Volume III

Contents:

Volume I

Contents:

Volume II

Contents:

Volume III

Contents:

Volume IV

Contents:

Volume V

.

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Towards a data assimilation method for blood circulation

Telma Guerra[1], Jorge Tiago[2] and Adélia Sequeira[2]

[1] *Departamento de Matemática, Instituto Superior Técnico - Universidade Técnica de Lisboa*

[2] *Departamento de Matemática, Escola Superior de Tecnologia - Instituto Politécnico de Setúbal*

emails: telma.guerra@estbarreiro.ips.pt, jorge.tiago@math.ist.utl.pt, adelia.sequeira@math.ist.utl.pt

Abstract

Data Assimilation can be used to achieve patient specific simulations of the blood flow. Integrating the data from medical imaging into the simulations can be done through an optimal control problem. We present an approach for this problem which includes dealing with the non-Newtonian character of the blood and WSS measurements. We finally present some numerical results for this approach.

Key words: blood circulation, non-Newtonian, optimal control

1 Introduction

The collaboration between the medical community and scientists offers an exchange of knowledge and data information between both strands. Such data can be used by the researchers in the numerical simulations to predict blood behavior under healthy or disease conditions. This process is called Data Assimilation. Although different techniques can be considered, in [3, 4] it has been shown that the so called variational approach can lead to better results when in the frame of hemodynamics modeling. Here we present some advances in this direction.

In the vascular system, one of the most frequently disease we can observe is the partial obstruction of the vessels, reducing the diameter which commits the normal behavior of the blood circulation. It is already known that hemodynamical issues can affect the progression

of this and others pathologies through the action of the shear stress exerted by the blood flow on the vessels walls, although it's influence is not fully known yet (see [1, 2]).

In normal situations the blood flow has a Newtonian behavior in most parts of the arterial system. In fact, when non-Newtonian effects are observed it may indicate that some abnormality can be occurring. Some blood effects can be observed, for instance stable recirculation of the flow downstream a stenosis or inside a saccular aneurysm. In such cases is important the study of non-Newtonian behavior as well as shear-thinning viscosity, thixotropy and eventually the yield stress of the blood.

In the same sense, it is believed that the development of aneurysms and its consequently rupture is also related to local hemodynamical factors and vessels structure. The size of the vessels, its radius, curvature and branching plays a role in the growth and rupture of the aneurysm. Some parameters like the wall shear stress (WSS) can also give some information about a patient condition. The WSS is measured in the vessel wall and high values are associated to disease conditions.

In Section 2 we define an control problem describing the data assimilation process, for a non-Newtonian dynamics. We use the inlet boundary condition as the control. In Section 2 we present some numerical results for solution of such an optimal control problem.

2 State equation and control problem formulation

We deal with generalized steady Navier-Stokes equations given by

$$-div\,(\tau(D\mathbf{y})) + (\mathbf{y}\cdot\nabla)\mathbf{y} + \nabla p = \mathbf{f} \tag{1}$$

with the divergence free condition

$$div\,\mathbf{y} = 0,$$

both defined in a domain Ω where the unknowns are the velocity field $\mathbf{y}$ and the pressure p, τ is the viscous stress tensor is represented by

$$\tau = 2\mu(\dot\gamma)D\mathbf{y}, \tag{2}$$

where μ is the kinematic viscosity and $\dot\gamma$ is the shear rate given by

$$\dot\gamma = \sqrt{\frac{1}{2}(\nabla\mathbf{y} + (\nabla\mathbf{y})^T) : (\nabla\mathbf{y} + (\nabla\mathbf{y})^T)}$$

and D is the strain tensor

$$D\mathbf{y} = \frac{1}{2}(\nabla\mathbf{y} + (\nabla\mathbf{y})^T).$$

The model considered for viscosity is the generalized Cross model given by

$$\mu(\dot\gamma) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (\lambda\,\dot\gamma)^b)^a} \tag{3}$$

 ISBN: 978-84-616-2723-3

where $a, b, \lambda > 0$. The constants μ_0 and μ_∞ are the asymptotic viscosity values at zero and at infinite shear rates.

