j}^j \neq 0$), draw a directed edge from j to i (resp. from i to j), as can be seen in Figure 3.

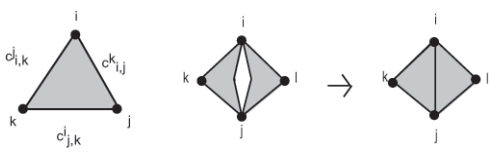


Figure 2: Full triangle and two triangles sharing an edge.

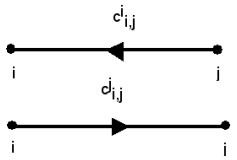


Figure 3: Directed edges.

 ISBN:978-84-615-5392-1

4 Vertex amalgamation

The amalgamation of two combinatorial structures consists in pasting both structures by identifying a vertex in both configurations. We determine under which conditions the structure obtained from this operation preserves the association with a Lie algebra.

4.1 Digraphs associated with Lie algebras and amalgamation

Proposition 1 *Let G be a digraph not associated with Lie algebras. Then, every digraph obtained from G by using vertex amalgamation is neither associated with Lie algebras.*

Proposition 2 *Let G and G' be two digraphs associated with the Lie algebras L and L' respectively. We consider the amalgamation of G and G' by an isolated vertex of G'. Then, there exists a unique Lie algebra associated with the amalgamation given by $L \oplus \bar{L}$, $\bar{L}$ is the Lie algebra associated with the subgraph $G' - \{v\}$ of G'.*

Proposition 3 *Let G and G' be two digraphs associated with Lie algebras. We consider the amalgamation by a non-isolated vertex. Then, the following statements hold*

1) *If G is an oriented 2-cycle, then no Lie algebra is associated with the amalgamation.*
2) *If G contains 3-cycles (structures from Theorem [1, Theorem 3.6]), then the amalgamation is associated with a Lie algebra if and only if the amalgamation vertex is a sink in G and G'. Moreover, either G and G' are digraphs of the same type or G' is a digraph P_n.*
3) *If G and G' do not contain 3- or 2-cycles, the amalgamation is associated with a Lie algebra if and only if the amalgamation vertex is of the same type in G and G'.*

4.2 Full triangles associated with Lie algebras and amalgamation

Lemma 1 *Let G and T be respectively a digraph and a triangular structure, both associated with Lie algebras. Then, the amalgamation of G and T by an isolated vertex v of G is associated with the Lie algebra $L \oplus \bar{L}$, where L and $\bar{L}$ are the Lie algebras associated with T and $G - \{v\}$, respectively.*

Proposition 4 *The amalgamation of a full triangle and a digraph by a non-isolated vertex k is associated with a Lie algebra if and only if k is a source and the opposite edge to k in the full triangle is ghost.*

 ISBN:978-84-615-5392-1

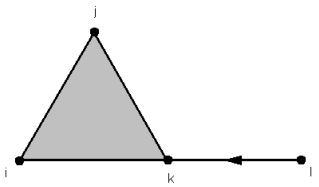


Figure 4: Amalgamation by a sink.

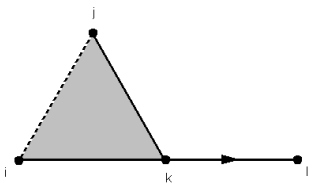


Figure 5: Amalgamation by a source.

Proposition 5 *The amalgamation of two full triangles by a vertex is associated with a Lie algebra if and only if either the amalgamation vertex is only incident with ghost edges and its opposite edges are full, or the edges not being incident with the amalgamation vertex are ghost.*

5 Algorithm for the amalgamation of two digraphs

In this section, we show an algorithmic procedure to compute the amalgamation of two digraphs associated with Lie algebras. We also study if the digraph obtained is associated with a Lie algebra. We consider the following two steps:

a) Compute the amalgamation of two digraphs associated with Lie algebras.

b) Check if the digraph obtained in the previous step is associated with a Lie algebra.

We have implemented this procedure in the symbolic computation package MAPLE 12, loading the libraries `DifferentialGeometry`, `LieAlgebras` and `GraphTheory`.

The first step is executed by the routine amalgamation, which receives two digraphs `G` and `H`. Both of them are defined with the order `Digraph(V,E)`, where `V` is a list with the vertices of `G` and `E` is a set whose elements are the edge (i.e. ordered pairs of vertices) with their weight. To implement this routine, several local variables are defined and a loop is programmed to compute the amalgamation.

```
> amalgamation:=proc(G,H)
> local U,V,A,B,W,C;U:=Vertices(G);V:=Vertices(H);A:=Edges(G,weights);B:=Edges(H,weights);
> W:=U; C:=A union B; for i from 1 to nops(V) do if member(V[i],W)=true then W:=W;
> else W:=[op(W),V[i]]; end if; end do; Ga:=Digraph(W,C); return Ga; end proc:
```

Now, the representation of this digraph can be obtained with the following sentence

```
> DrawGraph(amalgamation(G,H));
```

 ISBN:978-84-615-5392-1

Next, after computing the amalgamation, the routine `program` checks if the digraph obtained with this graph operation is associated with a Lie algebra. This routine builds a vector space associated with the digraph, which is the candidate to be its associated Lie algebra adding the bracket product. This routine receives the list `V` with the vertices of the digraph and the set `E` with its directed, weighted edges. As outputs, we obtain the vector space with basis $\{e_i\}_{i=1}^{n}$, where e_i corresponds to vertex i in the list `V`, and the brackets associated with the edges in the set `E`.

```
> program:=proc(V,E)
> local B, L; B:=[]; L:=[]; for x from 1 to nops(V) do B:=[op(B),e[x]];
> end do; for i from 1 to nops(E) do if E[i][1][1] < E[i][1][2] then
> L:=[op(L),[[E[i][1][1],E[i][1][2],E[i][1][2]],E[i][2]]];
> else L:=[op(L),[[E[i][1][2],E[i][1][1],E[i][1][2]],E[i][2]]];
> end if; end do; return _DG([["LieAlgebra",Alg1,[nops(V)]],L]);
> end proc:
```

Once we have implemented the routine `program`, we define the law by the sentence

```
> DGsetup(program(V,E));
```

After defining this vector space, saved as `Alg1`, we test if the Jacobi identities hold.

```
Alg1 > Query(Alg1,"Jacobi");
```

The vector space `Alg1`, defined by the output of `program`, is a Lie algebra if and only if the answer `true` is obtained for this question.

Acknowledgment

This work has been partially supported by MTM2010-19336 and FEDER.

References

[1] A. Carriazo, L.M. Fernández, J. Núñez, Combinatorial structures associated with Lie algebras of finite dimension, Linear Algebra Appl. 389 (2004), 43–61.

[2] M. Ceballos, J. Núñez, A.F. Tenorio, Complete triangular structures and Lie algebras, Int. J. Computer Math. 88:9 (2011), 1839–1851.

[3] M. Ceballos, J. Núñez, A.F. Tenorio, Study of Lie algebras by using combinatorial structures, Linear Algebra Appl. 436 (2012), 349–363.

[4] M. Ceballos, J. Núñez, A.F. Tenorio, Combinatorial structures and lie algebras of upper-triangular matrices, Appl. Math. Lett. 25 (2012), 514–519.

[5] R. Diestel, Graph Theory, 4th. Edition, Springer-Verlag, Heidelberg, 2010.

[6] L.M. Fernández, L. Martín-Martínez, Lie algebras associated with triangular configurations. Linear Algebra Appl. 407 (2005), 43–63.

[7] M. Primc, Basic representations for classical affine Lie algebras, J. of Algebra 228 (2000), 1–50.

[8] J.P. Serre, Algèbres de Lie Semi-Simples Complexes, Benjamin Inc., New York, 1996.

[9] V.S. Varadarajan, Lie Groups, Lie Algebras and Their Representations, Springer, New York, 1984.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Ecoepidemics with group defense and infected prey protected by the herd.

Elena Cagliero[1] and Ezio Venturino[1]

[1] *Dipartimento di Matematica "Giuseppe Peano", Università di Torino,*
via Carlo Alberto 10, 10123 Torino, Italy

emails: elenacagliero87@yahoo.it, ezio.venturino@unito.it[1]

Abstract

In this paper we consider a population model of predator-prey type in which prey gather together for defense purposes. A transmissible and unrecoverable disease is assumed to affect the prey. We characterize the system behavior, establishing that ultimately either only the susceptible prey survive, or the disease becomes endemic, but the predators are wiped out. Another alternative is that the disease is eradicated, with sound prey and predators thriving at an equilibrium or through persistent population oscillations. Finally, the two populations can thrive together, with the persisting disease still affecting the prey. The only alternative that in these circumstances is impossible, is the fact that predators can thrive just with infected prey. But this is a consequence of the model assumptions, in that infected prey are assumed to be too weak to sustain themselves. A peculiarity of the model is the singularity-free reformulation, which leads to three entirely new dependent variables to describe the system.

Key words: group defense, epidemics, predator-prey, disease transmission
MSC 2000: AMS codes 92D30, 92D25, 92D40

1 Introduction

The model we consider here is a prey-predator system in which the disease develops in prey. The latter gather together and live in a herd. Following recently introduced ideas, [2, 1], the large predators will hunt alone the herd and in it, it will be the individuals on the edge

[1]This paper was completed and written during a visit of the second author at the Max Planck Institut für Physik Komplexer Systeme in Dresden, Germany. The author expresses his thanks for the facilities provided.

of the bunch that will mostly bear the burden of the attack. In mathematical terms, the "size" of the prey population occupying the edge of the herd is proportional to the square root of the total population. Thus, instead of the standard mass action or Holling type II terms usually employed to model the predation mechanism, the predator-prey interactions are mathematically described via a term containing the square root of the prey population, coupled as usual with the predators' population. This is a different idea from the approach as the one used in [6], in which the defense mechanism is modelled via a suitable response function. In [12], these ideas are extended to another situation, in which a disease affects the prey. For further developments, see [4]. Thus, in this way the first ecoepidemic model of this sort is proposed. An idea of this kind had been presented for predators hunting in packs in [5].

Ecoepidemic models in fact contain a basic interacting population system on top of which a contagious disease is present. Models of this type are known since about a quarter of a century, [7, 3, 9] and are currently of wide interest among scientists, see [11] or [8] for an account of some of the developments of this branch of mathematical biology merging the two fields of population theory and epidemiology.

Coupling ecoepidemic systems with group defense is a step taken by one of the authors very recently, [12]. In the formulation of the model however, there is a kind of asymmetry in the way in which healthy and infected prey are dealt with by predators. Although both are hunted, the predation assumes in [12] two different mathematical forms, one containing the square root as discussed above, the other one the standard Holling type I interaction term. In fact, the additional basic assumption with respect to the standard predator-prey model of [1], which we will remove here, that has been formulated in [12] consists in the fact that the diseased prey are assumed to be left behind by the healthy herd. Therefore they are subject to hunting by predators on a one-to-one basis, a fact which is modeled as in the classical Lotka-Volterra system with the standard mass action term.

In this paper, we extend the model to encompass instead the situation in which the infected prey still remain in the herd, and mix with the healthy ones. Therefore they can occupy any position in the bunch, including the ones near the boundary. They are therefore subject to hunt as all the other susceptible prey. Mathematically speaking, the change amounts to the following: the square root term that formerly contained only healthy individuals, is now replaced by a square root term containing the whole prey population.

The paper is organized as follows. In the next Section we present the model. In Section 3 we redefine the basic variables to obtain a singularity-free system and adimensionalize it. The system's equilibria are assessed in Section 4. Section 5 contains their stability analysis. Hopf bifurcations are investigated in Section 6. The brief Section 7 summarizes the results interpretation in terms of the original model variables. Simulations are presented in Section 8 and a final discussion concludes the paper.

2 The model

Let S denote the healthy prey population, I be the infected prey and P the predators. We assume that the infection process running among the prey does not hinder them, so that infected individuals can still remain in the herd. The predators attack the prey, and the individuals at the edge of the bunch are the most likely to be captured by the predators. Since the infected do not remain behind the herd, they populate both the "inside" of the bunch as well as its boundary. Therefore they can be captured as well as the healthy prey.

Following the arguments expounded in [2, 1, 12], if we assume that the total prey population density $S + I$ is uniformly distributed on the land occupied by the herd, the number of the individuals staying on the border is proportional to the square root of this density. With these assumptions the system can be written as

$$\begin{aligned} \frac{dS}{dt} &= rS\left(1-\frac{S+I}{K}\right)-\sigma\frac{SI}{S+I}-qPS\frac{\sqrt{S+I}}{S+I} \\ \frac{dI}{dt} &= \sigma\frac{SI}{S+I}-wPI\frac{\sqrt{S+I}}{S+I}-\mu I \\ \frac{dP}{dt} &= -mP+gP\frac{S}{\sqrt{S+I}}+fP\frac{I}{\sqrt{S+I}} \end{aligned} \tag{1}$$

where all the parameters are assumed to be nonnegative. Here, r denotes the birth rate of healthy prey, σ is the disease incidence, q the predation rate on healthy prey, w the predation rate on infected prey, μ the natural plus disease-related mortality rate of infected prey, m the death rate of predators, e is the uptake due to predation for the predators, K the environment's carrying capacity.

The first equation shows that healthy prey follow a logistic growth, with intraspecific competition due also to the infected. Then there is the disease contagion mechanism, which is here assumed to be modelled by the standard incidence. Finally, healthy prey on the edge of the herd are captured by the predators, at rate q. Note that the last term expresses how many sound prey stay on the border. In fact, the population on the boundary is $\sqrt{S+I}$ as argued earlier. Of this, only the fraction $S(S+I)^{-1}$ is represented by healthy individuals. Note that the corresponding dual fraction $I(S+I)^{-1}$ gives the infected individuals on the boundary and is found in the second equation, in the predation term. Further, predation on infected prey occurs at rate w. The disease is assumed to be unrecoverable, for which the individuals that get it enter into the class I and can leave it only by being captured by predators, or via natural plus disease-related mortality. In the last equation the predators' dynamics transpires, which are dependent on the prey for their survival, otherwise they will die at rate m. Predators hunt the healthy and the infected prey alike, but at different rates.

In view of the assumptions stated above, some intrinsic relationships among the parameters hold. First of all $q \leqslant w$ and $g \leqslant f$ since predators hunt infected prey more easily

 ISBN:978-84-615-5392-1

than sound ones; further, $g < q$ and $f < w$, saying that not the whole captured prey are turned into new predators.

In view of singularities present in (1), we need to reformulate the system.

3 Preliminary steps - Reformulation

We proceed to the singularity elimination, via several steps. At first, we set $T = \sqrt{S+I}$ in order to remove the square root term. We thus obtain

$$\begin{aligned} \frac{dS}{dt} &= \sigma\frac{S^2}{T^2} - q\frac{PS}{T} + (r-\sigma)\,S - \frac{r}{K}ST^2 \\ \frac{dT}{dt} &= -\frac{\mu}{2}T - \frac{r}{2K}ST + \left(\frac{r}{2}+\frac{\mu}{2}\right)\frac{S}{T} + \left(\frac{w}{2}-\frac{q}{2}\right)\frac{PS}{T^2} - \frac{w}{2}P \\ \frac{dP}{dt} &= -mP + fPT + (g-f)\,\frac{PS}{T}. \end{aligned} \tag{2}$$

Then, let $V = ST^{-1}$ in place of S. The system (2) becomes

$$\begin{aligned} \frac{dV}{dt} &= \frac{r}{2K}V^2T + \left(\sigma - \frac{r}{2} - \frac{\mu}{2}\right)\frac{V^2}{T} + \left(\frac{q}{2}-\frac{w}{2}\right)\frac{PV^2}{T^2} \\ &\quad + \left(r - \sigma + \frac{\mu}{2}\right)V - \frac{r}{K}VT^2 + \left(\frac{w}{2}-q\right)\frac{PV}{T} \\ \frac{dT}{dt} &= -\frac{\mu}{2}T - \frac{r}{2K}VT^2 + \left(\frac{r}{2}+\frac{\mu}{2}\right)V + \left(\frac{w}{2}-\frac{q}{2}\right)\frac{PV}{T} - \frac{w}{2}P \\ \frac{dP}{dt} &= -mP + fPT + (g-f)\,PV. \end{aligned} \tag{3}$$

The third step introduces another new variable, $A = VT^{-1}$ replacing V, to reformulate (3) as

$$\begin{aligned} \frac{dA}{dt} &= (\sigma - r - \mu)\,A^2 + \frac{r}{K}A^2T^2 + (q-w)\,\frac{PA^2}{T} + (r+\mu-\sigma)\,A - \frac{r}{K}AT^2 + (w-q)\,\frac{PA}{T} \\ \frac{dT}{dt} &= -\frac{r}{2K}AT^3 - \frac{\mu}{2}T + \left(\frac{r}{2}+\frac{\mu}{2}\right)AT + \left(\frac{w}{2}-\frac{q}{2}\right)PA - \frac{w}{2}P \\ \frac{dP}{dt} &= -mP + fPT + (g-f)\,PAT. \end{aligned} \tag{4}$$

This is still unsatisfactory, in view of the presence of the variable T in the denominator. The next step introduces the variable $U = PT^{-1}$ in place of P, to get the new system with

 ISBN:978-84-615-5392-1

no singularities:

$$\begin{aligned}
\frac{dA}{dt} &= (\sigma - r - \mu)\, A^2 + \frac{r}{K} A^2 T^2 + (q - w)\, A^2 U \\
&\quad + (r + \mu - \sigma)\, A - \frac{r}{K} A T^2 + (w - q)\, AU, \\
\frac{dT}{dt} &= -\frac{r}{2K} A T^3 - \frac{\mu}{2} T + \left(\frac{r}{2} + \frac{\mu}{2}\right) AT - \frac{w}{2} UT + \left(\frac{w}{2} - \frac{q}{2}\right) AUT, \\
\frac{dU}{dt} &= \frac{w}{2} U^2 + \left(\frac{q - w}{2}\right) AU^2 + \left(\frac{\mu}{2} - m\right) U + (g - f)\, AUT \\
&\quad - \left(\frac{r + \mu}{2}\right) AU + fUT + \frac{r}{2K} AUT^2.
\end{aligned} \tag{5}$$

Combining all the substitutions made, we find the new variables definitions in terms of the original model variables, as follows

$$A = \frac{V}{T} = \frac{S}{T^2} = \frac{S}{S + I}, \quad U = \frac{P}{T} = \frac{P}{\sqrt{S + I}}, \quad T = \sqrt{S + I},$$

which allow an interpretation of their meanings. It follows indeed that A represents the fraction of healthy prey with respect to the total amount of prey, T is the total prey population on the edge of the herd and U denotes the ratio of predators over the total prey population occupying the edge of the area.

4 Equilibria

Note first of all that in eliminating singularities we had to divide by T, therefore this variable must be different from zero, in fact strictly positive, so that we exclude possible equilibria with $T = 0$. Mathematically, there is a second reason of geometric nature, as T represents the population of the herd on its boundary, and the latter is certainly never empty for a nonvanishing herd. There are thus only four possible equilibria.

Equilibrium $(A, T, U) = (0, +, 0)$ is infeasible since the second equation of (5) cannot be satisfied, as it does for $(A, T, U) = (0, +, +)$, so that we cannot accept this equilibrium either.

For $(A, T, U) = (+, +, 0)$, the first equation of (5) gives

$$\frac{r}{K} T^2 (A - 1) = (r + \mu - \sigma)(A - 1) \tag{6}$$

so that two cases arise.

If $A = 1$, from the second equation of (5) we have $T = \sqrt{K}$, giving the equilibrium $E_1 = (A_1, T_1, U_1) = \left(1, \sqrt{K}, 0\right)$ with unconditional feasibility.

 ISBN:978-84-615-5392-1

Alternatively, if $A < 1$, we find

$$T = \sqrt{\frac{K}{r}(r + \mu - \sigma)}, \quad A = \frac{\mu}{\sigma}.$$

We have thus found the equilibrium

$$E_2 = (A_2, T_2, U_2) = \left(\frac{\mu}{\sigma}, \sqrt{\frac{K}{r}(r + \mu - \sigma)}, 0\right)$$

under the conditions

$$r + \mu - \sigma > 0, \quad \mu < \sigma, \tag{7}$$

with the second one arising from the very definition of A.

Remark. If in E_2 we let $\mu = \sigma$, we reobtain E_1.

To find the equilibria with all nonvanishing components $(A, T, U) = (+, +, +)$ that we can call coexistence equilibria, we sum the second and the third equations of (5) to get

$$T = \frac{m}{(g - f)A + f}, \quad A \neq \frac{f}{f - g}. \tag{8}$$

From the first equation of (5) we have

$$(A - 1)\left[(\sigma - r - \mu) + \frac{r}{K}T^2 + (q - w)\right] = 0,$$

giving again two possibilities.

For $A = 1$ we get $T = mg^{-1}$ and the last equation of (5) then yields

$$U = \frac{r}{g^2 qK}\left(g^2 K - m^2\right),$$

which is positive if

$$K > \left(\frac{m}{g}\right)^2. \tag{9}$$

Thus we found the equilibrium

$$E_3 = (A_3, T_3, U_3) = \left(1, \frac{m}{g}, \frac{r}{g^2 qK}\left(g^2 K - m^2\right)\right)$$

with feasibility condition (9).

If instead $A \neq 1$ we solve the system

$$\begin{aligned} (\sigma - r - \mu) + \frac{r}{K}T^2 + (q - w)U &= 0, \\ -\frac{r}{2K}AT^2 - \frac{\mu}{2} + \left(\frac{r + \mu}{2}\right)A - \frac{w}{2}U + \left(\frac{w - q}{2}\right)AU &= 0, \end{aligned} \tag{10}$$

with T given by the first equation in (8). Now in the first equation (10) write U as a function of A:

$$(q-w)\,U = (r+\mu-\sigma) - \frac{r}{K}\frac{m^2}{[(g-f)\,A+f]^2}. \tag{11}$$

Now if $q-w=0$ the first equation of (10) simplifies to give

$$T = \sqrt{\frac{K}{r}\,(r+\mu-\sigma)}, \tag{12}$$

provided $r+\mu-\sigma>0$, an assumption that we are making from now on. Substituting into the second equation (10) we find

$$U = \frac{\sigma A-\mu}{w},$$

which is nonnegative if $A \geq \mu\sigma^{-1}$. From the first equation in (8) we then obtain

$$A = \frac{1}{(g-f)}\left[m\sqrt{\frac{r}{K\,(r+\mu-\sigma)}} - f\right].$$

Recalling the assumption $g<f$, A will be nonnegative if and only if

$$K > \left(\frac{m}{f}\right)^2 \frac{r}{r+\mu-\sigma}.$$

We finally have the explicit expression of U as follows,

$$U = \frac{\sigma}{w}\frac{1}{g-f}\left[m\sqrt{\frac{r}{K\,(r+\mu-\sigma)}} - f\right] - \frac{\mu}{w},$$

which is nonnegative when $A>\mu\sigma^{-1}$, i.e. for

$$m\sqrt{\frac{r}{K\,(r+\mu-\sigma)}} < \frac{\mu}{\sigma}\,(g-f)+f. \tag{13}$$

The right hand side is positive if $(\sigma-\mu)\,f+\mu g>0$ and from this the above restriction can be rewritten as

$$K > \left[\frac{m\sigma}{(\sigma-\mu)\,f+\mu g}\right]^2 \frac{r}{r+\mu-\sigma}.$$

In summary we found the equilibrium $E_4=(A_4,T_4,U_4)$ where, explicitly,

$$A_4 = \frac{1}{(g-f)}\left[m\sqrt{\frac{r}{K\,(r+\mu-\sigma)}} - f\right], \quad T_4 = \sqrt{\frac{K}{r}\,(r+\mu-\sigma)},$$
$$U_4 = \frac{\sigma}{w\,(g-f)}\left[m\sqrt{\frac{r}{K\,(r+\mu-\sigma)}} - f\right] - \frac{\mu}{w},$$

 ISBN:978-84-615-5392-1

with feasibility conditions $r+\mu-\sigma>0$, $q=w$ and

$$K>\left[\frac{m\sigma}{(\sigma-\mu)\,f+\mu g}\right]^2\frac{r}{r+\mu-\sigma},\quad (\sigma-\mu)\,f+\mu g>0,\quad K>\left(\frac{m}{f}\right)^2\frac{r}{r+\mu-\sigma}.$$

These conditions can be simplified, observing that from $g-f<0$ it follows

$$\frac{m\sigma}{f\sigma}<\frac{m\sigma}{(g-f)\,\mu+f\sigma}$$

so that if $(\sigma-\mu)\,f+\mu g>0$,

$$K>\left(\frac{m}{f}\right)^2\frac{r}{r+\mu-\sigma}$$

is implied by the condition

$$K>\left(\frac{m\sigma}{[(\sigma-\mu)\,f+\mu g]}\right)^2\frac{r}{r+\mu-\sigma}.$$

Thus feasibility conditions for E_4 become just the following ones

$$K>\left(\frac{m\sigma}{[(\sigma-\mu)\,f+\mu g]}\right)^2\frac{r}{r+\mu-\sigma},\quad (\sigma-\mu)\,f+\mu g>0,\quad r+\mu-\sigma>0,\quad q=w. \tag{14}$$

We now address the case $q-w<0$. In this situation from (11) we find

$$U=\frac{1}{(q-w)}\left[(r+\mu-\sigma)-\frac{r}{K}\frac{m^2}{[(g-f)\,A+f]^2}\right] \tag{15}$$

and substituting the values of T and U into the second equation of (10), we obtain

$$-\mu-\frac{w\,(r+\mu-\sigma)}{(q-w)}+\frac{rw}{(q-w)\,K}\frac{m^2}{[(g-f)\,A+f]^2}+\sigma A=0. \tag{16}$$

From this, with some algebra, we are led to the following cubic equation for A:

$$P\,(A)\equiv\sum_{k=0}^{3}b_kA^k=0, \tag{17}$$

where $b_3=1$,

$$\begin{aligned}
b_2&=\sigma\,(q-w)\,K\,(g-f)^2\,,\\
b_1&=2f\,(g-f)\,\sigma\,(q-w)\,K\,(w\sigma-rw-q\mu)\,K\,(g-f)^2\,,\\
b_1&=f^2\sigma\,(q-w)\,K+2f\,(g-f)\,(w\sigma-rw-q\mu)\,K,\\
b_0&=f^2\,(w\sigma-rw-q\mu)\,K+m^2rw.
\end{aligned}$$

 ISBN:978-84-615-5392-1

It has always a real root, and we seek now sufficient conditions for a nonnegative real root. Since $q \leqslant w$, it follows that

$$\lim_{A\to\infty} P(A) = -\infty,$$

so that if the constant term is positive, at least one positive real root must exist. This occurs if

$$f^2 (w\sigma - rw - q\mu) K > -m^2 rw, \tag{18}$$

which is trivial in case

$$w\sigma - rw - q\mu \geqslant 0, \tag{19}$$

otherwise it leads to

$$w\sigma - rw - q\mu < 0, \quad K < \left(\frac{m}{f}\right)^2 \frac{rw}{rw + q\mu - w\sigma}. \tag{20}$$

In summary the equilibrium $E_5 = (A_5, T_5, U_5)$ arises with first component given by the positive root of (17) and the remaining ones by (12) and (15), which need to be nonnegative, and further feasibility conditions given by (19) or (20).

5 Stability

The elements of the Jacobian matrix $J = (J_{ik})$, $i, k = 1, 2, 3$ are

$$J_{11} = 2(\sigma - r - \mu) A + 2\frac{r}{K} AT^2 + 2(q - w) AU + (r + \mu - \sigma) - -\frac{r}{K} T^2 + (w - q) U$$

$$J_{12} = 2\frac{r}{K} AT (A - 1) \quad J_{13} = (q - w) A (A - 1) \quad J_{21} = -\frac{r}{2K} T^3 + \frac{r + \mu}{2} T + \frac{w - q}{2} UT$$

$$J_{22} = -\frac{3r}{2K} AT^2 - \frac{\mu}{2} + \frac{r + \mu}{2} A - \frac{w}{2} U + \frac{w - q}{2} AU \quad J_{23} = -\frac{w}{2} T + \frac{w - q}{2} AT$$

$$J_{31} = \frac{q - w}{2} U^2 + (g - f) UT - \frac{r + \mu}{2} U + \frac{r}{2K} UT^2 \quad J_{32} = (g - f) AU + fU + \frac{r}{K} AUT$$

$$J_{33} = wU + (q - w) AU + \frac{\mu}{2} - m + (g - f) AT - \frac{r + \mu}{2} A + fT + \frac{r}{2K} AT^2$$

Observe that since $A = S(S + I)^{-1} \leqslant 1$ and $q < w$ two of the above terms have a fixed sign:

$$J_{12} \leqslant 0, \quad J_{13} \geqslant 0.$$

The Jacobian's eigenvalues at E_1 are $\lambda_1 = \sigma - \mu$, $\lambda_2 = -r$, $\lambda_3 = -m + g\sqrt{K}$, from which the stability conditions follow

$$\frac{\mu}{\sigma} > 1, \quad K < \left(\frac{m}{g}\right)^2. \tag{21}$$

 ISBN:978-84-615-5392-1

The Jacobian at E_2 gives one eigenvalue as

$$\lambda_1 = \sqrt{\frac{K}{r}\left(r+\mu-\sigma\right)}\left[f+\left(g-f\right)\frac{\mu}{\sigma}\right]-m.$$

from which the stability condition follows

$$K < \left[\frac{m\sigma}{\left(\sigma-\mu\right)f+g\mu}\right]^2 \frac{r}{r+\mu-\sigma} \tag{22}$$

having used the fact that $\left(\sigma-\mu\right)f+g\mu>0$ and the first condition (7). The other two eigenvalues are the roots of

$$\lambda^2+\frac{\mu}{\sigma}\left(r+\mu-\sigma\right)\lambda+\mu\left(1-\frac{\mu}{\sigma}\right)\left(r+\mu-\sigma\right)=0. \tag{23}$$

In view of the feasibility conditions (7), the Routh-Hurwitz stability conditions for (23) hold. Stability of E_2 is therefore regulated only by (22).

At E_3 again one eigenvalue is immediate,

$$\lambda_1 = \left(\sigma-\mu\right)+\frac{rw}{g^2qK}\left(m^2-g^2K\right).$$

It is negative if and only if $g^2K\left[q\left(\sigma-\mu\right)-rw\right]<-m^2rw$. But this cannot happen if $q\left(\sigma-\mu\right)-rw\geqslant 0$. Conversely, we are lead to the stability conditions

$$K>\left(\frac{m}{g}\right)^2\frac{rw}{rw+q\mu-q\sigma},\quad rw+q\sigma>q\mu. \tag{24}$$

The other eigenvalues come from the quadratic

$$\lambda^2+\frac{r}{2g^2K}\left(3m^2-g^2K\right)\lambda+\frac{mr}{2g^4K}\left(g^2K-m^2\right)\left(2mr+g^2K\right)=0. \tag{25}$$

From the (strict) feasibility conditions (9) for E_3, the constant term is always positive. Imposing that also the coefficient of the linear term is positive, we obtain the second stability condition,

$$K<3\left(\frac{m}{g}\right)^2. \tag{26}$$

In summary, E_3 is feasible and stable for

$$0<\max\left\{1,\frac{rw}{rw+q\mu-q\sigma}\right\}<K\left(\frac{g}{m}\right)^2<3. \tag{27}$$

 ISBN:978-84-615-5392-1

For the equilibrium E_4 some of the Jacobian entries, in view of the feasibility conditions (14) have fixed signs, as follows

$$J_{4_{12}} = \frac{2r}{K}\sqrt{\frac{K}{r}(r+\mu-\sigma)}\frac{1}{(g-f)}\left[m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right] \cdot\left[\frac{1}{(g-f)}\left(m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right)-1\right]<0,$$

$$J_{4_{21}} = \frac{\sigma}{2}\sqrt{\frac{K}{r}(r+\mu-\sigma)}>0, \quad J_{4_{22}} = -\frac{(r+\mu-\sigma)}{(g-f)}\left[m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right]<0,$$

$$J_{4_{23}} = -\frac{w}{2}\sqrt{\frac{K}{r}(r+\mu-\sigma)}<0,$$

$$J_{4_{32}} = \left\{\frac{\sigma}{w(g-f)}\left[m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right]-\frac{\mu}{w}\right\}\cdot\left[\left(m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right)+f + \frac{1}{(g-f)}\left(m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right)\sqrt{\frac{r}{K}(r+\mu-\sigma)}\right]>0,$$

while the remaining two must agree, since the same factor appears in the two elements, although the sign is not decided:

$$J_{4_{31}} = \frac{1}{w}\left[\sigma\left(m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right)-\mu(g-f)\right]\cdot \left[\sqrt{\frac{K}{r}(r+\mu-\sigma)}-\frac{\sigma}{2(g-f)}\right],$$

$$J_{4_{33}} = \frac{1}{2(g-f)}\left[\sigma\left(m\sqrt{\frac{r}{K(r+\mu-\sigma)}}-f\right)-\mu(g-f)\right].$$

We now study the signs of $J_{4_{31}}$ and $J_{4_{33}}$. Considering $J_{4_{31}}$ and using the feasibility condition (14) we find that for its positivity we must have

$$\frac{r}{K(r+\mu-\sigma)} > \left[\frac{\mu(g-f)+f\sigma}{m\sigma}\right]^2.$$

But this contradicts the feasibility condition (14), so that it must be negative. In summary we then have

$$J_{4_{31}}<0, \quad J_{4_{33}}<0.$$

Thus the resulting structure of the Jacobian matrix is

$$J_4 = \begin{pmatrix} 0 & - & 0 \\ + & - & - \\ - & + & - \end{pmatrix} \equiv \begin{pmatrix} 0 & Z & 0 \\ B & C & D \\ E & F & G \end{pmatrix}.$$

 ISBN:978-84-615-5392-1

The characteristic equation is now a cubic,

$$\sum_{k=0}^{3} a_k \lambda^k \equiv \lambda^3 - (C+G)\,\lambda^2 - (ZB + FD - CG)\,\lambda - Z\,(ED - BG) = 0. \tag{28}$$

Using the signs of Z, B, C, D, E, F and G all the coefficients a_k, $k = 0, \ldots, 3$ are positive. We can thus use the Liénard-Chipart criterion, a particular case of the Routh-Hurwitz criterion, thereby determining the sign of the eigenvalues imposing that the following determinant be positive:

$$\begin{aligned} D_2 = \begin{vmatrix} a_2 & a_0 \\ a_3 & a_1 \end{vmatrix} = \begin{vmatrix} -(C+G) & -Z\,(ED - BG) \\ 1 & -(ZB + FD - CG) \end{vmatrix} = \\ = (C+G)\,(ZB + FD - CG) + Z\,(ED - BG) > 0. \end{aligned} \tag{29}$$

We can conclude for this case that E_4 is stable if (29) holds.

Stability of E_5 is investigated numerically.

6 Bifurcations

Note that transcritical bifurcations further arise between E_1, E_2 and E_3.

We then try to establish if there are special parameter combinations for which Hopf bifurcations arise. For this purpose, we need purely imaginary eigenvalues. This is easy to assess for a quadratic characteristic equation, $\lambda^2 + b\lambda + c = 0$ since we need the linear term to vanish, $b = 0$, and the constant term to be negative, $c < 0$. For a generic cubic of the form

$$a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0 \tag{30}$$

instead, we need the following condition

$$a_1 a_2 - a_0 = 0.$$

Clearly at E_1 no bifurcation arises, since the eigenvalues are all real. At E_2 we need

$$b = \frac{\mu}{\sigma}\,(r + \mu - \sigma) = 0, \quad c = \mu\left(1 - \frac{\mu}{\sigma}\right)(r + \mu - \sigma) > 0 \tag{31}$$

but these conditions contradict each other. We conclude that at E_2 no Hopf bifurcations can carise.

At E_3 the characteristic equation factors, and the quadratic (25) from feasibility (9) has a positive constant term. Imposing that the linear term vanishes, we find the value

$$K^{\dagger} \equiv 3\left(\frac{m}{g}\right)^2 \tag{32}$$

for which a limit cycle appears. We will show this situation and investigate the remaining ones for E_4 and E_5 with simulations.

7 Equilibria interpretation

At equilibrium E_1, we have $U_1 = 0$ so that $P_1 = 0$ and the predators vanish. Further, $A_1 = 1$ implying that $I_1 = 0$. Thus only healthy prey survive, at the environment's carrying capacity, due to the model assumption of logistic growth, $T_1 = \sqrt{K}$ indeed implies in this case $S_1 = K$.

At E_2 the request that $A < 1$ tells us that neither healthy nor infected prey disappear from the system, while, as in the previous case, all the predators die since $U_2 = 0$. Therefore the disease remains endemic among the prey, while predators do not survive. Note once again that the the point E_2 becomes equilibrium E_1 if we assume that the disease transmission rate equals the disease mortality rate. In such case thus the disease can be eradicated.

At E_3 we have again that $A_3 = 1$, so that $I = 0$ and in this case the disease gets eradicated from the ecosystem, while the predators and healthy prey survive together. This is the only equilibrium for which we have proved analytically the existence of bifurcations, for the particular value of the prey carrying capacity $K^\dagger = 3m^2g^{-2}$.

At E_4 and E_5 we have coexistence, with the point E_3 being a particular case of the latter equilibria, when $A = 1$. Further E_4 and E_5 differ because in the first case $q = w$, i.e. the infected and healthy prey are hunted at the same rate by predators, and therefore it can be regarded as a special case of E_5. As for the latter, note that for the particular situation in which $f^2K\,(rw + q\mu - w\sigma) = m^2rw$ we find A_5, as the cubic (17) goes through the origin. This implies that the healthy prey are wiped out. Therefore in this situation the ecosystem thrives, with predators and only infected prey.

8 Simulations

The equilibrium E_1 represents the situation where the only population which survives in the habitat is represented by the healthy prey. The fact that infected individuals are extinguished is consistent with the stability conditions of E_1. In fact, the latter require that the disease incidence be lower than the disease-related mortality rate. Thus infected individuals die faster than they are recruited and ultimately there are not enough infectious individuals to propagate the disease. Its stable behavior is shown in Figure 1 for the parameter values $\sigma = 0.2$, $r = 0.5$, $K = 5$, $\mu = 0.4$, $q = 0.2$, $w = 0.5$, $m = 0.8$, $g = 0.1$, $f = 0.3$.

At the equilibrium E_2 predators get extinguished, but the disease remains endemic. In this situation the opposite condition of equilibrium E_1 must be verified, namely the disease-related mortality rate is lower than the disease incidence. This suggests that it is reasonable to expect that the population of infected prey survives. Figure 2 contains a simulation leading to this equilibrium for the parameter values $\sigma = 0.5$, $r = 0.5$, $K = 5$,

 ISBN:978-84-615-5392-1

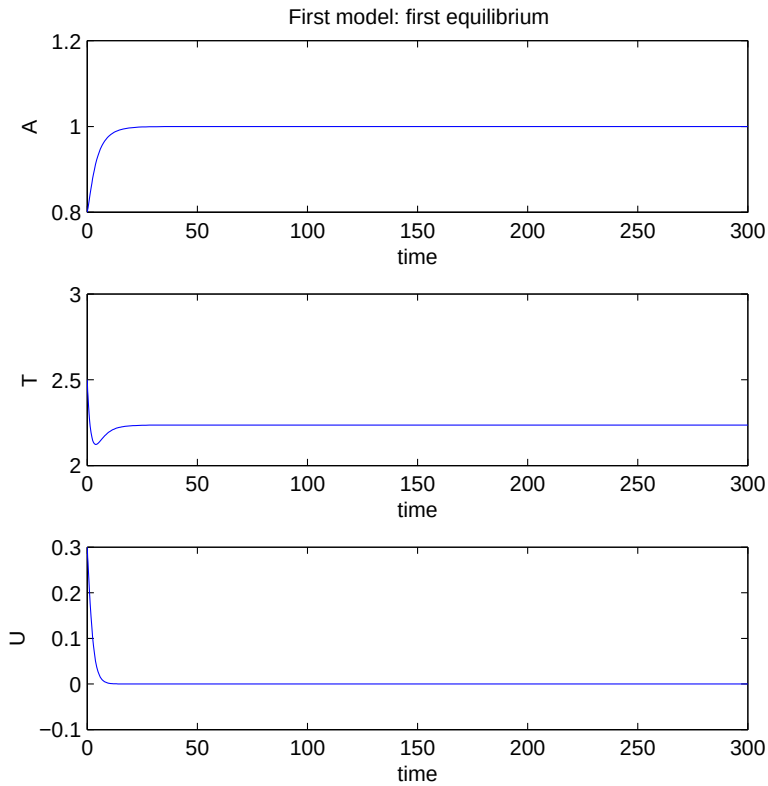


Figure 1: The stable equilibrium E_1 is achieved for the parameter values $\sigma = 0.2$, $r = 0.5$, $K = 5$, $\mu = 0.4$, $q = 0.2$, $w = 0.5$, $m = 0.8$, $g = 0.1$, $f = 0.3$.

$\mu = 0.4$, $q = 0.2$, $w = 0.5$, $m = 0.8$, $g = 0.1$, $f = 0.3$..

The third equilibrium E_3 shows coexistence of predators and healthy prey, with the disease eradicated. Figure 3 shows it for the parameter values $\sigma = 0.5$, $r = 0.5$, $K = 5$, $\mu = .4$, $q = 0.2$, $w = 0.5$, $m = 0.2$, $g = 0.1$, $f = 0.3$..

The equilibrium E_4 represents the possibility that all the populations in the system survive, i.e. E_4 represents the coexistence equilibrium. The simulation of Figure 4 shows it for the following parameter values: $\sigma = 0.4$, $r = 0.5$, $\mu = 0.2$, $q = 0.5$, $w = 0.5$, $m = 0.3$, $f = 0.2$, $g = 0.1$, $K = 10$..

In the study of the stability, focusing on the bifurcations, we discovered that, around E_3, limit cycles should arise when K crosses the threshold value $K^{\dagger} = 3m^2g^{-2}$. In Figure 5 we present a simulation of the two-dimensional limit cycle for the parameter values $\sigma = 0.5$, $r = 0.5, \mu = 0.4$, $q = 0.2$, $w = 0.5$, $m = 0.2$, $g = 0.1$, $f = 0.3$. Oscillations appear

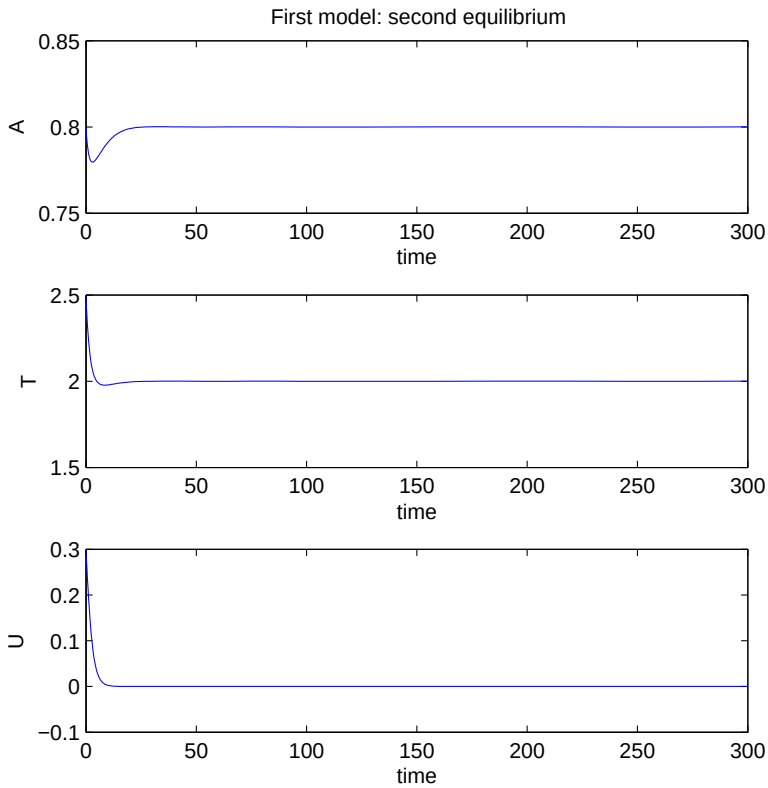


Figure 2: Equilibrium E_2 is obtained for the parameters: $\sigma = 0.5$, $r = 0.5$, $K = 5$, $\mu = 0.4$, $q = 0.2$, $w = 0.5$, $m = 0.8$, $g = 0.1$, $f = 0.3$..

only for the second and the third variables, while for first one remains at the fixed level $A = 1$, to mean that the system is disease-free. Predators survive together with the healthy individuals but with persistent oscillations of the two populations. In Figure 6 a three-dimensional phase-space portrait of the limit cycle is given.

For the equilibrium E_5 our extensive simulations seem to indicate its instability.

9 Conclusions

In this paper we studied an ecoepidemic model in which two popolations interact: the prey and the predators. We assumed that among the prey a disease develops, which spreads by contact. Further, the disease is unrecoverable, i.e. infected prey cannot heal from it.

 ISBN:978-84-615-5392-1

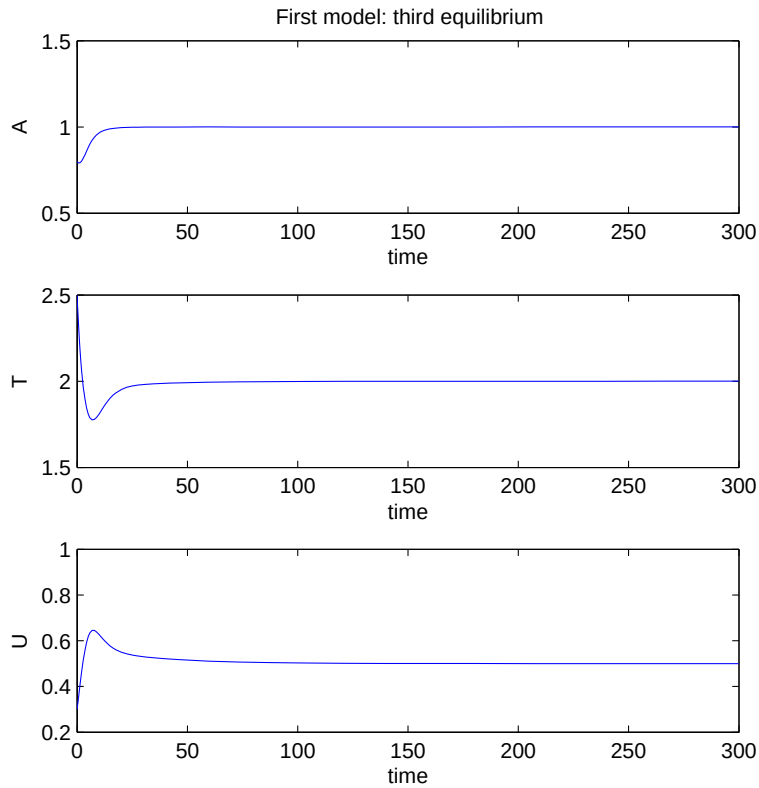


Figure 3: The equilibrium E_3 is here shown for the parameter values $\sigma = 0.5$, $r = 0.5$, $K = 5$, $\mu = .4$, $q = 0.2$, $w = 0.5$, $m = 0.2$, $g = 0.1$, $f = 0.3$..

Predation affects for healthy and infected prey. The prey group together in a herd and exert some defensive strategy, for which mainly the individuals on the boundary of the herd suffer from the attacks of the hunting population.

The original system formulation leads to possible singularities in the Jacobian, when the prey population vanishes. Therefore we have performed several changes of dependent variables to obtain a singularity-free reformulation. The newly obtained variables represent respectively the ratio of healthy prey over the total amount of prey, the number of predators per prey staying at the edge of the herd area and finally the number of prey occupying the edge of the herd.

We have discovered that there are four possibly stable equilibria, at which only the healthy prey thrive, or the disease remains endemic with only the prey population surviving, or healthy prey coexist, possibly with persistent oscillations, with the predators. The final

 ISBN:978-84-615-5392-1

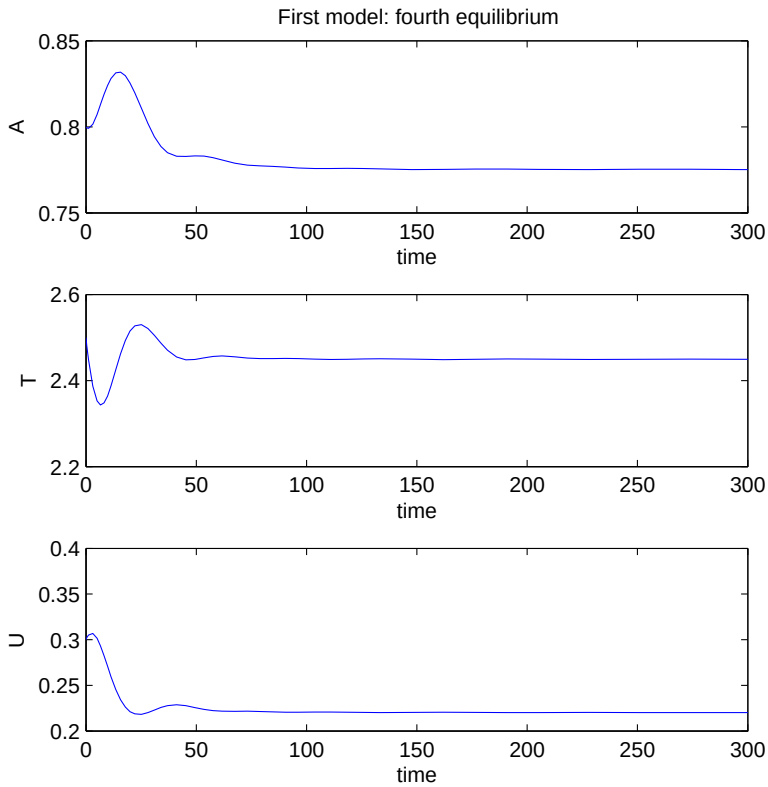


Figure 4: Equilibrium E_4 is obtained here for the parameters $\sigma = 0.4$, $r = 0.5$, $\mu = 0.2$, $q = 0.5$, $w = 0.5$, $m = 0.3$, $f = 0.2$, $g = 0.1$, $K = 10$..

coexistence equilibrium is also possible, with both populations and endemic disease.

The only alternative that in these circumstances is impossible, is the fact that predators can thrive just with infected prey. But this is a consequence of the model assumptions, in that infected prey are assumed to be too weak to sustain themselves.

References

[1] V. Ajraldi, M. Pittavino, E. Venturino, *Modelling herd behavior in population systems*, Nonlinear Analysis Real World Applications, **12** (2011) 2319-2338.

[2] V. Ajraldi, E. Venturino, *Mimicking spatial effects in predator-prey models with group defense*, Proceedings of the 2009 International Conference on Computational

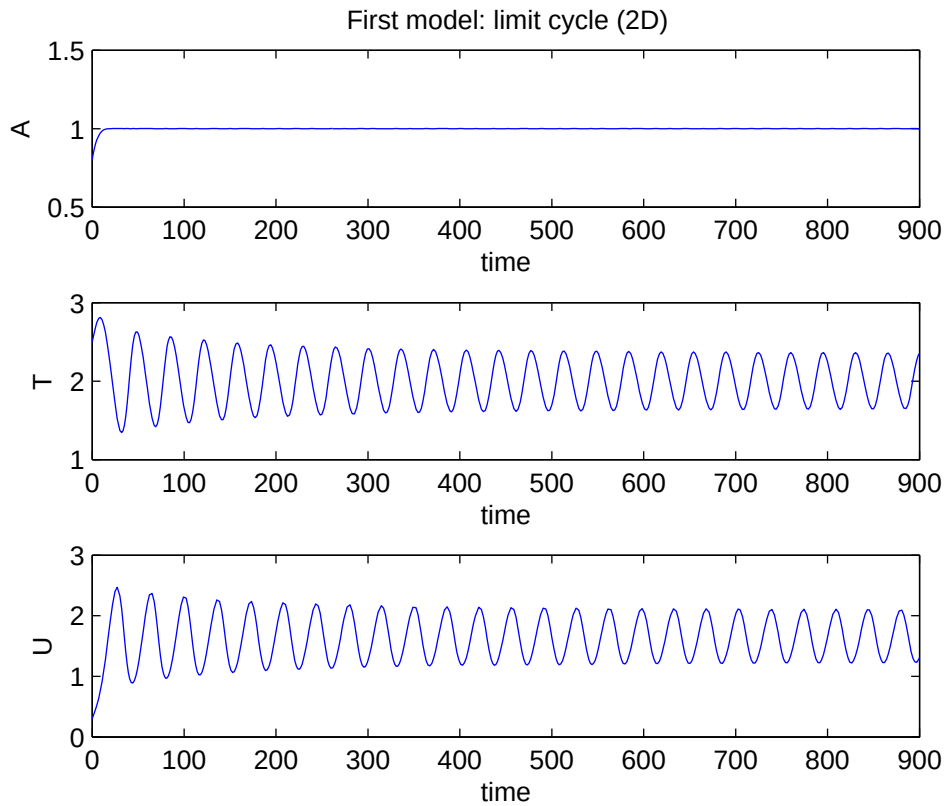


Figure 5: Two-dimensional limit cycle around E_4 obtained for the parameter values $\sigma = 0.5$, $r = 0.5, \mu = 0.4$, $q = 0.2$, $w = 0.5$, $m = 0.2$, $g = 0.1$, $f = 0.3$, for which $K = K^{\dagger} \equiv 3\left(\frac{m}{g}\right)^2$.

and Mathematical Methods in Science and Engineering, J. Vigo Aguiar, P. Alonso, S. Oharu, E. Venturino, B. Wade (Editors), Gijón, Asturias, Spain, June 30th - July 3rd (2009) 57-66.

[3] E. Beltrami, T. O. Carroll, *Modelling the role of viral disease in recurrent phytoplankton blooms*, J. Math. Biol. **32** (1994) 857-863.

[4] P. A. Braza, *Predatorprey dynamics with square root functional responses*, Nonlinear Analysis Real World Applications **13**(4) (2012) 1837-1843.

[5] C. Cosner, D. L. DeAngelis, J. S. Ault, D. B. Olson, *Effects of spatial grouping on the functional response of predators*, Theoretical Population Biology **56** (1999) 65-

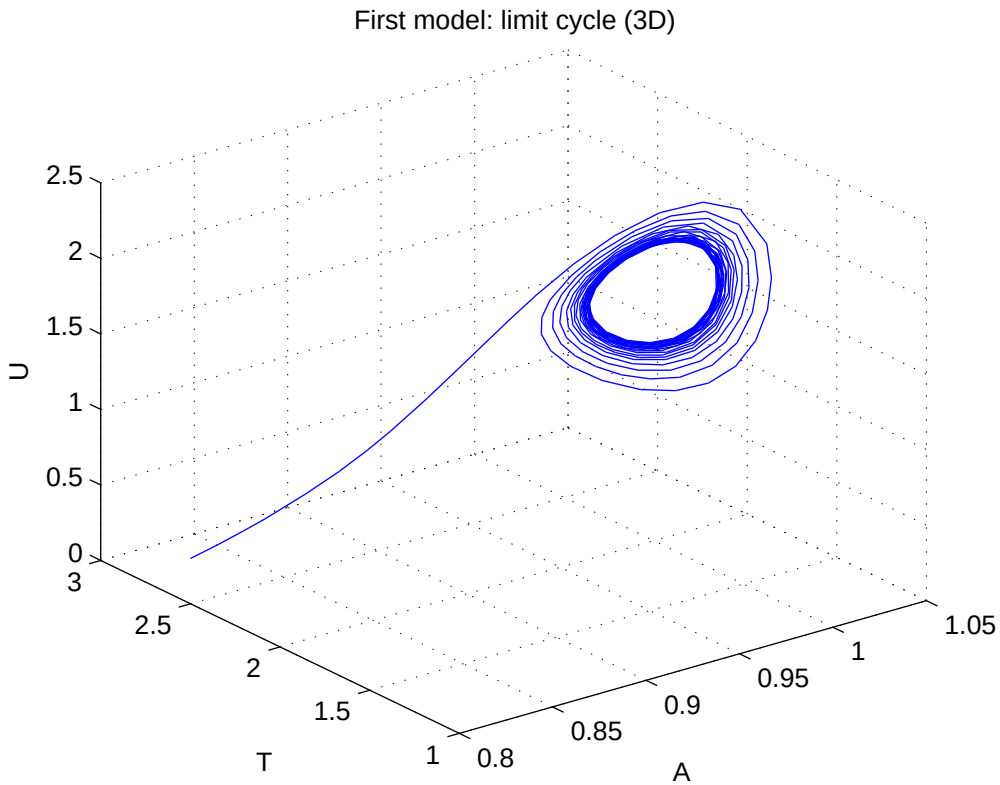


Figure 6: Phase space representation of the limit cycle of Figure 5 for the same parameter values.

75.

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

[7] K. P. Hadeler, H. I. Freedman, *Predator-prey populations with parasitic infection*, J. of Math. Biology **27** (1989) 609-631.

[8] H. Malchow, S. Petrovskii, E. Venturino, *Spatiotemporal patterns in Ecology and Epidemiology*, CRC, Boca Raton, 2008.

[9] E. Venturino, *Epidemics in predator-prey models: disease among the prey*, in O. Arino, D. Axelrod, M. Kimmel, M. Langlais: *Mathematical Population Dynamics:*

Analysis of Heterogeneity, Vol. one: Theory of Epidemics, Wuertz Publishing Ltd, Winnipeg, Canada, p. 381-393, 1995.

[10] E. Venturino, *Epidemics in predator-prey models: disease in the predators*, IMA Journal of Mathematics Applied in Medicine and Biology **19** (2002) 185-205.

[11] E. Venturino, *Ecoepidemiology 15 years later: a review*, Numerical Analysis and Applied Mathematics, T. Simos (Editor), Proceedings of ICNAAM 2007, AIP **936** (2007) 31-34.

[12] E. Venturino, *A minimal model for ecoepidemics with group defense*, J. of Biological Systems **19**(4) (2011) 763-785.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Applications of quantum thermal baths in vibrational spectroscopy

Florent Calvo[1]

[1] *CNRS, LASIM, University of Lyon, France*

emails: fcalvo@lasim.univ-lyon1.fr

Abstract

Quantum nuclear effects are important for weakly bound or light atoms and at low temperatures. They are manifested by residual energy stored as zero-point vibrations, tunneling, and possible exchange effects in bosonic systems. Unfortunately, the vibrational Schrodinger equation is difficult to solve except for very small systems. Quantum thermal baths (QTBs) have been proposed in the recent years as an alternative to the convenient, but still computationally expensive schemes based on path integrals. The QTB method relies on propagating a stochastic Langevin equation with a correlated (colored) noise designed to produced a power spectrum which satisfies the quantum fluctuation-dissipation theorem. As such, its numerical cost is close to that of a standard classical Langevin equation, making it a very promising technique for large-scale atomic and molecular systems.

In the present contribution, we discuss the application of the QTB approach for gas-phase systems and their equilibrium properties and vibrational spectroscopy. The implementation of the method is described, with an emphasis on specific computational aspects involved in the generation of the colored noise. The performance of the method is assessed by comparison with dedicated path-integral molecular dynamics simulations for simple ionic clusters, as well as polycyclic aromatic hydrocarbons.

Key words: Molecular dynamics, quantum nuclear effects, Langevin equation

  ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Application of Auto-Tuning Techniques to High-Level Linear Algebra Shared-Memory Subroutines

Jesús Cámara[1], Javier Cuenca[2], Domingo Giménez[1] and Antonio M. Vidal[3]

[1] *Department of Informatic and Systems, University of Murcia, 30071, Murcia, Spain*

[2] *Department of Engineering and Technology of Computers, University of Murcia, 30071, Murcia, Spain*

[3] *Department of Informatic Systems and Computation, Polytechnic University of Valencia, 46022, Valencia, Spain*

emails: jcm23547@um.es, jcuenca@um.es, domingo@um.es, avidal@dsic.upv.es

Abstract

The introduction of auto-tuning techniques in linear algebra shared-memory routines is analysed. Our goal is to study how auto-tuning techniques can be included in routines which call lower level routines, so using information from the installation of the low-level routine to take at run time some decisions to reduce the total execution time. The study is carried out with Cholesky factorization routines: the `potrf` LAPACK routine with internal calls to multithreaded MKL routines and a modified LAPACK version with a two-level implementation of its internal level-3 BLAS subroutines. For this modified LAPACK version, the experiments show that the number of threads to use at each parallelism level and the block size used internally by the `potrf` routine can be selected to automatically obtain satisfactory execution times.

Key words: autotuning, multithreading, linear algebra, OpenMP, MKL

1 Introduction

The appearance of multicore and cc-NUMA systems has led to the development of optimized software for these systems. Software optimization techniques are used in parallel routines to decide how to execute them to obtain the lowest execution time. Decisions are taken at run time as a result of the work performed at installation time, modelling the execution time of

the routines or by applying some empirical study of the behaviour of the routines. Moreover, these decisions may vary depending on the type of computational system used. Some of these decisions could be: select the appropriate number of threads to use at each level of parallelism, how to assign processes to processors or select the correct block size in the case of algorithms by blocks. In this work, the previous ideas of installing multithreaded basic linear algebra routines in large cc-NUMA systems [1] are combined and extended. In [1], auto-tuning is carried out by applying installation techniques to the BLAS-3 matrix multiplication routine (`dgemm`), which constitutes the basic subroutine for many computational routines. In this paper the same ideas are applied to the higher-level routine `potrf` of the LAPACK library: the Cholesky factorization of a symmetric positive definite matrix. The same methodology has been applied to other high-level routines (`dgesv`, `zsysv`, `dgetrf`, etc) with similar results. For that purpose, the matrix multiplication routine used in `potrf` is replaced by a parallel version (called `dgemm2L`) specially adapted for large multicore systems.

The rest of the paper is organized in the following way. Section 2 summarizes the auto-tuning methodology for linear algebra routines in large NUMA systems. Section 3 analyzes the behaviour of the Cholesky factorization when the auto-tuning methodology is applied to the `dgemm` routine, and the results are compared with those of the reference version of the LAPACK `potrf` routine when the number of threads and the block size are automatically selected. Finally, the conclusions and future research lines are shown in Section 4.

2 The auto-tuning methodology

The implementations of shared-memory linear algebra routines are normally not very scalable. This produces a degradation of the performance in large cc-NUMA systems. To improve the scalability of the routines, the auto-tuning methodology explained in [1] for the `dgemm` routine can be extended to higher-level routines. The goal of this methodology is to select the most appropriate number of threads to use at each level of parallelism.

This paper shows how this methodology can be applied to high-level linear algebra routines, using the Cholesky factorization as proof of concept. The Cholesky factorization is normally used to solve a linear system of the type $AX = B$, with A a symmetric positive definite matrix. To compute it, we can use the multithreaded MKL version of the `potrf` routine or the reference version included in the LAPACK library. Algorithm 1 shows the scheme used in the LAPACK `potrf` routine to compute the Cholesky factorization.

The installation of the routine in the system is made by executing the routine for each matrix size specified in the *installation set* [1] by varying the number of OpenMP and MKL threads at each level of parallelism, from one to the number of available cores in the system, and using a combination of threads not exceeding the maximum number of cores. Once the routine has been installed, the number of threads with which the lowest time is obtained

[1] An *installation set* consists of some problem sizes used to install the routine in the system.

```
* Compute the Cholesky factorization A = L*L**T.
*
      DO 20 J = 1, N, NB
*
*           Update and factorize the current diagonal block and test
*           for non-positive-definiteness.
*
            JB = MIN(NB, N-J+1)
            CALL dsyrk(...)
            CALL dpotf2(...)
            IF( J+JB.LE.N ) THEN
*
*              Compute the current block column.
*
               CALL dgemm(...)
               CALL dtrsm(...)
            END IF
   20   CONTINUE
}
```

Algorithm 1: Scheme of the LAPACK Cholesky (`potrf`) routine.

for each problem size is stored, and, at execution time, for a particular problem size, the number of threads to be used to solve the problem is selected by using the information stored during the installation phase.

In the Cholesky factorization, the auto-tuning methodology can be applied to its internal `dgemm` routine, which is used to perform all the matrix multiplications involved in the computation of the elements of the blocks (whose size is automatically determined by the LAPACK `ILAENV` function) of the lower triangular part of matrix A. Therefore, it is necessary to work directly with the reference `potrf` routine. The `dgemm` routine is then replaced by a parallel implementation that uses two levels of parallelism (`dgemm2L`) and the auto-tuning process is performed in order to select the most appropriate number of threads at each level of parallelism. The other routines internally used in the Cholesky factorization are called using their corresponding multithreaded MKL implementation.

Experiments have been carried out in a platform called Saturno, a shared-memory system with four hexa-cores (24 cores) and the *installation set* used is: {256, 768, 1280, 1792, 2304, 2816, 3328, 3840, 4352}. Different problem sizes are used for validation (*validation set*). At running time, the decisions for the problem sizes in the *validation set* are taken by applying an interpolation process to the information stored during the installation phase. Table 1 shows the execution times (in seconds) obtained with the auto-tuning methodology and the lowest execution times obtained experimentally by a perfect oracle. The number of OpenMP and MKL threads used at each level of parallelism is also shown. The number of threads most frequently used is 24 (the total number of cores of the system), but with different combinations (3-8, 4-6, 6-4). The times obtained with the auto-tuning methodology are normally close to the optimum, and the total number of threads used is also similar.

In larger systems (as those considered in [1]), differences in execution times would be higher.

N	Optimum OMP (threads)	Optimum MKL (threads)	Optimum Time (sec)	Auto-Tuning OMP (threads)	Auto-Tuning MKL (threads)	Auto-Tuning Time (sec)
512	1	16	0.001219	1	14	0.001447
1024	4	6	0.003989	3	8	0.004230
1536	4	6	0.007623	6	4	0.008004
2048	2	12	0.013438	4	6	0.014118
2560	7	3	0.032550	6	4	0.084170
3072	7	3	0.050523	6	4	0.083533
3584	3	8	0.078012	4	6	0.078633
4096	3	8	0.124691	4	6	0.127465

Table 1: Execution times (in seconds) obtained with the application of the auto-tuning methodology (Auto-Tuning) to the `dgemm` routine of `potrf` and lowest experimental execution time (Optimum), and number of OpenMP and MKL threads used in these executions.

3 Application to Cholesky factorization routines

In order to analyse the improvement achieved with this methodology when applied to linear algebra routines which call lower level routines, a comparative study of the execution time obtained by different implementations of the Cholesky `potrf` routine has been carried out. Table 2 shows the results obtained for the reference LAPACK routine, a `potrf` LAPACK routine which internally calls multithreaded MKL routines (`dsyrk`, `dpotf2` and `dtrsm`) and the modified LAPACK routine where `dgemm` is replaced by the auto-tuned `dgemm2L` routine. The last column shows the speed-up achieved by the modified LAPACK routine with auto-tuning respect to the lowest execution time obtained with the LAPACK routine with multithreaded MKL. The results of applying the auto-tuning methodology are satisfactory, but for some problem sizes a loss of performance occurs due to the interpolation applied to select the number of threads.

3.1 Selecting the block size

The Cholesky factorization of LAPACK (Algorithm 1) is computed by blocks. The size and form of these blocks vary depending on the value internally selected by the LAPACK `ILAENV` function. In this function, that value is selected using information of the problem size but not based in the number of threads used. Therefore, we can reduce the execution time even more by selecting the optimum block size for each value of the *installation set*. To

N	LAPACK	LAPACK+MKL	LAPACK+AutoTuning	Speed-Up
512	0.043130	0.003948 (9)	0.003793 (1,14)	1.04
1024	0.332920	0.012877 (12)	0.011624 (3,8)	1.10
1536	1.104757	0.024598 (24)	0.024420 (6,4)	1.00
2048	2.614030	0.075525 (24)	0.076562 (4,6)	0.99
2560	5.075289	0.109087 (24)	0.165639 (6,4)	0.66
3072	8.787374	0.202955 (21)	0.237618 (6,4)	0.85
3584	13.934977	0.279215 (21)	0.323004 (4,6)	0.86
4096	20.894776	0.390708 (21)	0.383885 (4,6)	1.02

Table 2: Execution times (in seconds) obtained with different versions of the `potrf` routine: the reference LAPACK routine (LAPACK), the LAPACK routine with multithreaded MKL kernels (LAPACK+MKL) and the LAPACK routine with the auto-tuning methodology (LAPACK+Auto-Tuning). The number of threads with which the lowest times are obtained is shown. The last column shows the speed-up achieved by the auto-tuning version with respect to the LAPACK+MKL. In brackets, the number of threads with which the execution times are obtained.

apply this idea to multithreaded routines, two parameters must be selected: the number of threads and the block size. The number of threads has been selected for the `dgemm` routine by applying the auto-tuning methodology. Now, for the selection of the block size it is necessary to work directly with the LAPACK `potrf` routine, so that the block size selected by the `ILAENV` function can be modified in order to select the best block size.

Table 3 compares, for different matrix sizes, the execution time obtained for the `potrf` routine when the block size is internally selected by `ILAENV` and the execution time when the block size is selected with the auto-tuning technique. All the experiments have been done in Saturno using the same *installation set*: {256, 768, 1280, 1792, 2304, 2816, 3328, 3840, 4352}, with block sizes power of 2 from 32 to 512 and a number of cores from 1 to 24. When the routine `potrf` uses the `ILAENV` function to select the block size, the same value is used for several matrix sizes regardless of the number of threads. When the number of threads and the block size are selected with the auto-tuning methodology, lower execution times are obtained and the speed-up achieved is higher than that obtained in table 2 by selecting only the number of OpenMP and MKL threads. For small matrix sizes the use of larger blocks is preferable, but for larger sizes a lower value than that selected by the `ILAENV` function (which does not consider the number of threads, only the matrix size) is more appropriate. The improvement achieved by selecting the appropriate block size is between 6% and 30% for most matrix sizes. Therefore, if we consider the block size parameter in the auto-tuning methodology, better results are obtained for the `potrf` routine. Similar results are obtained for other routines, and the advantage of the auto-tuning is more apparent in larger systems.

 ISBN:978-84-615-5392-1

	LAPACK-MKL with ILAENV		Auto-Tuning		
N	Block-Size	Time	Block-Size	Time	Speed-Up
512	32	0.003948 (9)	128	0,003440 (1,14)	1.15
1024	96	0.012877 (12)	128	0.011482 (3,8)	1.12
1536	192	0.024598 (24)	64	0.026508 (6,4)	0.93
2048	384	0.075525 (24)	128	0.062069 (4,6)	1.22
2560	384	0.109087 (24)	64	0.087751 (6,4)	1.24
3072	512	0.202955 (21)	64	0.145695 (6,4)	1.39
3584	512	0.279215 (21)	256	0.252449 (4,6)	1.11
4096	512	0.390708 (21)	256	0.364508 (4,6)	1.07

Table 3: Execution times (in seconds) obtained for the `potrf` LAPACK routine with a block size selected by the `ILAENV` function (LAPACK-MKL with ILAENV) and the LAPACK routine with auto-selection of the block size (Auto-Tuning). The last column shows the speed-up achieved with the auto-tuning methodology with respect to the use of the `ILAENV` function. In brackets, the number of OpenMP and MKL threads with which the lowest times are obtained.

4 Conclusions and future work

The experimental study carried out in this work has shown that the use of multithreaded routines in high-level routines together with an auto-tuning methodology capable of selecting the optimum block size and the appropriate number of OpenMP and MKL threads for each level of parallelism is a good technique to reduce the execution time. Therefore, if we apply this methodology to larger multicore systems, better results would be obtained. The main conclusions are:

- An appropriate selection of the number of threads to use at each level of parallelism substantially reduces the execution time, mainly when large matrices are used and the number of threads increases. For the system considered in this work, the best results are obtained when the maximum number of cores are used, but in larger systems this number may vary, as is shown in [1].
- The block size is an important parameter to take into account in routines by blocks in order to reduce the execution time. Therefore, an appropriate selection of its value together with an appropriate number of threads will allow us to reduce the total execution time even more. It is shown in the experiments, where a reduction of the execution time of up to 30% is achieved.

In this paper, the auto-tuning methodology has been applied to the Cholesky factorization, but it is been applied to other high-level routines (`dsysv`, `dgesv` and `dgetrf`) with

similar results. Preliminary results shows that the application of the auto-tuning methodology in larger systems produces more important reductions in the execution time, due to the higher scalability of two level routines and to the combined selection of the block size and the number of threads. We are investigating the application of the methodology to routines of the Parallel Linear Algebra for Scalable Multi-core Architectures (PLASMA, [3]), where the set of parameters to be tuned is different to those in LAPACK routines.

Acknowledgements

Partially supported by Fundación Séneca, Consejería de Educación de la Región de Murcia, 08763/PI/08, and the High-Performance Computing Network on Parallel Heterogeneus Architectures (CAPAP-H).

References

[1] Jesús Cámara, Javier Cuenca, Domingo Giménez and Antonio M. Vidal, *Empirical autotuning of two-level parallel linear algebra routines on large cc-NUMA systems*, ISPA, Madrid, 2012.

[2] Jesús Cámara, Javier Cuenca, Luis-Pedro García and Domingo Giménez, *Auto-tuned nested parallelism: a way to reduce the execution time of scientific software in NUMA systems*, PMAA, London, 2012.

[3] The PLASMA project: `http://icl.cs.utk.edu/plasma/`

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Observable variables and identifiability for chemical systems

B.Cantó[1], S.C. Cardona[2], C.Coll[1], J. Navarro-Laboulais[2] and E. Sánchez[1]

[1] *Institut de Matemàtica Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, España*

[2] *Departament d'Enginyeria Química i Nuclear, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, España*

emails: bcanto@mat.upv.es, scardona@iqn.upv.es, mccoll@mat.upv.es, jnavarla@iqn.upv.es, esanchezj@mat.upv.es

Abstract

In Chemistry, the dynamics of the composition of chemical species in reacting systems can be characterized by a set of autonomous differential equations derived from mass conservation principles and some elementary hypothesis related to chemical reactivity. These sets of ordinary differential equations (ODEs) are basically non-linear, its complexity grows as much increases the number of substances present in the reacting media and can be characterized by a set of phenomenological constants (kinetic rate constants) which contains all the relevant information about the physical system. The determination of these kinetic constants is critical for the design or control of chemical systems from a technological point of view but the non-linear nature of the ODEs implies that there are hidden correlations between the parameters which maybe can be revealed with a structural identifiability analysis. The chemical irreversible reactions can be expressed as a particular class of the more general chemical reversible reactions. Although the former are more common in chemical systems, the reversible ones have the advantage that can be approached, under some experimental circumstances, to linear systems. Then in this work we propose to analyze a reversible chemical reacting network, assuming that initially it remains stationary in an equilibrium state. Then, we will imagine an experiment where this system is perturbed and that it will return to its same initial state. Let us consider the chemical reversible reacting system given in figure 1.

In this figure the direct and the reverse kinetic rate constants, k_i and p_i respectively, are indicated on each reaction. This example of reacting system has been set because

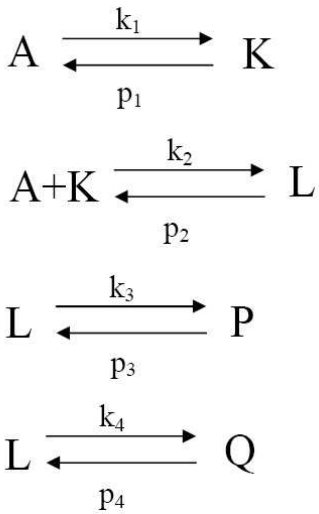


Figure 1: Chemical reversible reacting system.

includes several situations that can be encountered in typical chemical reaction mechanisms such as consecutive, competitive, first and second order chemical reactions. We expect that the identifiability analysis of such system will give some relevant information in more complex chemical systems about how the kinetic rate constants are related or the mathematical procedures that we will expect to need.

The problem of the structural identifiability of the model consists of the determination of all parameter sets which give the same input-output structure. A characterization of structural identifiability is given in [1].

The dynamic model describing the internal structure of the reactions given in figure 1 is formulated theoretically using nonlinear state-space mathematical equations, depending on unknown parameters. That is,

$$\begin{aligned}
\dot{z}_A &= -k_1 z_A + p_1 z_K - k_2 z_A z_K + p_2 z_L \\
\dot{z}_K &= k_1 z_A - p_1 z_K - k_2 z_A z_K + p_2 z_L \\
\dot{z}_L &= k_2 z_A z_K - (p_2 + k_3 + k_4) z_L + p_3 z_P + p_4 z_Q \\
\dot{z}_P &= k_3 z_L - p_3 z_P \\
\dot{z}_Q &= k_4 z_L - p_4 z_Q
\end{aligned}$$

where $z_A,\ z_K,\ z_L,\ z_P$ and z_Q are the concentrations of the reactive $A,\ K,\ L,\ P$ and Q at time t, respectively.

This system can be linearized around the equilibrium point of the system $\vec{z}_e = (A_e\ K_e\ L_e\ P_e\ Q_e)^T$, obtaining the following continuous linear system $\dot{\vec{x}} = A\vec{x}$ where

the matrix A is

$$A = \begin{pmatrix} -(k_1 + k_2K_e) & p_1 - k_2A_e & p_2 & 0 & 0 \\ k_1 - k_2K_e & -(p_1 + k_2A_e) & p_2 & 0 & 0 \\ k_2K_e & k_2A_e & -(p_2 + k_3 + k_4) & p_3 & p_4 \\ 0 & 0 & k_3 & -p_3 & 0 \\ 0 & 0 & k_4 & 0 & p_4 \end{pmatrix}.$$

In particular, the above equations can model a reversible chemical reacting network in a batch reactor, see [2]. The equilibrium point can be perturbed by the injection, in impulse, of a given concentration of either component A, K, L, P and Q. This injection is commonly employed as additional input variables. In this case, when we consider the impulse on one reactive the system is described by

$$\dot{\vec{x}} = A\vec{x} + Bu$$

where $B = e_i$ being e_i the canonical vector and it is important to know the identifiability of this system. Several authors have studied this topic using different techniques. For instance, if we consider the Markov parameters of this system $V_j = A^jB$, $j \geq 0$, we can prove that the system is identifiable, that is all the parameters of the model can be known using experimental data (see [3]).

Therefore, usually the parameters of the model are unknown and cannot be prespecified, and need to be estimated from data collected experimentally by measuring the observable variables. In this step, first we must know the number of variables that we can hope to measure. The number of directly observable variables may influence in the identifiability of the system and even sometimes can miss the identifiability. Then it is important to obtain the minimum number of variables that must be measured to identify the chemical process. Now, this process can be described by this continuous linear control system

$$\begin{aligned} \dot{\vec{x}} &= A\vec{x} + Bu \\ y &= C\vec{x}. \end{aligned}$$

In the above system the information on the observable or measured variables are obtained from the algebraic equations, that is, it is given by the structure of the matrix C.

In this work we study the identification problem associated to variables that can be measured. The main aim is to obtain the minimum number of rows of the matrix C to assure the identifiability of all parameters of the system.

Key words: identifiability, observability, chemical reaction
MSC 2000: 34,93

Acknowledgements

This work has been partially supported by MTM2010-18228.

References

[1] A. Ben-Zvi, P.J. McLellan, K.B. McAuley, *Ind. Eng. Chem. Res.* 42 (2003) 6607–6618.

[2] B. Cantó, S.C. Cardona, C. Coll, J. Navarro-Laboulais and E. Sánchez, *Dynamic optimization of a gas-liquid reactor*, J. Math. Chem. 50 (2012) 381–393.

[3] B. Cantó, C. Coll and E. Sánchez, *Identifiability of a class of discretized linear partial differential algebraic equations*, Math. Problems Eng. (2011) 1–12.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A new high-order well-balanced central scheme for 2D shallow water equations

M. T. Capilla[1] **and A. Balaguer-Beser**[1]

[1] *Department of Applied Mathematics, Universidad Politécnica de Valencia, Spain*

emails: tcapilla@mat.upv.es, abalague@mat.upv.es

Abstract

In this paper we present a new high-order well-balanced central scheme to solve the shallow water equations in two spatial dimensions. A Runge-Kutta scheme with a natural continuous extension has been applied for time discretization, using a Gaussian quadrature rule to evaluate time integrals. The reconstruction operator is of high order and non-oscillatory type. The central finite volume formulation requires to evaluate the flux integrals in one spatial dimension and to approach the 2D source term integrals. A new procedure for these integrals has been defined in order to verify the exact C-property, using the water surface elevation instead of the water depth as a variable. Numerical experiments have confirmed the high-resolution properties of our numerical scheme in 2D test problems.

Key words: Central scheme, well-balanced, non-oscillatory, shallow water equations.
MSC 2000: 120600

1 Introduction

The shallow water equations have initially been approached in the framework of upwind schemes. In this context, Bermúdez and Vázquez [3] proposed the idea of the exact C-property which means that the scheme is exact when applied to the stationary case. More recently, high-order central schemes have appeared which do not require to use the projection of the equations along characteristic directions (see [5]). In this paper we present a new central finite volume scheme which maintains the exact C-property and it is high-order accurate for the solutions of the shallow-water equations. For space discretization we will use the one-dimensional reconstruction procedure defined by Balaguer and Conde in [1] which uses centered three-degree polynomials with a modification on the slope at the midpoint of

the stencil, which has been designed so that the resulting polynomial has the same type of monotony that the data that are interpolated.

Central schemes described in [1] may be used to solve multidimensional hyperbolic systems of conservation laws in the framework of the two-dimensional central schemes described in Levy et al. [8]. In [6], we have applied the reconstruction polynomials of [1] to solve the shallow water equations in one spatial dimension, using the temporal integration scheme described by Caleffi et al. in [5]. In this paper we define a extension of the central schemes described in [6] and [8] for the two-dimensional shallow water equations. The new scheme has been designed in order to verify the exact C-property.

2 Two dimensional shallow water systems

The shallow water system in two space dimensions takes the form:

$$\begin{cases} h_t + (q_1)_x + (q_2)_y = 0\,, \\ (q_1)_t + \left(\dfrac{q_1^2}{h} + \dfrac{1}{2}gh^2\right)_x + \left(\dfrac{q_1\,q_2}{h}\right)_y = -gh(Z_b)_x\,, \\ (q_2)_t + \left(\dfrac{q_1\,q_2}{h}\right)_x + \left(\dfrac{q_2^2}{h} + \dfrac{1}{2}gh^2\right)_y = -gh(Z_b)_y\,, \end{cases} \tag{1}$$

which are the equations governing the flow of a shallow layer of homogeneous fluid in a two dimensional domain $D \subset \mathbb{R}^2$. In the equations, $h(x,y,t)$ is the water depth; $q_j(x,y,t)$ is the component of the discharge in the direction j, related to the velocity of the fluid $(v_1(x,y,t), v_2(x,y,t))$ by the expression $q_j(x,y,t) = h(x,y,t) \cdot v_j(x,y,t)$; $Z_b(x,y)$ is the function that specifies the bottom topography and g is the gravitational constant.

We manipulate the system in Equation (1) in order to use the water surface elevation $\eta(x,y,t) = h(x,y,t) + Z_b(x,y)$ instead of the water depth; then the shallow water system may be given by:

$$\begin{pmatrix} \eta \\ q_1 \\ q_2 \end{pmatrix}_t + \begin{pmatrix} q_1 \\ \frac{q_1^2}{\eta - Z_b} + \frac{1}{2}g(\eta - Z_b)^2 \\ \frac{q_1 q_2}{\eta - Z_b} \end{pmatrix}_x + \begin{pmatrix} q_2 \\ \frac{q_1 q_2}{\eta - Z_b} \\ \frac{q_2^2}{\eta - Z_b} + \frac{1}{2}g(\eta - Z_b)^2 \end{pmatrix}_y = \begin{pmatrix} 0 \\ -g(\eta - Z_b)(Z_b)_x \\ -g(\eta - Z_b)(Z_b)_y \end{pmatrix}. \tag{2}$$

System (2) can be expressed as:

$$u_t + f(u)_x + g(u)_y = s(x,y,u)\,, \tag{3}$$

where $u = (h, q_1, q_2)^T$ is the vector of conservative variables, $f(u)$ and $g(u)$ are the flux vector valued functions and $s(x,y,u)$ is the source term relative to the bottom slope. In [6] we presented a high-order well-balanced numerical scheme for solving the one dimensional

shallow water system and we added some comments about the two dimensional extension of the numerical scheme. In this paper, the numerical scheme of [6] is extended to two spatial dimensions using a uniform rectangular grid and following the methodology developed in [2] and [8]. In the next sections we develop the numerical model adapting it to the resolution of the two dimensional shallow water system (2). This involves designing a new source term treatment to verify the exact C-property.

3 Numerical scheme

We consider that the time interval is discretized into NT values, being Δt the time step, where $t^n = n \cdot \Delta t$ for $n = 0, 1, 2, ...,$ NT. The spatial discretization of the domain is based on the mesh sizes Δx and Δy in x and y directions respectively, so we use in the calculations the grid defined by the points $x_i = x_{i-1} + \Delta x$ and $y_j = y_{j-1} + \Delta y$, and the staggered grid defined by $x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}$ and $y_{j+\frac{1}{2}} = y_j + \frac{\Delta y}{2}$, for $i = 1, 2, ...,$ NX and $j = 1, 2, ...$ NY.

The central finite volume method integrates the system (2) with respect to the space and time variables over the control volume $I_{i+\frac{1}{2},j+\frac{1}{2}} \times [t^n, t^{n+1}]$, being $I_{i+\frac{1}{2},j+\frac{1}{2}} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, resulting with

$$\overline{\overline{u}}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \overline{\overline{u}}^{n}_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \left\{ \int_{y_j}^{y_{j+1}} \left[f_{i+1}(y,\tau) - f_i(y,\tau) \right] \, dy \right\} d\tau$$
$$- \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \left\{ \int_{x_i}^{x_{i+1}} \left[g_{j+1}(x,\tau) - g_j(x,\tau) \right] \, dx \right\} d\tau + \int_{t^n}^{t^{n+1}} \overline{\overline{s}}_{i+\frac{1}{2},j+\frac{1}{2}}(\tau) \, d\tau \,, \qquad (4)$$

where for simplicity of notation we have denoted $f_i(y,t) = f(u(x_i, y, t))$ and $g_j(x,t) = g(u(x, y_j, t))$. The first integral on the right-hand side (rhs) of (4) is the cell average of the function $u(x, y, t^n)$ on the staggered cell $I_{i+\frac{1}{2},j+\frac{1}{2}}$, given by

$$\overline{\overline{u}}^{n}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{\Delta x \Delta y} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} u(x, y, t^n) \, dy \, dx \,, \qquad (5)$$

and the cell average in the last time integral on the rhs of (4) is

$$\overline{\overline{s}}_{i+\frac{1}{2},j+\frac{1}{2}}(\tau) = \frac{1}{\Delta x \Delta y} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} s(u(x, y, \tau)) \, dy \, dx \,. \qquad (6)$$

In order to obtain the fourth order accuracy in time, a Gaussian quadrature with two integration nodes is selected to evaluate the time flux integrals in Equation (4), for example:

$$\int_{t^n}^{t^{n+1}} f(u(x_i, y_j, z)) dz = \frac{\Delta t}{2} \left(f(\hat{u}^{n+\beta_0}_{i,j}) + f(\hat{u}^{n+\beta_1}_{i,j}) \right) \,, \qquad (7)$$

 ISBN:978-84-615-5392-1

where $\hat{u}_{i,j}^{n+\beta_k} = u(x_i, y_j, t^n + \beta_k \Delta t)$, $k = 0, 1$, being

$$\beta_0 = \left(\frac{1 - 1/\sqrt{3}}{2} \right), \qquad \beta_1 = \left(\frac{1 + 1/\sqrt{3}}{2} \right).$$

According to [8], the following centered quadrature rule in space is used for the integrals in space:

$$\int_{y_j}^{y_{j+1}} f_i(y,t)dy = \frac{\Delta y}{24} \left[-f_i(y_{j+2},t) + 13 f_i(y_{j+1},t) + 13 f_i(y_j,t) - f_i(y_{j-1},t) \right]. \tag{8}$$

In this way, the quadrature rule for approximating the integrals of the fluxes involves nodes on the segments $(x_i, y_j) \times [t^n, t^{n+1}]$ where the solution remains smooth.

In the next sections we present a summary of the procedure involved in a computational time step to obtain the cell-averages $\overline{\overline{u}}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$ at the next time step t^{n+1}, starting from Equation (4).

3.1 Reconstruction at time t^n of point values and averaged values

At time t^n we start the reconstruction with a given fourth order approximation of the following cell averages:

$$\overline{\overline{u}}^{n}_{i,j} = \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x,y,t^n)\, dy\; dx\,. \tag{9}$$

We will denote by $I_{i,j}$ the cell centered around the grid point (x_j, y_j): $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{i-\frac{1}{2}}, y_{i+\frac{1}{2}}]$. A two-dimensional piecewise-polynomial reconstruction is computed from the data $\{\overline{\overline{u}}^{n}_{i,j}\}$ resulting with

$$R_{i,j}(x,y;\overline{\overline{u}}^{n}) \equiv R_{i,j}(x,y,t^n) = u(x,y,t^n) + O(h^4)\,, \qquad \forall\, x,y \in I_{i,j}\,, \tag{10}$$

where $h = \Delta x = \Delta y$ and $R_{i,j}(x,y,t^n)$ is a vector valued bicubic polynomial, obtained simply through the tensor product of two one-dimensional interpolating polynomials, which were defined and analyzed in [1]. Also, we used these 1D reconstruction polynomials in the numerical model and applications presented in a previous work [6]. The cell averages, $\overline{\overline{u}}^{n}_{i+\frac{1}{2},j+\frac{1}{2}}$ in Equation (5) can be approximated using the polynomials $R_{i+1,j+1}(x,y,t^n)$, $R_{i,j+1}(x,y,t^n)$, $R_{i+1,j}(x,y,t^n)$ and $R_{i,j}(x,y,t^n)$ on the corresponding quarter cells, using a gaussian quadrature rule. Let $I^m_{i,j}$, $m = 1, ..., 4$ denote the four quarters of the cell $I_{i,j}$, with $I^1_{i,j}$ being the upper-right quarter, while the other three quarters are numbered clockwise. A fourth-order computation of the cell averages in (5) is obtained through the averages of $R_{i,j}(x,y,t^n)$ over the four quarter cells:

$$\overline{\overline{R}}^{(m)}_{i,j} \simeq \frac{4}{\Delta x\, \Delta y} \int_{I^m_{i,j}} R_{i,j}(x,y,t^n) dx\, dy\,, \qquad m = 1, ..., 4\,. \tag{11}$$

Then the cell average of the solution in (5) is the sum of the four quarter-cell averages defined in (11):

$$\overline{\overline{u}}^{n}_{i+\frac{1}{2},j+\frac{1}{2}} = \overline{\overline{R}}^{(1)}_{i,j} + \overline{\overline{R}}^{(2)}_{i,j+1} + \overline{\overline{R}}^{(3)}_{i+1,j+1} + \overline{\overline{R}}^{(4)}_{i+1,j}\,.$$

Using the reconstruction polynomials $R_{i,j}(x,y,t^n)$, we also approximate the point-values $\hat{u}^n_{i,j}$ at time t^n on the non-staggered grid.

3.2 Source term integration

The second component of the cell average of the source term in (6) is given by

$$\overline{\overline{s}}^{[2]}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{\Delta y}\int_{y_j}^{y_{j+1}} \left(\frac{-1}{\Delta x}\int_{x_i}^{x_{i+1}} g(\eta - Z_b)\frac{\partial Z_b}{\partial x}\,dx\right)\,dy\,. \tag{12}$$

Following the procedure described in [6], an integration by parts with respect to the variable x is performed, in order to involve the spatial derivative of the free surface elevation instead of the bed elevation. This was suggested by Caleffi et al. in [5] for the one dimensional case. Thus, initially we compute:

$$\begin{aligned}
\overline{s}^{[2]}_{i+\frac{1}{2}}(y) &= \frac{-1}{\Delta x}\int_{x_i}^{x_{i+1}} g(\eta(x,y) - Z_b(x,y))\frac{\partial Z_b(x,y)}{\partial x}\,dx \\
&= \frac{g}{2\Delta x}\left[\hat{Z}^2_{b,i+1}(y) - \hat{Z}^2_{b,i}(y) - 2\hat{\eta}_{i+1}(y)\hat{Z}_{b,i+1}(y) + 2\hat{\eta}_i(y)\hat{Z}_{b,i}(y)\right] \\
&+ \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} \psi(x,y)\,dx\,,
\end{aligned} \tag{13}$$

where $\psi(x,y) = g\cdot Z_b(x,y)\cdot\frac{\partial\eta(x,y)}{\partial x}$. This formulation depends on the spatial derivative of the free surface elevation which is preferable to obtain accurate point-values of the variables at cell-centers and to prevent numerical errors in the solutions. The spatial integral in Equation (12) in the y variable is evaluated using a centered quadrature rule:

$$\overline{\overline{s}}^{[2]}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{24}\left[-\overline{s}^{[2]}_{i+\frac{1}{2}}(y_{j+2}) + 13\,\overline{s}^{[2]}_{i+\frac{1}{2}}(y_{j+1}) + 13\,\overline{s}^{[2]}_{i+\frac{1}{2}}(y_j) - \overline{s}^{[2]}_{i+\frac{1}{2}}(y_{j-1})\right]\,. \tag{14}$$

In a similar way we evaluate the third component of the averaged source term in (6), obtaining:

$$\begin{aligned}
\overline{s}^{[3]}_{j+\frac{1}{2}}(x) &= \frac{-1}{\Delta y}\int_{y_j}^{y_{j+1}} g(\eta(x,y) - Z_b(x,y))\frac{\partial Z_b(x,y)}{\partial y}\,dy \\
&= \frac{g}{2\Delta y}\left[\hat{Z}^2_{b,j+1}(x) - \hat{Z}^2_{b,j}(x) - 2\hat{\eta}_{j+1}(x)\hat{Z}_{b,j+1}(x) + 2\hat{\eta}_j(x)\hat{Z}_{b,j}(x)\right] \\
&+ \frac{1}{\Delta y}\int_{y_j}^{y_{j+1}} \phi(x,y)\,dy\,,
\end{aligned} \tag{15}$$

 ISBN:978-84-615-5392-1

where $\phi(x,y) = g \cdot Z_b(x,y) \cdot \frac{\partial \eta(x,y)}{\partial y}$ and

$$\int_{y_j}^{y_{j+1}} \phi(x,y)\ dy = \int_{y_j}^{y_j+\frac{\triangle y}{2}} P_{j,x}(y;\hat{\phi})\ dy + \int_{y_j+\frac{\triangle y}{2}}^{y_{j+1}} P_{j+1,x}(y;\hat{\phi})\ dy,$$

where the interpolating three-degree polynomial $P_{j,x}(y;\hat{\phi})$ approximates point values starting from point values (see [6]).

Again, we apply a centered quadrature rule for the integral with respect to the x variable:

$$\overline{\overline{s}}^{[3]}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{24}\left[-\overline{s}^{[3]}_{j+\frac{1}{2}}(x_{i+2}) + 13\,\overline{s}^{[3]}_{j+\frac{1}{2}}(x_{i+1}) + 13\,\overline{s}^{[3]}_{j+\frac{1}{2}}(x_i) - \overline{s}^{[3]}_{j+\frac{1}{2}}(x_{i-1})\right]. \tag{16}$$

The source term time integrals in (4) are also evaluated using a Gaussian quadrature rule with two nodes:

$$\int_{t^n}^{t^{n+1}} \overline{\overline{s}}_{i+\frac{1}{2},j+\frac{1}{2}}(\tau)\,d\tau = \frac{\Delta t}{2}\left(\overline{\overline{s}}^{n+\beta_0}_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{\overline{s}}^{n+\beta_1}_{i+\frac{1}{2},j+\frac{1}{2}}\right), \tag{17}$$

where

$$\overline{\overline{s}}^{n+\beta_k}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{\Delta x \Delta y}\int_{x_i}^{x_{i+1}}\int_{y_j}^{y_{j+1}} s(u(x,y,t^n+\beta_k\Delta t))\ dy\,dx\,, \qquad k=0,1\,. \tag{18}$$

3.3 Reconstruction of Runge-Kutta fluxes

In order to evaluate the time integrals of the source term (17) and the time flux integrals in (7) we have to predict the point-values of the solution at two intermediate states: $\hat{u}^{n+\beta_k}_{i,j} \equiv u(x_i, y_j, t^n + \beta_k\,\Delta t)$, $k = 0, 1$. The prediction of these intermediate values at times $t^{n+\beta_0}$ and $t^{n+\beta_1}$ is obtained by means of a Runge-Kutta scheme coupled with the natural continuous extension (NCE) [4]:

$$\hat{u}^{n+\beta_k}_{i,j} \equiv u(x_i, y_j, t^n + \beta_k\,\Delta t) = \hat{u}^n_{i,j} + \Delta t \sum_{l=1}^{4} b_l(\beta_k)\,k^{(l)}_{i,j}\,, \tag{19}$$

where the constants $b_l(\beta_k)$ are given in [6] and $k^{(l)}_{i,j}$, $1 \le l \le 4$, are the Runge-Kutta fluxes, which coincide with a numerical evaluation of $(-f_x - g_y + s)$ in the shallow water system (3). We use the point values of the solution $\{\hat{u}^{(l)}_{i,j}\}$ to calculate the functions F^j_i and G^i_j, defined by

$$F^j_i(x_k;\hat{u}) = -\left[f(\hat{u}_{k,j}) - f(\hat{u}_{i,j})\right] + \begin{bmatrix} 0 \\ \frac{1}{2}g\left[(\hat{\eta}_{i,j} - \hat{Z}_{b,k}(y_j))^2 - (\hat{\eta}_{i,j} - \hat{Z}_{b,i}(y_j))^2\right] \\ 0 \end{bmatrix},$$

 ISBN:978-84-615-5392-1

$$G_j^i(y_k;\hat{u}) = -\left[g(\hat{u}_{i,k}) - g(\hat{u}_{i,j})\right] + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2}g\left[(\hat{\eta}_{i,j} - \hat{Z}_{b,k}(x_i))^2 - (\hat{\eta}_{i,j} - \hat{Z}_{b,j}(x_i))^2\right] \end{bmatrix}, \quad (20)$$

where for simplicity we have omitted the index (l) from $\hat{u}_{i,j}^{(l)}$. Equation (20) is a two-dimensional extension of the definition presented in [5, 6]. This definition, (20), guarantees that the numerical scheme maintains the exact C-property. The evaluation of the Runge-Kutta fluxes is approximated using the interpolating polynomials that approximate the functions F_i^j and G_j^i, by means of:

$$k_{i,j}^{(l)} \equiv k_{i,j} = \frac{dP_i(x;F_i^j)}{dx} + \frac{dP_j(y;G_j^i)}{dy}, \quad (21)$$

where:

$$\begin{aligned} P_j(x;y^{(l)}) &= \hat{y}_i^{(l)} + \theta_i^{(l)}\left[dP_i^{(l)} \cdot \left(\frac{x-x_i}{\Delta x}\right) + \left(\frac{\hat{y}_{i-1}^{(l)} - 2\hat{y}_i^{(l)} + \hat{y}_{i+1}^{(l)}}{2}\right) \cdot \left(\frac{x-x_i}{\Delta x}\right)^2\right. \\ &+ \left.\left(\frac{-\hat{y}_{i-1}^{(l)} + \hat{y}_{i+1}^{(l)} - 2dP_i^{(l)}}{2}\right) \cdot \left(\frac{x-x_i}{\Delta x}\right)^3\right]. \end{aligned} \quad (22)$$

For more details about these polynomials we refer the reader to [1].

4 C-property verification

In this Section we will prove that our numerical scheme satisfies the exact C-property. It can be verified through straightforward calculations that in case of quiescent flow, starting from $\overline{\overline{\eta}}_{i,j}^n = \eta^* = \text{constant}$ and $\overline{\overline{q_k}}_{;i,j}^n = 0$, where $q_k = v_k h$, for $k = 1, 2$, gives $\overline{\overline{\eta}}_{i,j}^{n+1} = \hat{\eta}_{i,j}^{n+1} = \eta^*$ and $\overline{\overline{q_k}}_{;i,j}^{n+1} = \hat{q_k}_{;i,j}^{n+1} = 0$, $\forall\, n$. This means that, starting from the mentioned initial conditions, the scheme maintains the steadiness of the point values and the cell averages of the solution.

4.1 C-property for cell averages

The steadiness of the cell averages of the solution, i.e., $\overline{\overline{u}}_{i,j}^{n+1} = \overline{\overline{u}}_{i,j}^n$, in the case of quiescent flow, must be verified. To achieve this result, it is sufficient to show that in the following equation:

$$\begin{aligned} \overline{\overline{u}}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \overline{\overline{u}}_{i+\frac{1}{2},j+\frac{1}{2}}^{n} - \frac{1}{\Delta x \Delta y}\int_{t^n}^{t^{n+1}}\left[\int_{y_j}^{y_{j+1}}\left(f_{i+1}(y,\tau) - f_i(y,\tau)\right)\, dy\right. \\ \left. + \int_{x_i}^{x_{i+1}}\left(g_{j+1}(x,\tau) - g_j(x,\tau)\right)\, dx - \Delta x \Delta y\, \overline{\overline{s}}_{i+\frac{1}{2},j+\frac{1}{2}}(\tau)\right] d\tau, \end{aligned} \quad (23)$$

 ISBN:978-84-615-5392-1

the vector valued term in square brackets is zero, for water at rest.

It can be seen from Equation (2), that the first component of the term in square brackets in (23) is identically zero. We have to prove that the second and third components of this term are zero. Remembering Equation (13) and assuming $\hat{\eta}_i(y) = \hat{\eta}_{i+1}(y) = \eta^* = \text{constant}$, the second component of the term in square brackets in (23) may be written as:

$$\begin{aligned}
&\int_{y_j}^{y_{j+1}} \left[f^{[2]}_{i+1}(y,\tau) - f^{[2]}_{i}(y,\tau) \right] dy - \Delta x \int_{y_j}^{y_{j+1}} \overline{s}^{[2]}_{i+\frac{1}{2}}(y,\tau)\, dy \qquad (24)\\
&\quad = \int_{y_j}^{y_{j+1}} \frac{1}{2} g \left[(\eta^* - \hat{Z}_{b,i+1}(y))^2 - (\eta^* - \hat{Z}_{b,i}(y))^2 \right] dy\\
&\quad - \int_{y_j}^{y_{j+1}} \left[\frac{1}{2} g (\hat{Z}^2_{b,i+1}(y) - \hat{Z}^2_{b,i}(y)) - g\eta^* (\hat{Z}_{b,i+1}(y) - \hat{Z}_{b,i}(y)) + \Delta x\, \overline{\psi}_{i+\frac{1}{2}}(y) \right] dy\\
&\quad = -\Delta x \int_{y_j}^{y_{j+1}} \overline{\psi}_{i+\frac{1}{2}}(y)\, dy,
\end{aligned}$$

where $\overline{\psi}_{i+\frac{1}{2}}(y)$ is the cell average of the function $\psi(x,y) = g\, Z_b(x,y)\, \frac{\partial \eta}{\partial x}$ on the cell $[x_i, x_{i+1}]$.

In a similar way, it can be proved that the third component of the term in square brackets in (23) is equal to $-\Delta y \int_{x_i}^{x_{i+1}} \overline{\phi}_{j+\frac{1}{2}}(x)\, dx$, where $\phi(x,y) = g\, Z_b(x,y)\, \frac{\partial \eta}{\partial y}$, so if $\overline{\psi}_{i+\frac{1}{2}}(y)$ and $\overline{\phi}_{j+\frac{1}{2}}(x)$ are zero then the terms in square brackets are also zero. In the case of quiescent flow, if $\hat{\eta}_{i,j}$ are constants $\forall\, i,j$, then the free surface derivatives $\hat{\eta}'_{i,j} = 0$. Consequently, the point-values $\hat{\psi}_{i,j}$ and $\hat{\phi}_{i,j}$ are zero. Finally, the cell averages of the functions ψ and ϕ are approximated using the point-values $\hat{\psi}_{i,j}$ and $\hat{\phi}_{i,j}$, and therefore if $\hat{\psi}_{i,j} = \hat{\phi}_{i,j} = 0$ also the averaged values $\overline{\psi}_{i+\frac{1}{2}}(y)$ and $\overline{\phi}_{j+\frac{1}{2}}(x)$ are zero, and the C-property verification for cell averages is proved. In an analogous form we can prove the C-property for point-values.

5 Applications

In this Section we examine the behavior and accuracy of the numerical scheme, in several numerical tests. We show a numerical verification of the exact C-property and we present results for several standard tests proposed in the literature. In all cases, we consider $\Delta x = \Delta y$ for the spatial steps. In the case of the shallow water system with fixed bottom topography, the numerical stability of the scheme is assured by selecting a time step satisfying the CFL condition which is related to the numerical stability of the Runge-Kutta scheme used in the time integration (see [8]). We focus our attention on the behavior of the reconstruction method described in the previous Section. To assure that our numerical scheme obtains accurate results with different CFL numbers, we have taken a fixed time step in all simulations: $\Delta t = 0.0025\, \Delta x = 0.0025\, \Delta y$.

 ISBN:978-84-615-5392-1

5.1 Test for the exact C-property

This test, presented in [10], is used to verify numerically that our scheme maintains the exact C-property over a non flat bottom. The bottom topography is given by a two-dimensional hump:

$$Z_b(x,y) = 0.8\, e^{-50((x-0.5)^2+(y-0.5)^2)}\,, \qquad x,y \in [0,1]\,.$$

As initial condition for the water depth we set $h(x,y,0) = 1 - Z_b(x,y)$ and the initial velocity is set to be zero: $q_1(x,y,0) = q_2(x,y,0) = 0$. This surface should remain flat. We consider a rectangular mesh with $\Delta x = \Delta y = 0.01$ which corresponds to a NX=NY=100 uniform mesh. Table 1 contains the L^1 and L^∞ errors for the water height h and the discharges q_1 and q_2, at time $t = 0.05$ s.

Unknowns	h		q_1		q_2	
Precision	L^1	L^∞	L^1	L^∞	L^1	L^∞
	0	0	0	0	0	0

Table 1: L^1 and L^∞ errors for the C-property analysis

From these results, we can conclude that the exact C-property is verified.