Our purpose is to minimize the difference between the computed state variable and the data observations $\mathbf{y}_d$ registered a priori in part of the domain in such a way to capture the the correct behavior of the blood in the proposed domains. We intend to control the inlet boundary by introducing a Dirichlet control and to compare the obtained results with the data observations. The control problem can be formulated as

$$min\ \mathbf{J}(\mathbf{y},\mathbf{u}) = w_1 \int_{\Omega_{part}} |\mathbf{y}-\mathbf{y}_d|^2\, dx + w_2 \int_{\Gamma_{in}} |\nabla \mathbf{u}|^2\, dx + w_3 \int_{\Gamma_{wall}} |WSSm - WSSm_d|^2\, dx. \tag{4}$$

subject to

$$\begin{cases} -div\,\tau + (\mathbf{y}\cdot\nabla)\mathbf{y} + \nabla p = \mathbf{f} & \text{in } \Omega \\ div\,\mathbf{y} = 0 & \text{in } \Omega, \\ \mathbf{y} = 0 & \text{on } \Gamma_{wall} \\ \mathbf{y} = \mathbf{u} & \text{in } \Gamma_{in} \\ -\nu\nabla\mathbf{y}\cdot\mathbf{n} + p\mathbf{n} = 0 & \text{in } \Gamma_{out}. \end{cases} \tag{5}$$

The constants w_1, w_2 and w_3 are weights to balance the integrals in the cost function, $\mathbf{y}_d$ represents the data observations collected in a previous chosen part of the domain Ω, named Ω_{part}. The $WSSm$ is the magnitude of the WSS and the $WSSm_d$ is the data magnitude of the WSS both computed in the surface of the domain.

The WSS is the tangential stress exerted by the fluid in the vessels wall and is given by

$$WSS = \sigma_n - (\sigma_n \cdot \mathbf{n})\mathbf{n} \tag{6}$$

Where $\sigma = p\mathbf{I} - \tau$, $\mathbf{n}$ is the outward normal to the wall surface, σ_n is the normal component of the stress tensor.

3 Numerical results

After discretizing the optimal control problem using finite element methods we obtain a finite dimensional optimization problem corresponding to 35934 degrees of freedom. We solve this problem by a Sequential Quadratic Programming approach. We generate the data by solve the model using a Poiseiulle profile with the following parameter

$$\mu_0 = 0.004\, Pa.s, \quad \mu_\infty = 3.6e^{-3}\, Pa.s$$

 ISBN: 978-84-616-2723-3

$$a = 1.23, \quad b = 0.64, \quad \lambda = 8.2\, s$$
$$U_0 = 0.0662\, m/s, \quad \rho = 1059\, Kg/m^3.$$

where U_0 is the maximum velocity at the inlet.

We use an idealized geometry representing a stenosis. The velocity is represented in Figure 1.

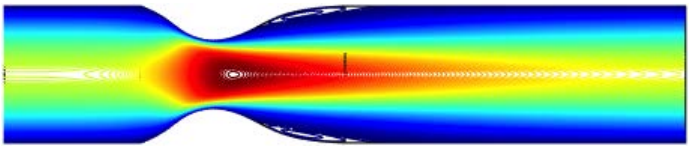


Figure 1: Non-Newtonian case: velocity contour highlighting a circulating region after the stenosed area

We chose to observe the data at the sections represented in Figure 2.

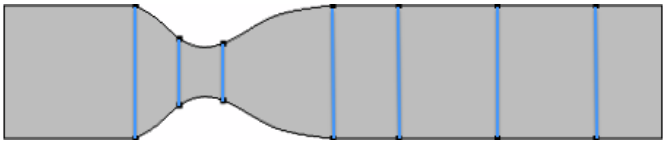


Figure 2: Ω_{part} representation

We test different values of w_1, w_2 and w_3. We show the results for such tests in Tables 1, 2 and 3. There $Y = y - y_d$ and $W = WSSm - WSSm_d$. The final cost function (CF) is also represented. We remark that for $w_3 = 0$ since no regularization term is present, the the corresponding cost J is minimized but the control is not the expected one. When $w_3 \neq 0$ we see that the first term in the cost J has a fundamental role in approximating the solution y from the data y_d. We see however that including the second term for the WSS improves the approximation.

 ISBN: 978-84-616-2723-3

Table 1: Cost functions components values fixing $w_2 = 1e^6$ and $w_3 = 1e^{-3}$ and varying w_1, non-Newtonian problem.