5.2 Circular dam-break problem

We consider a test problem described in [7] and [9]. The domain is the square $[0,2]\times[0,2]$ and the bottom topography is given by the function:

$$Z_b(x,y) = \begin{cases} \frac{1}{8}(\cos(2\pi(x-0.5))+1)(\cos(2\pi y)+1)\,, & \text{if } (x-1.5)^2+(y-1)^2 \le (0.5)^2\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

The water depth is initially given by:

$$h(x,y,0) = \begin{cases} 1.1 - Z_b(x,y)\,, & \text{if } (x-1.25)^2+(y-1)^2 \le (0.1)^2\,, \\ 0.6 - Z_b(x,y)\,, & \text{otherwise}\,, \end{cases}$$

and $q_1(x,y,0) = q_2(x,y,0) = 0$. We compute the numerical results using (200×200) grid points ($\Delta x = \Delta y = 0.01$). Figure 1 (left) shows the numerical solution for the free surface level at time $t = 0.15$ s. The analytical solution is not known for this problem, and therefore we have used our numerical scheme to obtain a reference solution in a mesh composed by (800×800) cells. In Figure 1 (right) we compare the numerical and the reference solutions for the free surface level at time $t = 0.15$ s, along the line $y = 1$.

These results can be compared with those in [7] and [9].

 ISBN:978-84-615-5392-1

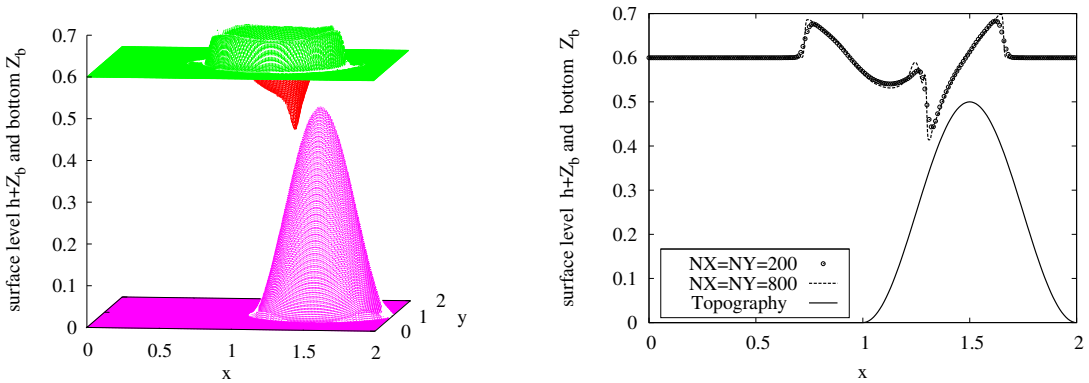


Figure 1: The circular dam-break problem: solution at $t = 0.15$ s. Left: free surface and topography. Right: a longitudinal section at $y = 1$.

5.3 Perturbation of a lake at rest in 2D

We show numerical results for a classical example of a small perturbation of a two-dimensional steady-state flow [10, 11, 9]. The perturbation occupies a small portion of the computational domain. The domain is the rectangle $[0, 2] \times [0, 1]$ and the topography is an isolated elliptical shaped hump:

$$Z_b(x, y) = 0.8\, e^{-5((x-0.9)^2-50(y-0.5)^2)}\,.$$

The initial conditions are $q_1(x, y, 0) = q_2(x, y, 0) = 0$ and

$$h(x, y, 0) = \begin{cases} 1.01 - Z_b(x, y)\,, & \text{if } 0.05 \le x \le 0.15\,, \\ 1 - Z_b(x, y)\,, & \text{otherwise}\,. \end{cases}$$

So the surface is almost flat except for $0.05 \le x \le 0.15$, where h is perturbed upward by 0.01. Figure 2 displays the results obtained with our numerical scheme on a uniform mesh with (200×100) nodes. The Figure shows the contours of the surface level $h + Z_b$ and a longitudinal section at $y = 0.5$, at two different ending times: $t = 0.24$ s and $t = 0.36$ s. These results can be directly compared with those in [10, 11, 9]. We observe that our scheme can resolve consistently this problem and no oscillations are observed.

6 Conclusions

In this paper we have extended the well-balanced central scheme described in [6] to solve the shallow water system in two space dimensions. The resulting scheme satisfies the exact

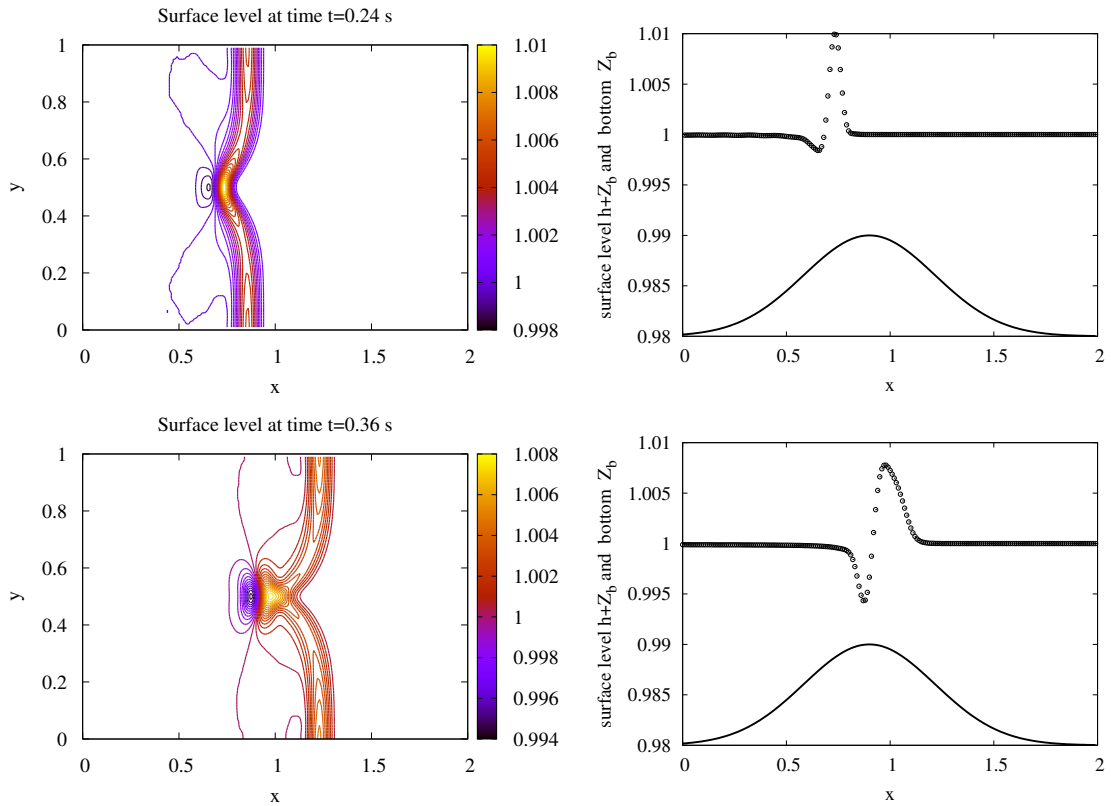


Figure 2: Perturbation of a lake at rest: Left: contours of the surface level $h + Z_b$ with 30 uniformly spaced contour lines. Right: solution along the line $y = 0.5$ ($Z_b = Z_b/80 + 0.98$).

C-property and it has been designed to have high-resolution and a non-oscillatory behavior. A circular dam-break problem and a test with small perturbation of a lake at rest are solved to demonstrate these theoretical properties. The capability of the numerical scheme in reproducing the space-time evolution of the variables has been proved, and the agreement between our results and what are reported in literature has been confirmed.

Acknowledgements

This work was partially funded by the "Programa de Apoyo a la Investigación y Desarrollo" (PAID-06-10) of the Universidad Politécnica de Valencia. Angel Balaguer-Beser thanks the support of the Spanish Ministry of Education and Science in the framework of the Projects

CGL2009-14220-C02-01 and CGL2010-19591.

References

[1] A. Balaguer and C. Conde, *Fourth-Order Non-oscillatory Upwind and Central Schemes for Hyperbolic Conservation Laws*, SIAM J. Numer. Anal. **43**(2) (2005) 455–473.

[2] A. Balaguer-Beser, *A new reconstruction procedure in central schemes for hyperbolic conservation laws*, Int. J. Numer. Meth. Engng. **86** (2011) 1481–1506.

[3] A. Bermúdez and M. E. Vázquez, *Upwind methods for hyperbolic conservation laws with source terms*, Computers and Fluids **23** (1994) 1049–1071.

[4] F. Bianco, G. Puppo, and G. Russo, *High-order central schemes for hyperbolic systems of conservation laws*, SIAM J. Sci. Comput. **21**(1) (1999), 294–322.

[5] V. Caleffi, A. Valiani, and A. Bernini, *Fourth-order balanced source term treatment in central WENO schemes for shallow water equations*, J. Comput. Phys. **218** (2006) 228–245.

[6] M. T. Capilla and A. Balaguer-Beser, *A well-balanced high-resolution shape-preserving central scheme to solve one-dimensional sediment transport equations*, Adv. Eng. Softw. (2012) http://dx.doi.org/10.1016/j.advengsoft.2012.04.003.

[7] M. J. Castro, E. D. Fernández-Nieto, A. M. Ferreiro, J. A. García-Rodríguez and C. Parés, *High order extensions of Roe schemes for two dimensional nonconservative fyperbolic systems*, J. Sci. Comput. **39** (2009) 67–114.

[8] D. Levy, G. Puppo, and G. Russo, *A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws*, SIAM J. Sci. Comput. **24** (2002) 480–506.

[9] A. Martínez-Gavara and R. Donat, *A hybrid second order scheme for shallow water flows*, J. Sci. Comput. **48** (2011) 241–257.

[10] Y. Xing and C. W. Shu, *High order finite difference WENO schemes with the exact conservation property for the shallow water equations*, J. Comput. Phys. **208** (2005) 206–227.

[11] Y. Xing and C. W. Shu, *A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms*, Commun. Comput. Phys. **1**(1) (2006) 100–134.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A faster than real-time simulator of motion platforms

Sergio Casas[1], Ricardo Olanda[1], Marcos Fernández[1] and José V. Riera[1]

[1] *Robotics Institute and Information Technology and Communications (IRTIC), University of Valencia*

emails: Sergio.Casas@uv.es, Ricardo.Olanda@uv.es, Marcos.Fernandez@uv.es, J.Vicente.Riera@uv.es

Abstract

Motion platforms have been used for several decades in real-time simulations for motion cueing generation. Although many motion cueing algorithms (MCA) have been proposed to generate appropriate signals for motion platforms, this software always needs to be set-up for the specific application and motion platform used. This process is both expensive and time consuming, as the tests are performed on expensive motion platforms and the parameters are often adjusted by successive approximations, which may take plenty of time. Moreover, these tests can perform harmful sequences than can damage the hardware and even hurt human testers in case of failure. In order to solve these problems, we present a generic real-time virtual simulator of motion platforms. A motion platform simulator emulates the behavior of a real counterpart without the need to actually build it, and unlike a real one, it can run faster than real time, allowing fast tests and parameterizations of MCA. This simulator is assessed with a comparative study with a 3-DoF (Degrees of Freedom) and a 6-DoF real motion platforms.

Key words: Motion platform, virtual prototyping, real-time simulation, MCA.

1 Introduction

A motion platform [11] is a powered, mechanical and self-contained motion generator system. Motion platforms have been used for motion cueing in different kind of applications, like real-time flight [21] and driving simulators [18], industrial equipment training [8] or medical rehabilitation [4]. Despite their widespread use, it is not easy to generate the appropriate signals for these manipulators in order to obtain the desired behavior [17]. A large set of tests for each motion-platform and for each specific use is necessary, requiring a

successive approximation adjustment of the parameters [14]. This process is both expensive and time consuming, because the motion platform has to be built and then, tested for a long time. Moreover, some tests perform severe and potentially harmful movements that can damage the motion platform, and more importantly, hurt human testers in case of software or hardware failure.

In order to solve these problems, a motion platform simulator can be used. This simulator performs a computer-based emulation of a real motion platform, receiving the same inputs than the real motion platform does, and generating the same (simulated) outputs. This way, multiple tests do not wear or hurt the platform hardware, ensuring the safety of human testers. In addition, depending on the hardware used to run the simulator, tasks performed by the virtual platform can be executed faster than the real ones in the real motion platform, saving time. For all these reasons, we think that the use of a motion platform simulator is justified.

Hence, in this paper, we present a faster than real-time generic physically-based motion platform simulator, which allows us to simulate different kinds of motion platforms. The rest of the paper is organized as follows: section 2 reviews previous works on motion platform simulation. In section 3 the simulator itself is discussed. Section 4 shows the simulation validation tests. Finally, section 5 shows the conclusions and outlines the future work.

2 Related Work

Motion platforms are widely used in order to add inertial cues to virtual reality applications, but the simulation of motion platforms is less frequent. Nevertheless, we can find some works that compare a virtual motion platform behavior to analytical results, like in Selvakumar et al. work [19] where a 3-DoF parallel manipulator simulated using ADAMS [13] is compared using MATLAB [10]. In Hajimirzaalian et al. work [6], a Stewart platform [20] is simulated using also ADAMS, and a comparison between the direct dynamics, provided by ADAMS, and the inverse solution, based on a Lagrangian formulation, is presented. In Gosselin et al. work [5], a simulator that allows interactive kinematic analysis of spherical parallel mechanisms is presented. In Lee et al. work [9], ADAMS is used to perform a kinematic and a dynamic simulation of a 3-DoF manipulator, in order to compare it to an inverse dynamic formulation based on a Newton-Euler approach.

However, all these works use analytical solutions in order to study the behavior of a specific motion platform. Our goal is to test MCA, and thus, we want to build a virtual motion platform that is able to simulate a generic motion platform physically and numerically, not kinematically nor analytically. A solution similar to our needs is presented in Hulme and Pancotti work[7], where a 6-DoF Moog 2000E platform [12] simulator is presented. However, the simulation is restricted to this model, not allowing a generic specification. This simulator uses a detailed CAD model to provide visual information, and allows "in the

loop" intervention. However, unlike ours, the simulation is kinematic. Kinematic solutions do not simulate the process of moving the motion platform form one pose to another, they just calculate final poses. Unluckily, this is not sufficient to test MCA.

3 Simulator Description

In this paper, we present a generic motion platform simulator. For simplicity, we will assume that the simulated motion platform uses rotational motors that move connecting rods, pistons and joints in order to fulfill a desired set of DoF. The extension to translational motors should be straightforward.

The main idea behind the simulator is to fully substitute the real counterpart. This means that the virtual motion platform should receive exactly the same inputs and provide, at least, the same outputs as the real motion platform would do. It also means that an external software (an MCA plus an inverse kinematics module) should be responsible for the generation of these inputs, as we are focusing exclusively on simulating the motion platform. Thus, the inputs of the virtual motion platform motors will be the desired target angles (as we focus on rotational motors) for the motors. Regarding the outputs, the virtual motion platform should provide, at least, the current angles (this information will be received by the MCA) and optionally the current state (DoF) of the motion platform. Moreover, a real time visualization of the virtual motion platform movements is also encouraged to make the simulator user friendly. Thus, the simulation works in a loop: the virtual motion platform receives the desired target angles, this makes the virtual manipulator move (DoF change as motors move), and this current state of the motion platform is the output of the system.

The most difficult part to define here is the required parameter list, because each real motion platform has their own. We will categorize them in: motor parameters (control parameters, maximum force, etc), geometrical design of the motion platform (piston lengths, rod dimensions, etc), physics magnitudes (masses, inertias, etc), physics constraints (joints types and limits), and visual representation. In order to provide values for these parameters to the simulator, a CAD model of the motion platform and a XML file are used.

For the CAD model, we chose Autodesk 3D Studio Max [3], because it is one of the most popular CAD packages, and because by using this software, we can control both the visual representation of the motion platform and its physics structure, allowing to integrate them directly in our simulator using Open Scene Graph (OSG)[16] visual representation (.ive format) and Nvidia PhysX[15] physics representation (.nxb format). This fact allows us to be able to design virtual motion platforms easier and faster.

3.1 Simulator Operation

The virtual motion platform simulator is a physically-based graphic simulation. The physics simulation is implemented with C++ using PhysX and the visual representation, with OSG.

 ISBN:978-84-615-5392-1

The simulator is structured in a three thread system:

- Communication thread: it receives the motor target angles (from outside) and passes them to the physics thread. It also receives the necessary outputs from the physics thread, and sends them out of the simulator.
- Visual thread: it is an OSG visual representation of the CAD model in motion.
- Physics thread: it reads the inputs collected by the communication thread (target angles) and feeds the virtual motors (controlled by virtual controllers) that move the virtual motion platform. It is responsible for calculating the outputs of the simulator.

The use of a threaded structure allows running each thread at a different frequency. This is important because the physics thread should simulate the motion platform faster than it is drawn, and faster than the inputs are fed into the simulator.

In order to speed-up the input gathering, we decided to implement the communication thread with shared memory. This means that the software responsible for generating these inputs (usually the MCA) should write this information into this shared memory. The communication thread reads the inputs at a fixed rate set at the XML parameter file.

The visual system is quite simple. It loads the CAD visual description (.ive file), and draws their different parts in the positions decided by the physics thread. The visual system frequency is set to 25 Hz. Figure 1 shows a real and a virtual motion platforms.

The physics thread is the main thread of the simulator. To perform the physics simulation, the physics thread loads the CAD physics description (.nxb file) and identifies in it the objects that it needs to simulate. It identifies 6 categories at loading time: motors (objects named MotorXX), joints (named JointXX), connecting rods (named RodXX), moving pistons (named PistonXX), the moving platform base that we are actually moving (named Base) and the load that lies on the platform base (named Load).

As we intend to make it platform-independent, there are some rules that the CAD file should follow in order to be readable for our simulator. First, the names of the objects should follow the aforementioned convention. Second, connecting rods, pistons, the moving platform base, and the load should be dynamic rigid bodies (NxActor). Any rigid body can be linked to any rigid body by a link (represented by a NxJoint). Which bodies are linked, and how, is something that depends on the particular motion platform, and it is defined in the CAD model by the PhysX joint specification [15]. Third, motors are kinematic bodies that do not move (NxActor with the flag NX_BF_KINEMATIC), linked by a revolute joint to a connecting rod (MotorX should always connect to RodX). These joints will be controlled by a virtual motor controller, which is in turn commanded by the simulators inputs, and are used to calculate the motor current angles. Finally, there can be only one load and one moving platform base. The moving platform base will be the rigid body we will use to calculate the motion platform DoF.

The geometrical design of the motion platform, the joints specific constraints and all the masses and inertias from all the objects, are read directly from the .nxb file. The only things not present in the CAD model are the motor controllers' features, which are parameterized in the XML parameter file. In addition, the masses and inertias of all the objects can also be changed in the XML file.

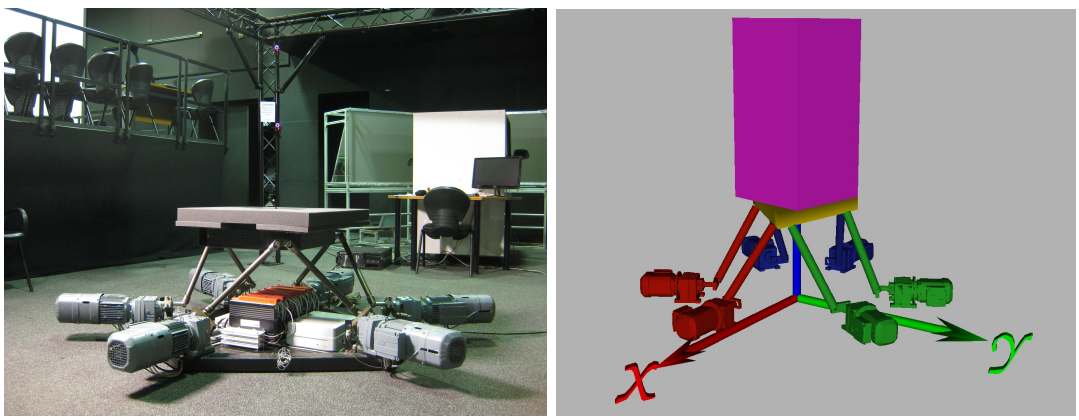


Figure 1: Real 6-DoF (left) vs virtual 6-DoF (right) motion platform

The motor controller needs some further explanation. In order to differentiate ourselves from kinematic approaches, we decided to simulate the motion of the motors. To control the motors, we decided to build a virtual PID control system. This type of controller provides good results when a good model of the controlled system is not available [1], which is the case. The tuning of the controller requires some motor operation knowledge but it can also be automated [2]. The motor parameters are: Kp (proportional constant), Ki (integral constant) and Kd (derivative constant) [1]. The input of the controller is the target angle and the output is the resulting torque on the rod joint. Once the motor controllers have calculated the resulting torque, this torque is applied to the motor-rod joint, and PhysX performs all the necessary calculations over the entire physics scene in order to simulate the positions of all the remaining objects. Once the platform base is repositioned, we can ask PhysX its position and orientation in the form of a 6-component pose. No direct kinematics is needed to achieve this information, as PhysX has performed all the dynamics (not kinematics) calculations for us.

The physics thread performs all these calculations at a parameterizable frequency (via XML). The higher the frequency, the more accurate the simulation, but the more time it consumes. If the frequency is too high, the simulation process could take longer than the time it is trying to simulate, resulting in a non real-time simulation. However, if the frequency is too low, simulation precision decreases due to an unstable physics integration.

4 Evaluation

In order to use the present simulator as a substitute for a real motion platform, it is necessary to assess its applicability. The best way to do so is to compare the response of the simulated motion platform with respect to a real one. As we had at our disposal both a 3-DoF and a 6-DoF real motion platforms, we decided to perform the assessment of the simulator by testing its performance against these two examples.

4.1 Simulator set-up

In order to perform this assessment, we need first to set-up the simulator for these two cases of study (3-DoF and 6-DoF). In order to do that, the particular designs of the motion platforms must be studied first. For the sake of brevity, we cannot make a full description of these two motion platforms, nor is it the purpose of this paper. Thus, we will just show their main features. The first motion platform is a heave-pitch-roll 3-DoF T1R2 (1 translational, 2 rotational) parallel manipulator. The second one is a 6-DoF T3R3 Stewart-like parallel manipulator. The T1R2 manipulator uses 3 motors, 3 connecting rods attached to the motors axes (this represents a revolute joint with respect to the motor body), 3 pistons connected to the rods by means of 3 ball-and-socket joints, and 1 moving base connected to the pistons by means of 3 more ball-and-socket joints. The moving base is also connected to a splined shaft through a prismatic-universal joint. The T3R3 design is different, but its constraint structure is very similar. This manipulator uses 6 motors, 6 connecting rods attached to the motors axes, 6 pistons connected to the rods through 6 ball-and-socket joints, and 1 moving base connected to the pistons through 6 more ball-and-socket joints. The motion ranges and CAD models of these two parallel manipulators are shown in table 1 and figure 2.

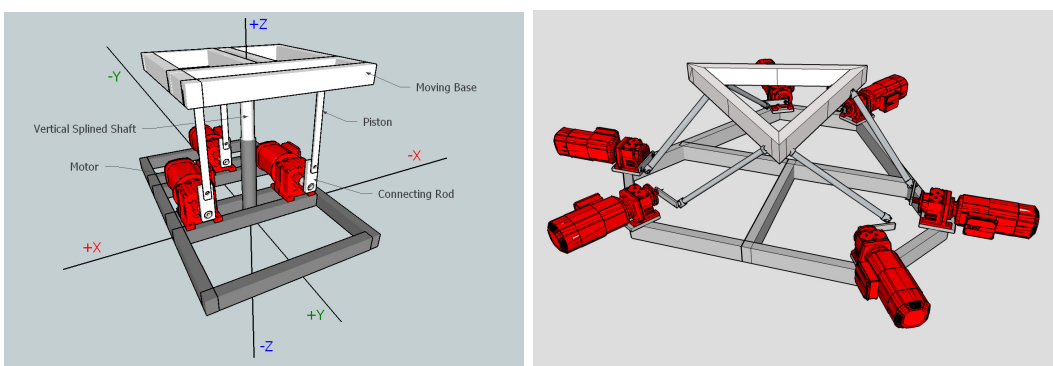


Figure 2: CAD design

As aforementioned, the first step for the simulator to work is to build a CAD model for

Table 1: DoF ranges of both real motion platforms

	3-DoF	3-DoF	6-DoF	6-DoF
	Min. Value	Max. Value	Min. Value	Max. Value
Surge range	-	-	-10.58 cm	+10.58 cm
Sway range	-	-	-9.77 cm	+13.78 cm
Heave range	-10.0 cm	+10.0 cm	-7.84 cm	+9.06 cm
Yaw range	-	-	-20.73 deg	+20.73 deg
Pitch range	-30.52 deg	+30.52 deg	-17.45 deg	+16.33 deg
Roll range	-26.21 deg	+26.21 deg	-17.58 deg	+17.58 deg

both of the manipulators. The restrictions of the CAD model for the simulator to work were explained on section 3.1. In this case, both of the designs are complaint with the required structure. The construction of the CAD model cannot be automated and few guidelines can be explained. The most important rule is to place every motor, every rigid object and every joint as in the real design. This is done using 3dsMax with the PhysX plugin [3]. Then, the last step is to choose the appropriate type of joint for each linkage. A full description of the joint types supported by NVidia PhysX is shown at [15]. Once the CAD model is correctly built, the next step is to set-up the parameters of the simulator. We can split this process in two parts. The first one is related with the physics magnitudes of the elements of the manipulator (dimensions, masses, etc). The second one is the tuning of the motors.

The physics magnitudes include positions, dimensions, masses and inertias of the motion platform elements. Positions and dimensions are defined within the CAD model. Masses can be set directly from the simulator via XML files. Inertias are computed internally by measuring the objects volumes (assuming a constant density situation). All these magnitudes were set to their corresponding real values, which can be seen in table 2. As positions and dimensions are implicitly defined within the CAD model, we only show the masses. The load is a variable parameter that can be set to arbitrarily. In these tests, we assume a load of 100.0 Kg with a boxed shape of $1m^3$.

The second part of the set-up includes the tuning of the virtual motors. This involves 4 parameters: motor maximum torque and the PID parameters: Kp, Ki, and Kd. The maximum torque is a feature of the motor, whose value is taken from the real one. This value includes the motor-reduction unit, and establishes a limit for the torque the motors can supply to the connecting rods. The tuning of the PID controller implies finding appropriate values for Kp, Ki and Kd. There are many methods to do this, but we opted for an empirical tuning [1] . For the sake of shortness we show only the final values (see Table 2).

 ISBN:978-84-615-5392-1

Table 2: Physics and motor parameters for both virtual motion platforms

Parameter	3-DoF	6-DoF
Motor mass	100.0 Kg	100.0 Kg
Connecting rod mass	5.0 Kg	5.0 Kg
Piston mass	1.0 Kg	1.5 Kg
Moving base mass	20.0 Kg	30.0 Kg
Splined outer shaft mass	20.0 Kg	-
Splined inner shaft mass	10.0 Kg	-
Load mass	100.0 Kg	100.0 Kg
Kp	11.2	10.1
Ki	1.35	1.53
Kd	0.19	0.27
Maximum torque	150.0 Nm	200.0 Nm

4.2 Simulator assessment

Once the simulator is set-up, we can start the assessment process. This process is a comparison between the motion platform simulator and its real counterpart. The most common way to do so is to compare them DoF by DoF separately with some test input signals [18]. Many different test signals can be used, but the most common ones are the sine and the step signals. A sine signal provides information on smooth motion, while the step signal provides information on sudden motion. The problem with these signals is that they provide a test in time domain, while no information of the frequency domain can be analyzed (unless we perform many different tests with different frequencies). To solve this, we decided to use the sine-chirp signal (a sine with increasing frequency) and the square-chirp (a square wave with increasing frequency). This way, we can compare the motion platforms both in time and frequency domain, at the same time. The last decision is the amplitude of the signals. Although the response of the motion platform is different from large displacements to small ones (because large displacements require greater speeds) it is interesting to test the performance of the compared motion platforms with respect to different amplitudes. However, by using chirp functions we are already demanding different speeds to the motion platforms and we can see the effect of a speed change on the platform behavior. Thus, we will use chirp functions of approximately maximum amplitudes for each DoF.

The chirp functions are defined as follows:

$$\text{sine-chirp}(t) = A \cdot \sin(2 \cdot \pi \cdot f_0(r^t - 1)/\log(r)) \tag{1}$$

$$\text{square-chirp}(t) = A \cdot sign(\sin(2 \cdot \pi \cdot f_0(r^t - 1)/\log(r))) \quad (2)$$

Where t is time, A is the amplitude of the signal, f_0 is its initial frequency, and r is the rate of change of the signal frequency.

For these tests, the rate of change was set to 1.2 and the initial frequency to 0.01 Hz. All the tests were performed using an external test software that feeds the inputs to the motion platform and then, reads its outputs. This test software needs to calculate the inverse kinematics of the motion platform, because the inputs of the tests are DoF signals, and the inputs of the motion platform are motor angles. However, for the sake of brevity, we cannot describe the inverse kinematics of the manipulator.

Figure 3 shows the results of plotting the response of both real motion platforms versus the simulated ones, given the same inputs. For the sake of shortness we only show heave (Z translation) for the T1R2, and pitch angle for the T3R3.

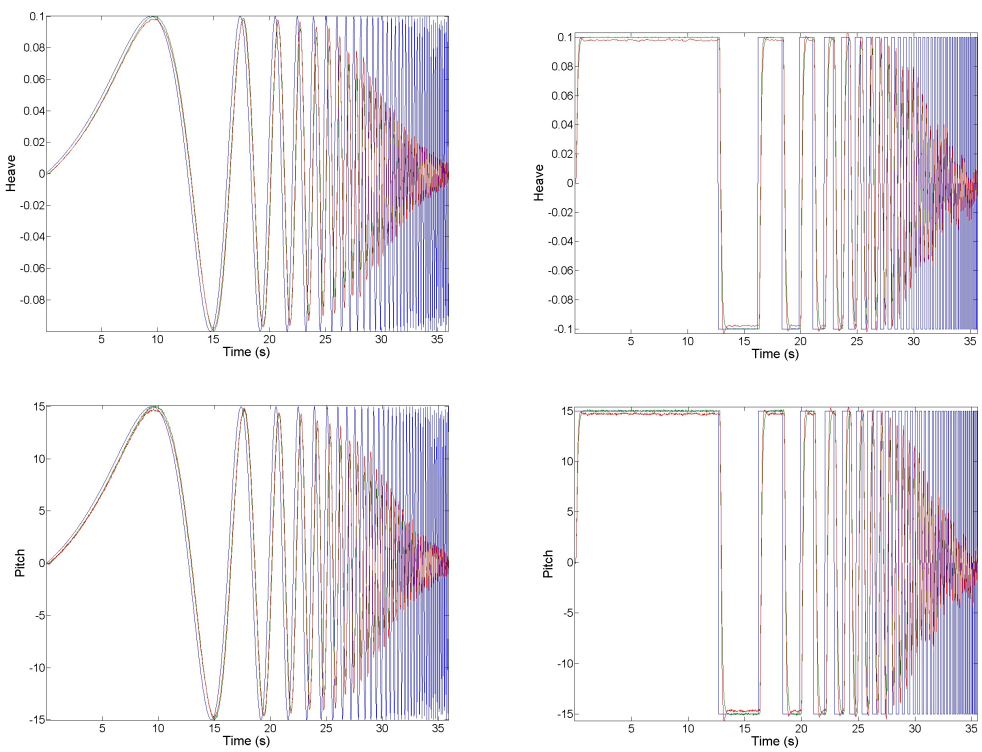


Figure 3: T1R2 heave (above) and T3R3 pitch (below) response (blue: input, green: real, red: simulated)

As we can see, in both cases the simulated and real outputs are quite similar. Moreover, their frequency response is similar. Both tend to have a progressive gain attenuation as the frequency is raised. This shows the typical low-pass frequency behavior present on many mechanical systems. The effect is somehow different for each DoF, but in all cases a sharp attenuation is found above 2 Hz. In both cases of study, and in all DoF, the simulated output is not more than 3% greater or lower, on average, than the real output, which means that both virtual motion platforms are very similar to the real ones.

As the motion platform simulator is intended to run in real-time, it is also necessary to demonstrate that this can be accomplished. However, this assertion depends highly on the computer in which the simulator runs. For this reason, the only thing than we can show is that real-time constraints are met with a particular hardware. Thus, we tested the motion platform simulator (with the parameter set-up explained before) with the T1R2 motion platform simulator on an Intel Core 2 Quad at 2.66 Ghz with 4 Gb of RAM, and with a Windows 7 operating system. The frequency of the draw thread was set to 25 Hz, the frequency of the communication thread to 50 Hz, and the frequency of the physics thread to 500 Hz, which is a common frequency for physics-based simulations. The simulator was run for 600 seconds and average times were measured. The time measurements showed that 1/500 seconds (2 ms) could be simulated in about 0.1 ms, which implies that the real-time constraints are easily met, and, if we want this simulation to run faster than real-time, the acceleration factor could reach up to approximately 20x. This result means that, with this hardware, we can test the motion platform 20 times faster than with a real motion platform. The same test was performed for the T3R3, and the acceleration factor was around 14x, mainly because the more joints, the more time it takes to simulate them.

5 Conclusions and future work

In this paper, we present a generic motion platform simulator. This simulator allows us to perform multiple tests without the need to use a real motion platform, avoiding damages on real motion platforms and humans. In order to assess our simulator, we have tested it with two different manipulators. A numerical evaluation was performed by comparing the outputs of our virtual motion platforms with respect to the real ones. In both cases, the results show that our simulator is able to substitute their real counterparts as the difference between real and virtual outputs (given the same inputs) is small. To prove the real-time features, computation times were measured, and although they depend highly on the hardware used to perform the tests, acceleration factors around 20x were reached using a fairly standard PC.

Although we consider that the simulator fulfills the goals by which it was designed, there are several research lines that can be followed in order to extend its functionality. The first one is the automation of the design process of the virtual motion platform. At the

present moment, this is a manual task guided by experts. If this mapping could be done automatically, a lot of time could be saved. Another research line is to visually improve the simulator. At the present moment, the simulator visual outputs are sufficient to evaluate the motion platform behavior. However, a more realistic visualization could enhance the understanding of the motion platform. Moreover, additional information can be added to the visual representation. For instance, we could add text to show the motor angles, the DoF, the acceleration factor, etc. Another possible improvement is to implement different virtual controllers, and not only a PID. Finally, we could perform processing optimizations by, for instance, executing simulator tasks on Graphics Processing Units (GPU), in order to increment the simulator performance and achieve greater acceleration factors.

References

[1] K. J. Astrom, T. Hagglund, *PID Controllers: Theory, Design and Tuning*, Instrument Society of America, Research Triangle Park (1995).

[2] K. J. Astrom, T. Hagglund, *Revisiting the ZieglerNichols step response method for PID control*, Journal of Process Control **14** (2004) 635–650.

[3] Autodesk, *3d Studio Max home page*, Retrieved April 12, 2012 from http://usa.autodesk.com/3ds-max/.

[4] J. Fung, F. Malouin, B. J. McFadyen, F. Comeau, A. Lamontagne, S. Chapdelaine, C. Beaudoin, D. Laurendeaum , L. Hugheym, C. L. Richardsm, *Locomotor rehabilitation in a complex virtual environment*, International Conference of the IEEE EMBS (2004) 4859–4861.

[5] C. Gosselin, L. Perreault, C. Vaillancourt, *Simulation and Computer-Aided Design of Spherical Parallel Manipulators*, IEEE International Conference on Robotics and Automation (1994) 145–151.

[6] H. Hajimirzaalian, H. Moosavi, M. Massah, *Dynamics analysis and simulation of parallel robot stewart platform*, Computer and Automation Engineering **5** (2010) 472–477.

[7] K. F. Hulme, A. Pancotti, *Development of a virtual 6 D.O.F motion platform for simulation and rapid synthesis*, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (2004) 19–22.

[8] H. Lau, L. Chan, R. Wong *A virtual container terminal simulator for the design of terminal operation*, International Journal on Interactive Design and Manufacturing (2007) 107–113.

[9] Y-W. Li, J. Wang, L-P. Wang, X-J. Liu, *Inverse dynamics and simulation of a 3-DOF spatial parallel manipulator*, IEEE international conference on robotics and automation (2003) 4092–4097.

[10] Mathworks, *Matlab home page*, Retrieved April 12, 2012 from http://www.mathworks.es/products/matlab/.

[11] J. P. Merlet, *Parallel manipulators - part I: theory, design, kinematics, dynamics and control*, Technical report 646, INRIA, Cedex, France (1987).

[12] Moog, *Moog 6 DOF 2000E Motion System User's Manual*, Part No.: c37970, Doc No.: LSF-0462 (2000).

[13] Mscsoftware, *Adams home page*, Retrieved April 12, 2012 from http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx.

[14] M. A. Nahon, L. D. Reid, *Simulator motion-drive algorithms: A designer's perspective*, Journal of Guidance **3** (1989) 356–362.

[15] Nvidia, *Nvidia Physx home page*, Retrieved April 12, 2012 from http://developer.nvidia.com/physx.

[16] OpenSceneGraph, *OpenSceneGraph home page*, Retrieved April 12, 2012 from http://www.openscenegraph.org/projects/osg.

[17] L. D. Reid, M. A. Nahon, *Flight Simulation Motion-Base Drive Algorithms, Part 2: Selecting the System Parameters*, Univ. Toronto (1986).

[18] G. Reymond, A. Kemeny, *Motion cueing in the Renault driving simulator*, Vehicle System Dynamics (2000) 249–259.

[19] A. Selvakumar, R. Sivaramakrishnan, S. Karthik, T. V. Valluri, S. Ramakrishna, S., B. Vinodh, *Simulation and workspace analysis of a tripod parallel manipulator*, World Academy of Science, Engineering and Technology **57** (2009).

[20] D. Stewart, *A platform with six degrees of freedom*, Institution of Mechanical Engineers **15** (1965) 371–384.

[21] M. White, P. Perfect, G. Padfield, A. Gubbels, A. Berryman, *Acceptance testing and commissioning of a flight simulator for rotorcraft simulation fidelity research*, Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering (2012).

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Sobolev orthogonal polynomials on the unit circle: Hessenberg matrices and zeros

Kenier Castillo[1], **Luis E. Garza**[2] **and Francisco Marcellán**[1]

[1] *Departamento de Matemáticas, Universidad Carlos III de Madrid, Spain*

[2] *Facultad de Ciencias, Universidad de Colima, Mexico*

emails: kcastill@math.uc3m.es, garzaleg@gmail.com, pacomarc@ing.uc3m.es

Abstract

In this contribution, we study some analytic properties of families of polynomials orthogonal with respect to the Sobolev inner product

$$\langle f, g\rangle_S := \int_{\mathbb{T}} f(z)\overline{g(z)}d\mu(z) + \lambda f^{(j)}(\alpha)\overline{g^{(j)}(\alpha)},$$

where μ is a nontrivial probability measure supported on the unit circle, $\alpha \in \mathbb{C}$, $\lambda \in \mathbb{R}^+\backslash\{0\}$, and $j \in \mathbb{N}$. We focus our attention on the relative asymptotic behavior of such polynomials with respect to the polynomials orthogonal associated with the measure μ, as well as on the distribution of their zeros in terms of the parameters n, λ and α. In this second problem our approach is based on the characterization of the zeros as eigenvalues of rank-one perturbations of the Hessenberg matrix (GGT matrix) associated with the measure μ. Finally, some numerical computations for specific examples are shown.

Key words: Sobolev orthogonal polynomials, Hessenberg matrices, zeros
MSC 2000: 42C05

1 Introduction

Let μ be a nontrivial probability measure supported on $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$ and $\{\phi_n\}_{n\geqslant 0}$ the sequence of polynomials with $\deg \phi_n = n$ such that

$$\int_{\mathbb{T}} \phi_m(z)\overline{\phi_n(z)}d\mu(z) = \delta_{m,n},$$

i.e., $\{\phi_n\}_{n\geqslant 0}$ is the sequence of orthonormal polynomials with respect to the measure μ (see [3], among others). The corresponding sequence of monic orthogonal polynomials will be denoted by $\{\Phi_n\}_{n\geqslant 0}$. The polynomial

$$K_n(z,y)=\sum_{k=0}^{n}\frac{\Phi_k(z)\overline{\Phi_k(y)}}{\|\Phi_k\|^2}$$

is called the reproducing kernel associated to $\{\phi_n\}_{n\geqslant 0}$. We will denote by $K_n^{(k,j)}(z,y)$ the $k-th$ and $j-th$ partial derivatives of $K_n(z,y)$ with respect to the variables z and y, respectively.

In this short paper we will consider the following discrete Sobolev inner product associated with a nontrivial probability measure μ supported on the unit circle

$$\langle f,g\rangle_S:=\int_{\mathbb{T}} f(z)\overline{g(z)}d\mu(z)+\lambda f^{(j)}(\alpha)\overline{g^{(j)}(\alpha)},\ \alpha\in\mathbb{C},\ \lambda\in\mathbb{R}^+\backslash\{0\},\ j\in\mathbb{N}, \tag{1}$$

where f,g belong to the Sobolev space

$$W^{j,2}\left[\mathbb{T};\mu\right]=\left\{f\in C_j(\mathbb{T})\cap L^2\left[\mathbb{T};\mu\right]:f^{(j)}\in L^2\left[\mathbb{T};\mu\right]\right\}.$$

Here $C_j(\mathbb{T})$ denotes the function space containing all functions $f:\mathbb{T}\to\mathbb{C}$ such that $f\in C^{j-2}$ and $f^{(j-1)}$ is absolutely continuous on $\mathbb{T}$. Notice that, since λ is a positive real number, there exists a family of polynomials orthonormal with respect to (1), which will be denoted by $\{\psi_n\}_{n\geqslant 0}$. The monic version will be denoted by $\{\Psi_n\}_{n\geqslant 0}$.

The structure of the manuscript is as follows. In Section 2, we show the connection between the Hessenberg matrices associated with the multiplication operator in terms of the bases $\{\Psi_n\}_{n\geqslant 0}$ and $\{\Phi_n\}_{n\geqslant 0}$, respectively. In Section 3, we study the location of the zeros of Ψ_n.

2 Hessenberg matrix

The matrix representation of the multiplication by the z operator with respect to the basis $\{\Phi_n\}_{n\geqslant 0}$ is given by $z\mathbf{\Phi}=\mathbf{H}_\Phi\mathbf{\Phi}$, where $\mathbf{\Phi}=\{\Phi_0,\Phi_1,\ldots\}^t$ and $\mathbf{H}_\Phi$ is a semi-infinite Hessenberg matrix with ones on the upper diagonal and whose remaining entries are given in terms of the Verblunsky coeficients $\{\Phi_n(0)\}_{n\geqslant 1}$ (see [3]). The study of the Hessenberg matrices is important because of their applications. As an example, it is well known (see [3]) that the zeros of the n-th orthogonal polynomial $\Phi_n(z)$ are the eigenvalues of the leading principal submatrix $n\times n$ of the Hessenberg matrix $\mathbf{H}_\Phi$, which we will denote by $(\mathbf{H}_\Phi)_n$.

The relation between $\{\phi_n\}_{n\geqslant 0}$ and $\{\psi_n\}_{n\geqslant 0}$ is (see [1])

Lemma 1 [1] *Let $\phi_n(z)=\alpha_n z^n+\ldots$ and $\psi_n(z)=\beta_n z^n+\ldots$ with $\alpha_n,\beta_n>0$. Then, $\{\psi_n\}_{n\geqslant 0}$ is the sequence of polynomials orthonormal with respect to* (1) *if and only if*

$$\psi_n(z)=\frac{\beta_n}{\alpha_n}\phi_n(z)-\lambda\psi_n^{(j)}(\alpha)K_{n-1}^{(0,j)}(z,\alpha),\qquad j=0,1,\ldots, \tag{2}$$

with

$$\frac{\beta_n}{\alpha_n} = \sqrt{\frac{1+\lambda K_{n-1}^{(j,j)}(\alpha,\alpha)}{1+\lambda K_n^{(j,j)}(\alpha,\alpha)}}, \tag{3}$$

and

$$\psi_n^{(j)}(\alpha) = \frac{\phi_n^{(j)}(\alpha)}{\sqrt{\left(1+\lambda K_{n-1}^{(j,j)}(\alpha,\alpha)\right)\left(1+\lambda K_n^{(j,j)}(\alpha,\alpha)\right)}}. \tag{4}$$

When λ tends to infinity, we get [2] the limit polynomial

$$S_n(z) = \Phi_n(z) - \frac{\Phi_n^{(j)}(\alpha)}{K_{n-1}^{(j,j)}(\alpha,\alpha)} K_{n-1}^{(0,j)}(z,\alpha). \tag{5}$$

It is easily seen that $S_n^{(j)}(\alpha) = 0$, as well as S_n is orthogonal to the linear space $span\{1, z-\alpha, \ldots, (z-\alpha)^{j-1}, (z-\alpha)^{j+1}, \ldots, (z-\alpha)^{n-1}\}$ in $\mathbb{P}_n$. We have proved the following extremal characterization for the limit polynomial.

Theorem 2 [2] *Let*

$$G_n = \min\left\{\int_{\mathbb{T}} |Q_n(z)|^2 d\mu(z) \quad : \quad Q_n = z^n + \textit{lower terms}, \quad Q_n^{(j)}(\alpha) = 0\right\}.$$

Then, $G_n = \|S_n\|^2$, where S_n is the limit polynomial defined by (5)*.*

In following theorem we show the relation between $\mathbf{H}_\Psi$, the Hessenberg matrix associated with the monic orthogonal polynomials $\{\Psi_n\}_{n\geqslant 0}$, and $\mathbf{H}_\Phi$.

Theorem 3 [2] *Let $(\mathbf{H}_\Phi)_n$ and $(\mathbf{H}_\Psi)_n$ be the $n \times n$ truncated Hessenberg matrices associated with $\{\Phi_n\}_{n\geqslant 0}$ and $\{\Psi_n\}_{n\geqslant 0}$, respectively. Then,*

$$(\mathbf{H}_\Psi)_n = \mathbf{L}_n[(\mathbf{H}_\Phi)_n - \mathbf{A}_n]\mathbf{L}_n^{-1}.$$

As a consequence, the zeros of Ψ_n are the eigenvalues of the matrix $(\mathbf{H}_\Phi)_n - \mathbf{A}_n$, a rank one perturbation of the matrix $(\mathbf{H}_\Phi)_n$, where

$$\mathbf{A}_n = \begin{pmatrix} 0 & \ldots & \ldots & 0 \\ \vdots & & & \vdots \\ 0 & \ldots & \ldots & 0 \\ l_{n,0} & \ldots & \ldots & l_{n,n-1} \end{pmatrix}$$

and

$$l_{m,k} = -\frac{\lambda \Phi_m^{(j)}(\alpha)\overline{\Phi_k^{(j)}(\alpha)}}{(1+\lambda K_{m-1}^{(j,j)}(\alpha,\alpha))\|\Phi_k\|^2}, \quad 1 \leqslant m \leqslant n, \quad 0 \leqslant k \leqslant m-1.$$

 ISBN:978-84-615-5392-1

3 Zeros

In this section, we analyze the behavior of the zeros of the polynomials orthogonal with respect to (1). Denote by $\{\phi_n(z; d\mu_{j+1})\}_{n\geqslant 0}$ the corresponding sequence of monic orthogonal polynomial with respect to

$$d\mu_j = |z-\alpha|^{2(j+1)} d\mu, \qquad j \in \mathbb{N}.$$

We have the following results.

Theorem 4 [2] *There is a positive integer n_0 such that, for $n \geqslant n_0$, the n-th Sobolev monic orthogonal polynomial $\psi_n(z)$ defined by* (2)*, with $|\alpha| > 1$, has exactly* 1 *zero in $|z| > 1$ accumulating in α, while the remaining zeros belong to $|z| < 1$.*

Theorem 5 [2] *Let $\{\psi_n\}_{n\geqslant 0}$ be the sequence of orthonormal polynomials with respect to* (1)*, with $j \in \mathbb{N}$. Then $\psi_n(z)$ can be expressed as a linear combination of $\phi_n(z)$, $(z-\alpha)\phi_{n-1}(z, d\mu_1)$, ..., $(z-\alpha)^{j+1}\phi_{n-j-1}(z, d\mu_{j+1})$. As a consequence, the zeros of $\psi_n(z)$ tend to the zeros of such a linear combination when $\lambda \to \infty$.*

Acknowledgements

The work of the first and third authors was supported by Dirección General de Investigación, Ministerio de Ciencia e Inovación of Spain, grant MTM2009-12740-C03-01. The work of the second author was supported by Consejo Nacional de Ciencia y Tecnología of México, grant 156668, and Promep.

References

[1] K. CASTILLO, L. E. GARZA, AND F. MARCELLÁN, Asymptotic behavior of Sobolev orthogonal polynomials on the unit circle, *Integral Transforms Spec. Funct.,* (2012), DOI:10.1080/10652469.2011.649751.

[2] K. CASTILLO, L. GARZA, AND F. MARCELLÁN, Zeros of Sobolev orthogonal polynomials on the unit circle, *Numer. Algorithms*, (2012). In press.

[3] B. SIMON, Orthogonal polynomials on the unit circle, 2 vols. *Amer. Math. Soc. Coll. Publ. Series*, **54**, Amer. Math. Soc. Providence, Rhode Island, 2005.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Analytical solvent accessible surface area calculation on GPUs

Eduardo Cepas-Quiñonero[1], Patrice Koehl[2], Horacio Pérez-Sánchez[1] and José M. García[1]

[1] *Computer Engineering Department, University of Murcia, Spain*

[2] *Department of Computer Science and Genome Center, University of California, Davis, California 95616, USA*

emails: ecepasqui@ditec.um.es, koehl@cs.ucdavis.edu, horacio@ditec.um.es, jmgarcia@ditec.um.es

Abstract

It is very important in drug discovery to determine the safety and effectiveness of current drugs and to accelerate findings in basic research (discovery of new leads and active compounds) into meaningful health outcomes. Both objectives imply the capacity to process the vast amount of protein structural data that are available in biological databases such as the PDB [1] or derived from genomic data using techniques such as homology modeling [9]. Screenings in lab and compound optimization are expensive and slow methods, but bioinformatics can significantly help clinical research by providing prediction of the toxicity of drugs and activity in non-tested targets, and by evolving discovered active compounds into drugs for the clinical trials. This can be achieved thanks to the availability of bioinformatics tools and Virtual Screening (VS) methods that allow to test all required hypotheses before clinical trials. Often however, VS methods fail to make good toxicity and activity predictions since they are constrained by the access to computational resources; even the nowadays fastest VS methods cannot process large biological databases in a reasonable time-frame. Thus, this imposes a serious limitation in many areas of translational research.

The Graphics Processing Unit (GPU) is a topic of significant interest in high performance computing. For applications that can benefit from parallelization, GPUs deliver higher peak computational throughput than latency-oriented CPUs, thus offering a tremendous potential performance uplift on massively parallel problems [4]. Of particular relevance to us are attempts to parallelize different kernels within VS methods on GPUs to allow for the introduction of improvements in the biophysical models that were not amenable in the past [5, 8].

Among the most relevant computationally intensive kernels present in current VS methods, we highlight the calculation of solvent accessible surface area (SASA) of a biomolecule. We can efficiently model molecular solvation in an implicit way by the calculation of SASA and posterior consideration of the hydrophobic and hydrophilic character of individual atoms [3]; this approach is widely applied nowadays in many areas like protein structure prediction and protein-ligand binding.

In order to improve the predictive capability of VS methods, it is necessary to use an exact method for the SASA calculation. Analytical methods describe the molecule as a union of pieces of balls, each defined by their center, radius, and arcs forming their boundary, and subsequently apply analytical geometry to compute the surface area and volume. The alpha shape theory solves this problem using Delaunay triangulations and their filtrations, as described by Edelsbrunner [2], and has been implemented for such purpose in the program UNIONBALL [6]. In order to speedup the necessary calculations and to reduce total running time, we present here our parallelization efforts of UNIONBALL, taking advantage of the last generation of massively parallel Graphics Processing Units (GPUs) and using the CUDA programming language [7].

Key words: Parallel Computing, GPUs, CUDA, Computational Geometry, Molecular Modeling

Acknowledgements

This research was supported by the Fundación Séneca (Agencia Regional de Ciencia y Tecnología, Región de MURCIA) under grant 15290/ PI/2010, by the Spanish MEC and European Commission FEDER under grants CSD2006-00046 and TIN2009-14475-C04 and a postdoctoral contract from the University of MURCIA (30th December 2010 resolution).

References

[1] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Res*, 28:235–242, 2000.

[2] H Edelsbrunner. The Union of Balls and Its Dual Shape. *Discrete & Computational Geometry*, 13:415–440, 1995.

[3] D Eisenberg and A D McLachlan. Solvation energy in protein folding and binding. *Nature*, 319(6050):199–203, 1986.

[4] Michael Garland, Scott Le Grand, John Nickolls, Joshua Anderson, Jim Hardwick, Scott Morton, Everett Phillips, Yao Zhang, and Vasily Volkov. Parallel computing experiences with cuda. *IEEE Micro*, 28:13–27, July 2008.

[5] G.D. Guerrero, H.E. Pérez-Sánchez, J.M. Cecilia, and J.M. García. Parallelization of virtual screening in drug discovery on massively parallel architectures. In *Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on*, pages 588 –595, 2012.

[6] Paul Mach and Patrice Koehl. Geometric measures of large biomolecules: surface, volume, and pockets. *Journal of Computational Chemistry*, 32(14):3023–3038, November 2011.

[7] NVIDIA. *NVIDIA CUDA C Programming Guide 4.1.1.* 2011.

[8] Horacio Perez Sanchez and Wolfgang Wenzel. Optimization methods for virtual screening on novel computational architectures. *Current Computer-Aided Drug Design*, 7(1):44–52, 2011.

[9] Roberto Sanchez and Andrej Sali. Large-Scale Protein Structure Modeling of the Saccharomyces cerevisiae Genome. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 95(23):13597–13602, November 1998.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A family of optimal fourth-order iterative methods and its dynamics

Francisco Chicharro[1], Alicia Cordero[1] and Juan R. Torregrosa[1]

[1] *Instituto de Matemáticas Multidisciplinar, Universitat Politècnica de València, Camino de Vera, s/n, 46022 València, Spain*

emails: frachilo@teleco.upv.es, acordero@mat.upv.es, jrtorre@mat.upv.es

Abstract

In this paper a family of new fourth-order optimal iterative methods for solving nonlinear equations is proposed. The classical King's family of fourth-order schemes is obtained as an special case. We also present results for describing the conjugacy classes and dynamics of some of the presented methods for complex polynomials of different degrees.

Key words: Iterative methods, order of convergence, rational map, basin of attraction, conjugacy classes.

1 Introduction

In this paper, we consider iterative methods and their dynamics for finding a simple root α of a nonlinear equation $f(x) = 0$. Newton's method [1] is the best known scheme for solving nonlinear equations, which is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

This method converges quadratically under some conditions.

To improve the local order of convergence, many modified methods have been proposed in the literature. For example, it is well-known (see [1]) that the following method, obtained from the doubled Newton scheme with "frozen" derivative

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(x_n)}, \tag{1}$$

where y_n is the Newton's iteration, has order of convergence three. This method requires three functional evaluations per step, so it is not an optimal method. By optimal method we mean a multipoint one without memory which requires $n+1$ functional evaluations per iteration, but achieves the order of convergence 2^n (see [2]).

In this manuscript we show a family of optimal fourth-order iterative methods derived by using in (1) the technique of weight functions. Some applications of this technique can be found for example in [3] and [4], where some kind of weight functions were used.

The dynamical study of the rational function associated to an iterative method gives important information about the convergence and stability of the scheme. The best known iterative method, under the dynamical point of view, is again Newton's scheme (see, for example, [5]). The dynamics of the K*ö*nig iteration methods [6], the Cauchy and Halley's methods [7] and an important number of root-finding methods including Jarratt and King families [8] were also studied in detail.

We present results which describe the conjugacy classes and dynamics of some new optimal fourth order methods for complex polynomials of degree two, three and four. The fact that one of our methods is not generally convergent for polynomials is investigated by constructing a particular polynomial such that the rational map arising from our method applied to this polynomial has an attracting periodic orbit of period 2. The basins of attraction of some elements of our family are presented. In order to do this, we recall some concepts.

Given a rational map $R:\hat{\mathbb{C}}\to\hat{\mathbb{C}}$, where $\hat{\mathbb{C}}$ is the Riemann sphere, the *orbit of a point* $z_0\in\hat{\mathbb{C}}$ is defined as:

$$\{z_0,\, R(z_0),\, R^2(z_0),...,R^n(z_0),...\}.$$

We are interested in the study of the asymptotic behavior of the orbits depending on the initial condition z_0, that is, we are going to analyze the phase plane of the map R defined by the different iterative methods.

To obtain these phase spaces, the first of all is to classify the starting points from the asymptotic behavior of their orbits. A $z_0\in\hat{\mathbb{C}}$ is called a *fixed point* if it satisfies: $R(z_0)=z_0$. A *periodic point* z_0 of period $p>1$ is a point such that $R^p(z_0)=z_0$ and $R^k(z_0)\neq z_0$, $k<p$. A *pre-periodic point* is a point z_0 that is not periodic but there exists a $k>0$ such that $R^k(z_0)$ is periodic. A *critical point* z_0 is a point where the derivative of rational function vanishes, $R'(z_0)=0$.

On the other hand, a fixed point z_0 is called *attractor* if $|R'(z_0)|<1$, *superattractor* if $|R'(z_0)|=0$, *repulsor* if $|R'(z_0)|>1$ and *parabolic* if $|R'(z_0)|=1$. The stability of a periodic orbit is defined by the magnitude (lower than 1 or not) of $|R'(z_1)\dots R'(z_p)|$, where $\{z_1,\dots,z_p\}$ are the points of the orbit of period p.

The basin of attraction of an attractor $\bar{z}$ is defined as the set of pre-images of any order:

$$\mathcal{A}(\bar{z})=\{z_0\in\hat{\mathbb{C}}\ :\ R^n(z_0)\to\bar{z},\ n\to\infty\}.$$

The set of points $z \in \hat{\mathbb{C}}$ such that their families $\{R^n(z)\}_{n\in\mathbb{N}}$ are normal in some neighborhood $U(z)$, is the *Fatou set,* $\mathcal{F}(R)$, that is, the Fatou set is composed by the set of points whose orbits tend to an attractor (fixed point, periodic orbit or infinity). Its complement in $\hat{\mathbb{C}}$ is the *Julia set,* $\mathcal{J}(R)$; therefore, the Julia set includes all repelling fixed points, periodic orbits and their pre-images. That means that the basin of attraction of any fixed point belongs to the Fatou set. On the contrary, the boundaries of the basins of attraction belong to the Julia set.

2 The methods and analysis of convergence

Based on (1), we consider the following two-step iteration scheme by using the weight functions technique:

$$\begin{array}{rcl} y_n & = & x_n - \beta \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} & = & y_n - H(\mu(x_n)) \frac{f(y_n)}{f'(x_n)}, \end{array} \tag{2}$$

where β is a real parameter, $H(t)$ represent a real-valued function and $\mu(x) = \dfrac{f(y)}{b_1 f(x) + b_2 f(y)}$, being b_1 and b_2 real parameters.

We show that without adding new functional evaluations, we increase the order of convergence to four. In the following theorem we prove that the method defined by (2) is of order 4 under some conditions on H.

Theorem 1 *Let $\alpha \in I$ be a simple zero of a sufficiently differentiable function $f : I \subseteq \mathbb{R} \to \mathbb{R}$ in an open interval I. Let H be any function with $H(0) = 1$, $H'(0) = 2b_1$ and $|H''(0)| < \infty$ and $\beta = 1$. Then the methods defined by (2), with $b_1 \neq 0$ and b_2 arbitrary real parameters, have fourth-order convergence, and its error equation is*

$$e_{n+1} = \left(\frac{\left(10b_1^2 + 4b_1b_2 - H''(0)\right) c_2^3 - 2b_1^2 c_2 c_3}{2b_1^2} \right) e_n^4 + O(e_n^5).$$

where $c_k = (1/k!)\frac{f^{(k)}(\alpha)}{f'(\alpha))}$, $k = 1, 2, \ldots$.

Some well-known methods are included in the family (2). For example, taking $H(t) = 1 + 2t$, and $b_2 = -2$ we obtain the Ostrowski's method, whose iterative expression is

$$\begin{array}{rcl} y_n & = & x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} & = & y_n - \frac{f(x_n)}{f(x_n)-2f(y_n)} \frac{f(y_n)}{f'(x_n)}. \end{array}$$

 ISBN:978-84-615-5392-1

On the other hand, if we take $H(t) = 1 + 2t$ and parameter b_2 remains arbitrary, we have the King's family (see [9])

$$\begin{array}{rcl} y_n & = & x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} & = & y_n - \frac{f(x_n)+(2+b_2)f(y_n)}{f(x_n)+b_2 f(y_n)} \frac{f(y_n)}{f'(x_n)}. \end{array}$$

In what follows, we give some concrete forms of iterative schemes (2).

Example 1 Taking function $H(t) = 1 + t$, $b_1 = 1/2$ and $b_2 = 0$ we obtain the iterative root-finding scheme

$$\begin{array}{rcl} y_n & = & x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} & = & x_n - \frac{f(x_n)^2+f(x_n)f(y_n)+2f(y_n)^2}{f(x_n)f'(x_n)}. \end{array} \tag{3}$$

Example 2 The function $H(t) = \frac{4}{4-2t-t^2}$, $b_1 = 1/4$ and $b_2 = 1/4$ satisfy the conditions of Theorem 1. A new fourth-order method is then obtained

$$\begin{array}{rcl} y_n & = & x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} & = & y_n - \frac{(f(x_n)+f(y_n))^2}{f(x_n)^2-5f(y_n)^2} \frac{f(y_n)}{f'(x_n)}. \end{array} \tag{4}$$

Example 3 In the case that $H(t) = 1 + t/2 + t^2/2$, $b_1 = 1/4$ and $b_2 = 0$ we obtain another optimal fourth-order method

$$\begin{array}{rcl} y_n & = & x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} & = & y_n - \left(1 + \frac{2f(y_n)}{f(x_n)} + \frac{8f(y_n)^2}{f(x)^2}\right) \frac{f(y_n)}{f'(x_n)}. \end{array} \tag{5}$$

3 Dynamical behavior

It has been proven in the last section that different choices of parameters b_1 and b_2 define distinct iterative methods. In this section we will present the effect that this selection has in the dynamical behavior of the resulting schemes, at least on polynomials of low degree. In all presented cases, an Scaling Theorem can be proved.

Theorem 2 *Let f be an analytic function on the Riemann sphere, and $A(z) = az + b$, with $a \neq 0$, an affine map. If $g(z) = \lambda(f \circ A)(z)$, where $\lambda \in \mathbb{C} - \{0\}$, then the fixed point operator M_f is analytically conjugated to M_g by A, that is, $(A \circ M_g \circ A^{-1})(z) = M_f(z)$.*

By means of the scaling theorem the analysis of the dynamics of a rational function becomes simpler, as it can be reduced to the study on simpler polynomials, as $p(z) = z^2 - c$ and $q(z) = z^3 + (c-1)z - c$, in case of quadratic and cubic polynomials, respectively.

 ISBN:978-84-615-5392-1

Proposition 1 *(a) Let $p(z) = az^2 + bz + c$ be a complex polynomial, with $a \neq 0$, and $q(z) = z^2 - d$. Then there is an analytic conjugacy between R_p and R_q.*

(b) Let $p(z) = (z - z_1)(z - z_2)(z - z_3)$ be an arbitrary complex polynomial of degree three and let $q(z) = z^3 + (\lambda - 1)z - \lambda$, $\lambda \in \mathbb{C}$. Then there is an analytic conjugacy between R_p and R_q.

Now, we study the dynamics of the rational map R_f arising from the scheme (3)

$$R_f(z) = z - \frac{f(z)^2 + f(z)f(y) + 2f(y)^2}{f(z)f'(z)}, \tag{6}$$

where $y = z - f(z)/f'(z)$, applied to a generic polynomial with simple roots.

As for most iterative root-finding methods, the roots of f are superattracting fixed points of R_f. The critical points that do not correspond to roots of f are called *free critical points*. The reason why free critical points are important is due to the following classical result.

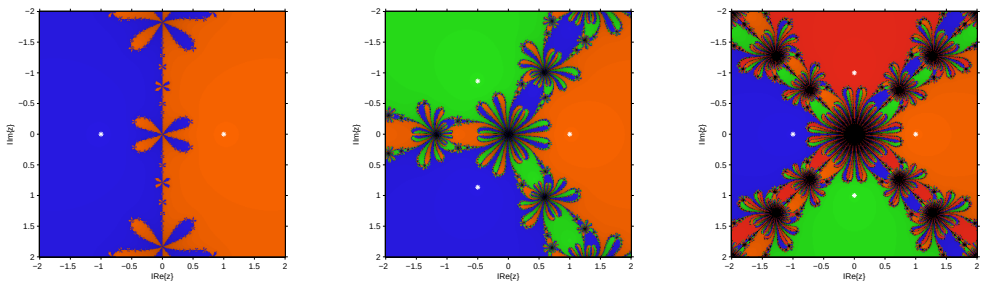
Theorem 3 *(Fatou-Julia) Let R be a rational function. Then the immediate basin of attraction of each attracting periodic point contains at least one critical point.*

As a consequence of this theorem, it is important to detect the existence of attracting periodic cycles because, in such a case there exists at least one critical point near the cycle, end the iterates of R_p starting with the critical point converge to that cycle and not to a root. To detect the existence of attracting periodic cycles, the orbits of the free critical points of R_p should be observed and the set of limit points determined.

The only attracting fixed points of $R_p(z)$ are the roots of the polynomial and these are also the critical points, so that $R_p'(z) = \dfrac{5\left(-c + x^2\right)^3}{16x^6}$. As in Newton's method, these roots are also critical points (in fact, they are the only critical points), so they are superattractive fixed points. There are also four strange fixed points, $\pm\sqrt{\frac{7c}{27} - \frac{4}{27}i\sqrt{2}c}$ and $\pm\sqrt{\frac{7c}{27} + \frac{4}{27}i\sqrt{2}c}$, which are repulsive, so they lie in Julia set.

Therefore, similarly to the Newton's method, the Fatou set consists of the basins of attraction of the two roots of the polynomial. That means that these methods never fail on quadratic polynomials when they are applied on an open set of the complex plane. The dynamical plane of this operator is similar as the one of Newton's method, but a bit more complex, as can be seen in Figure 1a for $c = 1$. In Figures 1b and 1c it can be observed that many preimages of the infinity appear with flower appearance. If more iterations are used (the images have been made by using 80 iterations as a limit), the black region in the center of each flower shrinks and petals are narrower in the center.

 ISBN:978-84-615-5392-1



(a) $p(z) = z^2 - 1$ (b) $p(z) = z^3 - 1$ (c) $p(z) = z^4 - 1$

Figure 1: Dynamical plane for (3) on some polynomials

Respect to the behavior of (3) on cubic or higher polynomials, the fixed points are the roots of a polynomial of degree at least 15 depending on the parameter c. Although no general result is concluded because of this high degree, some partial conclusions can be made, as the existence of specific polynomials whose dynamics includes periodic orbits. For the case presented in Figure 1b, that is, $p(z) = z^3 - 1$, there exist 12 strange fixed points that are repulsive and critical points are $\left\{\sqrt[3]{\frac{-185-3\sqrt{337}}{1114}}, \sqrt[3]{\frac{-185-3\sqrt{337}}{1114}}, -\sqrt[3]{-\frac{1}{1114}(-185-3\sqrt{337})}, \sqrt[3]{-\frac{185}{1114}+\frac{3\sqrt{337}}{1114}}, -\sqrt[3]{-1(-\frac{185}{1114}+\frac{3\sqrt{337}}{1114})}, \sqrt[3]{-\frac{185}{1114}+\frac{3\sqrt{337}}{1114}}\right\}$, which lie in the basin of attraction of the roots of the unity.

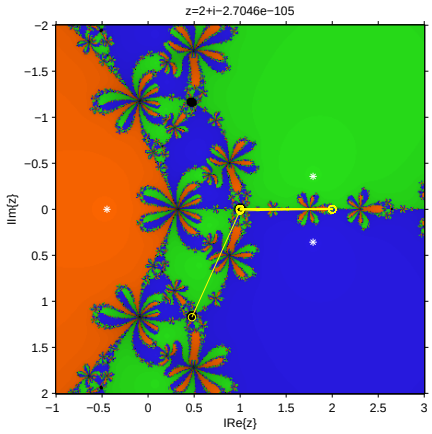
If a periodic behavior is pursued, a technique presented by Chun et al. in [10] allows us to establish the following result. For method (3) by constructing a specific polynomial $p(z)$ such that the rational map R_p applied to the polynomial has an attracting periodic orbit of period 2 at $z = 1$.

Proposition 2 *The scheme (3) is not globally convergent for cubic polynomials, as it presents a periodic orbit of period 2 for* $p(z) = z^3 - 3.13638x^2 + 1.73725x + 1.49038$.

To find these values of the coefficients of $p(z)$ a system of equations formed by the conditions $R_p(1) = 2$, $R_p(2) = 1$ and $R_p'(1) = 0$, has been solved. This is the only real polynomial that can be obtained as a solution of the system, but there exist a lot of complex polynomials verifying it. In Figure 2 this periodic orbit (with yellow lines in figure) is visualized, with black "holes" in the basin of attraction of the orbit.

When a similar study is made on (4), whose associated rational function is

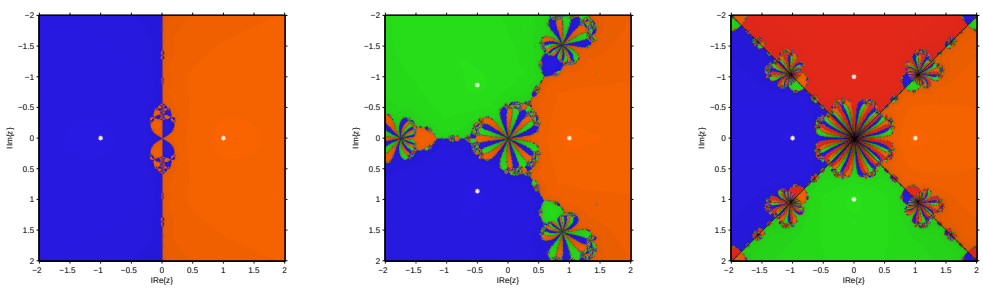
$$S_f(z) = y - \frac{(f(z)+f(y))^2}{f(z)^2 - 5f(y)^2}\frac{f(y)}{f'(z)},$$



(a) $p(z) = z^3 - 3.13638x^2 + 1.73725x + 1.49038$

Figure 2: Dynamical plane for (3) with a periodic orbit

it is observed that, for quadratic polynomials, six fixed points different from the roots can be found, $\pm\frac{i\sqrt{c}}{\sqrt{3}}$, $\pm\frac{\sqrt{3c-4\sqrt{2}c}}{\sqrt{23}}$ and $\pm\sqrt{\frac{3c}{23}+\frac{4\sqrt{2}c}{23}}$ whose character is repulsive. The free critical points are $\pm\sqrt{\frac{3}{11}}\sqrt{c}$, $\pm\frac{\sqrt{15c-8\sqrt{5}c}}{\sqrt{19}}$ and $\pm\sqrt{\frac{15c}{19}+\frac{8\sqrt{5}c}{19}}$. Moreover, the behavior of the critical points is stable, in the sense of their orbits remain in their original basin of attraction (or in Julia). Dynamical planes for low-degree polynomials can be seen in Figure 3 for roots of the unit. In Figures 3b and 3c we can see the dynamical behavior of method (4) for polynomials of degree three and four. Let us note that it seems more stable than (3) as the preimages of the infinity (black regions in the center of the flowers) are narrower. So, the



(a) $p(z) = z^2 - 1$ (b) $p(z) = z^3 - 1$ (c) $p(z) = z^4 - 1$

Figure 3: Dynamical plane for (4) on some polynomials

 ISBN:978-84-615-5392-1

dynamical analysis of the methods becomes an interesting tool in order to determine which members of a family behave better.

Acknowledgments: This research was supported by Ministerio de Ciencia y Tecnología MTM2011-28636-C02-02 and by Vicerrectorado de Investigación, Universitat Politècnica de València PAID-06-2010-2285.

References

[1] J.F. Traub, *Iterative methods for the solution of equations*, Chelsea Publishing Company, New York, 1977.

[2] H.T. Kung, J.F. Traub, *Optimal order of one-point and multipoint iteration*, J. Assoc. Comput. Math. **21** (1974) 643–651.

[3] W. Bi, H. Ren, Q. Wu, *Three-step iterative methods with eighth-order4 convergence for solving nonlinear equations*, J.Computational and Applied Mathematics **225** (2009) 105–112.

[4] X. Wang, L. Liu, *Two new families of sixth-order methods for solving nonlinear equations*, Applied Mathematics and Computation **213**(1) (2009) 73–78.

[5] P. Blanchard, *The dynamics of Newton's method*, Proc. of Symposia in Applied Math. **49** (1994) 139–154.

[6] V. Dracopoulos, *How is the dynamics of Koning iteration functions affected by their additional fixed points?*, Fractals **7**(3) (1999) 327–334.

[7] K. Kneisl, *Julia sets for the super-Newton method, Cauchy's method and Halley's method*, Chaos **11**(2) (2001) 359–370.

[8] S. Amat, S. Busquier and S. Plaza, *Review of some iterative root-finding methods from a dynamical point of view*, Scientia **10** (2004) 3–35.

[9] R.F. King, *A family of fourth order methodsfor nonlinear equations*, SIAM J. Numer. Anal. **10** (1973) 876–879.

[10] C. Chun, M. Y. Lee, B. Neta, J. Džunić, *On optimal fourth-order iterative methods free from second derivative and their dynamics*, Applied Mathematics and Computation **218** (2012) 6427–6438.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Uniform convergence of the Crank-Nicolson and central differences scheme for 1D parabolic singularly perturbed reaction–diffusion problems

C. Clavero[1], J.L. Gracia[1] and F. Lisbona[1]

[1] *Department of Applied Mathematics, University of Zaragoza*

emails: clavero@unizar.es, jlgracia@unizar.es, lisbona@unizar.es

Abstract

In this work we consider the numerical approximation of 1D parabolic singularly perturbed problems of reaction-diffusion type. The full discrete scheme combines the Crank-Nicolson method, defined on a uniform mesh, to discretize in time, and the standard central finite difference scheme, defined on special meshes condensing in the boundary layer regions, to discretize in space. The analysis proves that the numerical scheme is a second order in time and almost second order in space uniformly convergent method. The proof of the convergence is based on splitting the contribution to the error of the time and the spatial discretizations. Using an appropriate inductive argument, we also obtain the asymptotic behavior of the semidiscrete problems resulting after the time discretization, which is an interesting result by itself. We show the results obtained for a test problem, corroborating in practice the order of uniform convergence theoretically proved.

Key words: parabolic reaction-diffusion problems, uniform convergence, Crank-Nicolson method, special meshes
MSC 2000: 65N12, 65N30, 65N06

1 Introduction

We consider the 1D parabolic reaction-diffusion singularly perturbed problem

$$\left\{\begin{array}{ll} u_t + \mathcal{L}_{x,\varepsilon} u = f(x,t), & (x,t) \in Q = \Omega \times (0,T] \equiv (0,1) \times (0,T], \\ u(x,0) = 0, & x \in \overline{\Omega}, \\ u(0,t) = u(1,t) = 0, & t \in (0,T], \end{array}\right. \tag{1}$$

where the spatial differential operator is given by $\mathcal{L}_{x,\varepsilon}u \equiv -\varepsilon u_{xx} + \beta u$. The diffusion parameter $0 < \varepsilon \leq 1$ can be arbitrary small, β is a positive constant and we assume that sufficient regularity and compatibility conditions hold; then, the exact solution $u \in \mathcal{C}^{(6,3)}(\overline{Q})$ (see [7]). It is well–known (see [6, 8]) that the solution of (1) has a boundary layer at both $x = 0, 1$ of width $O(\sqrt{\varepsilon}|\ln \varepsilon|)$, and it satisfies

$$|u^{(k,m)}(x,t)| \leq C\left(1 + \varepsilon^{-k/2} B_\varepsilon(x)\right), \quad 0 \leq k + 2m \leq 6, \tag{2}$$

where

$$B_\varepsilon(x) = e^{-\sqrt{\beta/\varepsilon}\ x} + e^{-\sqrt{\beta/\varepsilon}\ (1-x)}.$$

These bounds show the asymptotic behavior, with respect to the diffusion parameter ε, of the exact solution. Moreover, u can be decomposed in the form $u = v + w$, where the regular component v is the solution of the problem

$$v_t + \mathcal{L}_{x,\varepsilon} v = f, \text{ in } Q, \ v(x,0) = 0, \text{ in } \overline{\Omega}, \tag{3}$$

and the boundary conditions at $x = 0$ and $x = 1$ are given by solving the IVP

$$z_t + \beta z = f(x,t), \ z(0) = 0, \quad t \in (0,T]. \tag{4}$$

On the other hand, the singular component w is the solution of the problem

$$\begin{cases} w_t + \mathcal{L}_{x,\varepsilon} w = 0, \text{ in } Q, \\ w(x,0) = 0, \text{ in } \overline{\Omega}, \\ w(0,t) = u(0,t) - v(0,t), \ w(1,t) = u(1,t) - v(1,t), \quad t \in (0,T]. \end{cases} \tag{5}$$

In practice, it is interesting to dispose of high order convergent methods because accurate approximations can be obtained with a low computational cost. In previous papers (see [1, 2]), a two stage process was introduced to analyze the uniform convergence for parabolic problem (1). In the first step, the problem is discretized only in time and defining appropriate auxiliary problems (see (7) below), adequate estimates for its local error are proved. In the second step, the stationary problems resulting from the time discretization are discretized in space. Then, the convergence of the numerical scheme is deduced by using the well-known results on fitted mesh methods for singularly perturbed steady problems.

To simplify the analysis given in [1, 2], the discretization of the singularly perturbed problems resulting from the time discretization, without any use of auxiliary problems, was considered in [3]. The new proof of the uniform convergence of the numerical scheme, which is based on an inductive argument, requires to know the asymptotic behavior of the exact solution of the semidiscrete problems. The implicit Euler method and the central finite difference scheme was used in [3], proving the uniform convergence of first order in time and almost second order in space. In this paper we analyze the uniform convergence of a higher

 ISBN:978-84-615-5392-1

order uniformly convergent method, which uses the Crank-Nicolson method to discretize in time and the central difference approximation in space.

Henceforth, C denotes a generic positive constant independent of the diffusion parameter ε and also of the discretization parameters N and M.

2 Time semidiscretization: uniform convergence and asymptotic behavior

The Crank-Nicolson method defined on the uniform mesh

$$\overline{\omega}^M \equiv \{t_k = k\tau,\ 0 \leq k \leq M, \tau = T/M\},$$

to discretize in time is given by

$$\left\{ \begin{array}{l} u^0(x) = 0,\ x \in \bar{\Omega}, \\ \left\{ \begin{array}{l} (I + (\tau/2)\mathcal{L}_{x,\varepsilon})u^n(x) = (\tau/2)(f(x,t_n) + f(x,t_{n-1})) + (I - (\tau/2)\mathcal{L}_{x,\varepsilon})u^{n-1}(x), \\ u^n(0) = u^n(1) = 0, \quad 1 \leq n \leq M. \end{array} \right. \end{array} \right. \tag{6}$$

In [4], the auxiliary problems

$$\left\{ \begin{array}{l} (I + (\tau/2)\mathcal{L}_{x,\varepsilon})\widehat{u}^n(x) = (\tau/2)(f(x,t_n) + f(x,t_{n-1})) + (I - (\tau/2)\mathcal{L}_{x,\varepsilon})u(x,t_{n-1}), \\ \widehat{u}^n(0) = \widehat{u}^n(1) = 0, \quad 1 \leq n \leq M, \end{array} \right. \tag{7}$$

were introduced to prove that $\|u(x,t_n) - \widehat{u}^n(x)\|_{\infty,\Omega} \leq C\tau^3,\ 1 \leq n \leq M$, where $\|\cdot\|_{\infty,\Omega}$ denotes the L_∞ norm on Ω. Using this result and the uniform stability satisfied by the Crank-Nicolson method, we can deduce its second order uniform convergence.

Theorem 1 *The global error associated to the method (6) satisfies*

$$\|u(x,t_n) - u^n(x)\|_{\infty,\Omega} \leq C\tau^2, \quad 1 \leq n \leq M. \tag{8}$$

For the analysis of the error associated to the spatial discretization, we need to know appropriate bounds on the derivatives of the exact solution of the semidiscrete problems (6). Similarly to the continuous case, we consider the decomposition $u^n = v^n + w^n$, for $1 \leq n \leq M$, where the regular component v^n is the solution of the problem

$$\left\{ \begin{array}{l} v^0(x) = 0,\ x \in \bar{\Omega}, \\ \left\{ \begin{array}{l} (I + (\tau/2)\mathcal{L}_{x,\varepsilon})v^n(x) = (\tau/2)(f(x,t_n) + f(x,t_{n-1})) + (I - (\tau/2)\mathcal{L}_{x,\varepsilon})v^{n-1}(x), \end{array} \right. \end{array} \right. \tag{9}$$

and the values at the boundaries $x = 0$ and $x = 1$ are given by

$$\left\{ \begin{array}{l} (I + (\tau/2)\beta)z^n(x) = (\tau/2)(f(x,t_n) + f(x,t_{n-1})) + (I - (\tau/2)\beta)z^{n-1}(x),\ 1 \leq n \leq M, \\ z^0 = 0, \end{array} \right.$$

 ISBN:978-84-615-5392-1

and the singular component w^n is the solution of the problem

$$\left\{\begin{array}{l} w^0(x)=0,\ x\in\Omega,\\ \left\{\begin{array}{l}(I+(\tau/2)\mathcal{L}_{x,\varepsilon})w^n(x)=(I-(\tau/2)\mathcal{L}_{x,\varepsilon})w^{n-1}(x),\quad 1\le n\le M,\\ w^n(0)=u^n(0)-v^n(0),\quad w^n(1)=u^n(1)-v^n(1).\end{array}\right.\end{array}\right. \tag{10}$$

Using similar ideas to these ones given in [3, 4], the following result can be proved.

Lemma 1 *The regular and the singular components satisfy*

$$\left|\frac{d^k v^n}{dx^k}(x)\right|\le C(1+\varepsilon^{1-k/2}),\ \left|\frac{d^k w^n}{dx^k}(x)\right|\le C\varepsilon^{-k/2}B_\varepsilon(x),\ 0\le k\le 4. \tag{11}$$

3 Uniform convergence of the fully discrete scheme

Let $\overline{\Omega}^N=\{0=x_0<\ldots<x_N=1\}$ a nonuniform mesh condensing the grid points in the boundary layers. On this mesh we discretize (6) by using the standard central difference scheme. Then, the fully discrete method is given by

$$\left\{\begin{array}{l} U_i^0=0,\quad 0\le i\le N,\\ \left\{\begin{array}{l}(I+(\tau/2)L_{x,\varepsilon}^N)U_i^n=(\tau/2)(f(x_i,t_n)+f(x_i,t_{n-1}))+(I-(\tau/2)L_{x,\varepsilon}^N)U_i^{n-1},\\ \qquad\qquad 1\le i\le N-1,\quad 1\le n\le M,\\ U_0^n=U_N^n=0,\end{array}\right.\end{array}\right. \tag{12}$$

where $L_{x,\varepsilon}^N Z_i\equiv-\varepsilon\delta^2 Z_i+\beta Z_i$, and

$$\delta^2 Z_i=\frac{2}{h_i+h_{i+1}}\left(\frac{Z_{i+1}-Z_i}{h_{i+1}}-\frac{Z_i-Z_{i-1}}{h_i}\right),\ h_i=x_i-x_{i-1},\ i=1,\cdots,N-1$$

is the standard approximation of second derivative on a nonuniform mesh. The convergence of this scheme is analyzed at each time level by using an inductive argument. In the proof some properties related with the discrete transition operator $R_{N,\tau}\equiv(I+(\tau/2)L_{x,\varepsilon}^N)^{-1}(I-(\tau/2)L_{x,\varepsilon}^N)$ are used.

Lemma 2 *Let $\Lambda_h=[-a_i\ c_i\ -b_i]$ the tridiagonal matrix associated to the the operator $L_{x,\varepsilon}^N$. Then, $a_i,c_i,b_i>0$ and it holds*

$$\left\|(I+(\tau/2)\Lambda_h)^{-1}\right\|_\infty\le\frac{1}{1+\beta\tau/2},$$

where $\|\cdot\|_\infty$ is the matrix maximum norm. Moreover, the eigenvalues λ_i of matrix $R_{N,\tau}$ are real and also they satisfy

$$-1<\lambda_i\le\frac{1-\beta\tau/2}{1+\beta\tau/2}<1.$$

Therefore, the spectral radius satisfies $\rho\,(R_{N,\tau})<1$.

 ISBN:978-84-615-5392-1

Remark 1 *Note that the bound on the spectral radius does not guarantee that it holds* $\| \left(R_{N,\tau}\right)^k \|_\infty < C,\ \forall\ k$, *which is the result that we need in the analysis of the uniform convergence of the method.*

Now let us define the special nonuniform meshes on which we construct the finite difference scheme. We consider two different meshes, both adapted to the asymptotic behavior of the solution of (1). These meshes are uniform when the diffusion parameter ε is large and they condense in the boundary layers otherwise.

The Shishkin mesh [8]. It is a piecewise uniform mesh with two transition points defined by means of the transition parameter

$$\sigma = \min\left\{1/4, \sigma_0\sqrt{\varepsilon}\ln N\right\}, \tag{13}$$

where σ_0 is constant. A uniform mesh is placed in $[0,\sigma]$, $[\sigma, 1-\sigma]$, and $[1-\sigma, 1]$, such that $x_0 = 0$, $x_{N/4} = \sigma$, $x_{3N/4} = 1-\sigma$, and $x_N = 1$ and therefore the mesh points are given by

$$x_i = \begin{cases} 4i\sigma/N, & i = 0, 1, \ldots, N/4, \\ \sigma + 2(i-N/4)(1-2\sigma)/N, & i = N/4+1, \ldots, 3N/4, \\ 1-\sigma+4(i-3N/4)\sigma/N, & i = 3N/4+1, \ldots, N. \end{cases} \tag{14}$$

The Vulanović mesh [9, 10]. This mesh is a generalized Shishkin mesh constructed by using a suitable generating function $\aleph$, which also depends on two transition points. To simplify, we use the same parameter σ given in (13). The grid points are defined by $x_i = \aleph(i/N),\ i = 0, 1, \ldots, N/2$, with $\aleph \in C^2[0, 1/2]$ and

$$\aleph(z) = \begin{cases} 4\sigma z, & z \in [0, 1/4], \\ p(z-1/4)^3 + 4\sigma(z-1/4) + \sigma, & z \in [1/4, 1/2]. \end{cases} \tag{15}$$

The coefficient p is such that $\aleph(1/2) = 1/2$ and the mesh is symmetric with respect to the point $x = 1/2$. Note that in $[0,\sigma]$ and $[1-\sigma, 1]$ the mesh points are the same than in the Shishkin mesh. However, in $[\sigma, 1-\sigma]$ it is nonuniform but the step sizes satisfy

$$|h_{i+1} - h_i| \le CN^{-2}, \quad i = N/4, \ldots, 3N/4. \tag{16}$$

To obtain appropriate estimates of the error of the spatial discretization, the solution U of the discrete problem (12) is decomposed into a regular and singular part, $U = V + W$, in a similar way than the solution of the continuous and semidiscrete problems. These two grid functions are the solution of the discrete problems

$$\begin{cases} V_i^0 = 0, \quad 0 \le i \le N, \\ \begin{cases} (I + (\tau/2)L_{x,\varepsilon}^N)V_i^n = (\tau/2)(f(x_i,t_n) + f(x_i,t_{n-1})) + (I - (\tau/2)L_{x,\varepsilon}^N)V_i^{n-1}, \\ \qquad\qquad 1 \le i \le N-1, \quad 1 \le n \le M, \\ V_0^n = v^n(0), \quad V_N^n = v^n(1), \end{cases} \end{cases} \tag{17}$$

 ISBN:978-84-615-5392-1

and

$$\left\{\begin{array}{l} W_i^0 = 0, \quad 0 \le i \le N, \\ \left\{\begin{array}{l} (I + (\tau/2)L_{x,\varepsilon}^N)W_i^n = (I - (\tau/2)L_{x,\varepsilon}^N)W_i^{n-1}, \ 1 \le i \le N-1, \ 1 \le n \le M, \\ W_0^n = w^n(0), \quad W_N^n = w^n(1), \end{array}\right. \end{array}\right. \tag{18}$$

respectively.

Theorem 2 *Let U^n be the numerical solution of (12) and u^n be the solution of (6), both at time level t_n. Let assume that $\sigma_0 \ge 2/\sqrt{\beta}$. Then, the error associated to the spatial discretization, on the Shishkin mesh satisfies*

$$\|U_i^n - u^n(x_i)\|_{\infty,\overline{\Omega}^N} \le C(N^{-1}\varepsilon + (N^{-1}\ln N)^2), \tag{19}$$

and on the Vulanović mesh it satisfies

$$\|U_i^n - u^n(x_i)\|_{\infty,\overline{\Omega}^N} \le C(N^{-1}\ln N)^2, \tag{20}$$

where $\|\cdot\|_{\infty,\overline{\Omega}^N}$ denotes the discrete maximum norm on $\overline{\Omega}^N$.

Proof. For the regular component v^n, the local error satisfies

$$\begin{aligned}(I + (\tau/2)L_{x,\varepsilon}^N)(V_i^n - v^n(x_i)) &= (\tau/2)\left[\left(L_{x,\varepsilon} - L_{x,\varepsilon}^N\right)\left(v^n(x_i) + v^{n-1}(x_i)\right)\right] \\ &+(I - (\tau/2)L_{x,\varepsilon}^N)(V_i^{n-1} - v^{n-1}(x_i)),\end{aligned}$$

and then, taking Taylor expansions and using estimates (11) it follows

$$(\tau/2)\left|\left(L_{x,\varepsilon} - L_{x,\varepsilon}^N\right)\left(v^n(x_i) + v^{n-1}(x_i)\right)\right| \le C\tau N^{-1}(N^{-1} + \varepsilon),$$

on the Shishkin mesh, and

$$(\tau/2)\left|\left(L_{x,\varepsilon} - L_{x,\varepsilon}^N\right)\left(v^n(x_i) + v^{n-1}(x_i)\right)\right| \le C\tau N^{-2},$$

on the Vulanovic mesh.

Now, using a recursive argument, taking into account that $\|(I + (\tau/2)L_{x,\varepsilon}^N)^{-1}\|_\infty \le C$ and assuming that the maximum norm of all the powers of the transition parameter $R_{N,\tau}$ is uniformly bounded, we can deduce that

$$\|V_i^n - v^n(x_i)\|_{\infty,\overline{\Omega}^N} \le CN^{-1}(N^{-1} + \varepsilon), \quad \|V_i^n - v^n(x_i)\|_{\infty,\overline{\Omega}^N} \le CN^{-2}, \tag{21}$$

on the Shishkin and Vulanovic meshes respectively.

For the singular component w^n the local error satisfies

$$\begin{aligned}(I + (\tau/2)L_{x,\varepsilon}^N)(W_i^n - w^n(x_i)) &= (\tau/2)\left[\left(L_{x,\varepsilon} - L_{x,\varepsilon}^N\right)\left(w^n(x_i) + w^{n-1}(x_i)\right)\right] \\ &+(I - (\tau/2)L_{x,\varepsilon}^N)(W_i^{n-1} - w^{n-1}(x_i)),\end{aligned}$$

 ISBN:978-84-615-5392-1

and from Taylor expansions and bounds (11) it follows

$$\left|\left(L_{x,\varepsilon}-L^N_{x,\varepsilon}\right)\left(w^n(x_i)+w^{n-1}(x_i)\right)\right| \leq C\tau \left\{ \begin{array}{ll} CN^{-2}, & x_i \in [\sigma, 1-\sigma], \\ C(N^{-1}\ln N)^2, & x_i \in (0,\sigma)\cup(1-\sigma,1), \end{array}\right.$$

and therefore

$$\|W_i^n - w^n(x_i)\|_{\infty,\overline{\Omega}^N} \leq C(N^{-1}\ln N)^2, \tag{22}$$

for both the Shishkin and the Vulanovic meshes.

Finally, from the triangular inequality and (21), (22) the result trivially follows. □

From Theorems 1 and 2, and using that $\|U_i^n - u(x_i,t_n)\|_{\infty,\overline{\Omega}^N} \leq \|U_i^n - u^n(x_i)\|_{\infty,\overline{\Omega}^N} + \|u^n(x_i) - u(x_i,t_n)\|_{\infty,\overline{\Omega}^N}$, we can easily obtain the main result of this work, giving the uniform convergence of the fully discrete scheme.

Theorem 3 *Let U be the numerical solution of (12) and u be the solution of (1). Let assume that $\sigma_0 \geq 2/\sqrt{\beta}$. Then, the error associated to the full discrete method, on the Shishkin mesh satisfies*

$$\|U_i^n - u(x_i,t_n)\|_{\infty,\overline{\Omega}^N} \leq C(N^{-1}\varepsilon + (N^{-1}\ln N)^2 + \tau^2), \tag{23}$$

and on the Vulanović mesh it satisfies

$$\|U_i^n - u(x_i,t_n)\|_{\infty,\overline{\Omega}^N} \leq C((N^{-1}\ln N)^2 + \tau^2). \tag{24}$$

4 Numerical experiments

We consider the test problem

$$u_t - \varepsilon u_{xx} + (2 + x - \cos(x))u = 4(e^{-t}-1) + 6txe^x,\ (x,t) \in (0,1)\times(0,1], \tag{25}$$

with homogeneous initial and boundary conditions, for which the exact solution is unknown. Figure 1 shows the numerical solution for $\varepsilon = 10^{-4}$; from it we see the boundary layers at both sides $x = 0, 1$.

Note that in this problem the reaction term is non constant, but the numerical results show the same order of uniform convergence proved for the constant case.

To approximate the numerical errors we use a variant of the double mesh principle (see [5]) and therefore at each time $t_n = n\tau,\ n = 0, 1, \ldots, M$ and for each mesh point $x_j,\ j = 0, 1, \ldots, N$, the error is estimated by

$$D_{j,n}^{\varepsilon,N,M} = \left|U_{j,n}^{\varepsilon,N,M} - U_{j,n}^{\varepsilon,2N,2M}\right|,$$

where $U_{j,n}^{\varepsilon,N,M}$ is the numerical solution given by the fully discrete method with a constant time step $\tau = 1/M$ and $(N+1)$ points in the spatial mesh, and $U_{j,n}^{\varepsilon,2N,2M}$ is the numerical

 ISBN:978-84-615-5392-1

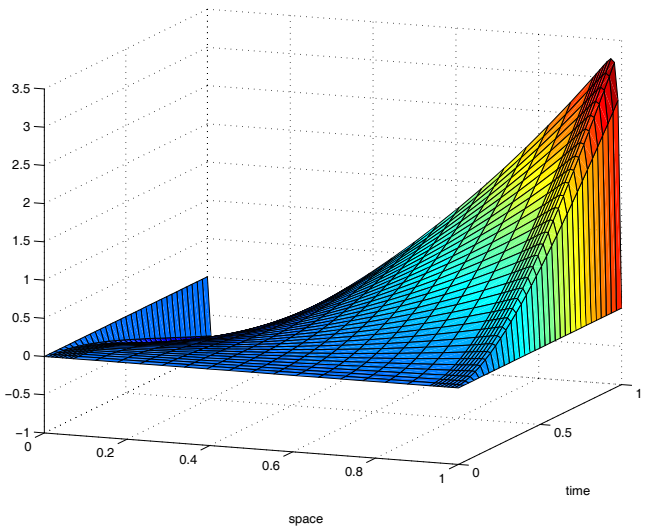


Figure 1: Solution for $\varepsilon = 10^{-4}$ with $N = M = 32$

solution computed by using the time step $\tau/2$ and $(2N+1)$ points in the spatial mesh, but with the same transition parameter than in the original mesh.

For each fixed value of ε, the maximum global errors are estimated by

$$D^{\varepsilon,N,M} = \max_{j,n} D^{\varepsilon,N,M}_{j,n}$$

and therefore, in standard way, the numerical orders of convergence are given by

$$q = \log\left(D^{\varepsilon,N,M}/D^{\varepsilon,2N,2M}\right)/\log 2.$$

From these values we obtain the ε-uniform errors and the ε-uniform orders of convergence by

$$D^{N,M} = \max_{\varepsilon} D^{\varepsilon,N,M}, \;\; q_{uni} = \log\left(D^{N,M}/D^{2N,2M}\right)/\log 2.$$

Table 1 displays the maximum and uniform errors and the numerical orders of convergence of scheme (12) defined on the Shishkin mesh (14), for the set of the values $\varepsilon \in \{2^{-4}, \ldots, 2^{-30}\}$. From these results we clearly deduce almost second order convergence in agreement with Theorem 3. For this and other test problems, similar results, about the maximum errors and the orders of convergence, have been obtained when the Vulanović mesh is used instead of the Shishkin mesh.

Table 2 displays the maximum matrix norm of the transition operator $R_{N,\tau}$ for $\varepsilon = 10^{-4}, 10^{-6}, 10^{-8}$ and some values of N and M. From it we can conclude that its maximum norm is less than 1. Nevertheless, taking a different range of values for the discretization

Table 1: Maximum errors and uniform orders of convergence

ε	N=32 $M=5$	N=64 $M=10$	N=128 $M=20$	N=256 $M=40$	N=512 $M=80$	N=1024 $M=160$
2^{-4}	0.979E-02 1.941	0.255E-02 1.995	0.639E-03 2.011	0.159E-03 2.011	0.394E-04 2.009	0.978E-05
2^{-6}	0.121E-01 1.914	0.322E-02 2.094	0.753E-03 1.997	0.189E-03 1.987	0.476E-04 2.006	0.118E-04
2^{-8}	0.389E-01 1.768	0.114E-01 2.105	0.265E-02 1.988	0.669E-03 1.981	0.169E-03 2.008	0.421E-04
2^{-10}	0.862E-01 1.067	0.411E-01 2.041	0.100E-01 1.973	0.255E-02 1.974	0.648E-03 2.005	0.161E-03
2^{-12}	0.860E-01 0.965	0.441E-01 1.621	0.143E-01 1.571	0.483E-02 1.636	0.155E-02 1.698	0.478E-03
2^{-14}	0.860E-01 0.966	0.440E-01 1.621	0.143E-01 1.571	0.482E-02 1.637	0.155E-02 1.698	0.477E-03
...	...	...	...	...	...	...
2^{-30}	0.859E-01 0.968	0.439E-01 1.621	0.143E-01 1.571	0.481E-02 1.637	0.155E-02 1.698	0.476E-03
$D^{N,M}$ q_{uni}	0.862E-01 0.967	0.441E-01 1.621	0.143E-01 1.571	0.483E-02 1.636	0.155E-02 1.698	0.478E-03

parameters (see Table 3), we observe that sometimes the maximum norm is greater than 1. Nevertheless, to obtain the bounds (23) and (24) for the error we only need that the norm of $R_{N,\tau}^p$ be bounded. Table 4 displays the value obtained for some values of ε in the most unfavorable case when N is large and M is small; from it we see that in all cases the maximum norm of the discrete transition operator is bounded and it decreases as p increases. So, we can conclude that from numerical point of view the condition of Remark 1 for the powers of $R_{N,\tau}$ holds.

Table 2: Maximum norm of the transition operator $R_{N,\tau}$

	N=16 $M=5$	N=32 $M=10$	N=64 $M=20$	N=128 $M=40$	N=256 $M=80$
$\varepsilon=10^{-4}$	0.8110442	0.9020340	0.9501518	0.9749084	0.9874263
$\varepsilon=10^{-6}$	0.8173460	0.9044277	0.9510904	0.9752613	0.9875600
$\varepsilon=10^{-8}$	0.8180901	0.9047215	0.9512043	0.9753031	0.9875756

Acknowledgements

This research was partially supported by the project MEC/FEDER MTM 2010-16917 and by the Diputación General de Aragón

Table 3: Maximum norm of the transition operator $R_{N,\tau}$ for $\varepsilon = 10^{-6}$

	$M = 1000$	$M = 200$	$M = 100$	$M = 20$	$M = 10$	$M = 5$
$N = 32$	0.99900	0.99500	0.99002	0.95105	0.90443	0.81745
$N = 64$	0.99900	0.99500	0.99003	0.95109	0.904447	0.81755
$N = 128$	0.99900	0.99501	0.99003	0.95111	0.90449	1.39948
$N = 256$	0.99900	0.99501	0.99004	1.26496	1.76229	2.08648
$N = 512$	0.99900	0.99501	0.99004	2.07472	2.34420	2.45748

Table 4: Maximum norm for $R^p_{N,\tau}$, with $N = 512, M = 5$

	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$
$\|R_{N,\tau}\|$	2.6651201	2.2341760	2.4574815	2.5331270
$\|R^3_{N,\tau}\|$	2.2756270	1.6986553	1.8661541	1.9127569
$\|R^5_{N,\tau}\|$	2.0543530	1.3372694	1.1191191	1.4958078
$\|R^7_{N,\tau}\|$	1.9503709	1.094808	1.1568691	1.1941271
$\|R^9_{N,\tau}\|$	1.8795689	0.9277669	0.9454566	0.9768708
$\|R^{11}_{N,\tau}\|$	1.8276318	0.7909523	0.8054321	0.8261720
$\|R^{13}_{N,\tau}\|$	1.7876380	0.6741852	0.6854494	0.6989921

References

[1] B. Bujanda, C. Clavero, J.L. Gracia and J.C. Jorge, *A high order uniformly convergent alternating direction scheme for time dependent reaction-diffusion singularly perturbed problems*, Num. Math. **107** (2007) 1–25.

[2] C. Clavero and J.L. Gracia, *High order methods for elliptic and time dependent reaction-diffusion singularly perturbed problems*, Appl. Math. Comp. **168** (2005) 1109–1127.

[3] C. Clavero and J.L. Gracia, *On the uniform convergence of a finite difference scheme for time dependent singularly perturbed reaction–diffusion problems*, Appl. Math. Comp. **216** (2010) 1478–1488.

[4] C. Clavero, J.L. Gracia and F.J. Lisbona, *Second order uniform approximations for the solution of time dependent singularly perturbed reaction–diffusion systems*, Int. J. Numer. Anal. Mod. **7** (2010) 428–443.

[5] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O'Riordan, E. and G.I. Shishkin, *Robust computational techniques for boundary layers*, Chapman & Hall (2000).

 ISBN:978-84-615-5392-1

[6] P.W. Hemker, G.I. Shishkin and L.P. Shiskina, *ε-uniform schemes with high-order time accuracy for parabolic singular perurbation problems*, IMA J. Numer. Anal. **20** (2000) 99–121.

[7] O.A. Ladyzhenskaya, V.A. Solonnikov and N.N. Ural'tseva, *Linear and quasi-linear equations of parabolic type*, Transactions of Mathematical Monographs, **23**, American Mathematical Society, 1968.

[8] H.-G. Roos, M. Stynes and L. Tobiska, *Robust numerical methods for singularly perturbed differential equations*, Springer Series in Computational Mathematics **24**, Springer-Verlag, Berlin, 2008.

[9] R. Vulanović, *A high order scheme for quasilinear boundary value problems with two small parameters*, Computing **67** (2001) 287–303.

[10] R. Vulanović, *An almost sixth-order finite-difference method for semilinear singular perurbation problems*, Comp. Meth. Appl. Math. **4** (2004) 368–383.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

A key agreement protocol for distributed secure multicast on a non-commutative ring

Joan-Josep Climent[1], Juan Antonio López-Ramos[2], Pedro R. Navarro[3] and Leandro Tortosa[3]

[1] *Departament d'Estadística i Investigació Operativa, Universitat d'Alacant*

[2] *Departamento de Álgebra y Análisis Matemático, Universidad de Almería*

[3] *Departament de Ciència de la Computació i Intel·ligència Artificial, Universitat d'Alacant*

emails: jcliment@ua.es, jlopez@ual.es, prnr63@gmail.com, tortosa@ua.es

Abstract

We introduce a key agreement protocol for secure communications. The protocol shows to be efficient for large audiences, making it applicable for the nowadays widely extended secure multicast communications and allows users to join or leave the communication group preserving forward and backward secrecy in an efficient way.

Key words: Secure Communications, Key exchange
MSC 2000: 68P25 94A60

1 Introduction

Key exchange protocols are a typical problem in Cryptography. Since Diffie and Hellman in [5] introduced their elegant solution based on the Discrete Logarithm Problem (DLP), many authors have dealt with this main issue in Cryptography (cf. [6, 8] and their references). Nowadays the increasing interest in group-oriented applications and protocols have lead the researchers to find efficient protocols for group communications and specially secure protocols that provide privacy and integrity.

Multicast is an efficient way to send contents from a source to multiple receivers [4]. The efficiency of IP-multicast is due mainly to the fact that the information is transmitted only

once and this reaches all those receivers that have shown their interest in receiving the information from a determined IP-address, going through each connection between two of these nodes corresponding to the receivers [3]. Although this scheme for distributing information is able to extend its operations capability as the group of users grows without decreasing its quality and/or efficiency, does not provide security for accessing the distributed information only to the group of authorized users. Multicast schemes that take into account this property are known as secure multicast schemes [12]. Concerning the authority that is in charge of rekeying, multicast protocols are divided into three categories [7]: centralized, those with a central authority that changes and multicast the session key; decentralized, where there exists a group of "detached users" that act as local authorities for key distribution and distributed key management protocols, where the member themselves carry out the key generation and our aim in this paper is to focus in the latter case.

Dynamic Peer Group are common in many applications such as data bases, video conferencing, etc and distributed protocols for multicasting are quite appropriate in this setting where group could be continuously changing. Steiner *et al.* [10] introduce a protocol for group key agreement based on the two-party Diffie-Hellman key exchange. This shows to behave much more efficiently than other protocols previously introduced and it is used later in a key agreement protocol, CLIQUES, in Dynamic Peer Groups [11].