$\bar{w} = (w_1, w_2, w_3)$	$\int_{\Omega_{part}} \lvert Y\rvert^2$	$\int_{\Gamma_{wall}} \lvert W\rvert^2$	$\int_{\Gamma_{in}} \lvert\nabla \mathbf{u}\rvert^2$	CF
$(0, 1e^6, 1e^{-3})$	$8.556603e^{-8}$	$9.245164e^{-14}$	3.680136	0.00368022846896
$(1e^5, 1e^6, 1e^{-3})$	9.201587^{-12}	$1.601673e^{-16}$	3.767962	0.00376888268568
$(1e^6, 1e^6, 1e^{-3})$	$9.965953e^{-14}$	$4.340147e^{-18}$	3.769641	0.00376974041077
$(1e^7, 1e^6, 1e^{-3})$	$1.011144e^{-15}$	$6.587033e^{-20}$	3.769821	0.00376983073395
$(1e^{10}, 1e^6, 1e^{-3})$	$1.014796e^{-21}$	$3.985640e{-}23$	3.769841	0.00376984085014

Table 2: Cost functions components values fixing w_1 and w_3 and varying w_2, non-Newtonian problem.

$\bar{w} = (w_1, w_2, w_3)$	$\int_{\Omega_{part}} \lvert Y\rvert^2$	$\int_{\Gamma_{wall}} \lvert W\rvert^2$	$\int_{\Gamma_{in}} \lvert\nabla \mathbf{u}\rvert^2$	CF
$(1e^6, 0, 1e^{-3})$	$1.012988e^{-13}$	$1.855193e^{-13}$	3.769638	0.00376973951987
$(1e^6, 1e^5, 1e^{-3})$	9.971377^{-14}	$4.018386e^{-16}$	3.769641	0.00376974037362
$(1e^6, 1e^7, 1e^{-3})$	$9.964534e^{-14}$	$4.425922e^{-20}$	3.769641	0.00376974041479
$(1e^6, 1e^{10}, 1e^{-3})$	$9.965343e^{-14}$	$4.421446e^{-26}$	3.769641	0.00376974041492

Table 3: Cost functions components values fixing w_1 and w_2 and varying w_3, non-Newtonian problem.

$\bar{w} = (w_1, w_2, w_3)$	$\int_{\Omega_{part}} \lvert Y\rvert^2$	$\int_{\Gamma_{wall}} \lvert W\rvert^2$	$\int_{\Gamma_{in}} \lvert\nabla \mathbf{u}\rvert^2$	CF
$(1e^6, 1e^6, 0)$	$3.689777e^{-16}$	$6.351068e^{-22}$	75.994451	$3.689783e^{-10}$
$(1e^6, 1e^6, 1e^{-2})$	9.220232^{-12}	$1.523402e^{-14}$	3.767958	0.03768881276207
$(1e^6, 1e^6, 1e^{-1})$	$7.593194e^{-10}$	$1.818445e^{-11}$	3.753622	0.37613975336735
$(1e^6, 1e^6, 1e^1)$	$2.095783e^{-6}$	$1.793444e^{-6}$	2.923831	33.1275360625329

4 Conclusions and future work

We propose an optimal control problem for the non-Newtonian model for blood flow that solves the Data Assimilation problem on an idealized stenosis. We tested the different parameters in the cost function to verify their influence in the approximation. In the future we should deal with noisy data and apply this method to 3D models.

 ISBN: 978-84-616-2723-3

Acknowledgements

This work was partially supported by CMA/FCT/UNL, under the project PEst-OE/MAT/UI0297/2011, PTDC/MAT109973/2009, by CEMAT-IST through FCT Funding Program and by FCT grant SFRH-BDP-66639-2009 POPH-FSE.

References

[1] T. Bodnár, A. Sequeira, *On the shear-thinning and viscolelastic effects of blood flow under various flow rates*, Applied Mathematics and Computation **217** (2011) 5055–5067.

[2] V. Calvez, J. G. Houot, N. Meunier, A. Raoult and G. Rusnakova, *Mathematical and Numerical Modeling or early atherosclerotic lesions*, ESAIM: PROCEEDINGS **30** (2010) 1–14.