However increasing of computing capabilities of actual computers and works as [1] and [9] have provoked that protocols and cryptosystems based on the Discrete Logarithm Problem (DLP) are no longer recommended.

Our aim in this paper is to introduce a solution for distributed key agreement over a non-commutative ring inspired by the two-party key exchange protocol given in [2], based on one of the currently considered problems for cryptography on non-commutative groups, more precisely, on the so-called Decomposition Problem (DP), i.e., given a pair of elements x and y in a noncommutative group G and $S \subseteq G$, the problem is to find z_1 and z_2 in S such that $y = z_1 x z_2$.

2 A Distributed Secure Multicast Scheme

The target scenario is the following: private communications must be established within a restricted group. There is not a central server that manages the key management issues. Components of this restricted group will manage all rekeying operations by themselves. From now on, we will refer to the clients as *members* or *users*. We are assuming a typical pattern of many-to-many communications as noted above although we can also consider a setting where one-to-many communications take place. However, as usually in this type of protocols, the number of users cannot be as larger as in a typical one-to-many communication as an IP-TV service due to the nature of distributed secure multicast protocols.

So let us assume that the set of users is given by $\{U_1, U_2, \ldots, U_h\}$. Users agree to use a noncommutative ring R and two elements $r \in R$ and $s \in R \setminus Z(R)$, where $Z(R)$ denotes the center of the ring R. Therefore r and s are public. Furthermore, if we consider $f(x), g(x) \in Z(R)[x]$ and u and v are positve integers, although R is not commutative, we have that

$$f(r)^u g(r)^v = g(r)^v f(r)^u. \tag{1}$$

This property allows us to establish the following protocol.

Protocol 1: Every user U_i, for $i = 1, 2, \ldots, h$ chooses a polynomial $f_i(x) \in Z(R)[x]$ and a pair of positive integers m_i and n_i. Then the private key for the user U_i is $(f_i(x), m_i, n_i)$. Assume also that $K_0 = s$.

(a) User U_1 computes the element K_1 of R given by

$$K_1 = f_1(r)^{m_1} K_0 f_1(r)^{n_1}. \tag{2}$$

User U_1 sends the element K_1 to user U_2.

(b) User U_2 computes the element K_2 of R given by

$$K_2 = f_2(r)^{m_2} K_1 f_2(r)^{n_2}. \tag{3}$$

User U_2 sends to user U_3 the vector of elements in R given by (K_1, K_2).

(c) In general, for $i = 3, 4, \ldots, h-2, h-1$, user U_i computes the element

$$K_i = f_i(r)^{m_i} K_{i-1} f_i(r)^{n_i}. \tag{4}$$

Then, user U_i sends to user U_{i+1} the vector $(K_1, K_2, \ldots, K_{i-1}, K_i)$ of elements in R.

(d) When user U_h receives the vector $(K_1, K_2, \ldots, K_{h-2}, K_{h-1})$, he/she computes the elements of R given by

$$L_l^{(h)} = f_h(r)^{m_h} K_l f_h(r)^{n_h}, \quad \text{for} \quad l = 0, 1, 2, \ldots, h-2, h-1. \tag{5}$$

Then user U_h sends back to user U_{h-1} the vector $\left(L_0^{(h)}, L_1^{(h)}, \ldots, L_{h-3}^{(h)}, L_{h-2}^{(h)}\right)$ of elements in R.

(e) When user U_{h-1} receives from user U_h the vector $\left(L_0^{(h)}, L_1^{(h)}, \ldots, L_{h-3}^{(h)}, L_{h-2}^{(h)}\right)$, he/she computes the elements of R given by

$$L_l^{(h-1)} = f_{h-1}(r)^{m_{h-1}} L_l^{(h)} f_{h-1}(r)^{n_{h-1}}, \quad \text{for} \quad l = 0, 1, 2, \ldots, h-3, h-2. \tag{6}$$

Then user U_{h-1} sends to user U_{h-2} the vector $\left(L_0^{(h-1)}, L_1^{(h-1)}, \ldots, L_{h-4}^{(h-1)}, L_{h-3}^{(h-1)}\right)$.

 ISBN:978-84-615-5392-1

(f) In general, for $i = 2, 3, \ldots, h-2, h-1$, when user U_{h-i} receives from user U_{h-i+1} the vector

$$\left(L_0^{(h-i+1)}, L_1^{(h-i+1)}, \ldots, L_{h-i-2}^{(h-i+1)}, L_{h-i-1}^{(h-i+1)}\right),$$

he/she computes the elements of R given by

$$L_l^{(h-i)} = f_{h-i}(r)^{m_{h-i}} L_l^{(h-i+1)} f_{h-i}(r)^{n_{h-i}}, \quad \text{for} \quad l = 0, 1, 2, \ldots, h-i-2, h-i-1. \tag{7}$$

Then user U_{h-i} sends to user U_{h-i-1} the vector $\left(L_0^{(h-i)}, L_1^{(h-i)}, \ldots, L_{h-i-3}^{(h-i)}, L_{h-i-2}^{(h-i)}\right)$.

(g) Each user U_{h-i}, for $i = 0, 1, 2, \ldots, h-2, h-1$, has computed the element $L_{h-i-1}^{(h-i)}$. Such element is the only one that has not been sent to user U_{h-i-1}. This is the element which is shared by all the users as we can see in the following theorem. □

Theorem 1: *For $i = 0, 1, 2, \ldots, h-2, h-1$ it follows that*

$$L_{h-i-1}^{(h-i)} = \left(\prod_{k=1}^{h} f_k(r)^{m_k}\right) K_0 \left(\prod_{k=1}^{h} f_k(r)^{n_k}\right).$$

PROOF: Assume that $i = 0, 1, 2, \ldots, h-2, h-1$. From expressions (7), (6), (5), and (1) we have that

$$\begin{aligned} L_{h-i-1}^{(h-i-1)} &= f_{h-i}(r)^{m_{h-i}} L_{h-i-1}^{(h-i+1)} f_{h-i}(r)^{n_{h-i}} \\ &= f_{h-i}(r)^{m_{h-i}} \left(f_{h-i+1}(r)^{m_{h-i+1}} L_{h-i-1}^{(h-i+2)} f_{h-i+1}(r)^{n_{h-i+1}}\right) f_{h-i}(r)^{n_{h-i}} \\ &= \cdots \\ &= \left(\prod_{k=h-i}^{h} f_k(r)^{m_k}\right) K_{h-i-1} \left(\prod_{k=h-i}^{h} f_k(r)^{n_k}\right) \end{aligned} \tag{8}$$

Now, from expressions (2), (3), (4), and (1) we have that

$$\begin{aligned} K_{h-i-1} &= f_{h-i-1}(r)^{m_{h-i-1}} K_{h-i-2} f_{h-i-1}(r)^{n_{h-i-1}} \\ &= f_{h-i-1}(r)^{m_{h-i-1}} \left(f_{h-i-2}(r)^{m_{h-i-2}} K_{h-i-3} f_{h-i-2}(r)^{n_{h-i-2}}\right) f_{h-i-1}(r)^{n_{h-i-1}} \\ &= \cdots \\ &= \left(\prod_{k=1}^{h-i-1} f_k(r)^{m_k}\right) K_0 \left(\prod_{k=1}^{h-i-1} f_k(r)^{n_k}\right) \end{aligned} \tag{9}$$

 ISBN:978-84-615-5392-1

Finally, from expressions (8), (9), and (1) we have that

$$L_{h-i-1}^{(h-i-1)} = \left(\prod_{k=h-i}^{h} f_k(r)^{m_k}\right)\left(\prod_{k=1}^{h-i-1} f_k(r)^{m_k}\right) K_0 \left(\prod_{k=1}^{h-i-1} f_k(r)^{n_k}\right)\left(\prod_{k=h-i}^{h} f_k(r)^{n_k}\right)$$
$$= \left(\prod_{k=1}^{h} f_k(r)^{m_k}\right) K_0 \left(\prod_{k=1}^{h} f_k(r)^{n_k}\right). \qquad \square$$

Following [7] we get that the parameters that are used to measure scalability of the precedent protocol are number of messages and rounds for every rekeying operation. In this case we get that the number of messages and rounds is exactly $2(h-1)$.

3 Joining and leaving the group

When a new user U_{h+1} joins the system a rekeying is needed in order to preserve backward secrecy. Then user U_h changes his/her secret information, i.e., chooses a new polynomial $\hat{f}_h \in Z(R)[x]$ and two new positive integers $\hat{m}_h$ and $\hat{n}_h$. Then he/she computes the element

$$\hat{K}_h = \hat{f}_h(r)^{\hat{m}_h} K_{h-1} \hat{f}_h(r)^{n_h}$$
$$= \left(\hat{f}_h(r)^{\hat{m}_h} \prod_{k=1}^{h-1} f_k(r)^{m_k}\right) K_0 \left(\hat{f}_h(r)^{\hat{n}_h} \prod_{k=1}^{h-1} f_k(r)^{n_k}\right).$$

User U_h sends to user U_{h+1} the vector $(K_1, K_2, \ldots, K_{h-1}, \hat{K}_h)$ of elements in R.

Then, user U_{h+1} chooses his/her private key $(f_{h+1}(x), m_{h+1}, n_{h+1})$, computes the elements of R given by

$$L_i^{(h+1)} = f_{h+1}(r)^{m_{h+1}} K_i f_{h+1}(r)^{n_{h+1}}, \quad \text{for} \quad i = 0, 1, 2, \ldots, h-2, h-1.$$
$$L_h^{(h+1)} = f_{h+1}(r)^{m_{h+1}} \hat{K}_h f_{h+1}(r)^{n_{h+1}},$$

and sends back to user U_h the vector $\left(L_0^{(h+1)}, L_1^{(h+1)}, \ldots, L_{h-2}^{(h+1)}, L_{h-1}^{(h+1)}\right)$. This fact corresponds to step (d) of Protolol 1 for user U_{h+1} instead of user U_h.

Next, all users follow the process described by steps (e) and (f) of Protocol 1, but starting from user U_h instead of user U_{h-1}.

Now if user U_i decides to leave the system, a rekeying is also needed in order to preserve forward secrecy. In this case, the process is also easy since it is enough that user U_{i-1} keeps recorded the message that he/she received from user U_{i-2}, changes his/her private information as before and starts the rekeying process from this message as in Protocol 1.

In case user U_h decides to leave the group, then user U_{h-1} changes his/her parameters and sends back the other users the message acting as the new last user.

 ISBN:978-84-615-5392-1

Let us remark finally that in the corresponding protocol given in [10] and [11], a simple division on a finite field would yield every power computed for every user and thus, to compromise every user's private key we have just to solve a DLP. In the case we are considering, this attack is not always possible since existence of inverses is not ensured.

Acknowledgements

The work of the first author was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Economía y Competitividad of the Gobierno de España. The work of the second author was partially supported by Spanish grants TEC2009-13763-C02-02 of the Ministerio de Ciencia e Innovación of the Gobierno de España and FQM 0211 of the Junta de Andalucía.

References

[1] D. Boneh, R. A. DeMillo and R. J. Lipton. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology*, **14**: 101–119 (2001).

[2] J.-J. Climent, P. R. Navarro and L. Tortosa. Key exchange protocols over non-commutative rings. The case $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$. In J. Vigo Aguiar (editor), *Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE 2011)*, pages 357–364. 2011.

[3] M. Cotton, L. Vegoda, ICANN and D. Meyer. IANA guidelines for IPv4 multicast address assignements. Internet Engineering Task Force (IETF), RFC5771, 2010. `http://tools.ietf.org/html/rfc5771`.

[4] S. E. Deering. Multicast routing in internetworks and extended LANs. In *Proceedings of the Symposium on Communications Architectures and Protocols (SIGCOMM '88)*, pages 15–64. Stanford, CA, 1988.

[5] W. D. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, **22(6)**: 644–654 (1976).

[6] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, 1996.

[7] S. Rafaeli and D. Hutchison. A survey of key management for secure group communication. *ACM Computing Surveys*, **35(3)**: 309–329 (2003).

[8] B. Schneier. *Applied Cryptography*. John Wiley & Sons, New York, NY, second edition, 1996.

[9] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, **26(5)**: 1484–1509 (1997).

[10] M. Steiner, G. Tsudik and M. Waidner. Diffie-Hellman key distribution extended to group communication. In *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pages 31–37. ACM, New York, NY, 1996.

[11] M. Steiner, G. Tsudik and M. Waidner. Key agreement in dynamic peer groups. *IEEE Transactions of Parallel and Distributed Systems*, **11(8)**: 769–780 (2000).

[12] S. Zhu and S. Jajodia. Scalable group key management for secure multicast: A taxonomy and new directions. In S. C.-H. Huang, D. MacCallum and D.-Z. Du (editors), *Network Security*, pages 57–75. Springer, New York, 2010.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

High Order Schemes for Solving Nonlinear Systems of Equations

Alicia Cordero[1], Juan R. Torregrosa[1] and María P. Vassileva[2]

[1] *Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera, s/n, 46022 València, Spain*

[2] *Instituto Tecnológico de Santo Domingo (INTEC), Avda. Los Próceres, Galá, Santo Domingo, República Dominicana*

emails: acordero@mat.upv.es, jrtorre@mat.upv.es, marip@intec.edu.do

Abstract

A set of multistep iterative methods with increasing order of convergence is presented, for solving systems of nonlinear equations. One of the main advantages of these schemes is to achieve high order of convergence with few Jacobian and functional evaluations, joint with the use of the same matrix of coefficients in the most of the linear systems involved in the process. Indeed, the application of the pseudocomposition technique on these proposed schemes allow us to increase their order of convergence, obtaining new high-order, efficient methods.

Key words: Nonlinear systems, Iterative methods, Jacobian matrix, Convergence order, Efficiency index.

1 Introduction

Many relationships in nature are inherently nonlinear, which according to these effects are not in direct proportion to their cause. In fact, a large number of such real-world applications are reduce to solve nonlinear systems numerically. Approximating a solution ξ of a nonlinear system $F(x) = 0$, $F : \mathbb{R}^n \to \mathbb{R}^n$, is a classical problem that appears in different branches of science and engineering.

Recently, for $n = 1$, many robust and efficient methods have been proposed with high convergence order, but in most of cases the method cannot be extended for several variables. Nevertheless, Babajee et al. in [1] design Chebyshev-like schemes for solving nonlinear

systems. In general, few papers for the multidimensional case introduce methods with high order of convergence. The authors design in [2] a modified Newton-Jarrat scheme of sixth-order; in [4] a third-order method is presented for computing real and complex roots of nonlinear systems; Darvishi et al. in [3] improve the order of convergence of known methods from quadrature formulae; Shin et. al. compare in [5] Newton-Krylov methods and Newton-like schemes for solving big-sized nonlinear systems; in [6] a general procedure to design high-order methods for problems in several variables is presented; moreover, the Adomian Decomposition has shown to be a useful tool to design new high order methods (see [7] and [8])

On the other hand, the pseudocomposition technique (see [9]) consists of the following: we consider a method of order of convergence p as a predictor, whose penultimate step is of order q, and then we use a corrector step based on the Gaussian quadrature. So, we obtain a family of iterative schemes whose order of convergence is $min\{q+p, 3q\}$. This is a general procedure to improve the order of convergence of known methods.

We denote $e_k = x^{(k)} - \xi$ the error in the kth iteration. The equation $e_{(k+1)} = L{e_k}^p + O[{e_k}^{p+1}]$, where L is a p-linear function $L \in \mathcal{L}(R^n \times \cdots \times R^n, R^n)$, is called the *error equation* and p is the *order of convergence.*

To analyze and compare the efficiency of the proposed methods we use the classic efficiency index $I = p^{1/d}$ due to Ostrowski [10], where d is the number of functional evaluations at each iteration.

In this paper, we present three new Newton-like schemes, of order of convergence four, six and eight, respectively. After the analysis of convergence of the new methods, we apply the pseudocomposition technique in order to get higher order procedures.

We have organized the rest of the paper as follows: in the next section, we present the new methods of order four, six and eight, respectively. Then, the pseudocomposition technique is applied on them and some new higher-order schemes are obtained, which have also more interesting properties. Finally, some concluding remarks are stated.

2 New high-order methods and their pseudocomposed partners

In the following, we will present a new multistep Newton-type scheme which reaches eighth-order of convergence with five steps, and we will denote it as M8. In the analysis of convergence, we proof that its first three steps are a fourth-order scheme, denoted by M4, and its four first steps become a sixth-order method that will be denoted by M6. The coefficients involved have been obtained optimizing the order the convergence and the whole scheme requires three functional evaluations of F and two of F' to attain eighth-order of convergence. Let us also note that no linear system must be solved at the second step and the linear systems to be solved in the last three steps have the same matrix. So, the number

of operations involved is not as high as it can seem.

Theorem 1 *Let $F : \Omega \subseteq R^n \to R^n$ be a sufficiently differentiable in a neighborhood of $\xi \in \Omega$ which is a solution of the nonlinear system $F(x) = 0$. We suppose that $F'(x)$ is continuous and nonsingular at ξ. Then, the sequence $\{x_k\}_{k\geq 0}$ obtained by*

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{1}{2}\left[F'\left(x^{(k)}\right)\right]^{-1} F\left(x^{(k)}\right), \\ z^{(k)} &= \frac{1}{3}\left(4y^{(k)} - x^{(k)}\right), \\ u^{(k)} &= y^{(k)} + \left[F'\left(x^{(k)}\right) - 3F'\left(z^{(k)}\right)\right]^{-1} F\left(x^{(k)}\right), \\ v^{(k)} &= u^{(k)} + 2\left[F'\left(x^{(k)}\right) - 3F'\left(z^{(k)}\right)\right]^{-1} F\left(u^{(k)}\right), \\ x^{(k+1)} &= v^{(k)} + 2\left[F'\left(x^{(k)}\right) - 3F'\left(z^{(k)}\right)\right]^{-1} F\left(u^{(k)}\right), \end{aligned} \tag{1}$$

converges to ξ with order of convergence eight. The error equation is:

$$e_{k+1} = \frac{1}{9}\left(C_3 - C_2^2\right)\left(C_4 - 9C_3C_2 + 9C_2^3\right)e_k^8 + O[e_k^9].$$

It is known (see [9]) that, by applying pseudocomposition, it is possible to design methods with higher order of convergence. We will see in the following how this technique modify the properties of the proposed schemes.

Theorem 2 *[9] Let $F : \Omega \subseteq R^n \to R^n$ be sufficiently differentiable in a neighborhood of $\xi \in \Omega$ and ξ a solution of the nonlinear system $F(x) = 0$. We suppose that $F'(x)$ is continuous and nonsingular at ξ. Let $y^{(k)}$ and $z^{(k)}$ be the penultimate and final steps of orders q and p, respectively, of a certain iterative method. Taking this scheme as a predictor we get a new approximation $x^{(k+1)}$ of ξ given by*

$$x^{(k+1)} = y^{(k)} - 2\left[\sum_{i=1}^{m} \omega_i F'(\eta_i^{(k)})\right]^{-1} F(y^{(k)}),$$

where $\eta_i^{(k)} = \frac{1}{2}\left[(1+\tau_i)z^{(k)} + (1-\tau_i)y^{(k)}\right]$ and τ_i, ω_i $i = 1, \ldots, m$ are the nodes and weights of the orthogonal polynomial corresponding to the Gaussian quadrature used. Then,

1. *the obtained set of families will have an order of convergence at least q;*
2. *if $\sigma = 2$ is satisfied, then the order of convergence will be at least $2q$;*

 ISBN:978-84-615-5392-1

3. if, also, $\sigma_1 = 0$ *the order of convergence will be* $min\{p+q, 3q\}$.

where $\sum_{i=1}^{n} \omega_i = \sigma$ *and* $\sum_{i=1}^{n} \frac{\omega_i \tau_i^j}{\sigma} = \sigma_j \quad with \quad j = 1, 2.$

Each of the families obtained will consist of subfamilies that are determined by the orthogonal polynomial corresponding to the Gaussian quadrature used. Furthermore, in these subfamilies there can be obtained various methods using different number of nodes corresponding to the orthogonal polynomial used (see Table 1). According to the proof of Theorem 2 the order of convergence of the obtained methods does not depend on the number of nodes used; so, the method will be more efficient as lower is the number of nodes employed.

	Quadratures							
Number of nodes	Chebyshev		Legendre		Lobatto		Radau	
	σ	σ_1	σ	σ_1	σ	σ_1	σ	σ_1
1	π	0	2	0	2	0	2	-1
2	π	0	2	0	2	0	2	0
3	π	0	2	0	2	0	2	0

Table 1: Quadratures used

Let us note that these methods, obtained by means of Gaussian quadratures, seem to be known interpolation quadrature schemes such as midpoint, trapezoidal or Simpson's method (see [11]). It is only a similitude, as they are not applied on the last iteration $x^{(k)}$, and the last step of the predictor, but on the two last steps of the predictor. In the following, we will use a midpoint-like as a corrector step, which corresponds to a Gauss-Legendre quadrature with one node; for this scheme the order of convergence will be at least $min\{q+p, 3q\}$, by applying Theorem 2. As this corrector on any of the new methods does only need a new functional evaluation of the Jacobian matrix, the efficiency of the resulting procedure will be maximum. So, by pseudocomposing on M6 and M8 there can be obtained two procedures of order of convergence 10 and 14 (denoted by PsM10 and PsM14), respectively. It is also possible to pseudocompose on M4, but the resulting scheme would be of third order of convergence, which is worst than the original M4, so it will not be considered.

Following the notation used in (1), the last step of PsM10 is

$$x^{(k+1)} = u^{(k)} - \left[F'\left(\frac{v^{(k)} + u^{(k)}}{2} \right) \right]^{-1} F(u^{(k)}), \tag{2}$$

 ISBN:978-84-615-5392-1

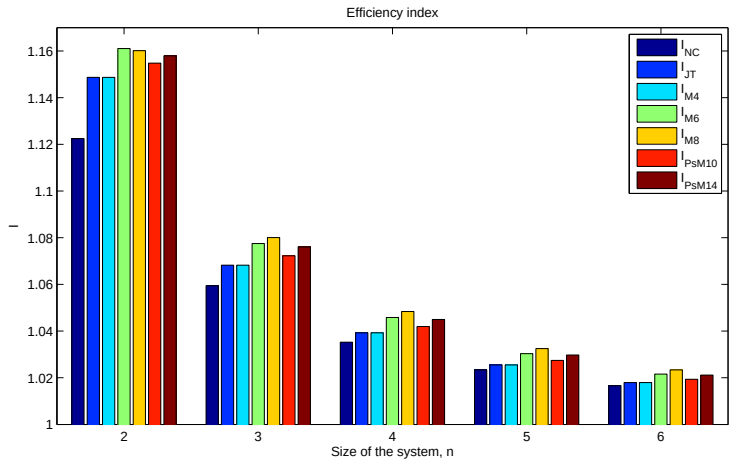


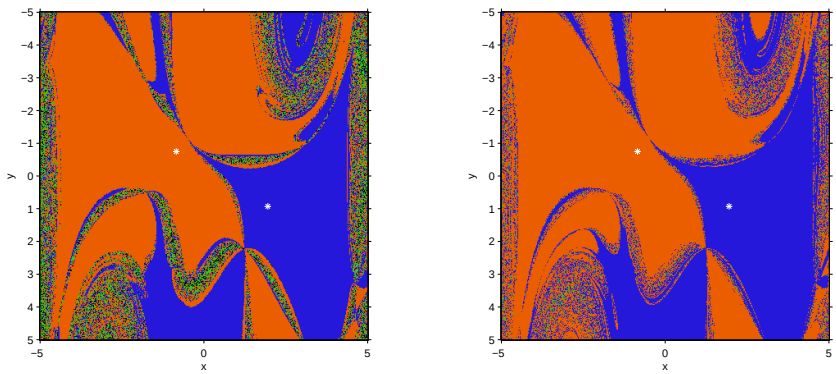
Figure 1: Efficiency index of the different methods for different sizes of the system

and the last three steps of psM14 can be expressed as

$$\begin{aligned} v^{(k)} &= z^{(k)} + \left[F'\left(x^{(k)}\right) - 3F'\left(y^{(k)}\right)\right]^{-1}\left[F\left(x^{(k)}\right) + 2F\left(u^{(k)}\right)\right], \\ w^{(k)} &= v^{(k)} - \frac{1}{2}\left[F'\left(x^{(k)}\right)\right]^{-1}\left[5F'\left(x^{(k)}\right) - 3F'\left(y^{(k)}\right)\right]\left[F'\left(x^{(k)}\right)\right]^{-1} F\left(v^{(k)}\right), \quad (3) \\ x^{(k+1)} &= v^{(k)} - \left[F'\left(\frac{w^{(k)} + v^{(k)}}{2}\right)\right]^{-1} F(v^{(k)}). \end{aligned}$$

If we analyze the efficiency indices (see Figure 1), we deduce the following conclusions: the new methods M4, M6 and M8 (and also the pseudocomposed PsM10 and PsM14) improve Newton and Jarratt's schemes (in fact, the indices of M4 and Jarratt's are equal). Indeed, for $n \geq 3$ the best index is that of M8. Nevertheless, none of the pseudocomposed methods improve the efficiency index of their original partners.

The advantage of pseudocomposition can be observed in Figures 2a, 2b (methods M6 and PsM10) and Figures 3a, 3b (methods M8 and PsM14) where the dynamical plane on R^2 is shown: let us consider a system of two equations and two unknowns (the case $\overline{F}(x_1, x_2) = (x_1^2 - x_1 - x_2^2 - 1, -\sin(x_1) + x_2$ is represented, being the solutions $\xi_1 \approx (-0.845257, -0.748141)^T$ and $\xi_2 \approx (1.952913, 0.927877)^T$, marked with a white star in Figures 2 and 3). For any initial estimation in R^2 represented by its position in the plane, a different color (blue or orange, as there exist only two solutions) is used for the different solutions found (marked by a white point in the figure). Black color represents an initial point in which the method converges to infinity, and the green one means that no convergence is found (usually because any linear system cannot be solved). It is clear that when



(a) $M6$ (b) $PsM10$

Figure 2: Real dynamical planes for system $\overline{F}$ and methods M6 and PsM10

many initial estimations tend to infinity (see Figure 3a), the pseudocomposition "cleans" the dynamical plane, making the method more stable as it can find one of the solutions by using starting points that do not allow convergence with the original scheme (see Figure 2b).

We conclude that the presented schemes M4, M6 and M8 show to be excellent, in terms of order of convergence and efficiency, but also that the pseudocomposition technique achieves to transform them in competent and more robust new schemes.

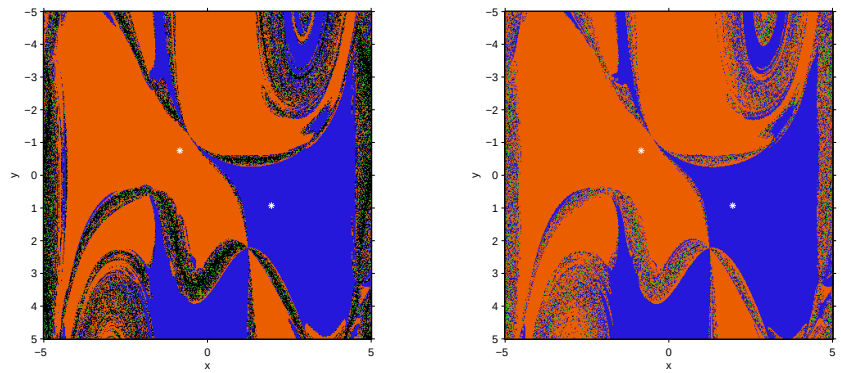
Acknowledgments: This research was supported by Ministerio de Ciencia y Tecnología MTM2011-28636-C02-02 and by Vicerrectorado de Investigación, Universitat Politècnica de València PAID-06-2010-2285.

References

[1] D.K.R. Babajee, M.Z. Dauhoo, M.T. Darvishi, A. Karami, A. Barati, *Analysis of two Chebyshev-like third order methods free from second derivatives for solving systems of nonlinear equations*, Journal of Computational and Applied Mathematics **233**(8) (2010) 2002–2012.

[2] A. Cordero, J.L. Hueso, E. Martínez, J.R. Torregrosa, *A modified Newton-Jarratt's composition*, Numer. Algor. **55** (2010) 87–99.

[3] M.T. Darvishi, A. Barati, *A fourth-order method from quadrature formulae to solve systems of nonlinear equations*, Applied Mathematics and Computation **188**(1) (2007) 257–261.



(a) $M8$ (b) $PsM14$

Figure 3: Real dynamical planes for system $\overline{F}$ and methods M8 and PsM14

[4] M. Nikkhah-Bahrami, R. Oftadeh, *An effective iterative method for computing real and complex roots of systems of nonlinear equations*, Applied Mathematics and Computation **215** (2009) 1813–1820.

[5] B. Shin, M.T. Darvishi, C. Kim, *A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems*, Applied Mathematics and Computation **217** (2010) 3190–3198.

[6] A. Cordero, J.L. Hueso, E. Martínez, J.R. Torregrosa, *Efficient high-order methods based on golden ratio for nonlinear systems*, Applied Mathematics and Computation **217**(9) (2011) 4548–4556.

[7] A. Cordero, E. Martínez, J.R. Torregrosa, *Iterative methods of order four and five for systems of nonlinear equations*, Journal of Computational and Applied Mathematics **231** (2009) 541–551.

[8] D.K.R. Babajee, M.Z. Dauhoo, M.T. Darvishi, A. Barati, *A note on the local convergence of iterative methods based on Adomian decomposition method and 3-node quadrature rule*, Applied Mathematics and Computation **200**(1) (2008) 452–458.

[9] M.P. Vassileva, *Métodos iterativos eficientes para la resolución de sistemas no lineales*, Ph.D. Universitat Politècnica de València 2011.

[10] A. M. Ostrowski, *Solutions of equations and systems of equations*, Academic Press New York-London, 1966.

[11] A. Cordero, J.R. Torregrosa, *On interpolation variants of Newton's method for functions of several variables*, Journal of Computational and Applied Mathematics **234** (2010) 34–43.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Cycles of period two in the family of Chebyshev-Halley type methods

Alicia Cordero[1], Juan R. Torregrosa[1] and Pura Vindel[2]

[1] *Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera, s/n, 46022 València, Spain*

[2] *Instituto de Matemáticas y Aplicaciones de Castellón, Universitat Jaume I, Av. de Vicent Sos Baynat s/n, 12071 Castellò de la Plana, Spain*

emails: acordero@mat.upv.es, jrtorre@mat.upv.es, vindel@uji.es

Abstract

In this paper, 2-cycles of the Chebyshev-Halley family are studied on quadratic polynomials. This analysis has been made on the basis of the parameter space, described as "cat set". Some regions of this set are the loci of values of the parameter α that give rise to iterative methods of the family with problems of convergence.

Key words: Nonlinear equations, iterative methods, complex dynamics.

1 Introduction

The application of iterative methods for solving nonlinear equations $f(z) = 0$, $f : \mathbb{C} \to \mathbb{C}$, gives rise to rational functions whose dynamical behavior provide us important information about the stability and reliability of the corresponding iterative scheme. The best known iterative method, under the dynamical point of view, is Newton's scheme (see, for example, [4]).

It is known that parameter space of Newton's process applied on $p(z) = z^2 + c$ gives the Mandelbrot set (see [7]). This set has been widely studied (see, for example [6]), analyzing the dynamical behavior of the method for values of the parameter c in the different regions of the parameter space. For example, the bulbs rounding the main body of Mandelbrot set contain values of c for which Newton's procedure has periodic orbits, of several periods.

This study has been extended by different authors to other point-to-point iterative methods for solving nonlinear equations (see, for example [1], [2] and, more recently, [8] and

[10]). Some of the classical iterative schemes for solving nonlinear equations are included in the parametric family of Chebyshev-Halley, whose dynamical analysis has been started in [5].

The family of Chebyshev-Halley type methods can be written as the iterative scheme

$$z_{n+1} = z_n - \left(1 + \frac{1}{2}\frac{L_f(z_n)}{1 - \alpha L_f(z_n)}\right)\frac{f(z_n)}{f'(z_n)}, \tag{1}$$

where

$$L_f(z) = \frac{f(z)\, f''(z)}{(f'(z))^2}$$

and α is a complex parameter.

The corresponding fixed point operator is

$$G(z) = z - \left(1 + \frac{1}{2}\frac{L_f(z)}{1 - \alpha L_f(z)}\right)\frac{f(z)}{f'(z)}. \tag{2}$$

In [5] the dynamics of this operator when it is applied on quadratic polynomial $p(z) = z^2 + c$ has been initially studied. For this polynomial, the operator (2) correspond to the rational function:

$$G_p(z) = \frac{z^4(-3 + 2\alpha) + 6cz^2 + c^2(1 - 2\alpha)}{4z(z^2(-2 + \alpha) + \alpha c)},$$

depending on parameters α and c.

The parameter c can be obviated by considering the conjugacy map

$$h(z) = \frac{z - i\sqrt{c}}{z + i\sqrt{c}},$$

with the properties $h(\infty) = 1$, $h(i\sqrt{c}) = 0$ and $h(-i\sqrt{c}) = \infty$.

Then, the operator becomes a one-parametric function described by

$$O_p(z) = z^3\frac{z - 2(\alpha - 1)}{1 - 2(\alpha - 1)z}. \tag{3}$$

Now, let us recall some basic concepts on complex dynamics (see [3]). Given a rational function $R : \hat{\mathbb{C}} \to \hat{\mathbb{C}}$, where $\hat{\mathbb{C}}$ is the Riemann sphere, the *orbit of a point* $z_0 \in \hat{\mathbb{C}}$ is defined as:

$$z_0,\, R(z_0),\, R^2(z_0), ..., R^n(z_0), ...$$

We are interested in the study of the asymptotic behavior of the orbits depending on the initial condition z_0, that is, we are going to analyze the phase plane of the map R defined by the different iterative methods.

To obtain these phase spaces, the first of all is to classify the starting points from the asymptotic behavior of their orbits.

 ISBN:978-84-615-5392-1

A $z_0 \in \hat{\mathbb{C}}$ is called a *fixed point* if it satisfies: $R(z_0) = z_0$. A *periodic point* z_0 of period $p > 1$ is a point such that $R^p(z_0) = z_0$ and $R^k(z_0) \neq z_0$, $k < p$. A *pre-periodic point* is a point z_0 that is not periodic but there exists a $k > 0$ such that $R^k(z_0)$ is periodic. A *critical point* z_0 is a point where the derivative of rational function vanishes, $R'(z_0) = 0$.

On the other hand, a fixed point z_0 is called *attractor* if $|R'(z_0)| < 1$, *superattractor* if $|R'(z_0)| = 0$, *repulsor* if $|R'(z_0)| > 1$ and *parabolic* if $|R'(z_0)| = 1$. The stability of a periodic orbit is defined by the magnitude (lower than 1 or not) of $|R'(z_1) \dots R'(z_p)|$, where $\{z_1, \dots, z_p\}$ are the points of the orbit of period p.

The basin of attraction of an attractor $\bar{z}$ is defined as the set of pre-images of any order:

$$\mathcal{A}(\bar{z}) = \{z_0 \in \hat{\mathbb{C}} \ : \ R^n(z_0) \to \bar{z}, \ n \to \infty\}.$$

The set of points $z \in \hat{\mathbb{C}}$ such that their families $\{R^n(z)\}_{n \in \mathbb{N}}$ are normal in some neighborhood $U(z)$, is the *Fatou set,* $\mathcal{F}(R)$, that is, the Fatou set is composed by the set of points whose orbits tend to an attractor (fixed point, periodic orbit or infinity). Its complement in $\hat{\mathbb{C}}$ is the *Julia set,* $\mathcal{J}(R)$; therefore, the Julia set includes all repelling fixed points, periodic orbits and their pre-images. That means that the basin of attraction of any fixed point belongs to the Fatou set. On the contrary, the boundaries of the basins of attraction belong to the Julia set.

The invariant Julia set for Newton's method is the unit circle S^1 and the Fatou set is defined by the two basins of atraction of the superattractor fixed points: 0 and ∞. On the other hand, the Julia set for Chebyshev's method applied to quadratic polynomials is more complicated than for Newton's method and it has been studied in [9]. These methods are two elements of the family (1).

2 Previous results on Chebyshev-Halley family

Fixed points of the operator $O_p(z)$ are $z = 0$, $z = \infty$, which correspond to the roots of the polynomial, and $z = 1$ and $z = \frac{-3+2\alpha \pm \sqrt{5-12\alpha+4\alpha^2}}{2}$, denoted by s_1 and s_2, respectively.

Moreover, $z = 0$ and $z = \infty$ are superattractors and the stability of the other fixed points is established in the following results, which appear in [5].

Proposition 1 *The fixed point $z = 1$ satisfies the following statements :*

i) If $\left|\alpha - \frac{13}{6}\right| < \frac{1}{3}$, then $z = 1$ is an attractor and, in particular, it is a superattractor for $\alpha = 2$.

ii) If $\left|\alpha - \frac{13}{6}\right| = \frac{1}{3}$, then $z = 1$ is a parabolic point.

iii) If $\left|\alpha - \frac{13}{6}\right| > \frac{1}{3}$, then $z = 1$ is a repulsive fixed point.

 ISBN:978-84-615-5392-1

Proposition 2 *The fixed points $z = s_i$, $i = 1, 2$, satisfy the following statements:*

i) If $|\alpha - 3| < \frac{1}{2}$, then s_1 and s_2 are two different attractive fixed points. In particular, for $\alpha = 3$, s_1 and s_2 are superattractors.

ii) If $|\alpha - 3| = \frac{1}{2}$, then s_1 and s_2 are parabolic points. In particular, for $\alpha = \frac{5}{2}$, $s_1 = s_2 = 1$.

iii) If $|\alpha - 3| > \frac{1}{2}$, then s_1 and s_2 are repulsive fixed points.

On the other hand, the critical points of $O_p(z)$ are $z = 0$, $z = \infty$ and

$$z = \frac{3 - 4\alpha + 2\alpha^2 \pm \sqrt{-6\alpha + 19\alpha^2 - 16\alpha^3 + 4\alpha^4}}{3\,(\alpha - 1)},$$

which are denoted by c_1 and c_2, respectively.

It is known that there is at least one critical point associated with each invariant Fatou component. It is shown in [5] that the critical points c_i, $i = 1, 2$, are inside the basin of attraction of $z = 1$ when it is attractive ($\frac{11}{6} < \alpha < \frac{5}{2}$) and coincide with $z = 1$ for $\alpha = 2$. Then, they move to the basins of attraction of s_1 and s_2 when these fixed points become attractive ($\frac{5}{2} < \alpha < \frac{7}{2}$), critical and fixed points coincide for $\alpha = 3$ and s_1 and s_2 become superattractors.

A powerful tool to analyze the dynamics of the rational function associated to an iterative method is the parameter space (see Figure 1): each point of the parameter plane is associated to a complex value of α, i.e., to an element of family (1). Every value of α belonging to the same connected component of the parameter space give rise to subsets of schemes of family (1) with similar dynamical behavior.

In this parameter space we observe a black figure (let us call it *the cat set*), with a certain similarity with the Mandelbrot set: for values of α outside this cat set the Julia set is disconnected. The two disks in the main body of the cat set correspond to the α values for those the fixed points $z = 1$ (the head) and s_1 and s_2 became attractive (the body). We also observe a curve similar to a circle that passes through the cat's neck, we call it *the necklace*. As we have proved in [5], the parameter space inside this curve is topologically equivalent to a disk.

The head of the cat corresponds to the values of the parameter for which the fixed point $z = 1$ became attractive, that is, for the interval defined by $\left|\alpha - \frac{13}{6}\right| < \frac{1}{3}$. In this case, the fixed point $z = 1$ is an attractor and the other two fixed points s_1 and s_2 are repulsors.

The body of the cat set corresponds to values of the parameter $|\alpha - 3| < \frac{1}{2}$. In this case, the fixed point $z = 1$ is a repulsor and s_1,s_2 are attractors and have their own basin of attraction, one critical point is in each basin.

Let us remark that the intersection point of the head and the body of the cat is in their common boundary and corresponds to $\alpha = \frac{5}{2}$.

 ISBN:978-84-615-5392-1

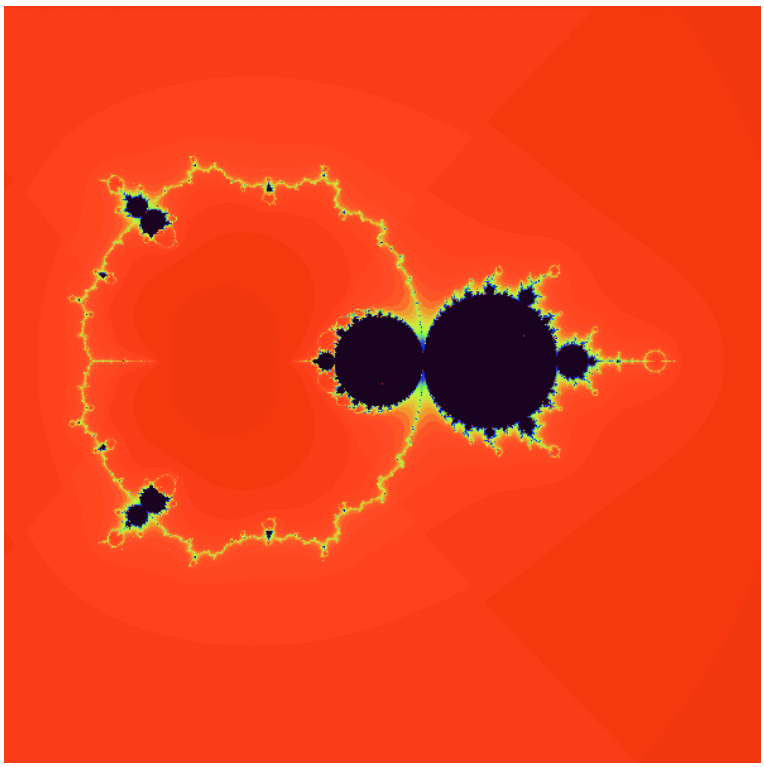


Figure 1: Parameter plane

In fact, similarly to what happen in the Mandelbrot set, the boundary of the cat set is exactly the bifurcation locus of the family of Chebyshev-Halley type family acting on quadratic polynomial; that is, the set of parameters for which the dynamics changes abruptly under small changes of α. Let us observe that the head and the body are surrounded by bulbs, of different sizes, that yield to the appearance of attractive cycles of different periods. In this paper, we are interested in the study of the bulbs involving cycles of period 2.

3 The bulb of period 2 of the head

It is easy to check that $z = 1$ is an hyperbolic point for all these values of α belonging to the circle $\left|\alpha - \frac{13}{6}\right| = \frac{1}{3}$, since they can be expressed $\alpha = \frac{13}{6} + \frac{1}{3}e^{i\theta}$ and

$$O_p'(1) = \frac{2e^{i\theta}+1}{2+e^{i\theta}}, \qquad \left|O_p'(1)\right| = 1.$$

 ISBN:978-84-615-5392-1

Therefore, for different θ values we find some bulbs with attractive cycles surrounding "the head of the cat". In this section we study the bulb of period 2, whose intersection with the head of the cat corresponds to $\alpha = \frac{11}{6}$.

As we have seen in Proposition 1, if $\alpha > \frac{11}{6}$ then $O_p'(1) < 1$, if $\alpha = \frac{11}{6}$ then $O_p'(1) = 1$, and when $\alpha < \frac{11}{6}$ then $O_p'(1) > 1$.

We are going to show that there is a doubling period bifurcation for $\alpha = \frac{11}{6}$. For $\alpha < \frac{11}{6}$ the periodic point $z = 1$ became repulsive and one attractive cycle of period 2 appears.

This cycle has to satisfy the equation:

$$O_p^2(z) = z.$$

The relation $O_p^2(z) - z = 0$ can be factorized as

$$z(-1+z)\left(1+3z-2az+z^2\right) f(\alpha,z)\, g(\alpha,z) = 0,$$

where

$$\begin{aligned} f(\alpha,z) &= \left(1+(3-2\alpha)z+(3-2\alpha)z^2+(3-2\alpha)z^3+z^4\right), \\ g(\alpha,z) &= 1+(3-4\alpha)z+\left(2-6\alpha+4\alpha^2\right)z^2+\left(3-6\alpha+4\alpha^2\right)z^3+ \\ &\quad +\left(9-22\alpha+20\alpha^2-8\alpha^3\right)z^4+\left(3-6\alpha+4\alpha^2\right)z^5+ \\ &\quad +\left(2-6\alpha+4\alpha^2\right)z^6+(3-4\alpha)z^7+z^8. \end{aligned}$$

As we have seen, the product $z(-1+z)\left(1+3z-2\alpha z+z^2\right)$ yields to the fixed points. So, 2-periodic points come from $f(\alpha,z) = 0$ or $g(\alpha,z) = 0$. We observe that $f\left(\frac{11}{6},z\right) = \frac{1}{3}\left(3z^2+4z+3\right)(z-1)^2$ so that the periodic points that collapse with the fixed point $z = 1$ for $\alpha = \frac{11}{6}$ come from the zeros of this function. In fact, we will focus our attention on function $f(\alpha,z)$, as it yields periodic orbits in the bulb of the head, while roots of $g(\alpha,z)$ give rise to 2-orbits in the bulb of the body whose intersection is $\alpha = \frac{7}{2}$.

We obtain a new factorization

$$f(\alpha,z) = f_1(\alpha,z)\, f_2(\alpha,z),$$

where

$$\begin{aligned} f_1(\alpha,z) &= 1+\frac{1}{2}\left(3-2\alpha-\sqrt{5-4\alpha+4\alpha^2}\right)z+z^2, \\ f_2(\alpha,z) &= 1+\frac{1}{2}\left(3-2\alpha+\sqrt{5-4\alpha+4\alpha^2}\right)z+z^2, \end{aligned}$$

and we observe that $f_1\left(\frac{11}{6},z\right) = (1-z)^2$ and $f_2\left(\frac{11}{6},z\right) = \frac{1}{3}\left(3-4z+3z^2\right)$. So, the cycle of period 2 that becomes attractive comes from $f_1(\alpha,z) = 0$.

 ISBN:978-84-615-5392-1

The two solutions are:

$$z_1 = -\frac{3}{4}+\frac{1}{2}\alpha+\frac{1}{4}\sqrt{5-4\alpha+4\alpha^2}+\frac{1}{4}\sqrt{-2-16\alpha+8\alpha^2+(-6+4\alpha)\sqrt{5-4\alpha+4\alpha^2}},$$
$$z_2 = -\frac{3}{4}+\frac{1}{2}\alpha+\frac{1}{4}\sqrt{5-4\alpha+4\alpha^2}-\frac{1}{4}\sqrt{-2-16\alpha+8\alpha^2+(-6+4\alpha)\sqrt{5-4\alpha+4\alpha^2}}.$$

Moreover, these two solutions are the cycle of period two because they satisfy:

$$O_p(z_1)=z_1^3\frac{z_1-2(\alpha-1)}{1-2(\alpha-1)z_1}=z_2,\quad O_p(z_2)=z_2^3\frac{z_2-2(\alpha-1)}{1-2(\alpha-1)z_2}=z_1.$$

The stability of this 2-cycle is a function of α:

$$S(\alpha) = O_p'(z_1)\cdot O_p'(z_2)= \tag{4}$$
$$= \frac{-54+132\alpha-166\alpha^2+112\alpha^3-40\alpha^4+6(\alpha-1)(3-\alpha+2\alpha^2)\sqrt{5-4\alpha+4\alpha^2}}{-9+26\alpha-34\alpha^2+23\alpha^3-8\alpha^4+2(\alpha-1)(2-3\alpha+2\alpha^2)\sqrt{5-4\alpha+4\alpha^2}}.$$

To know the size of the bulb where this cycle is attractive, we study the boundary where it is parabolic, that is, the values of α such that

$$|S(\alpha)|=\left|O_p'(z_1)\cdot O_p'(z_2)\right|=1.$$

If we consider α real, the above expression gives

$$(1-2\alpha)^2(6\alpha-11)\left(-19+22\alpha-20\alpha^2+8\alpha^3\right)=0.$$

So, $\alpha=\frac{11}{6}$ gives one real point of the bulb and the other real value is given by the only real solution of $-19+22\alpha-20\alpha^2+8\alpha^3=0$, that is,

$$\alpha^*=\frac{1}{6}\sqrt[3]{\left(134+18\sqrt{57}\right)}-\frac{4}{3\sqrt[3]{\left(134+18\sqrt{57}\right)}}+\frac{5}{6}\approx 1.704\,1.$$

Moreover, we prove that this cycle is attractive in the interval $\alpha^*<\alpha<\frac{11}{6}$ by drawing the function $S(\alpha)$ in this interval (see Figure 2).

It is known that there is one value where this cycle is superattractive, that coincides with the minimum of the function $S(\alpha)$,

$$S(\alpha)=0\Rightarrow 16(-1+2\alpha)^2(-9+12\alpha-11\alpha^2+4\alpha^3)^2=0.$$

The only real root in the interval $]\alpha^*,\frac{11}{6}[$ is

$$\alpha=\frac{1}{12}\left(11-\frac{23}{\sqrt[3]{899+36\sqrt{633}}}+\sqrt[3]{899+36\sqrt{633}}\right),\quad \alpha\approx 1.77383.$$

 ISBN:978-84-615-5392-1

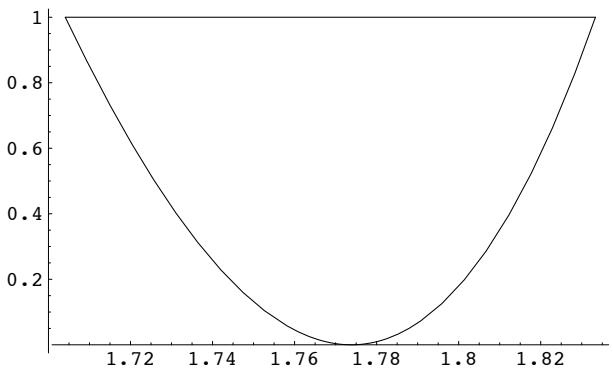


Figure 2: $S(\alpha)$ for α real

Let

$$\alpha_0^* = \frac{1}{2}\left(\frac{8}{3} - \frac{2\sqrt[3]{4}}{3\sqrt[3]{67+9\sqrt{57}}} + \frac{\sqrt[3]{67+9\sqrt{57}}}{3\sqrt[3]{4}}\right) \approx 1.76871,$$

be the middle point of α^* and $\frac{11}{6}$. It is known that the boundary of the bulb satisfy $|S(\alpha)| = 1$. It is not a circle but there exists a 2-ball centered in α_0^*, with radius $r = 0.064$ where the 2-cycles are attractive. We can see it by evaluating $|S(\alpha)|$ in the points $\alpha = \alpha_0^* + 0.064e^{it\pi}$, where $0 \leq t \leq 2$ and the step size is $h = 0.1$. The values are: $0.977843, 0.979721, 0.984619, 0.99065, 0.995595, 0.997797, 0.996763, 0.993236, 0.988793,$ $0.985214, 0.983854, 0.985214, 0.988793, 0.993236, 0.996763, 0.997797, 0.995595, 0.99065,$ $0.984619, 0.979721, 0.977843$.

A similar study can be made on $g(\alpha, z)$, in order to obtain cycles in the bulb of period 2 surrounding the body of the cat set.

4 Conclusions

The cat set as parameter space of the Chebyshev-Halley family on quadratic polynomials is dynamically very wealthy, as it happens with Mandelbrot set. The head and the body of the cat set are surrounded by bulbs of different sizes. Some of them have been analyzed in this work, obtaining cycles of period two for several values of the parameter, that is, for different members of the family of iterative methods.

Acknowledgments: This research was supported by Ministerio de Ciencia y Tecnología MTM2011-28636-C02-02 and by Vicerrectorado de Investigación, Universitat Politècnica de València PAID-06-2010-2285.

References

[1] S. Amat, C. Bermúdez, S. Busquier and S. Plaza, *On the dynamics of the Euler iterative function*, Applied Mathematics and Computation **197** (2008) 725–732.

[2] S. Amat, S. Busquier and S. Plaza, *A construction of attracting periodic orbits for some classical third-order iterative methods*, J. of Computational and Applied Math. **189** (2006) 22–33.

[3] P. Blanchard, *Complex Analytic Dynamics on the Riemann Sphere*, Bull. of the AMS **11**(1) (1984) 85–141.

[4] P. Blanchard, *The Dynamics of Newton's Method*, Proc. of Symposia in Applied Math. **49** (1994) 139–154.

[5] A. Cordero, J.R. Torregrosa, P. Vindel, *Dynamics of a family of Chebyshev-Halley-type method*, Applied Mathematics and Computation submitted.

[6] R.L. Devaney, *An introduction to chaotic dynamical systems*, Addinson-Wesley Publishing Company, 1989.

[7] R.L. Devaney, *The Mandelbrot Set, the Farey Tree and the Fibonacci sequence*, Am. Math. Monthly **106**(4) (1999) 289–302.

[8] J. M. Gutiérrez, M. A. Hernández and N. Romero, *Dynamics of a new family of iterative processes for quadratic polynomials*, J. of Computational and Applied Math. **233** (2010) 2688–2695.

[9] K. Kneisl, *Julia sets for the super-Newton method, Cauchy's method and Halley's method*, Chaos **11**(2) (2001) 359–370.

[10] S. Plaza and N. Romero, *Attracting cycles for the relaxed Newton's method*, J. of Computational and Applied Math. **235** (2011) 3238–3244.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Modelling the dynamics of the students academic performance in the German region of North Rhine-Westphalia

Juan-Carlos Cortés[1], Matthias Ehrhardt[2], Almudena Sánchez-Sánchez[1], Francisco-José Santonja[3] and Rafael-Jacinto Villanueva[1]

[1] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Spain*

[2] *Fachbereich C - Mathematik und Naturwissenschaften, Angewandte Mathematik und Numerische Analysis, Bergische Universität Wuppertal, Germany*

[3] *Departamento de Estadística e Investigación Operativa, Universitat de València, Spain*

emails: jccortes@imm.upv.es, ehrhardt@math.uni-wuppertal.de, alsncsnc@posgrado.upv.es, francisco.santonja@uv.es, rjvillan@imm.upv.es

Abstract

Academic underachievement is a concern of paramount importance in Europe, where around 15% of the students in the last courses in high school do not achieve the minimum knowledge academic requirement. In this paper, we propose a model based on a system of differential equations to study the dynamics of the students academic performance in the German region of North Rhine-Westphalia. This approach is supported by the idea that both, good and bad study habits, are a mixture of personal decisions and influence of classmates. This model may permit to forecast trends in the next few years.

Key words: Academic Performance, Modelling, System of Differential Equations, Forecasting in Social Sciences.

1 Introduction

In many countries of the European Union, in the last courses of high school, the rates of academic underachievement are at very worrying levels [1, 2, 3, 4, 5]. The concern about the high level of academic underachievement is completely justified, not only by the high

rates but also by the negative effects on the country's economic development, especially in the unemployment and its serious consequences. Nowadays, the job opportunities of people depend on their qualification, their ability to acquire, use and interprete the information, including their skills to adapt the new knowledge to a very demanding and competitive society in constant change. In order to acquire them, students go to basic schools first and high schools later, learning the contents determined in the corresponding legislations.

The main goal of the last high school courses is to provide the students a proper educational training to consolidate the intellectual maturity of the pupils, increasing their specific knowledge as well as boosting the development of abilities that help them to join up either the labor market or higher studies. For all these reasons, this educational level is considered a milestone to students because it represents a period to make important decisions about academic and professional future.

According to the Vygotsky learning theories [6, 7] and the recent studies published by Christakis and Fowler [8], habits and behavior may be socially transmitted, in particular, academic and study habits.

Taking into account this approach, in this paper, we are going to focus in the German region of North Rhine-Westphalia and propose a model to study the evolution of the students academic performance in the last three courses of the high school (levels 11, 12 and 13) before accessing to the university by most of the students, using techniques of mathematical epidemiology. This approach may be of relevant interest because a new studies plan will come into force next year in North Rhine-Westphalia and the model forecasting academic results could be compared to the real ones corresponding to the new plan in order to evaluate if the change has been as good as expected.

Some examples of social problems approached using type-epidemiological mathematical models are encountered in obesity [9, 10], alcoholism [11], drug abuse [12], shopaholism [13], spread of ideas [14], evaluation of law effects on societies [15], and so on.

2 Model building

2.1 Available data

We say that a student *promotes* if, in case the course finishes now, he or she will pass to the next level or graduate satisfying the current legislation into force in North Rhine-Westphalia. Otherwise, this student is in *non-promote* group. The legislation establishes that the grades in North Rhine-Westphalia are "very good" (1), "good" (2), "satisfactory" (3), "sufficient" (4), "bad" (5) and "very bad" (6). A student in level 11 and 12 does not promote to next level if he/she has in 2 or more main subjects (like Maths, Physics, German, English) or in 3 or more minor subjects (like music, arts, sports), a grade of 5 or 6. In case the student is in the last level (level 13), he/she has to pass all the subjects to obtain the grade [16, 17].

The available data that we have considered in this paper correspond to the academic results belonging to the students of the last three courses of high schools during the academic years from $2006-2007$ to $2010-2011$, in both, state and private high schools all over North Rhine-Westphalia, divided by gender, level and promote/non-promote. The corresponding data can be seen in Table 1 [18].

GIRLS		2006–2007	2007–2008	2008–2009	2009–2010	2010–2011
Level 11	% Promote	19.37	19.09	19.1	19.24	18.27
	% Non–Promote	0.81	0.67	0.59	0.53	0.44
Level 12	% Promote	18.23	17.96	18.15	17.77	18.29
	% Non–Promote	0.75	0.68	0.58	0.47	0.47
Level 13	% Promote	15.34	15.96	15.94	16.25	16.44
	% Non–Promote	0.25	0.25	0.19	0.19	0.17
BOYS		2006–2007	2007–2008	2008–2009	2009–2010	2010–2011
Level 11	% Promote	16.05	15.92	15.95	16.3	15.87
	% Non–Promote	0.96	0.88	0.81	0.73	0.6
Level 12	% Promote	14.7	14.73	14.77	14.72	15.21
	% Non–Promote	0.85	0.81	0.67	0.67	0.64
Level 13	% Promote	12.38	12.77	13.04	12.94	13.39
	% Non–Promote	0.31	0.28	0.21	0.19	0.21

Table 1: The available data corresponding to levels 11, 12 and 13, in both, state and private high schools all over North Rhine-Westphalia from academic year $2006-2007$ to $2010-2011$ divided by gender, level and promote/non-promote over the total number of students in the three levels.

2.2 The type-epidemiological model

We build our mathematical model following an epidemiological approach considering that the academic performance of a student, Girl (G) or Boy (B), is a mixture of her/his own study habits and his/her classmates study habits, good or bad. In our model, we assume that the transmission of good and bad academic habits is caused by the social contact between students who belong to the same academic level [8, 7, 19].

The subpopulations of the model will be (time t in years and $i = 1$ for level 11, $i = 2$ for level 12 and $i = 3$ for level 13):

- $G_i = G_i(t)$ is the number of girls of level i who promote at time instant t.
- $B_i = B_i(t)$ is the number of boys of level i who promote at time instant t.
- $\overline{G}_i = \overline{G}_i(t)$ is the number of girls of level i who do not promote at time instant t.
- $\overline{B}_i = \overline{B}_i(t)$ is the number of boys of level i who do not promote at time instant t.

Furthermore, we consider the following assumptions to design the model:

 ISBN:978-84-615-5392-1

- Let us assume a homogeneous population mixing, i.e., each student can contact with any other student in his/her class [20].

- *Negative autonomous decision*: For each academic level, $i = 1, 2, 3$, students belonging to the promotable groups G_i or B_i may change their personal study habits and this change may lead them to obtain bad academic results, moving to $\overline{G}_i$ or $\overline{B}_i$. We assume that this transition is proportional to the number of pupils in G_i and B_i, and it is modelled by the linear terms $\alpha_i^G G_i$ and $\alpha_i^B B_i$. According to educational experts, it is assumed that the academic attitude is different in the same educational level depending on gender: girls are usually more responsible for their academic performance than boys [21]. This leads us to suppose the following restrictions:

$$\alpha_1^G < \alpha_1^B,\ \alpha_2^G < \alpha_2^B,\ \alpha_3^G < \alpha_3^B. \tag{1}$$

  In addition we will assume that:

$$\alpha_1^G > \alpha_2^G > \alpha_3^G,\ \alpha_1^B > \alpha_2^B > \alpha_3^B, \tag{2}$$

  because students in the higher levels are more mature than their mates in the lower levels [21].

- *Negative habits transmission*: For each academic level, $i = 1, 2, 3$, students in G_i or B_i may move to the non–promotable group, $\overline{G}_i$ or $\overline{B}_i$ respectively, due to the negative influence transmitted by encounters between students (girls and boys) in the non–promotable group in the same academic level. Hence, these transitions are modelled by the nonlinear terms $\beta_i^{G\overline{G}} G_i \overline{G}_i + \beta_i^{G\overline{B}} G_i \overline{B}_i$ and $\beta_i^{B\overline{G}} B_i \overline{G}_i + \beta_i^{B\overline{B}} B_i \overline{B}_i$, where $\beta_i^{G\overline{G}}$, $\beta_i^{G\overline{B}}$, $\beta_i^{B\overline{G}}$ and $\beta_i^{B\overline{B}}$ are the corresponding transmission rates where the first letter in the superindexes denotes the group susceptible to acquire bad study habits and the second one denotes the group that transmit those bad study habits. All specific factors and social encounters involved in the transmission of the bad academic habits are embedded in β parameters.

- *Positive autonomous decision*: Analogously to *negative autonomous decision*, students belonging to the non–promotable groups may change their personal behavior towards their study habits and this change may lead the students to improve their academic results, moving to G_i or B_i. We assume that this transition is proportional to the number of pupils in $\overline{G}_i$ and $\overline{B}_i$, and it is modelled by the linear terms $\gamma_i^G \overline{G}_i$ and $\gamma_i^B \overline{B}_i$.

- *Positive habits transmission*: Students in non–promotable group may move to the promotable groups due to the positive influence transmitted in the encounters between students (girls and boys) in the promotable group in the same academic level.

Hence, these transitions are modelled by the nonlinear terms $\delta_i^{\overline{G}G}\overline{G}_i G_i + \delta_i^{\overline{G}B}\overline{G}_i B_i$ and $\delta_i^{\overline{B}G}\overline{B}_i G_i + \delta_i^{\overline{B}B}\overline{B}_i B_i$. The interpretation of the transmission rate parameters is the same as in the *negative habits transmission*.

- *Passing courses and graduation*: The students in G_i and B_i, in September, transit automatically to next level G_{i+1} and B_{i+1}, respectively, for $i = 1, 2$. Students in G_3 and B_3 will graduate in September. These transitions are modelled by εG_1, εG_2, εG_3, εB_1, εB_2, εB_3, where

$$\varepsilon = \begin{cases} 1 & \text{if} \quad \dfrac{9}{12} + j \leq t \leq \dfrac{10}{12} + j, \\ 0 & \text{otherwise,} \end{cases}$$

  where $j = 0, 1, 2, 3, 4$, correspond to academic years 2006–2007, ..., 2010–2011, respectively.

- *Abandon*: For each academic level, $i = 1, 2, 3$, a proportion of the students in $\overline{G}_i$ or $\overline{B}_i$ with bad academic results may leave their studies by autonomous decision. This situation is modelled by the linear terms $\eta_i^G \overline{G}_i$ and $\eta_i^B \overline{B}_i$. We also assume that these transitions are proportional to the number of pupils in $\overline{G}_i$ and $\overline{B}_i$.

- *Access*: New students enter into the level 11 in the month of September in the promotable groups of girls and boys. It is modelled by the functions

$$\sigma^G = \begin{cases} \tau^G & \text{if} \quad \dfrac{9}{12} + j \leq t \leq \dfrac{10}{12} + j, \\ 0 & \text{otherwise,} \end{cases} \quad \sigma^B = \begin{cases} \tau^B & \text{if} \quad \dfrac{9}{12} + j \leq t \leq \dfrac{10}{12} + j, \\ 0 & \text{otherwise,} \end{cases}$$

  where $j = 0, 1, 2, 3, 4$, correspond to academic years 2006–2007, ..., 2010–2011, respectively, and τ^G and τ^B to be determined.

Thus, under the above assumptions we build the nonlinear system of ordinary differential equations (3)-(5) in order to describe the dynamics of students academic performance in the German region of North Rhine-Westphalia.

 ISBN:978-84-615-5392-1

$$
\begin{aligned}
G_1'(t) &= \sigma^G - \varepsilon G_1(t) - \alpha_1^G G_1(t) + \gamma_1^G \overline{G}_1(t) \\
&- \left[\beta_1^{G\overline{G}} G_1(t)\frac{\overline{G}_1(t)}{T(t)} + \beta_1^{G\overline{B}} G_1(t)\frac{\overline{B}_1(t)}{T(t)}\right] + \left[\delta_1^{\overline{G}G}\overline{G}_1(t)\frac{G_1(t)}{T(t)} + \delta_1^{\overline{G}B}\overline{G}_1(t)\frac{B_1(t)}{T(t)}\right], \\
\overline{G}_1'(t) &= \alpha_1^G G_1(t) - \gamma_1^G \overline{G}_1(t) - \eta_1^G \overline{G}_1(t) \\
&+ \left[\beta_1^{G\overline{G}} G_1(t)\frac{\overline{G}_1(t)}{T(t)} + \beta_1^{G\overline{B}} G_1(t)\frac{\overline{B}_1(t)}{T(t)}\right] - \left[\delta_1^{\overline{G}G}\overline{G}_1(t)\frac{G_1(t)}{T(t)} + \delta_1^{\overline{G}B}\overline{G}_1(t)\frac{B_1(t)}{T(t)}\right], \\
G_2'(t) &= \varepsilon G_1(t) - \varepsilon G_2(t) - \alpha_2^G G_2(t) + \gamma_2^G \overline{G}_2(t) \\
&- \left[\beta_2^{G\overline{G}} G_2(t)\frac{\overline{G}_2(t)}{T(t)} + \beta_2^{G\overline{B}} G_2(t)\frac{\overline{B}_2(t)}{T(t)}\right] + \left[\delta_2^{\overline{G}G}\overline{G}_2(t)\frac{G_2(t)}{T(t)} + \delta_2^{\overline{G}B}\overline{G}_2(t)\frac{B_2(t)}{T(t)}\right], \\
\overline{G}_2'(t) &= \alpha_2^G G_2(t) - \gamma_2^G \overline{G}_2(t) - \eta_2^G \overline{G}_2(t) \\
&+ \left[\beta_2^{G\overline{G}} G_2(t)\frac{\overline{G}_2(t)}{T(t)} + \beta_2^{G\overline{B}} G_2(t)\frac{\overline{B}_2(t)}{T(t)}\right] - \left[\delta_2^{\overline{G}G}\overline{G}_2(t)\frac{G_2(t)}{T(t)} + \delta_2^{\overline{G}B}\overline{G}_2(t)\frac{B_2(t)}{T(t)}\right], \\
G_3'(t) &= \varepsilon G_2(t) - \varepsilon G_3(t) - \alpha_3^G G_3(t) + \gamma_3^G \overline{G}_3(t) \\
&- \left[\beta_3^{G\overline{G}} G_3(t)\frac{\overline{G}_3(t)}{T(t)} + \beta_3^{G\overline{B}} G_3(t)\frac{\overline{B}_3(t)}{T(t)}\right] + \left[\delta_3^{\overline{G}G}\overline{G}_3(t)\frac{G_3(t)}{T(t)} + \delta_3^{\overline{G}B}\overline{G}_3(t)\frac{B_3(t)}{T(t)}\right], \\
\overline{G}_3'(t) &= \alpha_3^G G_3(t) - \gamma_3^G \overline{G}_3(t) - \eta_3^G \overline{G}_3(t) \\
&+ \left[\beta_3^{G\overline{G}} G_3(t)\frac{\overline{G}_3(t)}{T(t)} + \beta_3^{G\overline{B}} G_3(t)\frac{\overline{B}_3(t)}{T(t)}\right] - \left[\delta_3^{\overline{G}G}\overline{G}_3(t)\frac{G_3(t)}{T(t)} + \delta_3^{\overline{G}B}\overline{G}_3(t)\frac{B_3(t)}{T(t)}\right],
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
B_1'(t) &= \sigma^B - \varepsilon B_1(t) - \alpha_1^B B_1(t) + \gamma_1^B \overline{B}_1(t) \\
&- \left[\beta_1^{B\overline{G}} B_1(t)\frac{\overline{G}_1(t)}{T(t)} + \beta_1^{B\overline{B}} B_1(t)\frac{\overline{B}_1(t)}{T(t)}\right] + \left[\delta_1^{\overline{B}G}\overline{B}_1(t)\frac{G_1(t)}{T(t)} + \delta_1^{\overline{B}B}\overline{B}_1(t)\frac{B_1(t)}{T(t)}\right], \\
\overline{B}_1'(t) &= \alpha_1^B B_1(t) - \gamma_1^B \overline{B}_1(t) - \eta_1^B \overline{B}_1(t) \\
&+ \left[\beta_1^{B\overline{G}} B_1(t)\frac{\overline{G}_1(t)}{T(t)} + \beta_1^{B\overline{B}} B_1(t)\frac{\overline{B}_1(t)}{T(t)}\right] - \left[\delta_1^{\overline{B}G}\overline{B}_1(t)\frac{G_1(t)}{T(t)} + \delta_1^{\overline{B}B}\overline{B}_1(t)\frac{B_1(t)}{T(t)}\right], \\
B_2'(t) &= \varepsilon B_1(t) - \varepsilon B_2(t) - \alpha_2^B B_2(t) + \gamma_2^B \overline{B}_2(t) \\
&- \left[\beta_2^{B\overline{G}} B_2(t)\frac{\overline{G}_2(t)}{T(t)} + \beta_2^{B\overline{B}} B_2(t)\frac{\overline{B}_2(t)}{T(t)}\right] + \left[\delta_2^{\overline{B}G}\overline{B}_2(t)\frac{G_2(t)}{T(t)} + \delta_2^{\overline{B}B}\overline{B}_2(t)\frac{B_2(t)}{T(t)}\right], \\
\overline{B}_2'(t) &= \alpha_2^B B_2(t) - \gamma_2^B \overline{B}_2(t) - \eta_2^B \overline{B}_2(t) \\
&+ \left[\beta_2^{B\overline{G}} B_2(t)\frac{\overline{G}_2(t)}{T(t)} + \beta_2^{B\overline{B}} B_2(t)\frac{\overline{B}_2(t)}{T(t)}\right] - \left[\delta_2^{\overline{B}G}\overline{B}_2(t)\frac{G_2(t)}{T(t)} + \delta_2^{\overline{B}B}\overline{B}_2(t)\frac{B_2(t)}{T(t)}\right], \\
B_3'(t) &= \varepsilon B_2(t) - \varepsilon B_3(t) - \alpha_3^B B_3(t) + \gamma_3^B \overline{B}_3(t) \\
&- \left[\beta_3^{B\overline{G}} B_3(t)\frac{\overline{G}_3(t)}{T(t)} + \beta_3^{B\overline{B}} B_3(t)\frac{\overline{B}_3(t)}{T(t)}\right] + \left[\delta_3^{\overline{B}G}\overline{B}_3(t)\frac{G_3(t)}{T(t)} + \delta_3^{\overline{B}B}\overline{B}_3(t)\frac{B_3(t)}{T(t)}\right], \\
\overline{B}_3'(t) &= \alpha_3^B B_3(t) - \gamma_3^B \overline{B}_3(t) - \eta_3^B \overline{B}_3(t) \\
&+ \left[\beta_3^{B\overline{G}} B_3(t)\frac{\overline{G}_3(t)}{T(t)} + \beta_3^{B\overline{B}} B_3(t)\frac{\overline{B}_3(t)}{T(t)}\right] - \left[\delta_3^{\overline{B}G}\overline{B}_3(t)\frac{G_3(t)}{T(t)} + \delta_3^{\overline{B}B}\overline{B}_3(t)\frac{B_3(t)}{T(t)}\right],
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
T(t) &= G_1(t) + \overline{G}_1(t) + B_1(t) + \overline{B}_1(t) + G_2(t) + \overline{G}_2(t) + B_2(t) + \overline{B}_2(t) \\
&+ G_3(t) + \overline{G}_3(t) + B_3(t) + \overline{B}_3(t).
\end{aligned}
\tag{5}
$$

The flow diagram, associated to the above model, is plotted in Figure 1.

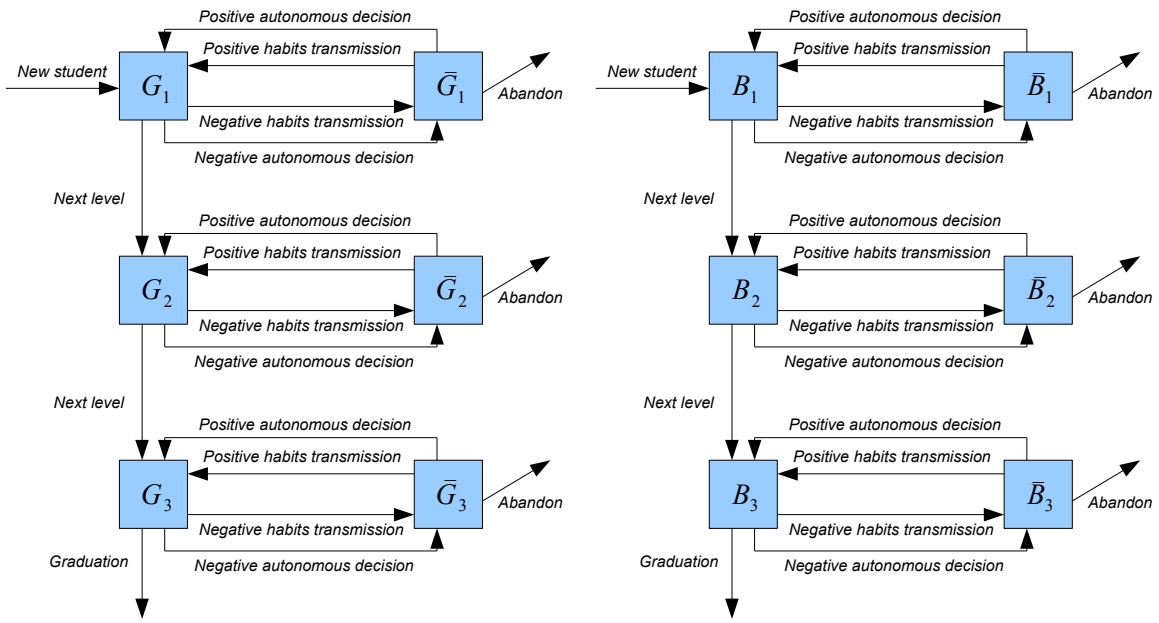


Figure 1: Flow diagram of the model (3)-(5). The boxes represent the students depending on their gender, level and academic results. The arrows denote the transit of students labelled by the cause of the flow.

3 Scaling, fitting and predictions

Data in Table 1 are in percentages meanwhile model (3)-(5) is referred to number of students. It leads us to transform (scaling) the model into the same units as data in order to fit the model with the data. To do that, we follow the techniques developed in [22, 23] about how to scale models where the population is varying in size. Here, we are not going to show the process and the scaled model because it is a technical transformation, the resulting equations are more complex and longer and does not provide extra information about the model. Moreover, the scaled model has the same parameters as the non-scaled model with the same meaning. In order to avoid introducing new notation, we are going to consider that the subpopulations $G_1(t)$, $\overline{G}_1(t)$, $B_1(t)$, $\overline{B}_1(t)$, $G_2(t)$, $\overline{G}_2(t)$, $B_2(t)$, $\overline{B}_2(t)$, $G_3(t)$, $\overline{G}_3(t)$, $B_3(t)$, $\overline{B}_3(t)$ correspond to the percentage of Girls and Boys in the promotable and non–promotable groups in the levels 11, 12 and 13.

Now, compute the model parameters that best fit the scaled model with the available data collected in Table 1 in the mean square sense. Computations have been carried out with *Mathematica 8.0* [24] and the estimated model parameters are:

- *Negative autonomous decision*:
  - Girls per level: $\alpha_1^G = 0.00257431$, $\alpha_2^G = 0.000479681$, $\alpha_3^G = 0.0000980351$.
  - Boys per level: $\alpha_1^B = 0.000518445$, $\alpha_2^B = 0.000462886$, $\alpha_3^B = 0.0000783883$.
- *Negative habits transmission*:
  - Girls per level: $\beta_1^{G\overline{G}} = 0.128823$, $\beta_1^{G\overline{B}} = 0.146999$, $\beta_2^{G\overline{G}} = 0.115597$, $\beta_2^{G\overline{B}} = 0.0940018$, $\beta_3^{G\overline{G}} = 0.128018$, $\beta_3^{G\overline{B}} = 0.0465132$.
  - Boys per level: $\beta_1^{B\overline{G}} = 0.124969$, $\beta_1^{B\overline{B}} = 0.0247756$, $\beta_2^{B\overline{G}} = 0.0406373$, $\beta_2^{B\overline{B}} = 0.0893315$, $\beta_3^{B\overline{G}} = 0.115285$, $\beta_3^{B\overline{B}} = 0.0713746$.
- *Positive autonomous decision*:
  - Girls per level: $\gamma_1^G = 0.0598649$, $\gamma_2^G = 0.138232$, $\gamma_3^G = 0.00441141$.
  - Boys per level: $\gamma_1^B = 0.0254583$, $\gamma_2^B = 0.0407112$, $\gamma_3^B = 0.143022$.
- *Positive habits transmission*:
  - Girls per level: $\delta_1^{G\overline{G}} = 0.0628747$, $\delta_1^{G\overline{B}} = 0.117906$, $\delta_2^{G\overline{G}} = 0.0162307$, $\delta_2^{G\overline{B}} = 0.0217844$, $\delta_3^{G\overline{G}} = 0.064252$, $\delta_3^{G\overline{B}} = 0.0722602$.
  - Boys per level: $\delta_1^{B\overline{G}} = 0.0831484$, $\delta_1^{B\overline{B}} = 0.0396256$, $\delta_2^{B\overline{G}} = 0.14784$, $\delta_2^{B\overline{B}} = 0.0560535$, $\delta_3^{B\overline{G}} = 0.0199681$, $\delta_3^{B\overline{B}} = 0.0505348$.
- *Abandon*:
  - Girls per level: $\eta_1^G = 0.0899652$, $\eta_2^G = 0.0620594$, $\eta_3^G = 0.118145$.
  - Boys per level: $\eta_1^B = 0.111194$, $\eta_2^B = 0.0445628$, $\eta_3^B = 0.0235689$.
- *Access*:
  - Girls: $\tau^G = 0.121096$.
  - Boys: $\tau^B = 0.12517$.

Once the parameters are estimated, we are able to give predictions of each group and level over the next few years by computing the solutions of the model for values of time t in the forthcoming future. The results can be seen in Figure 2.

In Table 2 we present the prediction of percentage of non–promote students for the next four courses.

Out[45]//TableForm=

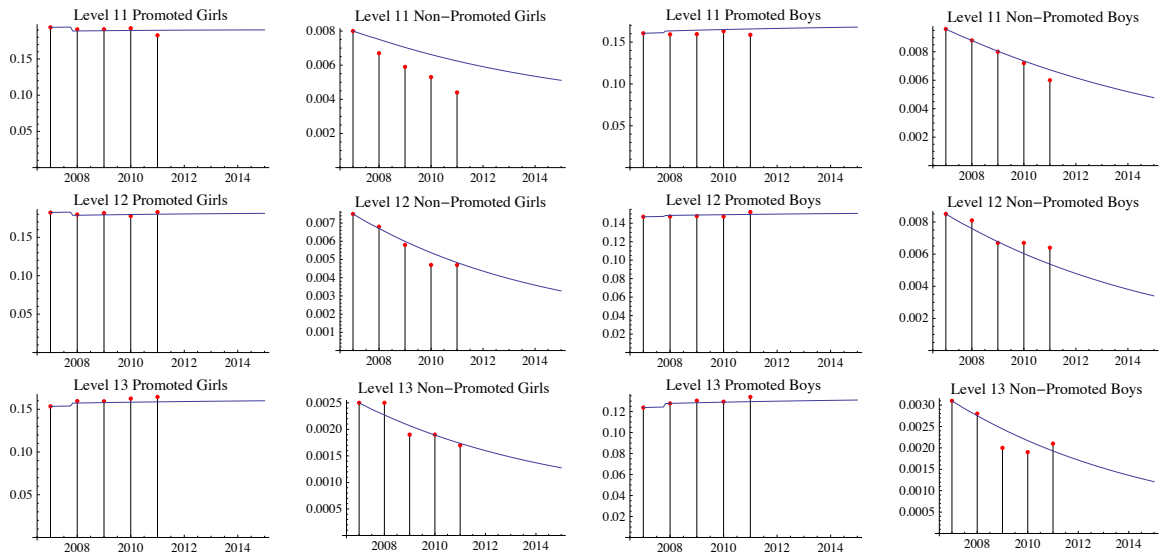


Figure 2: Graph representing the model fitting and the predictions until the course 2014-2015. Note that there is a decreasing trend in the non-promotable groups.

4 Conclusion

In this paper we present a model to study the dynamics of the students academic performance in the German region of North Rhine-Westphalia. In this model we divide the students by gender and academic levels, and it is based on the assumption that both, good and bad study habits, are a mixture of personal decisions and influence of classmates. Using data of the students academic performance, we estimate the model parameters fitting the model with the data. Thus, we can predict the students academic performance in the next few years. In Figure 2, it is expected that the decreasing trend in all non–promotable groups continues in the next years. For instance, in the course 2014-2015 less than 2% of the students will not promote (see Table 2).

This model will allow us to compare the performance of the coming new academic plan to this one in order to evaluate if the change is as good as expected.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness grant MTM2009-08587 and Universitat Politecnica de Valencia grant PAID06-11-

	2011–2012	2012–2013	2013–2014	2014–2015
Level 11, Non–Promote girls	0.591 %	0.561 %	0.535 %	0.511 %
Level 12, Non–Promote girls	0.436 %	0.395 %	0.359 %	0.327 %
Level 13, Non–Promote girls	0.16 %	0.148 %	0.137 %	0.127 %
Level 11, Non–Promote boys	0.617 %	0.566 %	0.52 %	0.477 %
Level 12, Non–Promote boys	0.479 %	0.427 %	0.381 %	0.34 %
Level 13, Non–Promote boys	0.171 %	0.152 %	0.136 %	0.121 %
TOTAL	2.455 %	2.25 %	2.067 %	1.905 %

Table 2: Prediction for the next four courses of the percentage of non-promoted students per gender and level, and the total. Note that there is a decreasing trend over the time in all levels with independence of gender. Also, the percentages decrease when the level increases. There are minor differences between boys and girls figures.

2070.

References

[1] A. Marchesi, C. Hernández, *El Fracaso Escolar. Perspectiva Internacional [Academic Underachievement. International Perspective]*, Alianza, Madrid, 2003. (In Spanish).

[2] *http://www.rlp.com.ni/noticias/93005*.

[3] *http://www.euractiv.com/en/enterprise-jobs/eu-youth-job-strategy-under-fire-news-497858*.

[4] *http://ec.europa.eu/news/culture/110202_en.htm*.

[5] *http://ec.europa.eu/education/school-education/doc/earlycom_en.pdf*.

[6] H. Daniels, M. Cole, J. Wertsch *Cambridge Companion to Vygotski*, Cambridge University Press, Cambridge, 2007.

[7] L.S. Vygotsky, *Mind in Society: The Development of Higher Mental Processes*, Harvard University Press, Cambridge, 1978.

[8] N.A. Christakis, J.H. Fowler, *Connected: The Surprising Power Of Our Social Networks And How They Shape Our Lives*, Brown and Company, 2009.

[9] N.A. Christakis, J.H. Fowler, *The spread of obesity in a large social network over 32 years*, The New England Journal of Medicine, **357** (2007) 370–379.

[10] F.J. Santonja, A. Morales, R.J. Villanueva, J.C. Cortés, *Analysing the effect of public health campaigns on reducing excess weight: A modelling approach for the*

Spanish Autonomous Region of the Community of Valencia, Evaluation and Program Planning, **35**(1) (2012) 34–39.

[11] F.J. Santonja, E. Sánchez, M. Rubio, J.L. Morera *Alcohol consumption in Spain and its economic cost: A mathematical modelling approach*, Mathematical and Computer Modelling, **52**(7-8) (2010) 999–1003.

[12] E. Sánchez, R.J. Villanueva, F.J. Santonja, M. Rubio *Predicting cocaine consumption in Spain, A mathematical modelling approach*, Drugs: Education, Prevention and Policy, **18**(2) (2011) 108–115.

[13] I. García, L. Jódar, P. Merello, F.J. Santonja *A discrete mathematical model for addictive buying: predicting the affected population evolution*, Mathematical and Computer Modelling, **54**(78) (2011) 1634-1637.

[14] L.M.A. Bettencourt, A. Cintron-Arias, D.I. Kaiser, C. Castillo-Chavez *The power of a good idea: Quantitative modelling of the spread of ideas from epidemiological models*, Physica A, **364** (2006) 513–536.

[15] M. Peco, F.J. Santonja, A.C. Tarazona, R.J. Villanueva, J. Villanueva-Oller, *The effect of the Spanish Law of Political Parties (LPP) on the attitude of the Basque Country population towards ETA: A dynamic modelling approach*, Mathematical and Computer Modelling, (2011) (To be published. Doi:10.1016/j.mcm.2011.11.007).

[16] B. Landner, *Bildungsreport Nordrhein-Westfalen 2010 [Education Report North Rhine-Westphalia 2010]*, Statistische Analysen und Studien, Band 68. 2011 (in German).

[17] *Regionalverband Ruhr [Education Report Ruhr Region]*, Bildungsbericht Ruhr, Waxmann 2012 (in German).

[18] *Kommunales Bildungsmonitoring: Tab. D13.2 Bestand an Schülerinnen und Schülern sowie Klassenwiederholungen [Municipal Education Monitoring: Table D13.2 Inventory of students and class repetition]*, Landesbetrieb Information und Technik Nordrhein-Westfalen (IT.NRW), Düsseldorf, 2012 (in German).

[19] K.R. Wentzel, D.E. Watkins, *Peer relationships and collaborative learning as contexts for academic enablers*, School Psychology Review, **31**(3) (2002) 366–377.

[20] J.D. Murray *Mathematical Biology*, Springer, 2002.

 ISBN:978-84-615-5392-1

[21] C. Fierro-Hernández, *Patrón de rasgos personales y comportamiento escolar en jóvenes [Personal characteristic patterns and scholar behavior in young people]*. Revista de Educación, **332** (2000) 291–304 (In Spanish).

[22] M. Martcheva, C. Castillo-Chavez *Diseases with chronic stage in a population with varying size*, Mathematical Biosciences, **182**(1) (2003) 1–25.

[23] J. Mena-Lorca, H.W. Hethcote, *Dynamic models of infectious diseases as regulators of population sizes*, Journal of Mathematical Biology, **30** (1992) 693–716.

[24] *http://www.wolfram.com/products/mathematica*.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Quadratic B-splines on criss-cross triangulations for solving elliptic diffusion-type problems

Isabella Cravero[1], Catterina Dagnino[1] and Sara Remogna[1]

[1] *Department of Mathematics, University of Torino, via C. Alberto, 10 - 10123 Torino, Italy*

emails: isabella.cravero@unito.it, catterina.dagnino@unito.it, sara.remogna@unito.it

Abstract

In this paper we propose a method for the solution of elliptic diffusion-type problems based on bivariate quadratic B-splines on criss-cross triangulations. This technique considers the weak form of the differential problem and the Galerkin method to approximate the solution. As finite-dimensional space, we choose the space of quadratic splines on a criss-cross triangulation and we use its local basis both for the reconstruction of the physical domain and for the representation of the solution.

Beside the theoretical description, we provide some numerical examples.

Key words: elliptic diffusion-type problem, bivariate B-spline, criss-cross triangulation

MSC 2000: 65D07; 65N99

1 Introduction

Let $\Omega \subset \mathbb{R}^2$ be an open, bounded and Lipschitz domain, whose boundary $\partial\Omega$ is partitioned into two relatively open subsets, Γ_D and Γ_N, i.e. they satisfy $\emptyset \subseteq \Gamma_D, \Gamma_N \subseteq \partial\Omega$, $\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. In this paper, we consider an elliptic diffusion-type problem with mixed boundary conditions

$$\begin{cases} -\nabla \cdot (K \nabla u) = f, & \text{in } \Omega, \\ \dfrac{\partial u}{\partial \mathbf{n}_K} = g_N, & \text{on } \Gamma_N, \quad \text{(Neumann conditions)} \\ u = g, & \text{on } \Gamma_D, \quad \text{(Dirichlet conditions)} \end{cases} \tag{1}$$

where $K \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite matrix, $\mathbf{n}_K = K\mathbf{n}$ is the outward conormal vector on Γ_N, $f \in L^2(\Omega)$, $g_N \in L^2(\Gamma_N)$ and g is the trace on Γ_D of an $H^1(\Omega)$ function, i.e. $g \in H^{1/2}(\Gamma_D)$ (see [7]).

As noticed in [11], the diffusion type problem (1) arises in a variety of applications such as the temperature equation in heat conduction, the pressure equation in flow problems, and also mesh smoothing algorithms. If K is the identity matrix, (1) simplifies to Poisson's problem.

A standard method to find the approximate solution of (1) is the Finite Element Method (see e.g. [7]) and, over the last years, the Isogeometric Analysis (IGA) (see e.g. [5]). Usually IGA is based on NURBS defined by B-splines of tensor product type (see e.g. [1, 4]) or, recently, on quadratic Powell-Sabin splines (see [10]). In this paper we propose an IGA approach for (1), based on bivariate quadratic B-splines on criss-cross triangulations. As remarked in [13], functions having total degree are preferable, in some cases, to tensor product ones that may have some inflection points, due to their higher coordinate degree.

The paper is organized as follows. In Section 2 we recall definitions and properties of bivariate quadratic B-splines on criss-cross triangulations and, in Section 3, we use them for the solution of (1). Finally, in Section 4 we give some numerical examples.

2 Quadratic B-splines on criss-cross triangulations

In order to have a self-contained presentation, in this section we briefly recall definitions and properties of unequally smooth bivariate quadratic B-splines on criss-cross triangulations (for details see [3, 13] and the references therein).

Let $\Omega_0 = \{(s,t) \mid 0 \le s,t \le 1\}$ and m, n be positive integers. We consider the sets $\bar{\xi} = (\xi_i)_{i=0}^{m+1}$ and $\bar{\eta} = (\eta_j)_{j=0}^{n+1}$, with $0 = \xi_0 < \xi_1 < \ldots < \xi_{m+1} = 1$, $0 = \eta_0 < \eta_1 < \ldots < \eta_{n+1} = 1$, that partition Ω_0 into $(m+1)(n+1)$ rectangular cells. By drawing both diagonals for each cell, we obtain a non-uniform criss-cross triangulation $\mathcal{T}_{mn}$, made of $4(m+1)(n+1)$ triangular cells. Let $\mathcal{S}_2^{(\bar{\mu}^\xi,\bar{\mu}^\eta)}(\mathcal{T}_{mn})$ be the space of bivariate quadratic piecewise polynomials on $\mathcal{T}_{mn}$, where

$$\bar{\mu}^\xi = (\mu_i^\xi)_{i=1}^m \quad \text{and} \quad \bar{\mu}^\eta = (\mu_j^\eta)_{j=1}^n \tag{2}$$

are vectors whose elements can be either 1 or 0 and denote the smoothness C^1, C^0, respectively, across the inner grid lines $s-\xi_i = 0$, $i = 1,\ldots,m$ and $t-\eta_j = 0$, $j = 1,\ldots,n$, while the smoothness across all oblique mesh segments[1] is C^1.

Let L_s^0 and L_t^0 be the number of grid lines $s-\xi_i = 0$, $i = 1,\ldots,m$ and $t-\eta_j = 0$, $j = 1,\ldots,n$, respectively, across which we want $S \in \mathcal{S}_2^{(\bar{\mu}^\xi,\bar{\mu}^\eta)}(\mathcal{T}_{mn})$ has C^0 smoothness. We

[1] According to [12], we call *mesh segments* the line segments that form the boundary of each triangular cell of $\mathcal{T}_{mn}$.

 ISBN:978-84-615-5392-1

recall that (see [3])

$$\dim \mathcal{S}_2^{(\bar{\mu}^\xi, \bar{\mu}^\eta)}(\mathcal{T}_{mn}) = mn + 3m + 3n + 8 + (n+2)L_s^0 + (m+2)L_t^0.$$

We remark that, if $S \in \mathcal{S}_2^{(\bar{\mu}^\xi, \bar{\mu}^\eta)}(\mathcal{T}_{mn})$ is globally C^1 (i.e. $L_s^0 = L_t^0 = 0$), we obtain the well-known dimension of $\mathcal{S}_2^1(\mathcal{T}_{mn})$ (see [12]).

Furthermore, we can provide a local basis for $\mathcal{S}_2^{(\bar{\mu}^\xi, \bar{\mu}^\eta)}(\mathcal{T}_{mn})$ (see [3]). In order to do it, we set $M = 3 + \sum_{i=1}^m (2 - \mu_i^\xi)$ and $N = 3 + \sum_{j=1}^n (2 - \mu_j^\eta)$, where μ_i^ξ, μ_j^η are defined as in (2). Let $\bar{s} = (s_i)_{i=-2}^M$, $\bar{t} = (t_j)_{j=-2}^N$ be the nondecreasing sequences of knots, obtained from $\bar{\xi}$ and $\bar{\eta}$ by the following two requirements:

(i) $s_{-2} = s_{-1} = s_0 = \xi_0 = 0, \qquad s_{M-2} = s_{M-1} = s_M = \xi_{m+1} = 1,$
$t_{-2} = t_{-1} = t_0 = \eta_0 = 0, \qquad t_{N-2} = t_{N-1} = t_N = \eta_{n+1} = 1;$

(ii) for $i = 1, \ldots, m$, the number ξ_i occurs exactly $2 - \mu_i^\xi$ times in $\bar{s}$ and for $j = 1, \ldots, n$, the number η_j occurs exactly $2 - \mu_j^\eta$ times in $\bar{t}$.

For the above sequences $\bar{s}$ and $\bar{t}$, we consider the following set of functions belonging to $\mathcal{S}_2^{(\bar{\mu}^\xi, \bar{\mu}^\eta)}(\mathcal{T}_{mn})$

$$\mathcal{B} = \{B_{ij}(s,t)\}_{(i,j)\in\mathcal{K}_{MN}}, \tag{3}$$

where $\mathcal{K}_{MN} = \{(i,j) : 0 \le i \le M-1, 0 \le j \le N-1\}$. If both/either $\bar{s}$ and/or $\bar{t}$ have/has double knots, then the B_{ij} smoothness will change and the support will change as well. Moreover, the B_{ij}'s have a local support, are non negative and form a partition of unity. In $\mathcal{B}$ we find different types of spline functions. There are $\rho = 2M + 2N - 4$ unequally smooth functions, that we call *boundary B-splines*, whose restrictions to $\partial\Omega_0$ are univariate quadratic B-splines. The remaining $MN - \rho$ functions, called *inner B-splines*, are such that their restrictions to $\partial\Omega_0$ are equal to zero. The supports and the BB-coefficients of such B-splines are reported in [2].

Since $\sharp\mathcal{B} > \dim \mathcal{S}_2^{(\bar{\mu}^\xi, \bar{\mu}^\eta)}$, the functions belonging to $\mathcal{B}$ are linearly dependent. Let:

(i) $\{\Omega_{0,r}\}_{r=1}^\gamma$ be a partition of Ω_0 into rectangular subdomains, generated by the grid lines with associated C^0 smoothness, with $\gamma = (L_s^0 + 1)(L_t^0 + 1)$;

(ii) $\mathcal{B}$ be defined as in (3);

(iii) $\mathcal{B}_1 \subset \mathcal{B}$ be the set of inner B-splines with C^1 smoothness everywhere or with C^0 smoothness only on the boundary of their support;

(iv) $\{\mathcal{B}^{(r)}\}_{r=1}^\gamma$ be a partition of $\mathcal{B}_1$, where each $\mathcal{B}^{(r)}$ contains B-splines with support in $\Omega_{0,r}$.

 ISBN:978-84-615-5392-1

Then, we can prove that a B-spline basis for $\mathcal{S}_2^{(\bar{\mu}^\xi,\bar{\mu}^\eta)}(\mathcal{T}_{mn})$ can be extracted from $\mathcal{B}$, by removing γ B-splines, one in each $\mathcal{B}^{(r)},\ r=1,\dots,\gamma$ (see [3]). We denote the corresponding set of indices of the B-spline basis by $\bar{\mathcal{K}}_{MN}$.

We remark that, if $\mathcal{S}_2^{(\bar{\mu}^\xi,\bar{\mu}^\eta)}(\mathcal{T}_{mn}) \equiv \mathcal{S}_2^1(\mathcal{T}_{mn})$, then, from [9] and standard arguments in approximation theory, for all $H \in C^3(\Omega_0)$ there exist a constant $C>0$ such that

$$\inf_{S\in\mathcal{S}_2^1(\mathcal{T}_{mn})} \|H-S\|_\infty \le Ch^3 \max\left\{\|D^{\alpha_1,\alpha_2}f\|_\infty : \alpha_1+\alpha_2=3\right\}$$

where $h=\max\{diam(T) \mid T \text{ is a triangle of } \mathcal{T}_{mn}\}$.

Since we are interested in the application of bivariate quadratic B-splines to the solution of (1), given the physical domain $\Omega \subset \mathbb{R}^2$, defined as in Section 1, we assume that such a domain can be exactly described through a parametrization of the form

$$\mathbf{G}:\Omega_0 \to \bar{\Omega}, \qquad \mathbf{G}(s,t)=\begin{pmatrix} x \\ y \end{pmatrix} \tag{4}$$

expressed as quadratic B-spline surface

$$\mathbf{G}(s,t)=\sum_{(i,j)\in\mathcal{K}_{MN}} \mathbf{P}_{ij}\ B_{ij}(s,t), \tag{5}$$

where $\{\mathbf{P}_{ij}\}_{(i,j)\in\mathcal{K}_{MN}}$ is a bidirectional net of control points, with $\mathbf{P}_{ij}\in\mathbb{R}^2$. We assume $p_{ij}=(s_i^p,t_j^p)\in\Omega_0$ as the pre-image of $\mathbf{P}_{ij}$, with

$$s_i^p=\frac{s_{i-1}+s_i}{2}, \qquad t_j^p=\frac{t_{j-1}+t_j}{2}. \tag{6}$$

We remark that, in order to construct the surface, it is not necessary to work with the basis, but we can use all the functions in the spanning set $\mathcal{B}$. In this case, the surface (5) has both the convex hull property and the affine transformation invariance one.

The proposed parametrization (5) is able to exactly reproduce domains whose boundary is made of linear and parabolic sections. In order to do it, the control points are obtained either by interpolation or quasi-interpolation spline operators (see [9, 12]).

Since the domains of interest in engineering problems are often described by conic sections, a possible extension of the current paper is to consider bivariate NURBS based on the B-splines here presented and we are working on it.

3 The Galerkin method based on bivariate quadratic B-splines

In this section, we consider an elliptic diffusion-type problem (1), where, for the sake of simplicity, we first assume homogeneous Dirichlet conditions, i.e. $g\equiv 0$. The weak formulation of (1) (see e.g. [5, 7]) is to find $u\in\mathbb{V}$ such that

$$a(u,v)=F(v), \quad \forall v\in\mathbb{V}, \tag{7}$$

 ISBN:978-84-615-5392-1

where:

- $\mathbb{V} = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$ is the space of functions with vanishing trace on Γ_D;
- $a : \mathbb{V} \times \mathbb{V} \to \mathbb{R}$ is the bilinear form given by $a(u, v) = \int_\Omega (K\nabla u) \cdot \nabla v \; d\Omega$;
- $F : \mathbb{V} \to \mathbb{R}$ is the linear functional given by $F(v) = \int_\Omega fv \; d\Omega + \int_{\Gamma_N} g_N v \; d\Gamma_N$.

In the Galerkin method to approximate the solution of (7), we replace the infinite dimensional space $\mathbb{V}$ by a finite-dimensional subspace $\mathbb{V}_h \subset \mathbb{V}$, with the subscript h indicating the relation to a spatial grid. Then, the discretized problem is to find $u_h \in \mathbb{V}_h$ such that

$$a(u_h, v_h) = F(v_h), \quad \forall v_h \in \mathbb{V}_h, \tag{8}$$

where $a(u_h, v_h) = \int_\Omega (K\nabla u_h) \cdot \nabla v_h \; d\Omega$ and $F(v_h) = \int_\Omega f v_h \; d\Omega + \int_{\Gamma_N} g_N v_h \; d\Gamma_N$.

Since, in (4), we have introduced the parametrization $\mathbf{G}$, we consider

$$\mathbb{V}_h = \left\{ v_h \in \mathbb{V} : \; v_h = v_{0,h} \circ \mathbf{G}^{-1}, \; v_{0,h} \in \mathbb{V}_{0,h} \right\},$$

where $\mathbb{V}_{0,h}$ is the discrete space in the parametric domain, that has to be chosen. In this paper, we consider $\mathbb{V}_{0,h}$ as an opportune subspace of $\mathcal{S}_2^{(\bar{\mu}^\xi, \bar{\mu}^\eta)}(\mathcal{T}_{mn})$.

Let N_h be the dimension of the spaces $\mathbb{V}_h$ and $\mathbb{V}_{0,h}$, and let $\{\Phi_l\}_{l=1}^{N_h}$ be a basis for $\mathbb{V}_{0,h}$. Then, we can define a basis for $\mathbb{V}_h$ as $\left\{\varphi_l = \Phi_l \circ \mathbf{G}^{-1}\right\}_{l=1}^{N_h}$ and the approximate solution u_h is given by

$$u_h = \sum_{l=1}^{N_h} q_l \varphi_l = \sum_{l=1}^{N_h} q_l (\Phi_l \circ \mathbf{G}^{-1}),$$

with unknown coefficients $q_l \in \mathbb{R}$. Therefore, (8) gives rise to

$$\sum_{l=1}^{N_h} q_l a(\varphi_l, \varphi_i) = F(\varphi_i), \quad i = 1, \ldots, N_h, \tag{9}$$

that is equivalent to the linear system $A\mathbf{q} = \mathbf{f}$, where

- $A \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix with elements

$$A_{il} = a(\varphi_l, \varphi_i) = \int_\Omega (K\nabla\varphi_l) \cdot \nabla\varphi_i \; d\Omega, \qquad i, l = 1, \ldots, N_h; \tag{10}$$

- $\mathbf{f} \in \mathbb{R}^{N_h}$ is the vector with components

$$\mathbf{f}_i = F(\varphi_i) = \int_\Omega f\varphi_i \; d\Omega + \int_{\Gamma_N} g_N \varphi_i \; d\Gamma_N = \mathbf{f}_i^{(1)} + \mathbf{f}_i^{(2)}, \qquad i = 1, \ldots, N_h; \tag{11}$$

 ISBN:978-84-615-5392-1

- $\mathbf{q} \in \mathbb{R}^{N_h}$ is the vector of unknown coefficients q_l, $l = 1, \ldots, N_h$.

Here, we assume that the parametrization $\mathbf{G}$ is given by (5) and consequently, we get $\mathbb{V}_h \subset \text{span}\left\{B_{ij} \circ \mathbf{G}^{-1}\right\}_{(i,j)\in\bar{\mathcal{K}}_{MN}}$. Note that the boundary condition $u = 0$ has also to be considered and for this reason we write $\mathbb{V}_h$ as a subset of the span. Then, the approximate solution is obtained taking into account the homogeneous boundary conditions.

The integrals A_{il} in (10) and $\mathbf{f}_i^{(1)}$ in (11), can be transformed as follows

$$\begin{aligned} A_{il} &= \int_{\Omega_0} \left(K\left[J^{-T}\nabla\Phi_l\right]\right) \cdot \left[J^{-T}\nabla\Phi_i\right] |\text{det}J| \; d\Omega_0, \quad i,l = 1,\ldots,N_h, \\ \mathbf{f}_i^{(1)} &= \int_{\Omega_0} (f \circ \mathbf{G})\, \Phi_i \, |\text{det}J| \; d\Omega_0, \quad i = 1,\ldots,N_h, \end{aligned} \tag{12}$$

with J the Jacobian matrix of the parametrization $\mathbf{G}$ given in (4) and (5)

$$J = J(s,t) = \begin{bmatrix} \frac{\partial x(s,t)}{\partial s} & \frac{\partial x(s,t)}{\partial t} \\ \frac{\partial y(s,t)}{\partial s} & \frac{\partial y(s,t)}{\partial t} \end{bmatrix}.$$

To evaluate the boundary term $\mathbf{f}_i^{(2)}$ in (11), we first define the mapping $\mathbf{G}_b : I := (0,1) \to \Gamma_N$ as the restriction of $\mathbf{G}$ to the subset of $\partial\Omega_0$ mapped into Γ_N, assuming that each side of Ω_0 is completely mapped into Γ_N or Γ_D. Then,

$$\mathbf{f}_i^{(2)} = \int_I (g_N \circ \mathbf{G}_b)\, \Phi_i \left|\mathbf{G}_b'\right| \; dI. \tag{13}$$

In order to compute $\nabla\Phi_i$, $i = 1,\ldots,N_h$, and J in (12), we obtain the values of the B-spline derivatives by means of their BB-coefficients (see [8]).

For the evaluation of the integrals in (12), we use a composite Gaussian Quadrature on triangular domains (see [6]) implemented by the Matlab function `triquad` (see [14]). Given in input the integer p and the vertices of a triangle of $\mathcal{T}_{mn}$, this procedure computes the p^2 nodes and the corresponding weights of the rule, whose precision degree is $2p-1$. In the numerical tests proposed in Section 4, we use $p = 2$. When $\mathbf{G}$ is the identity map (i.e. $\bar{\Omega} \equiv \Omega_0$) then, in (12),

$$A_{il} = \int_{\Omega_0} (K\nabla\Phi_l) \cdot \nabla\Phi_i \; d\Omega_0, \quad i,l = 1,\ldots,N_h,$$

and it is exactly computed, since in each triangle of $\mathcal{T}_{mn}$ the integrand function is a bivariate quadratic polynomial. To evaluate the integral in (13), we use a classical composite Gaussian rule with precision degree three, inherited from the one defined in the whole domain.