[3] M. D' Elia, M. Perego, A. Veneziani, *A Variational Data Assimilation Procedure for the Incompressible Navier-Stokes Equations in Hemodynam*, ics. Journal of Scientific Computing (2011)

[4] M. D' Elia, M. Perego, A. Veneziani, *A Data Assimilation technique for including noisy measurements of the velocity field into Navier-Stokes simulations*, Proceedings of V European Conference on Computational Fluid Dynamics, ECCOMAS (2010)

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Mathematical modeling of Timed-Arc Petri Nets as Dynamical Systems

Juan L. G. Guirao[a], Fernando L. Pelayo[b] and Jose C. Valverde[c]

[a] *Department of Applied Mathematics, Polytechnic University of Cartagena*

[b] *Department of Computing Systems, University of Castilla - La Mancha*

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

emails: Juan.Garcia@upct.es, FernandoL.Pelayo@uclm.es*, Jose.Valverde@uclm.es

Abstract

This paper presents an step towards applying the theory of discrete dynamical systems to the analysis of Concurrent Computing Systems. In order to do that, a restricted model of Timed - Arc Petri Net, TAPN, where the labels of the arcs connecting places to transitions are non-negative integer numbers, has been encoded as a discrete dynamical systems, so defining the corresponding phase space, which has been endowed with the evolution operator of the system.

Key words: Formal Computing Science, Timed-Arc Petri Net, Quasi-Pseudometric, Discrete Dynamical System.

1 Timed-Arc Petri Nets

Given the TAPN $N = (P, T, F, times)$ where:

- P is the finite *set of places*
- T is the finite *set of transitions*
- $P \cap T = \emptyset$. In the classical representation of PNs, places are circles and transitions are rectangles
- F is the *flow relation* which relates places and transitions by arcs connecting them.

- $F \subseteq (P \times T) \cup (T \times P)$
- $dom(F) \cup cod(F) = P \cup T$
- $times : F|_{P \times T} \longrightarrow \mathbb{N}$
- $M : P \longrightarrow \mathbb{N}$ is a Marking of N.

  Markings of TAPNs are graphically represented by including in places either non-negative integers so representing tokens with that age, or, nothing when there is no token in that place.

Example 1. *Fig. 1 shows the MTAPN which models the classical Producer/Consumer problem with a buffer of capacity 1, where transition t_1 represents "producing" an item so lasting 5 units of time, t_2 "putting" the item into the buffer immediately , t_3 "removing" an item from the buffer as soon as a token appears in places p_3 and p_6, and, t_4 "consuming" the item so elapsing at least 4 units of time.*

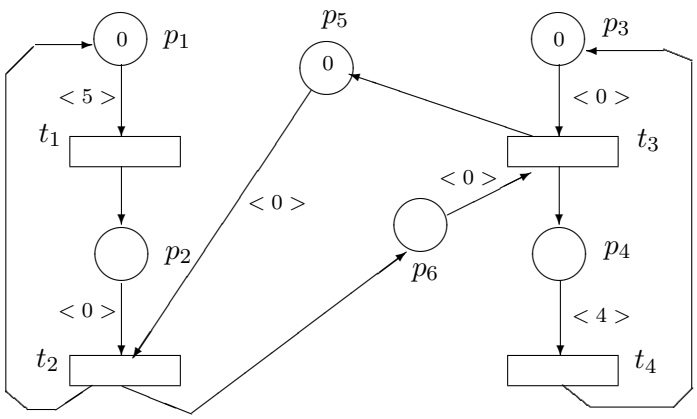


Figure 1: M_0 of the TAPN modeling the Producer/Consumer with a single capacity buffer

Once elapsed 4 units of time, the new Marking can be seen in fig. 2:
One unit of time later is shown in figs. 3, 4 and definitively 5
Four units of time more have elapsed in fig. 6

Given a MTAPN $(M, P, T, F, times)$ with $P = \{p_1, \ldots, p_n\}$, a Marking $\mathcal{M}$ of it which has m tokens in places $p_{i_1}, \ldots, p_{i_m}$ $(m \leq n)$ with ages $x_{i_1}, \ldots, x_{i_m}$, will be codified by a n-tuple containing in every i_j-position $j \in \{1 \ldots m\}$ the age of the token in place p_{i_j} and the remainder $n - m$ positions contain $\emptyset$.