In case of non-homogeneous Dirichlet boundary conditions, the boundary degrees of freedom, i.e. the control variables associated with basis functions that do not vanish on Γ_D, have to be computed and we have to change the right term in the linear system (9)

 ISBN:978-84-615-5392-1

(see [5, 7]). The implementation of the Dirichlet boundary conditions is not trivial and it is still a matter of research (see e.g. [4] and the reference therein). In this paper we propose some examples of the above kind, where we choose the control variables associated with basis functions that do not vanish on Γ_D as the solution of a univariate spline interpolation problem.

4 Numerical examples

In this section we propose some numerical examples to show the performance of the bivariate quadratic B-splines on criss-cross triangulations for the solution of Poisson's problems with mixed boundary conditions. We perform h-refinement by adding at every step a middle knot in each interval of the partitions. With the global geometry function defined in (5), we reproduce the physical domain and this initial exact representation is retained during the refinement process.

In each table we give the number of subintervals $m+1$ and $n+1$ in the two directions s and t, respectively and the discrete L^2-norm of the error $(u-u_h)$, computed on a 35×35 grid of evaluation points in Ω_0, denoted by Ψ.

Example 1

Firstly we consider a very simple example, where $\bar{\Omega}\equiv\Omega_0$

$$\begin{cases} -\Delta u = f, & \text{in } (0,1)^2, \\ u = g, & \text{on } x=0,\ y=0 \\ \dfrac{\partial u}{\partial \mathbf{n}} = g_N, & \text{on } x=1, y=1, \end{cases}$$

with f, g and g_N obtained from the exact solution $u(x,y)=3x^2+2y^2$. In order to reproduce the domain, we consider the coarse knot partitions $\bar{\xi}=\bar{\eta}=(0,1)$. Therefore, we have $M=N=3$, $\mathcal{K}_{MN}=\mathcal{K}_{33}=\{(i,j):0\le i,j\le 2\}$ and $\mathbf{G}(s,t)=\sum_{(i,j)\in\mathcal{K}_{33}}\mathbf{P}_{ij}B_{ij}(s,t)$, with $(s,t)\in\Omega_0$. Since $\mathbf{G}$ is the identity map, the control points are the nine points $\mathbf{P}_{ij}=\{(s_i^p,t_j^p),\ 0\le i,j\le 2\}$, defined as in (6). Then, we perform h-refinement, considering $m,n=1$, 3, 7, 15, 31 and smoothness vectors $\bar{\mu}^{\xi}$, $\bar{\mu}^{\eta}$ with elements equal to one. We report the results in Table 1. According to Section 2, we remark that we have to neglect one inner B-spline either with C^1 smoothness everywhere or with C^0 smoothness only on the boundary of its support, in order to obtain a basis.

$m+1=n+1$	2	4	8	16	32
L^2-error	4.0(-15)	2.5(-15)	3.6(-15)	1.3(-15)	1.6(-14)

Table 1: Example 1. Error in L^2-norm versus interval number per side.

 ISBN:978-84-615-5392-1

We can notice that the solution, i.e. a quadratic polynomial, is reproduced. The computation of derivatives and integrals is stable, because there is not deterioration of the approximation error increasing the refinement.

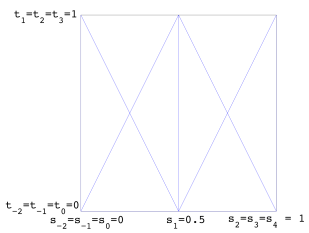
Example 2

In this example we consider the Poisson's problem in the L-shape domain shown in Fig. 1(b)

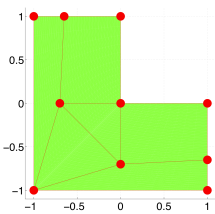
$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \Gamma_D = \partial\Omega, \end{cases}$$

where f is obtained from the exact solution $u(x,y) = \sin(\pi x)\sin(\pi y)$. In order to reproduce the domain, we can use two approaches, as in [11]. To introduce a discontinuity in the first derivative and create the corners, we can place two control points at the same location in physical space or we can use suitable double knots in $\bar{s}$ and $\bar{t}$. In the first case, we ensure that the basis has C^1 continuity throughout the interior of the domain. The only place where the basis is not C^1 is on the boundary itself, at the location of the repeated control points. We consider both cases in order to compare the corresponding results.

Approach 1: Double control point. We start with the coarse knot partitions $\bar{\xi} = (0, \frac{1}{2}, 1)$, $\bar{\eta} = (0,1)$ and we assume $\bar{\mu}^{\bar{\xi}} = (1)$. Therefore, we have $M = 4$, $N = 3$, $\mathcal{K}_{MN} = \mathcal{K}_{43} = \{(i,j) : 0 \le i \le 3,\ 0 \le j \le 2\}$ and $\mathbf{G}(s,t) = \sum_{(i,j)\in\mathcal{K}_{43}} \mathbf{P}_{ij} B_{ij}(s,t)$, with $(s,t) \in \Omega_0$ and the control points given in Fig. 1(c). In Fig. 1(a) we show the parameter domain Ω_0, with the associated knot sequences and, in Fig. 1(b), the corresponding physical domain Ω, with the control points.



(a)



(b)

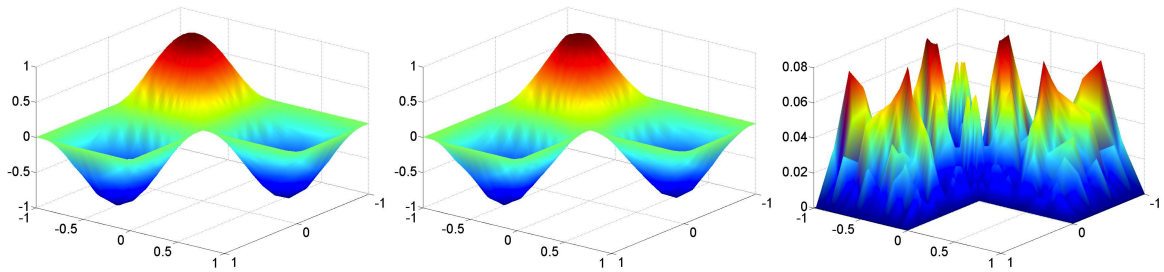
i	$\mathbf{P}_{i0}$	$\mathbf{P}_{i1}$	$\mathbf{P}_{i2}$
0	$(-1,1)$	$(-0.65,1)$	$(0,1)$
1	$(-1,-1)$	$(-0.7,0)$	$(0,0)$
2	$(-1,-1)$	$(0,-0.7)$	$(0,0)$
3	$(1,-1)$	$(1,-0.65)$	$(1,0)$

(c)

Figure 1: Example 2, Approach 1. (a) Parameter domain Ω_0, (b) physical domain Ω and (c) control points.

Then, we perform h-refinement, considering $m = 1, 3, 7, 15, 31$, $n = 0, 1, 3, 7, 15$, the smoothness vectors $\bar{\mu}^{\xi}$, $\bar{\mu}^{\eta}$ with elements equal to one and we report the results in the second row of Table 2.

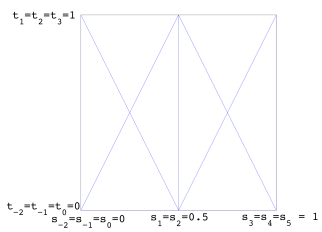
In Figs. 2(a)$\div$(c) we give the graphs of the exact solution, the approximation computed with $m = 7$, $n = 3$ and the discrete L^∞-norm error computed on Ψ.



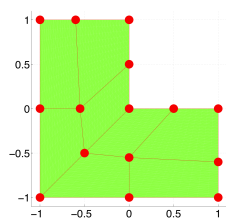
(a) (b) (c)

Figure 2: Example 2, Approach 1. The graphs of (a) the exact solution, (b) the approximation computed with $m = 7$, $n = 3$, (c) the discrete L^∞-norm error computed on Ψ.

Approach 2: Double knot. We start with the coarse knot partitions $\bar{\xi} = (0, \frac{1}{2}, 1)$, $\bar{\eta} = (0, 1)$ and we assume $\bar{\mu}^{\bar{\xi}} = (0)$. Therefore, we have $M = 5$, $N = 3$, $\mathcal{K}_{MN} = \mathcal{K}_{53} = \{(i, j) : 0 \le i \le 4,\ 0 \le j \le 2\}$ and $\mathbf{G}(s, t) = \sum_{(i,j)\in\mathcal{K}_{53}} \mathbf{P}_{ij} B_{ij}(s, t)$, with $(s, t) \in \Omega_0$ and the control points given in Fig. 3(c). In Fig. 3(a) we show the parameter domain Ω_0, with the associated knot sequences and, in Fig. 3(b), the corresponding physical domain Ω, with the control points.



(a)



(b)

i	$\mathbf{P}_{i0}$	$\mathbf{P}_{i1}$	$\mathbf{P}_{i2}$
0	$(-1, 1)$	$(-0.6, 1)$	$(0, 1)$
1	$(-1, 0)$	$(-0.55, 0)$	$(0, 0.5)$
2	$(-1, -1)$	$(-0.5, -0.5)$	$(0, 0)$
3	$(0, -1)$	$(0, -0.55)$	$(0.5, 0)$
4	$(1, -1)$	$(1, -0.6)$	$(1, 0)$

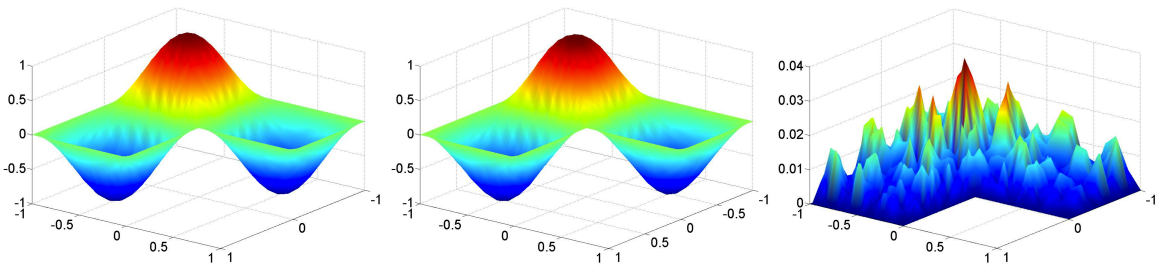
(c)

Figure 3: Example 2, Approach 2.(a) Parameter domain Ω_0, (b) physical domain Ω and (c) control points.

Then, we perform the same h-refinement of Approach 1. In this case the smoothness vector $\bar{\mu}^\eta$ has elements equal to one, while $\bar{\mu}^\xi$ has all of the elements equal to one except the element corresponding to $s = \frac{1}{2}$, that is equal to zero. We report the results in the third row of Table 2. In order to obtain a basis, according to Section 2, we remark that we have to neglect two inner B-splines either with C^1 smoothness everywhere or with C^0 smoothness only on the boundary of their support, because in this case the domain Ω is subdivided into two subdomains.

In Figs. $4(a) \div (c)$ we give the graphs of the exact solution, the approximation computed

with $m = 7$, $n = 3$ and the discrete L^∞-norm error computed on Ψ.



(a) (b) (c)

Figure 4: Example 2, Approach 2. The graphs of (a) the exact solution, (b) the approximation computed with $m = 7$, $n = 3$, (c) the discrete L^∞-norm error computed on Ψ.

$(m+1, n+1)$	(2,1)	(4,2)	(8,4)	(16,8)	(32,16)
L^2-error for Approach 1	7.1(-1)	4.5(-1)	5.3(-2)	6.4(-3)	6.2(-4)
L^2-error for Approach 2	8.3(-1)	2.2(-1)	1.7(-2)	1.6(-3)	1.8(-4)

Table 2: Example 2. Error in L^2-norm versus interval number per side.

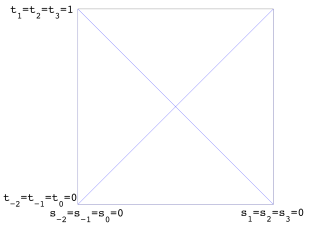
In [11] the authors solve the same problem, by considering the two above approaches, but they use a method based on biquadratic tensor product B-splines. If we analyse our results and theirs, we can conclude that the two methods are comparable.

Example 3

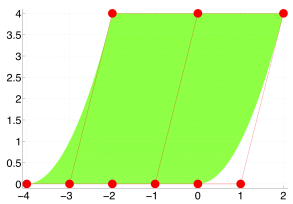
In this example we consider the Poisson's problem in the domain shown in Fig. $5(b)$

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \Gamma_D, \\ \dfrac{\partial u}{\partial \mathbf{n}} = g_N, & \text{on } \Gamma_N, \end{cases}$$

where Γ_N is given by the two segments with endpoints (-4,0), (0,0) and (-2,4), (2,4), respectively and Γ_D is given by the two parabolic sections with endpoints (-4,0), (-2,4) and (0,0), (2,4). The functions f, g and g_N are obtained from the exact solution $u(x, y) = \frac{\sin(x^2+y^2-1)}{5}$. In order to reproduce the domain, we consider the coarse knot partitions $\bar{\xi} = \bar{\eta} = (0, 1)$. Therefore, we have $M = 3$, $N = 3$, $\mathcal{K}_{MN} = \mathcal{K}_{33} = \{(i, j) : 0 \le i, j \le 2, \}$ and $\mathbf{G}(s, t) = \sum_{(i,j)\in\mathcal{K}_{33}} \mathbf{P}_{ij} B_{ij}(s, t)$, with $(s, t) \in \Omega_0$ and the control points given in Fig. $5(c)$. In Fig. $5(a)$ we show the parameter domain Ω_0, with the associated knot sequences and, in Fig. $5(b)$, the corresponding physical domain Ω, with the control points.



(a)



(b)

i	$\mathbf{P}_{i0}$	$\mathbf{P}_{i1}$	$\mathbf{P}_{i2}$
0	$(-4,0)$	$(-3,0)$	$(-2,4)$
1	$(-2,0)$	$(-1,0)$	$(0,4)$
2	$(0,0)$	$(-1,0)$	$(2,4)$

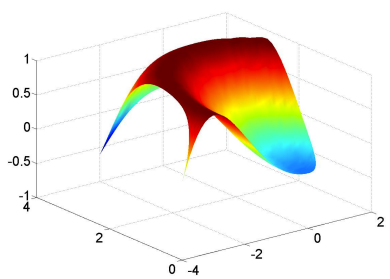
(c)

Figure 5: Example 3.(a) Parameter domain Ω_0, (b) physical domain Ω and (c) control points.

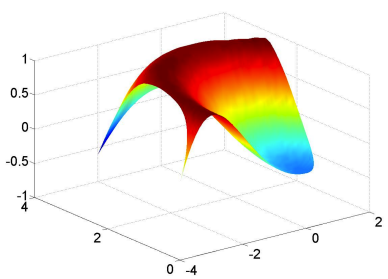
Then, we perform h-refinement, considering $m, n = 1,\ 3,\ 7,\ 15,\ 31$ and, in Table 3, we report the results. In Figs. $6(a) \div (c)$ we give the graphs of the exact solution, the approximation computed with $m = n = 7$ and the discrete L^∞-norm error computed on Ψ.

$m+1 = n+1$	2	4	8	16	32
L^2-error for Case 1	9.9(-1)	1.3(-1)	3.4(-2)	4.3(-3)	4.5(-4)

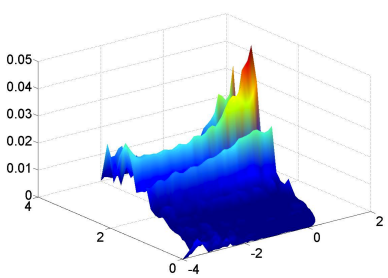
Table 3: Example 3. Error in L^2-norm versus interval number per side.



(a)



(b)



(c)

Figure 6: Example 3. The graphs of (a) the exact solution, (b) the approximation computed with $m = n = 7$, (c) the discrete L^∞-norm error computed on Ψ.

References

[1] P. Costantini, C. Manni, F. Pelosi, M. L. Sampoli, *Quasi-interpolation in isogeometric analysis based on generalized B-splines*, Comput. Aided Geom. Design **27** (2010) 656–668.

[2] C. Dagnino, P. Lamberti, 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://hdl.handle.net/2318/434

[3] C. Dagnino, P. Lamberti, S. Remogna, *On unequally smooth bivariate quadratic spline spaces*, Computational and Mathematical Methods in Science and Engineering (J. Vigo Aguiar Ed.), (2009) 350–359.

[4] C. de Falco, A. Reali, R. Vázquez, *GeoPDEs: a research tool for Isogeometric Analysis of PDEs*, Adv. Eng. Softw. **42** (2011) 1020–1034.

[5] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, *Isogeometric Analysis. Toward integration of CAD and FEA*, WILEY, 2009.

[6] J. N. Lyness, R. Cools, *A Survey of Numerical Cubature over Triangles*, Mathematics and Computer Science Division, Argonne National Laboratory **III** 1994.

[7] A. Quarteroni, *Numerical Models for Differential Problems, Modeling, Simulation & Applications*, Springer, 2009.

[8] P. Sablonnière, *Quadratic B-splines on non-uniform criss-cross triangulations of bounded rectangular domains of the plane*, Report IRMAR **03-14**, University of Rennes (2003)

[9] P. Sablonnière, *Quadratic spline quasi-interpolants on bounded domains of* $\mathbb{R}^d, d = 1, 2, 3$, Rend. Sem. Mat. Univ. Pol. Torino **61** (2003) 229–238.

[10] H. Speleers, C. Manni, F. Pelosi, M. L. Sampoli, *Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems*, Comput. Methods Appl. Mech. Engrg. **221-222** (2012) 132–148.

[11] A. V. Vuong, Ch. Heinrich, B. Simeon, *ISOGAT: a 2D tutorial MATLAB code for Isogeometric Analysis*, Comput. Aided Geom. Design **27** (2010) 644–655.

[12] R. H. Wang, *Multivariate Spline Functions and Their Application*, Science Press, Beijing/New York, Kluwer Academic Publishers, Dordrecht/Boston/London, 2001.

[13] R. H. Wang C. J. Li, *A kind of multivariate NURBS surfaces*, J. Comp. Math. **22** (2004) 137–144.

[14] G. von Winckel, *Matlab procedure* `triquad`, http://www.mathworks.com/matlabcentral/fileexchange/9230-gaussian-quadrature-for-triangles.

 ISBN:978-84-615-5392-1

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Image filtering with generalized fractional integrals

E. Cuesta[1], A. Duran[1], M. Kirane[2] and S. A. Malik[2]

[1] *Department of Applied Mathematics, University of Valladolid, Spain*

[2] *Laboratoire de Mathématiques, Image et Applications, University of La Rochelle, France*

emails: eduardo@mat.uva.es, angel@mac.uva.es, mokhtar.kirane@univ-lr.fr, salman.malik@univ-lr.fr

Abstract

In this work a new PDE technique based on fractional calculus for image restoration is proposed. The fractional order parameter, which controls the diffusion from classical Gaussian filtering to the absence of smoothing, is chosen, according to the variation of the gradient of the image at each pixel, by using new convolution kernels in the corresponding Volterra equations. The new implementation is described and some numerical experiments are shown.

Key words: Image processing, Fractional integrals and derivatives, Volterra equations, Convolution quadrature methods.
MSC 2000: 44A35, 44K05, 45D05, 65R20, 68U10, 94A08.

1 Introduction

This paper presents a numerical technique for image filtering based on the fractional partial differential equation model

$$\begin{cases} \partial_t^\alpha u(t,\mathbf{x}) &= \Delta u(t,\mathbf{x}), & (t,\mathbf{x}) \in [0,T]\times\Omega \\ u(0,\mathbf{x}) &= u_0(\mathbf{x}), & \mathbf{x}\in\Omega, \\ \partial_t u(0,\mathbf{x}) &= 0, & \mathbf{x}\in\Omega, \\ \dfrac{\partial u}{\partial \eta}(t,\mathbf{x}) &= 0, & (t,\mathbf{x})\in[0,T]\times\partial\Omega, \end{cases} \tag{1}$$

where ∂_t^α stands for the fractional time derivative of order $1<\alpha<2$ in the sense of Riemann–Liouville, Ω is a domain in $\mathbb{R}^2$ with boundary $\partial\Omega$, $\partial/\partial\eta$ stands for the outward

normal derivative and u_0 is the noisy image from which the objective is to restore the original (ideal) image. Thus $u(t, \mathbf{x})$ stands for the restored image at the time level t. The range of the parameter α implies that (1) interpolates the linear parabolic heat equation ($\alpha = 1$) and the linear hyperbolic wave equation ($\alpha = 2$), which suggests a suitable adaptation of the model to image filtering processes, in order to handle the diffusion following some dependence of α on the image gradient variation. The purpose of this work, based on the idea of applying (1) for each pixel of the image following the behaviour of the corresponding gradient [6], is to introduce an improvement of the selection of the viscosity parameter α, which is updated at every time step of the simulation. This leads to a better adaptation of the model (1) to the evolution of the image gradient.

Among the different methods presented in the literature for image processing, PDEs based models play a relevant role (see [1, 19] and references therein). In the particular case of restoration, use of the linear heat equation, $\alpha = 1$ in (1), lead to several investigations, aiming to control the diffusion process. They include:

1. Diffusion models with edge stopping functions [12], which incorporates a gradient-dependent diffusion coefficient to control the process and avoid blurring effect in edges and corners provided by the classical diffusion given by the heat equation.

2. Variational numerical algorithms, based on the minimization of the total variation of the image subject to constraints involving statistical parameters of the noise [15].

3. Neighborhood filters [9, 20], consisting of local averaging, with different techniques, to smooth the image noise

4. Anisotropic approach [3, 18], where most of the previous models (and the fractional PDE based ones below) are enhanced to incorporate non canonical directions of diffusion.

As an alternative to the nonlinear models described above, fractional calculus approach for image filtering has been considered in e. g. [2, 5]. In the last reference which, along with [6], is based on, the heat equation is generalized to models of the form (1). The fractional parameter $\alpha \in (1, 2)$ plays the role of a viscosity term with limiting values that make (1) interpolate the linear parabolic heat equation and the linear hyperbolic wave equation. This makes the selection of α being a particularly interesting task to control the image diffusion. On the other hand, in [6] the authors propose a refinement to handle the diffusion in a not uniform way over the whole image. This consists of applying (1) with a possibly different value of α for each single pixel of the image. The corresponding Volterra equation is well-posed for all $t > 0$ (which is not guaranteed in some nonlinear models) and the numerical results are efficient and competitive.

Considered here is an approach which goes more deeply into this topic. The selection of α on each pixel is modified. The new technique allows to modify this parameter during

the simulation according to the dependence on the image gradient variation. The numerical resolution is carried out with convolution quadrature techniques from the new convolution kernels for the Volterra approach.

The paper is structured as follows. In Section 2 the approach explained in [6] is recalled, the discretization of our proposal for (1) is introduced and the corresponding implementation is described. The performance of the new procedure is shown in Section 3, with numerical experiments on some noisy images.

2 Fractional models

2.1 Volterra equation and discretization

We first briefly describe the pixel by pixel technique, introduced in [6] and that will be the starting point for our study. Problem (1) is first written in the integral form

$$u(t,\mathbf{x}) = u_0(\mathbf{x}) + \int_0^t k_\alpha(t-s)\Delta u(s,\mathbf{x})\,\mathrm{d}s, \quad (t,\mathbf{x}) \in [0,T]\times\Omega, \tag{2}$$

$$k_\alpha(t) := t^{\alpha-1}/\Gamma(\alpha), \qquad t>0, \tag{3}$$

supplemented with homogeneous Neumann boundary condition. Now (2) is discretized in space and time with a different value of α for each pixel of the image. On a uniform $M\times M$ pixel mesh of Ω, with mesh length $h>0$, the Laplacian is approximated with second order central differences, leading to a $M^2\times M^2$ pentadiagonal matrix Δ_h of a pattern shown in Figure 1

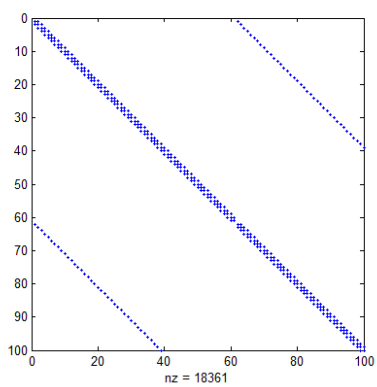


Figure 1: Sparsity pattern of the discretized Laplacian

 ISBN:978-84-615-5392-1

Considered now is a set of values $1 < \alpha_j < 2,\ j = 1, 2, \ldots, M^2$ and the semi-discrete approximation

$$\mathbf{u}(t) = \mathbf{u}_0 + \int_0^t \mathbf{K}(t-s)\mathbf{u}(s)\,\mathrm{d}s, \qquad 0 \le t \le T, \tag{4}$$

where $\mathbf{u}(t)$ stands for the $M^2 \times 1$ vector-image function at time t and at the pixel mesh, $\mathbf{u}_0$ is the initial (typically noisy) data and $\mathbf{K}$ the convolution kernel $\mathbf{K}(t) = I(t) \cdot \Delta_h$ with

$$I(t) = \begin{bmatrix} \dfrac{t^{\alpha_1}}{\Gamma(\alpha_1+1)} & 0 & \ldots & 0 \\ 0 & \dfrac{t^{\alpha_2}}{\Gamma(\alpha_2+1)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \dfrac{t^{\alpha_{M^2}}}{\Gamma(\alpha_{M^2}+1)} \end{bmatrix} \tag{5}$$

$$1 < \alpha_j < 2, \quad j = 1, 2, \ldots, M^2.$$

Formula (4) is finally discretized in time by means of a convolution quadrature method. A brief description is as follows (see [4, 10, 11] for the necessary explanations and details). The convolution integral in (4) is written in the form

$$\int_0^t \mathbf{K}(t-s)\mathbf{u}(s)\,\mathrm{d}s = \frac{1}{2\pi i}\int_\gamma \tilde{\mathbf{K}}(\lambda)Y(\lambda,t)\,\mathrm{d}\lambda, \tag{6}$$

where $\tilde{\mathbf{K}}(\lambda)$ stands for the Laplace transform of $\mathbf{K}$, γ is a suitable integration path in $\mathbb{C}$ connecting $-i\infty$ to $+i\infty$ (where Bromwich inversion formula for $\mathbf{K}$ is applied) and $Y(\lambda, t)$ stands for the solution of the ordinary differential equation

$$\mathbf{y}'(t) = \lambda\mathbf{y}(t) + \mathbf{u}(t),\ 0 \le t \le T, \quad \text{with} \quad \mathbf{y}(0) = 0. \tag{7}$$

Now the discretization of (7) provides numerical approximations of (4) via (6). The corresponding formulas are written in terms of the generating power series

$$\tilde{\mathbf{K}}\left(\frac{\delta(\xi)}{\tau}\right) = \sum_{j=0}^{+\infty} \mathbf{Q}_j^{(\alpha)}\xi^j, \tag{8}$$

where τ is the time-step of the discretization and $\delta(\xi)$ is the quotient of the generating polynomials of the underlying numerical method used in (6) which, in our case, will be the backward Euler formula [7]. Thus, $\delta(\xi) = 1 - \xi$, and if $\mathbf{u}_n$ is an approximation to $\mathbf{u}(t_n)$, for the time level $t_n = n\tau, n \ge 0$, then (4) is approximated by the convolution quadrature method

$$\mathbf{u}_n = \mathbf{u}_0 + \sum_{j=0}^{n} \mathbf{Q}_{n-j}^{(\alpha)}\mathbf{u}_j, \quad n \ge 1, \tag{9}$$

where for this method the quadrature weights have the form

$$\mathbf{Q}_j^{(\alpha)} = \tau^\alpha \begin{bmatrix} \binom{\alpha_1}{j} & 0 & \dots & 0 \\ 0 & \binom{\alpha_2}{j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \binom{\alpha_{M^2}}{j} \end{bmatrix} \cdot \Delta_h, \quad j = 0, 1, 2, \dots, M^2 \times M^2. \tag{10}$$

In practice, these matrix coefficients are computed in an efficient way by using FFT techniques [11].

2.2 Description of the modification

The implementation in [6] establishes values of α, for each single pixel, in the first time level and they will be remained fixed during the rest of the computation. A more realistic approach would be to consider nonconstant fractional orders (viscosity parameter). To this end, a coherent definition of

$$\partial^{-\alpha(t)} g(t), \qquad t \geq 0, \tag{11}$$

where $g : [0, +\infty) \to X$ is absolutely continuous on a Banach space X is required. To our knowledge, the first approach in this sense is given in [16], where

$$\partial^{\alpha(t)} g(t) := \frac{1}{\Gamma(\alpha(t))} \int_0^t (t-s)^{\alpha(t)-1} g(s)\, \mathrm{d}s, \qquad t \geq 0, \tag{12}$$

is adopted. Unfortunately, (12) lacks a convolution structure. A natural extension to the constant case would consists of replacing (3) by

$$k(t) := \frac{t^{\alpha(t)-1}}{\Gamma(\alpha(t))}, \qquad t \geq 0, \tag{13}$$

as the convolution kernel and therefore

$$\partial^{-\alpha(t)} g(t) := k * g(t) = \int_0^t k(t-s) g(s)\, \mathrm{d}s. \tag{14}$$

The main objection to this choice is that the Laplace transform of (13), which is necessary in (6) (as well as in many other procedures), cannot be explicitly computed. Finally, the most widely adopted definition (and which is our choice in this paper) for (11) is obtained from taking $k(t)$ as the inverse Laplace transform of

$$\tilde{K}(\alpha, z) = \frac{1}{z^{\bar{\alpha}(z)\, z}}, \tag{15}$$

 ISBN:978-84-615-5392-1

where $\bar{\alpha}(z)$ stands for the Laplace transform of $\alpha(t)$ (see [14]). Notice that if $\alpha(t) = \alpha$ is a constant, then this and (13) coincide with (3). Some numerical experiments, carried out in this sense for a future work, show that the results for the last two choices of the kernel $k(t)$ do not differ notably.

This time-dependent fractional order strategy is introduced in the formulation of [6], by considering different functions $\alpha^{(j)}(t), j = 1, \ldots, M^2$ for each single pixel, leading to a semidiscrete approximation (9) with a convolution kernel $\mathbf{K}(t) = I(t) \cdot \Delta_h$ where (5) is replaced by

$$I(t) = \begin{bmatrix} k_1(t) & 0 & \ldots & 0 \\ 0 & k_2(t) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & k_{M^2}(t) \end{bmatrix}$$

and, $k_j(t), j = 1, 2, \ldots, M^2$ is given by the inverse Laplace transform of $\tilde{K}(\alpha^{(j)}, z)$, following (15).

In order to use (8), a computable expression of the Laplace transform of $\mathbf{K}$ is required. This can be obtained as follows. Note that the time discretization (9) only requires to have the values of the $\alpha^{(j)}$ at the time levels $0 = t_0 < t_1 < t_2 < \ldots < t_N = T$. Then we can set

$$\alpha^{(j)}(t) = \alpha_0^{(j)} + \sum_{m=1}^{N} (\alpha_m^{(j)} - \alpha_{m-1}^{(j)}) U(t - t_m),$$

for $1 \leq j \leq M^2$ and $0 \leq t \leq T$, where $U : [0, T] \to \mathbb{R}$ stands for the Heaviside function. Observe that this defines a stair function. Now, the Laplace transform of $\alpha^{(j)}(t)$ is

$$\tilde{\alpha}^{(j)}(z) = \frac{\alpha_0^{(j)}}{z} + \sum_{m=1}^{N-1} \frac{(\alpha_m^{(j)} - \alpha_{m-1}^{(j)}) \mathrm{e}^{-z t_m}}{z}, \quad 1 \leq j \leq M^2;$$

therefore the Laplace transform of each component of the matrix valued function I reads simply as

$$\tilde{K}(\alpha^{(j)}, z) = z^{-\{\alpha_0^{(j)} + \sum_{m=1}^{N-1} (\alpha_m^{(j)} - \alpha_{m-1}^{(j)}) \, \mathrm{e}^{-z t_m}\}}.$$

2.3 Implementation

It is of interest to make some comments on the implementation. They essentially concern the automatic selection of the fractional parameter α, in order to improve the performance in the preservation of edges and corners, the noise filtering and the adaptation to the evolution of the diffusion. The main requirements about the first two points [6] are summarized as follows :

(i) The choice of α is determined, at each pixel, as an increasing function of the gradient. Pixels where the gradient is large should be associated with values of α close to 2 and pixels with lower gradients should be associated with values of α close to 1.

(ii) Extreme cases of noisy pixels (low gradient variation) and corners and edges (very high) determine the form of the function close to 1 and 2 respectively. Thus, in practice, the range of α is limited to an interval $[\alpha_{min}, \alpha_{max}]$, $1 < \alpha_{min} < \alpha_{max} < 2$, avoiding in this way the extreme cases $\alpha = 1, \alpha = 2$, only reserved for special situations. Since α_{min} and α_{max} can be very close to 1 and 2 respectively, this is not restrictive for our model. Furthermore, this assumption guarantees the solvability and uniqueness of solution of (4), as well as the corresponding problem for a larger class of operators than the Laplacian [13].

These criteria introduce a degree of freedom in the selection of α as function of $\nabla \mathbf{u}$. In our case, the pattern for α has been taken, for simplicity, as in Figure 2. Other distributions of α are indeed possible.

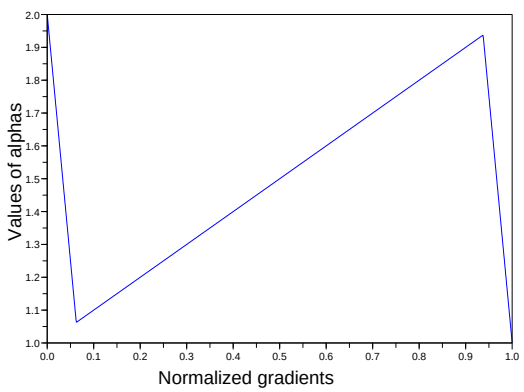


Figure 2: Profile distribution of α's.

According to these criteria, we describe the implementation of the numerical method introduced in Section (2.2)

(A) For a fixed integer $N > 1$, the interval $[\alpha_{min}, \alpha_{max}]$ is initially partitioned

$$\alpha_{min} \leq \alpha_1^{(0)} < \cdots < \alpha_N^{(0)} < \alpha_{max}, \tag{16}$$

and the corresponding weights $\mathbf{Q}_j^{(\alpha_k^{(0)})}$ are computed.

 ISBN:978-84-615-5392-1

(B) When advancing in time in (9), a choice of α, following (i)-(ii), is required. A first step consists of computing $\alpha_n = \left(\alpha_n^{(1)}, \alpha_n^{(2)}, \ldots, \alpha_n^{(M^2)}\right)^T$, the value of α corresponding to the M^2 pixels at the time step n, by using the function of Figure 2. Then α_n is transformed according to a mapping $\alpha_n \mapsto \widetilde{\alpha}_n$, in a simple way: for $k = 1, \ldots, M^2$, one identifies the interval (α_i, α_{i+1}) of (16) containing $\alpha_n^{(k)}$. Then, the k-th component of $\widetilde{\alpha}_n$ is α_i.

As in [6], the high computational cost in the calculus of the α forces to take the initial values (16), from which the quadrature weights are computed, and then using this group of quadratures in the whole computation, following the strategy described in (B). In the experiments below, we have taken $\alpha_j = 1 + \frac{j}{N}, 1 \leq j \leq N$ for a fixed integer N.

3 Numerical results

The performance of the previously described algorithm is analyzed in this section, by comparison with other techniques presented in the literature. Specifically, we show the quality of restoration for two different images provided by the following methods:

(PM): A nonlinear Perona–Malik model where the stopping function turns out to be $c(t) = \mathrm{e}^{-t}$, $t \geq 0$ (see [1] for more details).

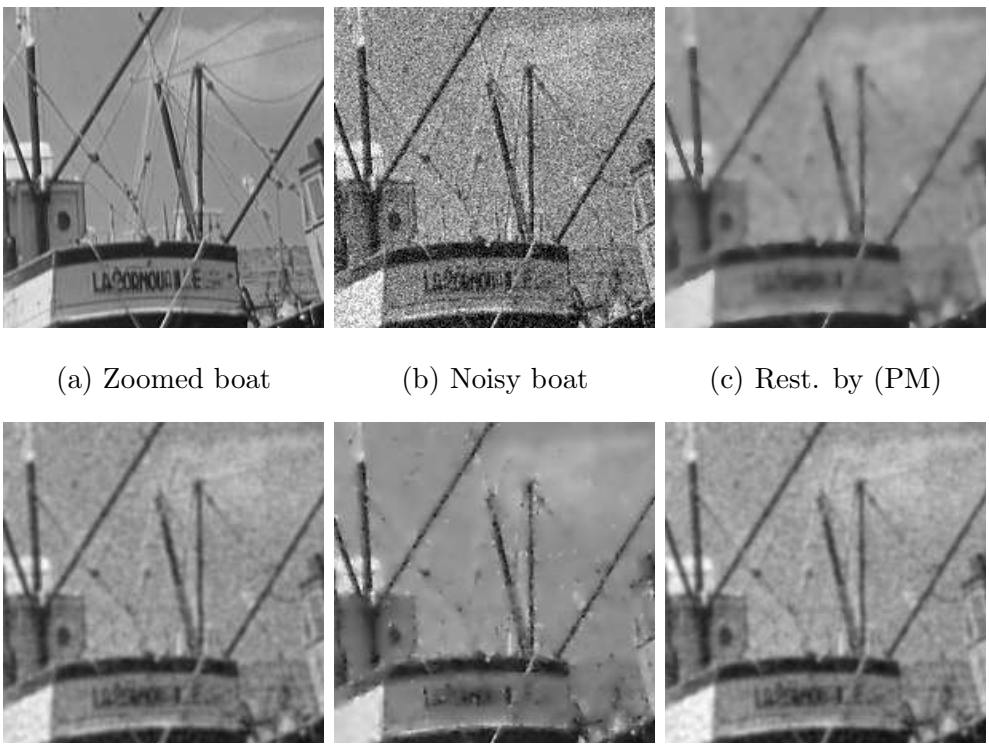
(VEV): The Volterra based model of fractional type described in [6].

(PG): The nonlinear Perona–Malik based model recently proposed in [8].

(VODO): The model proposed in the present manuscript.

The four algorithms were run up to a final time $T = 2$ and, in the cases of (VEV) and (VODO), the interval for the fractional order is $[\alpha_{min}, \alpha_{max}] = [1 + 10^{-3}, 2 - 10^{-3}]$. For the experiments below, the original images have been perturbed with an additive noise of Gaussian type and different values of the standard deviation σ, from 15 to 35, see Table 1.

 ISBN:978-84-615-5392-1

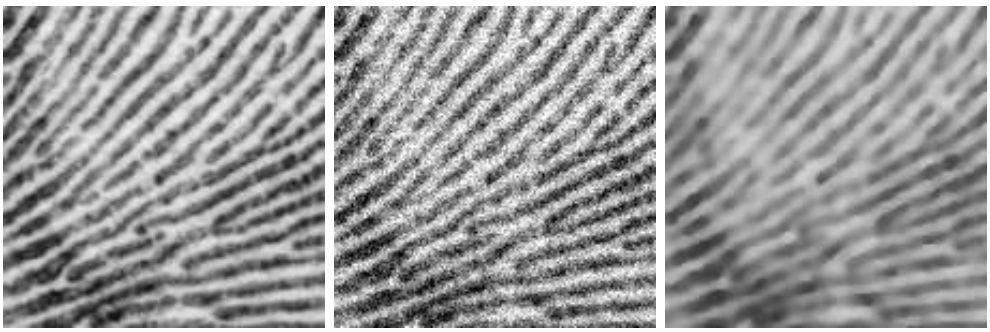


(a) Zoomed boat (b) Noisy boat (c) Rest. by (PM)

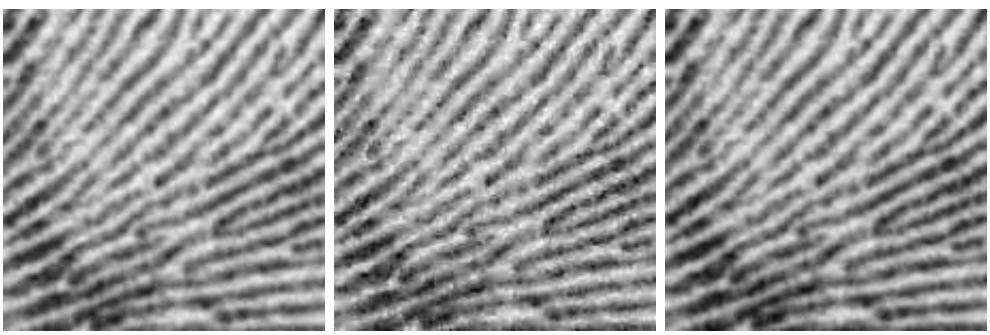
(d) Rest. by (VEV) (e) Rest. by (PG) (f) Rest. by (VODO)

Figure 3: Experiment 1.

 ISBN:978-84-615-5392-1



(a) Zoomed fingerprint (b) Noisy fingerprint (c) Rest. by (PM)



(d) Rest. by (VEV) (e) Rest. by (PG) (f) Rest. by (VODO)

Figure 4: Experiment 2.

Figures 3 and 4 display original and noisy images, along with the corresponding restoration given by the four methods considered. In both experiments, the appearance of the results gives the best quality restorations to (PG) and (VODO). This is verified in Table 1 (see the data in bold type), which shows, for each experiment and different σ, the values of the index SNR corresponding to the original image and the four methods at the end of the restoration. This is a standard index of measurement of quality in image processing (see e. g. [17] for the details). As far as the first experiment is concerned (Boats), the most competitive methods are, in this order, (PG), (VODO) and (VEV), and with a relatively close distance between them. In the case of the second image, the texture plays an important role and the restoration is a more stressful test for the methods. Table 1 shows that here the best choice is (VODO), followed by (PG) and (VEV) and, compared to the previous experiment, with a longer distance between them. These results conclude that our proposal gives very competitive performance and its construction, based on linear and well-posed

problems, makes it be preferable to other nonlinear models in many situations.

Table 1: SNR analysis

σ	15	20	25	30	35	15	20	25	30	35
Input SNR	9.88	7.40	5.50	4.00	2.70	10.60	8.20	6.30	4.80	3.50
Method	Boats (512×512)					Fingerprint (512×512)				
(PM)	13.69	12.26	11.57	9.72	9.74	11.97	10.26	9.10	7.90	8.15
(VEV)	14.10	12.65	11.32	10.32	9.42	11.40	10.9	9.50	9.3	8
(PG)	**14.87**	**13.66**	**12.77**	**11.95**	**11.32**	13.45	12.00	10.90	9.99	9.23
(VODO)	14.47	12.92	12.23	11.53	10.96	**15.42**	**13.90**	**12.89**	**12.00**	**11.24**

Acknowledgements

This research has been supported by MICINN project MTM2010-19510/MTM.

References

[1] J. Aubert, P. Kornprobst, *Mathematical Problems in Image Processing*, Springer, Berlin, 2001.

[2] J. Bai and X. Chu Feng, *Fractional anisotropic diffusion for image denoising*, IEEE Trans. Image Process. **16** (2007) 2492–2502.

[3] S. Bartels and A. Prohl, *Stable discretization of scalar and contrained vectorial Perona-Malik equation*, Interfaces Free Bound. **4** (2007) 431–453.

[4] M. P. Calvo and E. Cuesta and C. Palencia, *Runge-Kutta convolution quadrature methods for well-posed equations with memory*, Numer. Math. **107** (2007) 589–614.

[5] E. Cuesta and J. Finat, *Image processing by means of a linear integro-differential equation*, IASTED (2003) 438–442.

[6] E. Cuesta, M. Kirane and S. A. Malik, *Image structure preserve denoising using generalized fractional time integrals*, Signal Process. **92** (2012) 553–563.

[7] E. Cuesta and C. Palencia, *A numerical method for an integro-differential equation with memory in Banach spaces*, SIAM J . Numer. Anal. **41** (2003) 1232–1241.

 ISBN:978-84-615-5392-1

[8] P. Guidotti and James V. Lambers, *Two new nonlinear diffusion for noise reduction*, J. Math. Imaging Vis. **33** (2009) 25–37.

[9] J. S. Lee, *Digital image smoothing and the sigma filter*, Comput. Vis. Graph. Image Process. **24** (1983) 253–269.

[10] Ch. Lubich, *Convolution quadrature and discretized operational calculus I*, Numer. Math. **52** (1988) 129–145.

[11] Ch. Lubich, *Convolution quadrature and discretized operational calculus II*, Numer. Math. **52** (1988) 413–425.

[12] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. on Pattern Anal. and Mach. Intell. **12** (1990) 629–639.

[13] J. Pruss, *Evolutionary Integral Equations and Applications*, Birkhäuser Verlag, Basel, 1993.

[14] B. Ross and S. Samko, *Fractional integration operator of variable order in the holder spaces $h^{\lambda(x)}$*, Int. J. Math. and Math. Sci. **18** (1995) 777–788.

[15] L. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithm*, Phys. D **60** (1992) 259–268.

[16] G. Scarpi, *Sulla possibilita di un modello reologico di tipo evolutivo*, Rend. Sc. nat. fis. mat. **II** (1972) 570–575.

[17] J. L. Starck, E. J. Cands and D. L. Donoho, *The Curvelet Transform for Image Denoising*, IEEE Transactions on Image Processing **11** (2000) 670–684.

[18] J. Weickert, *A review of nonlinear diffusion filtering*, Lecture Notes in Computer Science - Scale Space Theory in Computer Science, Springer, Berlin, 1997.

[19] J. Weickert, *Anisotropic Diffusion in Image Processing*, B.G. Teubner, Stuttgart, 1998.

[20] L. P. Yaroslavsky, *Digital Picture Processing. An Introduction*, Springer-Verlag, New York, 1985.

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Modelling parameterized shared-memory hyperheuristics for auto-tuning

José-Matías Cutillas-Lozano[1], Domingo Giménez[1] and Luis-Gabino Cutillas-Lozano[2]

[1] *Departamento de Informática y Sistemas, University of Murcia*

[2] *Aguas Municipalizadas, Alicante*

emails: josematias.cutillas@um.es, domingo@um.es,
lgabino.cutillas@aguasdealicante.es

Abstract

This paper tackles the problem of developing and modelling shared-memory hyperheuristics based on parameterized shared-memory metaheuristic schemes. The same parallel, parameterized scheme used for metaheuristics development and tuning can be used for hyperheuristics, but in this case the execution time greatly increases due to the continuous application of different metaheuristics given by different values of the parameters in the scheme, and so the necessity of using a model of the execution time to decide at running time the number of threads to use to obtain a reduced execution time becomes more apparent. The model of the execution time and consequently the optimum number of threads depend on a number of factors: the problem to be solved, the metaheuristics used, the hyperheuristic scheme and the implementation of the basic functions in it, the computational system where the problem is being solved, etc. So, obtaining a satisfactory model of the execution time of the functions in the scheme and an auto-tuning methodology is not an easy task. This paper presents an auto-tuning methodology for shared-memory parameterized metaheuristic schemes that can in turn be applied to shared-memory hyperheuristics developed on top of them. The applicability of the proposal is shown with a problem of obtaining satisfactory metaheuristics for solving a problem of minimization of electricity consumption in exploitation of wells. The model and the methodology work satisfactorily, which allows us to reduce the execution time for the selection of satisfactory metaheuristics.

Key words: Parameterized metaheuristic schemes, parallel hyperheuristics, modelling
shared-memory schemes, auto-tuning

1 Introduction

The use of a unified parameterized scheme for metaheuristics facilitates the development of metaheuristics and their application. The scheme has been applied successfully: to obtain satisfactory Simultaneous Equation Models from a set of values of the variables [3]; for a tasks-to-processes assignation problem with independent tasks and memory constrains [2]; and for the optimization of power consumption in operation of wells [7]. Three pure metaheuristics (GRASP, Scatter Search and Genetic algorithms) and their combinations/hybridizations were considered in these studies, and the scheme has been extended by introducing a new metaheuristic (Tabu Search), with the inclusion in the scheme of four new parameters.

Parallelizing the scheme for shared-memory will reduce the execution time. But having a parallel routine does not ensure it will be used correctly, and the execution time of the parallel routine may be far from the optimum (or may be even larger than the sequential time) if the number of threads used in the application of the routine is not appropriate. The auto-tuning problem of sequential and parallel routines has been studied in different fields [4, 5, 6], and in this paper the application of auto-tuning methodologies to parameterized shared-memory metaheuristics and hyperheuristics is considered. Our aim is to develop hyperheuristics which satisfactory select metaheuristics or combinations/hybridations from a parametrized scheme by obtaining appropriate values of the metaheuristic parameters. Having a scheme with auto-tuning would be very useful because a large number of experiments would be necessary, and the experimentation time can be reduced with a good selection of the number of threads to use in different parts of the parallel scheme.

The rest of the paper is organized as follows. Section 2 presents the main ideas of hyperheuristics based on parameterized metaheuristics, and Section 3 the parameterized shared-memory scheme for metaheuristics that can in turn be used as hyperheuristic scheme. In section 4 the modelling of the execution time of the different basic and combined/hybridised metaheuriscs in the scheme is analysed theoretically and experimentally. Section 5 concludes the paper and shows some future research lines.

2 Hyperheuristics based on parameterized metaheuristics

The use of a unified parameterized scheme for metaheuristics (Algorithm 1) facilitates the development of metaheuristics and their application [11]. However, selecting the appropriate values of parameters ($ParamX$) to apply a satisfactory metaheuristic to a particular problem can be difficult and is computationally demanding. The selection of these values can be made through a hyperheuristic method also developed with the parameterized metaheuristic scheme. The range of possible values of the parameters of the metaheuristic can be large, and it is necessary to establish work intervals for the hyperheuristic. For clarity,

hereafter, we refer to the metaheuristic scheme directrly aplied to an optimization problem as MS, and HMS refers to hyperheuristic based on a metaheuristic scheme for selecting the appropriate values of metaheuristic parameters.

Algorithm 1 Parameterized metaheuristic scheme

```
Initialize(S,ParamIni)
while ( not EndCondition(S,ParamEnd)) do
  SS=Select(S,ParamSel)
  SS1=Combine(SS,ParamCom)
  SS2=Improve(SS1,ParamImp)
  S=Include(SS2,ParamInc)
end while
```

The application and auto-tuning of the hyperheuristic is analysed with a problem of cost optimization of electric consumption [9]. The application of parameterized metaheuristics to this problem has been studied [8], and now the problem to optimize by the hyperheuristic is the metaheuristic itself, to obtain a satisfactory metaheuristic for the electricity consumption problem. In the hyperheuristic, using the notation for evolutionary algorithms, an individual or element is represented by an integer vector $MetaheurParam$ of size 20 that encodes the set of parameters that characterizes a metaheuristic using the scheme in Algorithm 1. The set of individuals constitutes the reference set, which means a set of metaheuristics, with each metaheuristic the combination/hybridation of basic metaheuristics (GRASP, Scatter Search, Genetic algorithm and Tabu Search) given by the values in $MetaheurParam$. The fitness value in the hyperheuristic for an element $MetaheurParam$ is the fitness value obtained when the metaheuristic with the parameters in $MetaheurParam$ is applied to the electricity consumption problem. Our objective is to minimize the fitness function and so obtain the combination of the metaheuristic parameters which gives the lowest electricity consumption for a problem (a set of problem inputs could also be studied but only one problem has been considered for simplicity). Thus, when executing the hyperheuristic, a lot of metaheuristics are applied to different inputs of the electricity problem. The execution time is very large and it is necessary to use parallelism. Parallel metaheuristics can be used to reduce the execution time, but it is also possible, and preferable, to use parallelism at a higher level, for which the parameterised shared-memory metaheuristic scheme is used for the hyperheuristic, and the same auto-tuning techniques are valid for the metaheuristics and the hyperheuristic.

 ISBN:978-84-615-5392-1

3 A parameterized shared-memory scheme for metaheuristics and hyperheuristics

When developing hyperheuristics with the same scheme used for metaheuristics (HMS), the same parallelization techniques for metaheuristics [1] are applicable for hyperheuristics. In our approach, the parameterized scheme in Algorithm 1 becomes a parameterized shared-memory scheme just by independently parallelizing each basic function in the scheme (Algorithm 2) with new parallelism parameters ($ThreadX$) indicating the number of threads to use in each part of the algorithm. A parallel hyperheuristic is obtained by selecting the values of the metaheuristic parameters ($ParamX$) and the parallelism parameters ($ThreadsX$), which can be selected with some auto-tuning technique to obtain low execution times.

Algorithm 2 Parameterized shared-memory metaheuristic scheme

```
Initialize(S,ParamIni,ThreadsIni)
while ( not EndCondition(S,ParamEnd)) do
  SS=Select(S,ParamSel)
  SS1=Combine(SS,ParamCom,ThreadsCom)
  SS2=Improve(SS1,ParamImp,ThreadsImp)
  S=Include(SS2,ParamInc,ThreadsInc)
end while
```

Two basic parallel schemes are identified in [3] for the functions in Algorithm 2:

- In the first scheme the elements of a set are treated independently, and the number of threads to work in a loop are selected. This scheme appears, for example, when combining elements in a Genetic algorithm or when randomly generating an initial set of elements. Thus, $ThreadsIni$ and $ThreadsCom$ contain a parallelism parameter indicating the number of threads to use in the generation of the initial set and for the combination of the selected elements, and these values can be different, so obtaining different values of the parallelism parameters in each function.