 ISBN: 978-84-616-2723-3

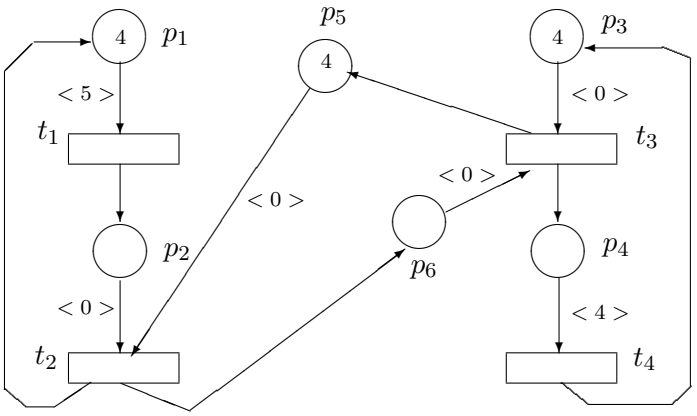


Figure 2: M_4 of the TAPN modeling the Producer/Consumer with a single capacity buffer

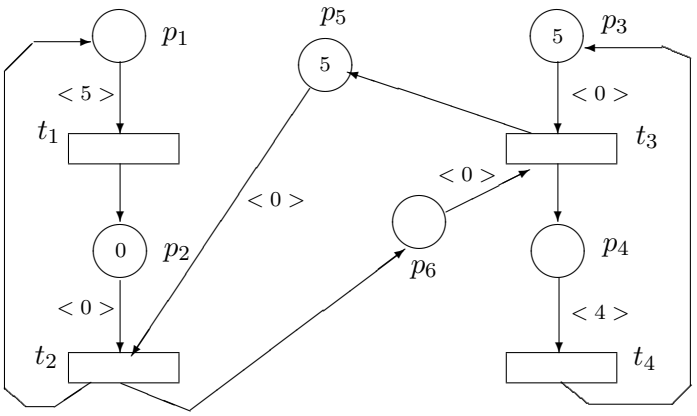


Figure 3: M_5 of the TAPN 1

 ISBN: 978-84-616-2723-3

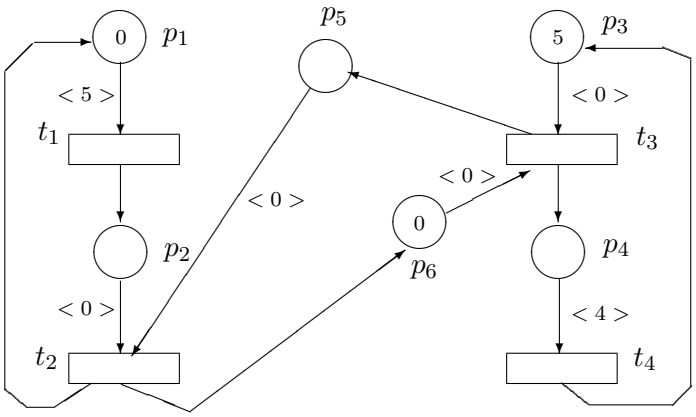


Figure 4: M_5 of the TAPN 2

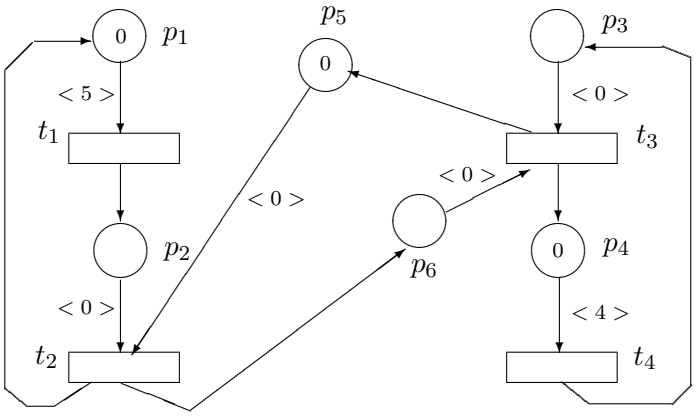


Figure 5: M_5 of the TAPN 3

 ISBN: 978-84-616-2723-3

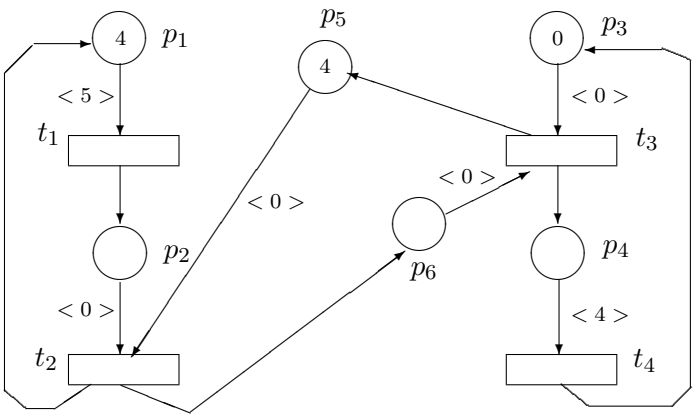


Figure 6: M_9 of the TAPN

The codification of the MTAPN modeling the initial state of the classical Producer/Consumer problem with a buffer of capacity 1, as shown in fig.1, appears enumerated as 1., so that, the codification once elapsed i units of time appears enumerated as $i+1$. as follows:

1. $(0,\emptyset,0,\emptyset,0,\emptyset)$ (see fig.1)
2. $(1,\emptyset,1,\emptyset,1,\emptyset)$
3. $(2,\emptyset,2,\emptyset,2,\emptyset)$
4. $(3,\emptyset,3,\emptyset,3,\emptyset)$
5. $(4,\emptyset,4,\emptyset,4,\emptyset)$ (see fig.2)
   (a) $(5,\emptyset,5,\emptyset,5,\emptyset)$ t_1 can be fired
   (b) $(\emptyset,0,5,\emptyset,5,\emptyset)$ t_2 can be fired (see fig.3)
   (c) $(0,\emptyset,5,\emptyset,\emptyset,0)$ t_3 can be fired (see fig.4)
6. $(0,\emptyset,\emptyset,0,0,\emptyset)$ (see fig.5)
7. $(1,\emptyset,\emptyset,1,1,\emptyset)$
8. $(2,\emptyset,\emptyset,2,2,\emptyset)$
9. $(3,\emptyset,\emptyset,3,3,\emptyset)$
   (a) $(4,\emptyset,\emptyset,4,4,\emptyset)$ t_4 can be fired
10. $(4,\emptyset,0,\emptyset,4,\emptyset)$ (see fig.6)

 ISBN: 978-84-616-2723-3

Given a MTAPN $(M, P, T, F, times)$, a transition $t \in T$ is fired as soon as $\forall p \in {}^\bullet t$, p has an enabled token for t.

A token in a place $p \in P$ ($p \in {}^\bullet t$), x_p, is enabled for that transition $t \in T$ when $x_p \geq times(p,t)$.

Let $N = (M, P, T, F, times)$ be a MTAPN and $t \in T$:

- t is fired at M iff $\forall p \in {}^\bullet t . x_p \geq times(p,t)$
- When t is fired at M and, consequently, marking M' is reached, we denote $M[t\rangle M'$
  - $M'(p) = M(p) - C^-(p,t) + C^+(t,p),\ \forall p \in P$
    - $C^-(p,t) = \begin{cases} x_p & \text{when } p \in {}^\bullet t, x_p \geq times(p,t) \wedge x_p \in M(p) \\ \emptyset & \text{otherwise} \end{cases}$
    - $C^+(t,p) = \begin{cases} \emptyset & \text{when } p \notin t^\bullet \\ 0 & \text{otherwise} \end{cases}$

Thus, from each precondition place of t, the corresponding enabled token is removed, and a new token (with age 0) appears on each postcondition place of t.

Let $N = (M, P, T, F, times)$ be a MTAPN and $R \subseteq T$ be a subset of transitions. It is said that all transitions in R can be fired at marking M iff

$$\forall t \in R . \forall p \in {}^\bullet t . M(p) \geq times(p,t)$$

We would like to note that since a place can belong to the precondition set of more than one different transition, a token in it could potentially be enabled for more than one transition and, after firing one of them (transitions), more than one different marking can be reached.

This fact has lead us to consider as Phase Space not the set of n-tuples of either natural numbers or $\emptyset$'s but the set of all its subsets, in order to properly capture these cases.