- The second scheme has two parallelism levels and can be used to obtain fine or grained parallelism. The number of threads at each parallelism level is established. This type of parallelism appears in improvement and mutation functions, where some elements are selected (first level) and each element is improved by analysing its neighbourhood (second level).

The number of threads (one value or several values) is established for each function in the parameterized shared-memory scheme. The number of parallelism parameters for each function depends on the particular implementation of the functions in the unified

scheme (Algorithm 2), but the methodology is common to different metaheuristics and parallel implementations. For example, some metaheuristics include an improvement part in the initialization, and the number of threads in the two levels of this improvement are added to the number of threads for the initialization of the reference set, so obtaining $ThreadsIni = \{threads-scheme1, threads-level1-scheme2, threads-level2-scheme2\}$.

The metaheuristic scheme is used at two levels: for the hyperheuristic (HMS) and for the application of the metaheuristics determined from the metaheuristic parameters (*MetaheurParam*) in each element of the reference set in the metaheuristic (MS) with which the hyperheuristic is implemented. Thus, parallelism can be applied in the hyperheuristic and in the metaheuristics, with a total of four parallelism levels, but it will be preferable to parallelize at a high level, and normally parallelism is applied only in the hyperheuristic.

4 Modelling and auto-tuning of the shared-memory parameterized scheme

To reduce the execution time it is necessary to select the values of the parallelism parameters ($ThreadsIni$, $ThreadsCom$, $ThreadsImp$ and $ThreadsInc$) appropriately, which means a model of the execution time must be obtained for each function, and the number of threads of a loop or the number of threads in the first and the second parallelism levels must be established. So, the value of some parameters (20 in our experiments with GRASP, Scatter Search, Genetic algorithms and Tabu Search as basic algorithms) must be selected, and an auto-tuning methodology is systematically applied. A scheme of the auto-tuning process is shown in Figure 1. It is divided in three phases:

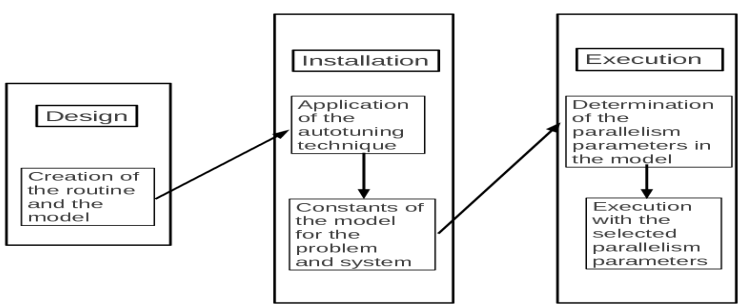


Figure 1: Phases of the auto-tuning process

- *First phase: Design.* The routine is developed together with its theoretical execution time. A model is obtained for each basic routine. Because two types of parallelism

have been identified in these routines, two basic models can be used, one for one-level routines and another for nested parallelism. For example, the generation of the initial population in function Initialize with an initial number of elements in the reference set $INEIni$ can be modelled:

$$t_{1-level} = \frac{k_g \cdot INEIni}{p} + k_p \cdot p \tag{1}$$

where k_g represents the cost of generating one individual; k_p the cost of generating one thread; and p is the number of threads. And the improvement of a percentage $PEIIni$ of the initial elements with an intensification (extension of improvement for each neighbour) $IIEIni$ is modelled:

$$t_{2-levels} = \frac{k_i \cdot \frac{INEIni \cdot PEIIni \cdot IIEIni}{100}}{p_1 \cdot p_2} + k_{p,1} \cdot p_1 + k_{p,2} \cdot p_2 \tag{2}$$

where k_i represents the cost of improving one element; $k_{p,1}$ and $k_{p,2}$ the cost of generating threads at the first and second level; and p_1 and p_2 the number of threads at each level.

For each of the other basic functions in Algorithm 2, the corresponding metaheuristic parameters are determined, and the model of the execution time is obtained as a function of those parameters and the parallelism parameters (the number of threads to select to be used in each routine and subroutine).

- *Second phase: Installation.* When the shared-memory parameterized scheme is being installed in a particular system, the value of the parameters influenced by the system are estimated. The parameters k_g, k_i, k_p, $k_{p,1}$ and $k_{p,2}$ used in step 1 are some of those parameters, as are the corresponding parameters for the other basic routines in Algorithm 2. We summarize the results of the installation of the scheme in an *HP Integrity Superdome SX2000* with 128 cores of *Intel Itanium-2 dual-core Montvale* with shared-memory. The optimum number of threads varies with the number of elements, and we are interested in the selection at running time of a number of threads close to the optimum. The model in equation 1 is used, and parameters k_g and k_p in the model are obtained by least-squares for a one-level routine, with a small number of elements (in order to have low installation time). In the experiments with a hyperheuristic with $INEIni = 5$, the values obtained are $k_g = 5.77 \cdot 10^{-1}$ and $k_p = 4.91 \cdot 10^{-2}$, all in seconds.

  For a two-level routine, like the routine to improve elements after the initial generation or after combining or mutation, the values of the parallelism parameters are obtained by least-squares with experiments with parameters for the hyperheuristic $INEIni = 10$, $PEIIni = 100$ and $IIEIni = 1$. The results are $k_i = 1.21$, $k_{p,1} = 1.04 \cdot 10^{-1}$

 ISBN:978-84-615-5392-1

and $k_{p,2} = 9.89 \cdot 10^{-2}$ seconds. By substituting these values in the theoretical model of the execution time (equation 2), the behaviour of the routine in the system is well predicted, as can be seen in Figure 2, where the theoretical and experimental speed-ups in the improvement of the initial population are represented for the hyperheuristic parameter combination $INEIni = 50$, $PEIIni = 50$ and $IIEIni = 1$.

- *Third phase: Execution.* At execution time the number of threads in each basic function is selected from the theoretical execution time (equations 1 and 2) with the values of the hyperheuristic parameters being those of the hyperheuristic we are experimenting with and the values of the system parameters those estimated in the installation phase. The number of threads which gives the theoretical mimimum execution time is obtained by minimizing the corresponding equation after substituting in it the values of the hyperheuristic and system parameters. For example, for the initial generation of the reference set:

$$p_{opt.} = \sqrt{\frac{k_g}{k_p} \cdot INEIni} = 3.43 \cdot \sqrt{INEIni} \tag{3}$$

and for the improvement of the generated elements:

$$p_{1,opt.} = 4.79 \cdot 10^{-1} \cdot \sqrt[3]{INEIni \cdot PEIIni \cdot IIEIni} \tag{4}$$

$$p_{2,opt.} = 5.05 \cdot 10^{-1} \cdot \sqrt[3]{INEIni \cdot PEIIni \cdot IIEIni} \tag{5}$$

To validate the auto-tuning methodology the optimum number of threads and the maximum speed-up achieved are calculated from the model for different hyperheuristic parameters using the system parameters obtained in the installation. Tables 1 and 2 compare the results for the initial generation of the reference set and for the improvement of elements for two parameter combinations. The number of threads selected with the auto-tuning methodology is not far from the experimental optimum and, as a consequence, the speed-up achieved with auto-tuning is not far from the maximum and the auto-tuning methodology is useful for the reduction of the execution time of hyperheuristics, which have a high cost caused by the application of a large number of metaheuristics.

We can compare the results obtained for the hyperheuristic using the auto-tuning metodology with those achieved when directly applying individual metaheuristics to a problem of optimization of electrical costs. Since the metaheuristic scheme is the same, similar results would be expected in both cases, although there may be differences due to different implementations. For example, in the improvement function of the MS, the second level was used to start more threads to work on the improvement of the fitness function (more

 ISBN:978-84-615-5392-1

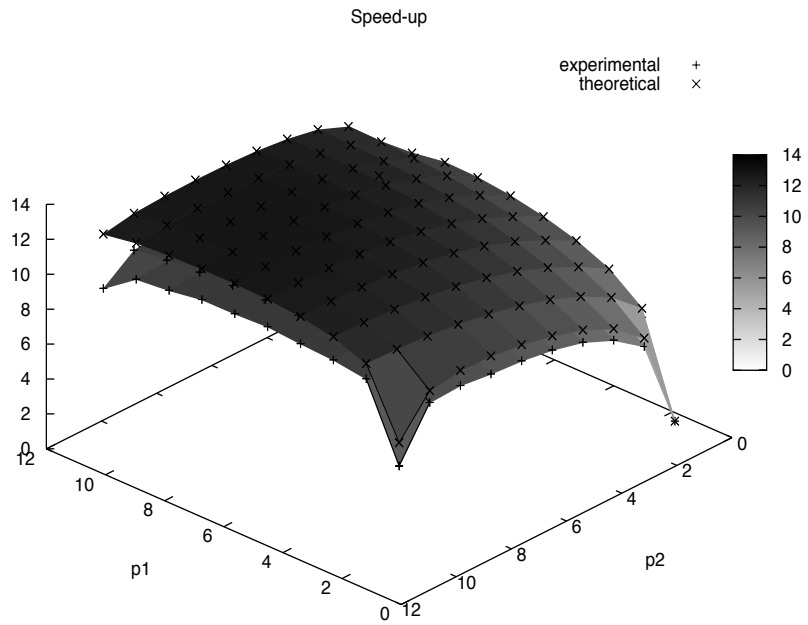


Figure 2: Theoretical and experimental speed-ups when varying the number of threads of the first and second level of parallelism for one parameter combination in a two-level parallel routine when applying the HMS.

Table 1: Speed-up and number of threads for $INEIni = 20$ and 100 in the one-level parallel routine when applying the HMS. Optimum experimental values (optimum), values obtained with auto-tuning (model) and experimental speed-up values obtained from threads given by model (opt.-mod.).

	threads		speed-up		
$INEIni$	optimum	model	optimum	model	opt.-mod.
20	22	15	11	8	8
100	24	34	12	17	12

 ISBN:978-84-615-5392-1

neighbours are analysed) but not to reduce the execution time, from which the number of threads of second level could be taken as constant. So, in this case the model is slightly different. In the function of the initial generation of elements there are no differences in the implementation. The behaviour of the one-level routine when applying the MS was well predicted, as can be seen in Figure 3, where the theoretical and experimental speed-up are represented.

Tables 3 and 4 compare the results for the initial generation of the reference set and for the improvement of elements for two parameter combinations using the MS. As in the case of the HMS, the number of threads and the speed-up selected with the auto-tuning methodology was not far from the experimental optimum. It can be seen that the technique applied for MS is also valid for HMS.

Table 2: Speed-up and number of threads for other two parameter combinations in the two-level parallel routine when applying the HMS. Optimum experimental values (optimum), values obtained with auto-tuning (model) and experimental speed-up values obtained from threads given by model (opt.-mod.).

			threads 1-level		threads 2-levels		speed-up		
$INEIni$	$PEIIni$	$IIEIni$	optimum	model	optimum	model	optimum	model	opt.-mod.
50	50	1	9	6	8	7	14	15	11
100	50	1	9	8	4	9	15	24	14

Table 3: Speed-up and number of threads for $INEIni = 100$ and 500 in the one-level parallel routine when applying the MS. Optimum experimental values (optimum), values obtained with auto-tuning (model) and experimental speed-up values obtained from threads given by model (opt.-mod.).

	threads		speed-up		
$INEIni$	optimum	model	optimum	model	opt.-mod.
100	55	35	22	18	21
500	64	78	44	39	44

 ISBN:978-84-615-5392-1

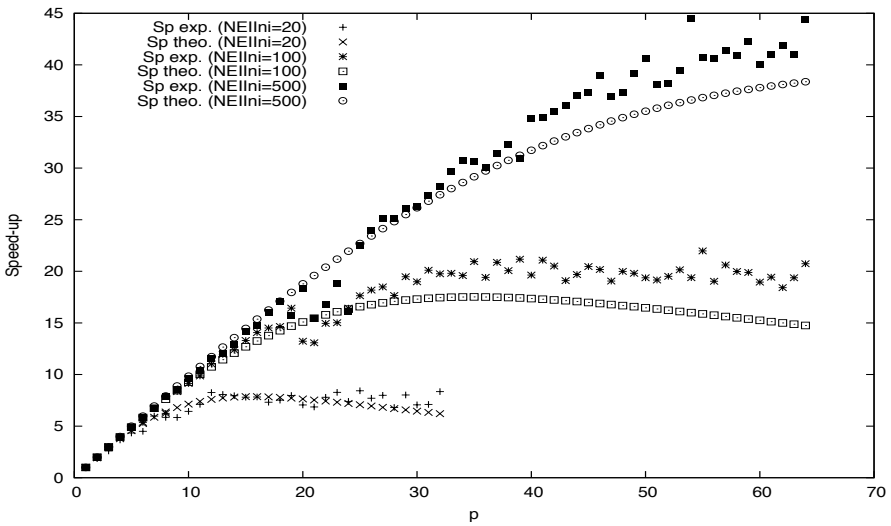


Figure 3: Theoretical and experimental speed-up when varying the number of threads for three parameters in a one-level parallel routine when applying the MS.

Table 4: Speed-up and number of threads for other parameter combinations in the two-level parallel routine when applying the MS. Optimum experimental values (optimum), values obtained with auto-tuning (model) and experimental speed-up values obtained from threads given by model (opt.-mod.).

			threads		speed-up		
$INEIni$	$PEIIni$	$IIEIni$	optimum	model	optimum	model	opt.-mod.
100	50	10	30	26	15	11	13
500	100	5	32	59	29	27	29

 ISBN:978-84-615-5392-1

5 Conclusions and future work

The auto-tuning methodology previously applied to parameterized shared-memory metaheuristic schemes can be applied to hyperheuristics based on metaheuristic schemes. The auto-tuning in this case offers additional difficulties to those in previous works, but the benefit is more apparent, due to the large execution time of the resulting hyperheuristic. The applicability of the methodology has been shown with a problem of minimization of electricity consumption in wells exploitation and in a large shared-memory system. The methodology provides satisfactory values for the number of threads to use in the application of the parallel hyperheuristic, which has been shown for two basic functions in the hyperheuristic scheme, but the auto-tuning works the same way for the other functions.

The auto-tuning methodology has not yet been integrated in the metaheuristic scheme, and this is the most immediate step. Another possibility to improve the application of the hyperheuristic is to determine search ranges for each metaheuristic parameter, so reducing the possible values of the elements in the metaheuristic with which the hyperheuristic is implemented.

Similar parameterized, parallel metaheuristic schemes together with the corresponding auto-tuning methodology should be developed for message-passing or GPU, which may be preferable for the application of hyperheuristics with large reference sets or with a high cost of the fitness function (application of the underlying methodologies).

Acknowledgements

Partially supported by Fundación Séneca, Consejería de Educación de la Región de Murcia, 08763/PI/08, and the High-Performance Computing Network on Parallel Heterogeneus Architectures (CAPAP-H). The authors gratefully acknowledge the computer resources and assistance provided by the Supercomputing Centre of the Scientific Park Foundation of Murcia.

References

[1] Alba, E.: Parallel Metaheuristics: A New Class of Algorithms. Wiley Interscience (2005).

[2] Almeida, F., Cuenca, J., Giménez, D., Llanes-Castro, A., Martínez-Gallar, J.P.: A framework for the application of metaheuristics to tasks-to-processors assignation problems. The Journal of Supercomputing. Published on-line (2009).

[3] Almeida, F., Giménez, D., López-Espín, J.J.: A parameterised shared-memory scheme for parameterised metaheuristics. The Journal of Supercomputing 58(3): 292–301 (2011).

[4] Clinton Whaley, R., Petitet, A., Dongarra, J.: Automated empirical optimizations of software and the ATLAS project. Parallel Computing **27**, 3–35 (2001).

[5] Cuenca, J., Giménez, D., González, J.: Architecture of an automatic tuned linear algebra library. Parallel Computing **30**, 187–220 (2004).

[6] Cuenca, J., Giménez, D., Martínez-Gallar, J.P.: Heuristics for work distribution of a homogeneous parallel dynamic programming scheme on heterogeneous systems. Parallel Computing **31**, 717–735 (2005).

[7] Cutillas-Lozano, J.M., Cutillas-Lozano, L.G., Giménez, D.: Modeling shared-memory metaheuristic schemes for electricity consumption. In: DCAI, 33–40 (2012).

[8] Cutillas-Lozano, J.M., Cutillas-Lozano, L.G., Giménez, D.: Resolución de un problema de optimización de consumo eléctrico en explotación de pozos por medio de metaheurísticas parametrizadas. In: MAEB, 391–398 (2012).

[9] Cutillas-Lozano, L.G.: Metaheurística aplicada a la optimización de los criterios de producción de aguas subterráneas. Sondea Project. Final-year dissertation, University of Alicante, 2008.

[10] López-Espín, J.J., Giménez, D.: Genetic algorithms for simultaneous equation models. In: DCAI, 215–224 (2008).

[11] Raidl, G.R.: A unified view on hybrid metaheuristics. Hybrid Metaheuristics Third International Workshop, LNCS **4030**, 1–12 (2006).

Proceedings of the 12th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2012
July, 2-5, 2012.

Evaluating the impact of cell renumbering of unstructured meshes on the performance of finite volume GPU solvers

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

In this work, we study the impact of renumbering the cells of unstructured triangular finite volume meshes on the performance of CUDA implementations of several finite volume schemes to simulate two-layer shallow water systems. We have used several numerical schemes with different demands of computational power whose CUDA implementations exploit the texture and L1 cache units of the GPU multiprocessors. Two different reordering schemes based on reducing the bandwidth of the adjacency matrix for the volume mesh have been used. Several numerical experiments performed on a Fermi-class GPU show that enforcing an ordering which enhances the data locality can have a significant impact on the runtime, and this impact is higher when the numerical scheme is computationally expensive.

1 Introduction

Currently, Graphics Processing Units (GPUs) are being used extensively to accellerate considerably numerical simulations in science and engineering. These platforms make it possible to achieve speedups of an order of magnitude over a standard CPU in many applications and are growing in popularity [16]. In particular, GPUs have been used in many applications based on finite volume numerical schemes [1, 2, 3, 7]. Currently, most of the GPU implementations of numerical schemes are based on the CUDA framework [14] which includes an extension of the C/C++ language to facilitate the programming of NVIDIA GPUs for general purpose applications.

Although the performance of finite volume computations for unstructured meshes could be substantially improved by using GPU platforms, the irregularity of the memory access patterns hampers this goal.

Obtaining high performance on a CUDA-enabled GPU implementation of unstructured mesh computations is not easy because mesh data cannot be easily laid out so as to enable coalescing [15]. However, the renumbering of the data cells of a mesh has proved to be an important way to improve the performance in parallel numerical computations which work on unstructured meshes [4] because a suitable data ordering optimizes the cache usage. In GPU, this approach could be possible if the ordering enables enough locality to use textures and to improve the L1 cache usage.

In modern CUDA-enabled GPUs, reads from texture memory are cached in a manner that preserves spatial locality, meaning that data reads from nearby points in space will possibly be cache hits. On the other hand, in Fermi class GPUs, the same on-chip memory can be dedicated mostly as L1 cache for each kernel call to reduce bandwidth demand [15]. A better access to texture and global memory can be achieved by renumbering the elements in an unstructured mesh such that the elements nearby in the mesh remain nearby in texture and global memory, enabling a better exploitation of the texture and L1 cache. Thus, one can obtain substantial performance improvements without changing the code.

In this paper, we study the impact of renumbering the cells of unstructured triangular finite volume meshes on the GPU performance for CUDA implementations of several finite volume two-layer shallow water solvers. These CUDA solvers have been implemented to take advantage of the texture and L1 cache units of a Fermi-class GPU, and exhibit different numerical intensity profiles. In order to apply a cell reordering which enhances the data locality, two reordering schemes based on reducing the bandwidth of the adjacency matrix for the mesh are used. Our goal is to evaluate the effect of these reordering techniques on the runtime of the finite volume CUDA solvers.

The outline of the article is as follows: the next section describes the underlying mathematical model and presents three finite volume numerical schemes to solve it. In Section 3 the CUDA implementation of the schemes is briefly described. Next, two bandwidth reduction techniques which will be used as renumbering strategies are introduced in Section 4. Section 5 shows and analyzes the performance results obtained when the different CUDA solvers are applied to two test problems on a NVIDIA GTX 580 GPU using different ordering strategies. Finally, conclusions are drawn in Section 6.

2 Mathematical model and numerical schemes

The two-layer shallow water system [5] is a system of partial differential equations which governs the 2d flow of two superposed immiscible layers of shallow fluids in a subdomain $\Omega \subset \mathbb{R}^2$. This system has been used as the numerical model to simulate ocean and estuarine

currents, oil spills, ... and has the following form:

$$\frac{\partial W}{\partial t}+\frac{\partial F_1}{\partial x}(W)+\frac{\partial F_2}{\partial y}(W)=B_1(W)\frac{\partial W}{\partial x}+B_2(W)\frac{\partial W}{\partial y}+S_1(W)\frac{\partial H}{\partial x}+S_2(W)\frac{\partial H}{\partial y}, \quad (1)$$

where $W=\begin{pmatrix} h_1 & q_{1,x} & q_{1,y} & h_2 & q_{2,x} & q_{2,y}\end{pmatrix}^T$,

$$F_1(W)=\begin{pmatrix} q_{1,x} & \frac{q_{1,x}^2}{h_1}+\frac{1}{2}gh_1^2 & \frac{q_{1,x}q_{1,y}}{h_1} & q_{2,x} & \frac{q_{2,x}^2}{h_2}+\frac{1}{2}gh_2^2 & \frac{q_{2,x}q_{2,y}}{h_2}\end{pmatrix}^T,$$

$$F_2(W)=\begin{pmatrix} q_{1,y} & \frac{q_{1,x}q_{1,y}}{h_1} & \frac{q_{1,y}^2}{h_1}+\frac{1}{2}gh_1^2 & q_{2,y} & \frac{q_{2,x}q_{2,y}}{h_2} & \frac{q_{2,y}^2}{h_2}+\frac{1}{2}gh_2^2\end{pmatrix}^T,$$

$$S_k(W)=\begin{pmatrix} 0 & gh_1(2-k) & gh_1(k-1) & 0 & gh_2(2-k) & gh_2(k-1)\end{pmatrix}^T, \quad k=1,2,$$

$$B_k(W)=\begin{pmatrix} \mathbf{0} & \mathcal{P}_{1,k}(W) \\ r\mathcal{P}_{2,k}(W) & \mathbf{0}\end{pmatrix}, \quad \mathcal{P}_{l,k}(W)=\begin{pmatrix} 0 & 0 & 0 \\ -gh_l(2-k) & 0 & 0 \\ -gh_l(k-1) & 0 & 0\end{pmatrix}, \quad l=1,2.$$

Index 1 in the unknowns makes reference to the upper fluid layer and index 2 to the lower one; g is the gravity and $H(\boldsymbol{x})$, the depth function measured from a fixed level of reference; $r=\rho_1/\rho_2$ is the ratio of the constant densities of the layers ($\rho_1<\rho_2$) which, in realistic oceanographical applications, is close to 1. Finally, $h_i(\boldsymbol{x},t)$ and $\boldsymbol{q}_i(\boldsymbol{x},t)$ are, respectively, the thickness and the mass-flow of the i-th layer at the point $\boldsymbol{x}$ at time t, and they are related to the velocities $\boldsymbol{u}_i(\boldsymbol{x},t)=(u_{i,x}(\boldsymbol{x},t),u_{i,y}(\boldsymbol{x},t))$, $i=1,2$ by the equalities: $\boldsymbol{q}_i(\boldsymbol{x},t)=\boldsymbol{u}_i(\boldsymbol{x},t)h_i(\boldsymbol{x},t)$, $\quad i=1,2$.

To discretize System (1), the computational domain D is divided into L cells or finite volumes $V_i\subset\mathbb{R}^2$, which are assumed to be triangles. Given a finite volume V_i, $N_i\in\mathbb{R}^2$ is the barycenter 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 neighbouring 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 V_j [5] (see Fig. 1).

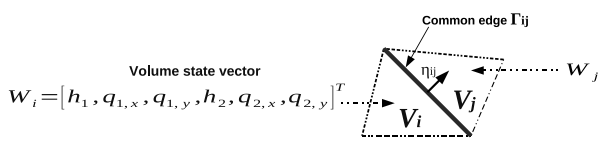


Figure 1: Finite volumes

 ISBN:978-84-615-5392-1

Assume that the approximations at time t^n, W_i^n $(i = 1, \ldots, L)$, have already been computed. To advance in time, with Δt^n being the time step, all the numerical schemes which will be used have the following general form:

$$W_i^{n+1} = W_i^n - \frac{\Delta t^n}{|V_i|} \sum_{j \in \aleph_i} |\Gamma_{ij}| \, F_{ij}^-(W_i^n, W_j^n, H_i, H_j) \tag{2}$$

where $|V_i|$ is the area of V_i.

The computation of $F_{ij}^-(W_i^n, W_j^n, H_i, H_j) \in \mathbb{R}^6$ in (2) depends on the particular numerical scheme. Here, three different numerical schemes will be presented: the classical Roe scheme [5], the IR-Roe scheme [2] and the PVM-IFCP scheme [6].

To compute the n-th time step, the following condition can be used:

$$\Delta t^n = \min_{i=1,\ldots,L} \left\{ \left[\frac{\sum_{j \in \aleph_i} \mid \Gamma_{ij} \mid\mid\mid \mathcal{D}_{ij} \mid\mid_\infty}{2\gamma \mid V_i \mid} \right]^{-1} \right\} \tag{3}$$

where γ, $0 < \gamma \leq 1$, is the CFL (Courant-Friedrichs-Lewy) parameter.

2.1 The classical Roe scheme

In the classical Roe scheme, F_{ij}^- (herein called F_{ij}^{ROE-}) is computed as follows:

$$\begin{aligned}
\mathcal{F}_{ij}^{ROE^-}(W_i^n, W_j^n, H_i, H_j) &= P_{ij}^- \left(A_{ij} \left(W_j^n - W_i^n \right) - S_{ij} \left(H_j - H_i \right) \right) + F_{\boldsymbol{\eta}_{ij}}(W_i^n) \,, \\
\mathcal{F}_{ij}^{ROE^+}(W_i^n, W_j^n, H_i, H_j) &= P_{ij}^+ \left(A_{ij} \left(W_j^n - W_i^n \right) - S_{ij} \left(H_j - H_i \right) \right) - F_{\boldsymbol{\eta}_{ij}}(W_j^n) \,.
\end{aligned}$$

$F_{ij}^{ROE^-}$ and $F_{ij}^{ROE^+}$ are the contributions of the edge Γ_{ij} to the state of the volumes V_i and V_j, respectively, where $F_{\boldsymbol{\eta}_{ij}}(W) = F_1(W)\,\eta_{ij,x} + F_2(W)\,\eta_{ij,y}$ and $H_\alpha = H(N_\alpha)$ with $\alpha = i, j$. $A_{ij} \in \mathbb{R}^{6\times 6}$ and $S_{ij} \in \mathbb{R}^6$ depends on W_i^n and W_j^n (see [5] for more details). The matrix $P_{ij}^\pm$ is calculated as:

$$P_{ij}^\pm = \frac{1}{2} \mathcal{K}_{ij} \cdot (I \pm \operatorname{sgn}(\mathcal{D}_{ij})) \cdot \mathcal{K}_{ij}^{-1}$$

where I is the identity matrix, $\operatorname{sgn}(\mathcal{D}_{ij})$ is a diagonal matrix whose coefficients are the sign of the eigenvalues of A_{ij}, and the columns of $\mathcal{K}_{ij} \in \mathbb{R}^{6\times 6}$ are the associated eigenvectors (see [5] for more details).

2.2 The IR-Roe scheme

The IR-Roe scheme exploits that system (1) verifies the property of rotational invariance (see [2] for more details) to reduce the computational costs without losing excessive accuracy. The resultant formula for $F_{ij}^\pm$ (herein called $F_{ij}^{IR-ROE\pm}$) reads as follows:

$$F_{ij}^{IR-ROE\pm} = T_{\eta_{ij}}^{-1} \left[\left(\Phi^{\pm}_{\boldsymbol{\eta}_{ij}}\right)_{[1]} \quad \left(\Phi^{\pm}_{\boldsymbol{\eta}_{ij}}\right)_{[2]} \quad \left(\Phi^{\pm}_{\boldsymbol{\eta}^{\perp}_{ij}}\right)_{[1]} \quad \left(\Phi^{\pm}_{\boldsymbol{\eta}_{ij}}\right)_{[3]} \quad \left(\Phi^{\pm}_{\boldsymbol{\eta}_{ij}}\right)_{[4]} \quad \left(\Phi^{\pm}_{\boldsymbol{\eta}^{\perp}_{ij}}\right)_{[2]} \right]^T.$$

where:

- $T_{\eta_{ij}} = \begin{pmatrix} R_{\eta_{ij}} & \mathbf{0} \\ \mathbf{0} & R_{\eta_{ij}} \end{pmatrix}, \quad R_{\eta_{ij}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \eta_{ij,x} & \eta_{ij,y} \\ 0 & -\eta_{ij,y} & \eta_{ij,x} \end{pmatrix}.$

- $\left(\Phi^{\pm}_{\boldsymbol{\eta}_{ij}}\right)_{[l]}$ is the l-th component of the vector $\Phi^{\pm}_{\boldsymbol{\eta}_{ij}} \in I\!R^4$ which is defined as follows:

  $\Phi^{-}_{\boldsymbol{\eta}_{ij}} = P^{-}_{ij} \left(\mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},j}) - \mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},i}) - \mathcal{B}_{ij} - \mathcal{S}_{ij}\,(H_j - H_i) \right) + \mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},i})\,,$
  $\Phi^{+}_{\boldsymbol{\eta}_{ij}} = P^{+}_{ij} \left(\mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},j}) - \mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},i}) - \mathcal{B}_{ij} - \mathcal{S}_{ij}\,(H_j - H_i) \right) - \mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},j})\,.$

  Here $P^{\pm}_{ij} = \frac{1}{2}\mathcal{K}_{ij} \cdot (I \pm \mathrm{sgn}(\mathcal{D}_{ij})) \cdot \mathcal{K}_{ij}^{-1}$, where I is the identity matrix, $\mathcal{K}_{ij}$ is the matrix whose columns are the eigenvectors of a Roe matrix $\mathcal{A}_{ij} \in \mathbb{R}^{4\times 4}$ which depends on $W^n_{\boldsymbol{\eta}_{ij},i}$ and $W^n_{\boldsymbol{\eta}_{ij},j}$ (see [2] for more details), being $W^n_{\boldsymbol{\eta}_{ij},\alpha} = T_{\boldsymbol{\eta}_{ij}} W^n_{\alpha}$ ($\alpha = i, j$). $\mathrm{sgn}(\mathcal{D}_{ij})$ is the diagonal matrix whose coefficients are the signs of the eigenvalues of $\mathcal{A}_{ij}$. $\mathcal{F}_1(W^n_{\boldsymbol{\eta}_{ij},\alpha}) = F_1(W^n_{\boldsymbol{\eta}_{ij},\alpha})_{[1,2,4,5]}$ and $\mathcal{S}_{ij} = S_1(W^n_{\boldsymbol{\eta}_{ij},ij})_{[1,2,4,5]}$, $\mathcal{B}_{ij} = \left(B_1(W^n_{\boldsymbol{\eta}_{ij},ij}) \left(W^n_{\boldsymbol{\eta}_{ij},j} - W^n_{\boldsymbol{\eta}_{ij},i}\right)\right)_{[1,2,4,5]}$ ($\alpha = i, j$), where $W^n_{\boldsymbol{\eta}_{ij},ij}$ is the 'Roe intermediate state' corresponding to $W^n_{\boldsymbol{\eta}_{ij},i}$ and $W^n_{\boldsymbol{\eta}_{ij},j}$ (see [5] for more details), and $W_{[i_1,\cdots,i_s]}$ is the vector defined from vector W, using its i_1-th, ..., i_s-th components.

- $\Phi^{\pm}_{\boldsymbol{\eta}^{\perp}_{ij}} = \mp \left[\left(\Phi^{-}_{\boldsymbol{\eta}_{ij}}\right)_{[1]} u^*_{1,\boldsymbol{\eta}^{\perp}_{ij}} \quad \left(\Phi^{-}_{\boldsymbol{\eta}_{ij}}\right)_{[3]} u^*_{2,\boldsymbol{\eta}^{\perp}_{ij}} \right]^{\mathrm{T}}$,where $u^*_{k,\boldsymbol{\eta}^{\perp}_{ij}}$ is defined as follows:

$$u^*_{k,\boldsymbol{\eta}^{\perp}_{ij}} = \begin{cases} \dfrac{\left(W^n_{\boldsymbol{\eta}_{ij},i}\right)_{[k(k+2)]}}{h_{k,i}} & \text{If } \left(\Phi^{-}_{\boldsymbol{\eta}_{ij}}\right)_{[2k-1]} > 0 \\ \dfrac{\left(W^n_{\boldsymbol{\eta}_{ij},j}\right)_{[k(k+2)]}}{h_{k,j}} & \text{Otherwise} \end{cases}, \quad k = 1, 2.$$

2.3 The PVM-IFCP-scheme

The PVM (Polynomial Viscosity Matrix) schemes are a family of numerical schemes for non conservative hyperbolic systems [6] which are defined in terms of viscosity matrices obtained from the polynomial evaluation of a Roe matrix. The main advantage of these methods is that they only need some information about the eigenvalues of the system and the spectral decomposition of the Roe matrix is not needed unlike the previous schemes.

For a PVM scheme, $F^{-}_{ij}(W^n_i, W^n_j, H_i, H_j)$ is obtained by applying a similar process to that described in 2.2 to derive $\mathcal{F}^{IR-ROE^{\pm}}_{ij}$. However, the flux $\Phi^{\pm}_{\boldsymbol{\eta}_{ij}}$ is obtained by:

$$\begin{aligned}
\Phi^{-}_{\boldsymbol{\eta}_{ij}} &= \frac{1}{2}\Big(\mathcal{F}_1(W_{\boldsymbol{\eta}_{ij},j}) - \mathcal{F}_1(W_{\boldsymbol{\eta}_{ij},i}) - \mathcal{B}_{ij} - \mathcal{S}_{ij}\left(H_j - H_i\right) \\
&\quad - Q_{ij}\left((W_{\boldsymbol{\eta}_{ij},j} - W_{\boldsymbol{\eta}_{ij},i})_{[1,2,4,5]} - \mathcal{A}_{ij}^{-1}\mathcal{S}_{ij}\left(H_j - H_i\right)\right)\Big) + \mathcal{F}_1(W_{\boldsymbol{\eta}_{ij},i})\,, \\
\Phi^{+}_{\boldsymbol{\eta}_{ij}} &= \frac{1}{2}\Big(\mathcal{F}_1(W_{\boldsymbol{\eta}_{ij},j}) - \mathcal{F}_1(W_{\boldsymbol{\eta}_{ij},i}) - \mathcal{B}_{ij} - \mathcal{S}_{ij}\left(H_j - H_i\right) \\
&\quad + Q_{ij}\left((W_{\boldsymbol{\eta}_{ij},j} - W_{\boldsymbol{\eta}_{ij},i})_{[1,2,4,5]} - \mathcal{A}_{ij}^{-1}\mathcal{S}_{ij}\left(H_j - H_i\right)\right)\Big) - \mathcal{F}_1(W_{\boldsymbol{\eta}_{ij},j})\,, \qquad (4)
\end{aligned}$$

where $\mathcal{F}_1(W_{\boldsymbol{\eta}_{ij}})$ and $\mathcal{B}_{ij}$ are defined in 2.2. Q_{ij} is the viscosity matrix defined as $Q_{ij} = \alpha_0^{ij} I + \alpha_1^{ij}\mathcal{A}_{ij} + \alpha_2^{ij}\mathcal{A}_{ij}^2 + \ldots + \alpha_l^{ij}\mathcal{A}_{ij}^l$, where I is an identity matrix and α_k^{ij}, $k = 0, \ldots, l$, are particular coefficients of the PVM scheme.

The PVM-IFCP (Intermediate Field Capturing Parabola) scheme [9] is defined by the coefficients α_k, $k = 0, 1, 2$, obtained by solving the following system:

$$\begin{pmatrix} 1 & \lambda_1 & (\lambda_1)^2 \\ 1 & \lambda_n & (\lambda_4)^2 \\ 1 & \chi_{int} & (\chi_{int})^2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} |\lambda_1| \\ |\lambda_4| \\ |\chi_{int}| \end{pmatrix} \qquad (5)$$

being $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$ the eigenvalues of the matrix $\mathcal{A}_{ij}$ and

$$\chi_{int} = \mathcal{S}_{ext} \cdot \max_{2 \le i \le 3}(|\lambda_i|) \text{ with } \mathcal{S}_{ext} = \begin{cases} \operatorname{sgn}(\lambda_1 + \lambda_4) & \text{si } (\lambda_1 + \lambda_4) \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

The choice of these coefficients provokes that this scheme is linearly L^∞ stable with the aforementioned CFL condition (see Equation (3)).

3 CUDA implementation of the schemes

The general structure of the CUDA implementation of the three numerical schemes exposed in Section 2 is the same for all the schemes. This implementation is a variant of the implementation described in [7], Section 7.3 and [2], Section 5. The general steps of the implementation are depicted in Figure 2. Each step executed on the GPU is assigned to a CUDA kernel. Next, we briefly describe each step:

- Build finite volume mesh: Volume data is stored in two arrays of L `float4` elements as 1D textures, where each element contains the data (state, depth and area) of a cell. We have used textures because each edge (thread) only needs the data of adjacent cells and texture memory is especially suited for each thread to access its closer environment in texture memory by exploiting the texture cache. Edge data is stored in two arrays in global memory with a size equal to the number of edges: an array of `float2` elements for storing

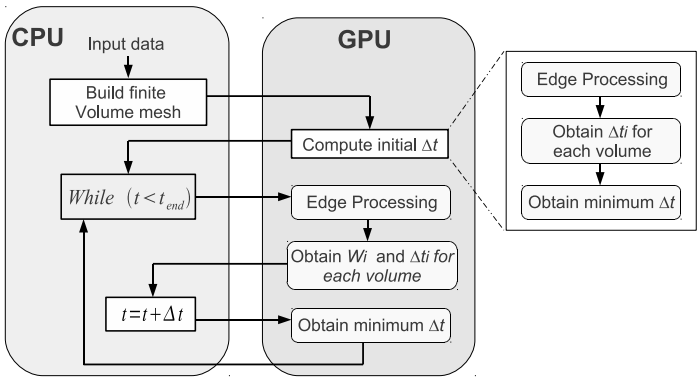


Figure 2: Structure of the CUDA Implementation for all the schemes

the normals, and another array of `int4` elements for storing, for each edge, the positions of the neighboring volumes in the volume textures and the positions of the two accumulators where the edge must write its contributions to the state of the neighboring volumes.

- Edge Processing: In this step each thread represents an edge $\Gamma_{ij}, \quad i,j \in 1,\ldots,L$, and computes the contribution $F_{ij}^{\pm}$ of the edge to their adjacent volumes V_i and V_j. This is the most costly computation phase and the particular calculations performed in this step depends on the particular numerical scheme.

The threads contribute to a particular cell by means of six accumulators, each one being an array of L `float4` elements stored in global memory (see [7] for more details). The ordering of the elements at each accumulator matches with the ordering of the elements in the volume data array.

- Compute W_i and Δt_i for each volume: In this step, each thread represents a volume and computes the local Δt_i of the volume V_i in accordance with (3) and also updates the state W_i of that volume in accordance with (2) (see [7] for more details).

Since the 1D textures containing the volume data are stored in linear memory, we update the textures by writing directly into them.

- Obtain minimum Δt: This phase finds the minimum of the local Δt_i of the volumes by applying the most optimized kernel of the reduction sample included in the CUDA Software Development Kit [14]. The global step size Δt obtained will be used in the next iteration.

4 Reordering techniques based on bandwidth reduction

Many algorithms to reduce the bandwidth of a sparse symmetric matrix have been published in the literature. The Cuthill-McKee [8] and the Gibbs, Poole and Stockmeyer (GPS) [11]

algorithms are two of the most popular. The Reverse Cuthill-McKee algorithm (RCM) [10] is a modification of the original algorithm where the resulting index numbers are reversed. RCM algorithm usually generates a more reduced profile than the original algorithm and consequently is most widely used.

Note that the application of a bandwitdh reduction algorithm to reorder the elements of the volume arrays enables adjacent volumes in the mesh to be in closer positions in the arrays which store the volume data. As a consequence, these algorithms can improve the data locality and enable an optimization of the cache usage. In this work, we will analyze the impact of ordering the mesh cells according to RCM and GPS algorithms.

5 Numerical Experiments

In this section we will study how the ordering of the volumes in the arrays which store the volume data affects the GPU execution times obtained with different solvers. We will consider two test problems:

Test 1 This test consists in an internal circular dambreak problem in the $[-5,5]\times[-5,5]$ domain. Depth is given by $H(x,y)=6$ and the initial condition is:

$$W_i^0(x,y)=\big(h_1(x,y),\quad 0,\quad 0,\quad h_2(x,y),\quad 0,\quad 0\big)^{\mathrm{T}},\qquad \text{where}$$

$$h_1(x,y)=\begin{cases} 4.0 & \text{if } \sqrt{x^2+y^2}>1.5 \\ 0.5 & \text{otherwise}\end{cases},\qquad h_2(x,y)=H-h_1(x,y)$$

The ratio of densities is $r=0.5$ and CFL paramenter is $\gamma=0.9$.

Test 2 This test represents two unstable water layers in the $[-5,5]\times[-5,5]$ domain. Depth function is $H(x,y)=1-1.5\cdot e^{-x^2-y^2}$ and the initial state is:

$$W_i^0(x,y)=\big(h_1(x,y),\quad 0,\quad 0,\quad h_2(x,y),\quad 0,\quad 0\big)^{\mathrm{T}},\qquad \text{where}$$

$$h_1(x,y)=\begin{cases} 4.0 & \text{if } x\geq 0 \\ 0.5 & \text{otherwise}\end{cases},\quad h_2(x,y)=\begin{cases} 0.5 & \text{if } x\geq 0 \\ 4.0 & \text{otherwise}\end{cases}$$

The ratio of densities is $r=0.98$ and CFL parameter is $\gamma=0.9$.

For both test problems, the simulation time is 0.1 seconds and wall boundary conditions ($q_1\cdot\boldsymbol{\eta}=0,\ q_2\cdot\boldsymbol{\eta}=0$) are considered. All the CUDA programs have been executed on an Intel Xeon server with 8 GB RAM containing a GeForce GTX 580 card and the GNU compiler has been used to derive the executables. We assign a size of 48 KB to L1 cache

and 16 KB to shared memory in all the kernels excepting the kernel used to obtain the minimum Δt. The edge processing kernel has been executed using a one dimensional grid of blocks with a blocksize equals to 64 threads.

	Classical Roe			IR-Roe			PVM-IFCP		
Volumes	MATLAB	RCM	GPS	MATLAB	RCM	GPS	MATLAB	RCM	GPS
4000	0.069	0.072	0.071	0.020	0.022	0.021	0.0048	0.0050	0.0050
16000	0.44	0.38	0.43	0.092	0.081	0.084	0.020	0.024	0.023
64000	3.13	2.36	2.71	0.65	0.51	0.57	0.14	0.15	0.15
256000	22.91	15.12	17.04	4.68	3.26	3.70	1.01	1.00	1.04
1024000	167.2	99.98	108.9	33.97	21.34	23.70	7.81	7.47	7.69
2080560	625.6	384.6	480.0	127.1	78.58	94.61	29.37	27.98	28.74

Table 1: Execution times in seconds for test 1 before and after applying the RCM and GPS algorithms using a GeForce GTX 580.

	Classical Roe			IR-Roe			PVM-IFCP		
Volumes	MATLAB	RCM	GPS	MATLAB	RCM	GPS	MATLAB	RCM	GPS
4000	0.059	0.061	0.061	0.022	0.021	0.021	0.0042	0.0045	0.0044
16000	0.37	0.34	0.37	0.11	0.089	0.098	0.017	0.020	0.020
64000	2.71	2.19	2.44	0.79	0.53	0.59	0.11	0.12	0.12
256000	20.97	14.80	16.46	6.06	3.31	3.82	0.85	0.86	0.88
1024000	166.9	103.5	113.4	46.07	21.35	24.48	6.73	6.37	6.56
2080560	623.8	395.7	502.2	169.2	75.40	100.3	25.15	23.40	24.93

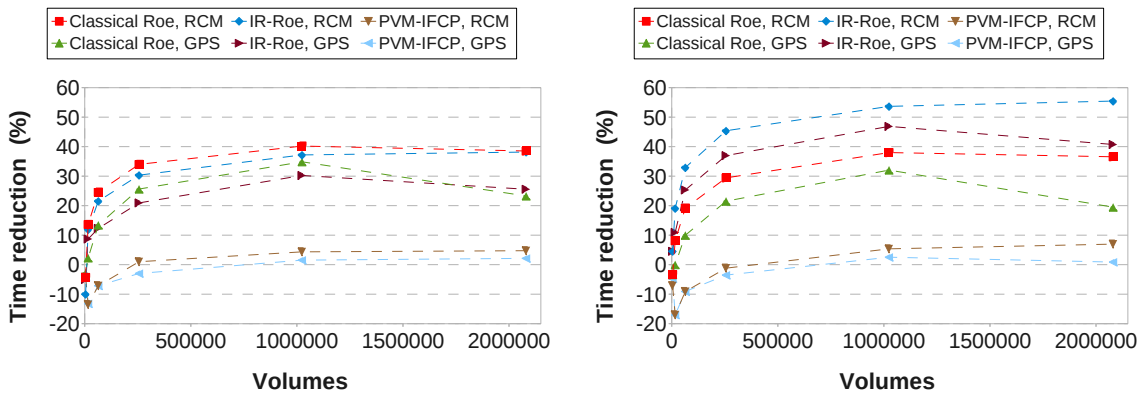
Table 2: Execution times in seconds for test 2 before and after applying the RCM and GPS algorithms using a GeForce GTX 580.

We have generated several triangular meshes using the Partial Differential Equation Toolbox for MATLAB [13]. In order to compare the different orderings of the volume arrays, we will execute the single precision CUDA programs of the classical Roe, IR-Roe and PVM-IFCP schemes using the two former test problems and three different volume orderings: the original provided by MATLAB and the resulting of applying the RCM and the GPS algorithms to the original meshes. The `symrcm` MATLAB function and the Fortran code given by the 582 TOMS algorithm [12] have been used to apply the RCM and GPS algoritms, respectively.

For all the volume orderings, Tables 1 and 2 show the GPU runtimes in seconds for tests 1 and 2, respectively, Figure 3 shows the time reduction obtained with all the numerical schemes and test problems, Figure 4 depicts graphically the adjacency matrices of the 4000 volumes mesh, and Table 3 shows the bandwidth of the adjacency matrices of all meshes.

We can see that, for the biggest meshes, the RCM ordering has given the best runtimes in all cases. Specifically, with the classical Roe and IR-Roe schemes, execution times have

 ISBN:978-84-615-5392-1



(a) (b)

Figure 3: GPU time reduction obtained with respect to the original meshes provided with MATLAB: (a) Test 1; (b) Test 2.

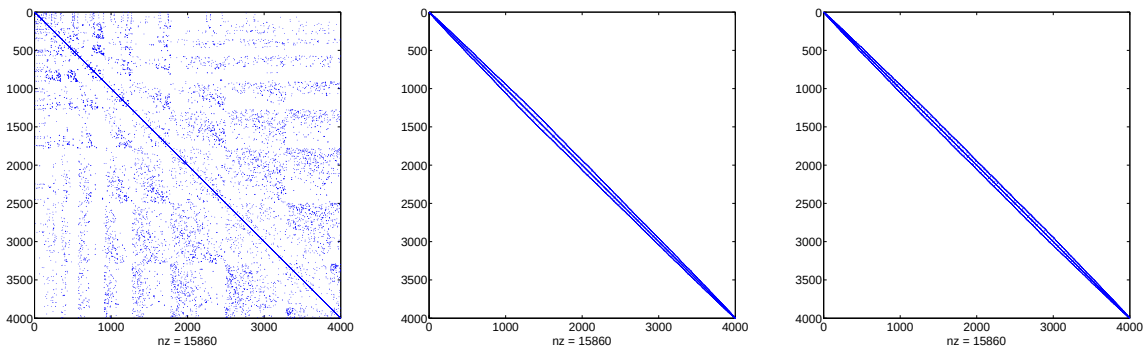
Number of volumes	MATLAB	RCM	GPS
4000	3786	73	62
16000	12000	147	126
64000	48000	294	249
256000	192000	598	500
1024000	768000	1206	1004
2080560	1996159	1728	1378

Table 3: Bandwidth of the meshes before and after applying RCM and GPS algorithms.

reduced approximately between 37 and 55 % for the meshes with more than one million volumes in both test problems, whereas using the PVM-IFCP scheme the runtimes have reduced the 5 %. The GPS ordering, although providing a lower bandwidth than RCM, has given worse execution times than RCM for the biggest meshes in all cases.

6 Conclusions

The performance optimization of finite volume shallow water solvers for unstructured meshes running on GPUs has been dealt. The reordering of the volumes in the arrays which store the volume data in GPU by using bandwidth reduction techniques makes it possible to reduce substantially the execution times obtained because the L1 and texture cache usage is optimized. The highest reduction has been achieved achieved when the reverse Cuthill-



(a) (b) (c)

Figure 4: Adjacency matrices for the mesh with 4000 volumes before and after applying RCM and GPS algorithms. `nz` denotes the number of non-zero elements: (a) original mesh obtained with MATLAB; (b) after applying RCM; (c) after applying GPS.

McKee ordering is applied to these arrays and we obtain an improvement of 5 to 55% approximately in the GPU simulation times (depending on the numerical scheme) with respect to the ordering provided when the volume mesh is generated by the MATLAB PDE toolkit. The impact seems to be higher when the numerical intensity of the solver and the problem size grow.

Acknowledgements

This research has been partially supported by the Spanish Government Research projects MTM09-11923, MTM2009-07719, and MTM2011-27739-C04-02.

References

[1] M. de la Asunción, J. M. Mantas and M. J. Castro, *Programming CUDA-based GPUs to simulate two-layer shallow water flows*, Euro-Par 2010 Ischia (Italy) (2010).

[2] M. de la Asunción, José M. Mantas, M. J. Castro, E.D. Fernández-Nieto, *An MPI-CUDA implementation of an improved Roe method for two-layer shallow water systems*, Journal of Parallel and Distributed Computing **in press** DOI: http://dx.doi.org/10.1016/j.bbr.2011.03.031 (2011).

[3] A. R. Brodtkorb, M. L. Sætra, and M. Altinakar, *Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation*, Computers & Fluids **55** (2011) 1–12.

[4] D.A. Burgess and M.B. Giles, *Renumbering unstructured grids to improve the performance of codes on hierarchical memory machines*, Adv. in Eng. Software **28 (3)** (1997) 189–201.

[5] M. J. Castro, J. A. García-Rodríguez, J. M. González-Vida and C. Parés, *A parallel 2D finite volume scheme for solving systems of balance laws with nonconservative products: Application to shallow flows*, Comput. Meth. Appl. Mech. Eng. **195** (2006) 2788–2815.

[6] M. J. Castro and E. D. Fernández-Nieto, *A class of computationally fast first order finite volume solvers: PVM methods*, Submitted to SIAM J. of Sci. Computing.

[7] M. J. Castro, S. Ortega, M. de la Asunción, J. M. Mantas and J. M. Gallardo, *GPU computing for shallow water flow simulation based on finite volume schemes*, Comptes Rendus Mécanique **339** (2011) 165–184.

[8] E. Cuthill and J. McKee, *Reducing the bandwidth of sparse symmetric matrices*, Proc. of the 24th Nat. Conf. ACM (1969), 157–172.

[9] E. D. Fernández-Nieto, M. J. Castro and C. Parés, *On an intermediate field capturing Riemann solver based on a parabolic viscosity matrix for the two-layer shallow water system*, Journal of Scientific Computing **48** (2011) 117–140.

[10] J. A. George and J. W-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, 1981.

[11] N. E. Gibbs, W. G. Poole and P. K. Stockmeyer, *An algorithm for reducing the bandwidth and profile of a sparse matrix*, SIAM J. on Num. Anal. **13** (1976) 236–250.

[12] J. G. Lewis, *Algorithm 582: The Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms for Reordering Sparse Matrices*, ACM Trans. Math. Softw. **8 (2)** (1982) 190–194.

[13] MathWorks, MATLAB R2011b, http://www.mathworks.com/products/matlab.

[14] NVIDIA Corporation, *CUDA Zone*, http://www.nvidia.com/object/cuda_home_new.html.

[15] NVIDIA Corporation, *CUDA C Best Practices Guide 4.1*, (2012).

[16] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone and J.C. Phillips, *GPU Computing*, Proceedings of the IEEE **96** (2008) 879–899.