Moreover, as once a transition is fired there will appear tokens with age 0 in all its postcondition places, that, potentially could be enabled for other (sometimes different) transitions; we assume that when modeling the elapsing of a single unit of time, the Net fires as much transitions (eventually sequentially ordered multisets of transitions) as possible until no transitions can be fired, see figs. 3, 4 and definitively 5

The firing of a sequentially ordered multiset of transitions RR at the marking M until no more transitions can be fired, as explained above, generates a new marking M', so denoted by $M[RR\rangle M'$. This represents the complete evolution of the Net without time elapsing. Therefore, we call the evolution of the Net in a single step, the sequence of, first aging all its tokens (increasing in 1 their ages) and then, completely evolve the marking reached after such 1-aging. It can be better understood in the codification of the example of the Producer/Consumer.

 ISBN: 978-84-616-2723-3

2 A restricted version of Timed-Arc PNs as Discrete Dynamical Systems

We have considered a slight variation of the well known Timed-Arc Petri Nets formalism, that modifies the temporal behaviour of the time required for firing, which meanwhile in the former it was uniformly randomly distributed in the time interval labeling each arc (p,t), for the sake of fulfilling a compulsory restriction of Dynamical Systems, we have considered just non-negative integers so labeling these arcs connecting places and transitions, thus resulting the model with the dynamical behaviour previously explained within this paper.

The discrete dynamical system which encodes the MTAPN $N=(M,P,T,F,times)$ is the triple (X,τ,Φ), where:

- $X=\mathcal{P}(\{\mathbb{N},\emptyset\}^n)$ is the set of all subsets of $\{\mathbb{N},\emptyset\}^n$, being n the number of places of the MTAPN. Note that $\mathbb{N}$ could be changed by Z, and, $\emptyset$ by a special value, as e. g. -1, without loss of generality.
- τ is the monoid $\mathbb{N}\cup\{0\}$
- $\Phi:\tau\times X\to X$ is the evolution operator Φ verifying:
  1. $\Phi(0,A)=A \quad \forall A\in X$, i.e., $\Phi_0=id_X$
  2. $\Phi(1,A)=B \quad A,B\in X$ where:
     - $A=\{z_1,\ldots,z_k\}$ where $z_i\in\{\mathbb{N},\emptyset\}^n$ encodes Markings of the MTAPN N
     - $\forall t\in\{1\ldots k\}\ .\ x_t=1+z_t$ capturing one unit of time elapsed, in this way: $\forall i\in\{1\ldots n\}\ .\ x_{t_i}=\begin{cases}\emptyset & \text{when } z_{t_i}=\emptyset\\ z_{t_i}+1 & \text{otherwise}\end{cases}$
     - $B=\cup_{i=1}^{k}B_i$
     - $B_i=\cup_{j=1}^{t}\{y_i^j\}$, i.e. the union of all (t) possible reachable markings from x_i, defined by $x_i[RR_i\rangle y_i^j$ being RR_i the maximal ordered multiset of transitions of the net that can be fired from the marking x_i
  3. $\Phi(t,\Phi(s,A))=\Phi(t+s,A) \quad \forall t,s\in\tau,\ \forall A\in X$

3 Conclusions and future work

We have presented a codification of a "urgent" version of Timed-Arc Petri Nets as a Discrete Dynamical System.

Our next step is to provide this Phase Space with a metric such that would fit well with the behavioral meaning of the this type of Timed PNs, and, that would give this set a sound topological structure to analyze the key elements of the underlying Discrete Dynamical System.

 ISBN: 978-84-616-2723-3

Acknowledgements

This work has been supported by project CGL2010-20787-C02-02

References

[1] D. K. Arrowsmith and C. M. Place, *An Introduction to Dynamical Systems*, Cambridge University Press (1990).

[2] J.W. Baker and E.P. Vink, *A metric approach to control flow semantics*, Annals of the New York Academy of Sciences 806, 11–27 (1996).

[3] J.W. Baker and E.P. Vink, *Denotational models for programming languages: Applications of Banach's fixed point theorem*, Topology Appl. 85, 35–52, (1998).

[4] J.L. Guirao, F.L. Pelayo and J.C. Valverde, *Modeling dynamics of Concurrent Computing Systems*, Comput. Math. Appl. 61:5, 1402-1406 (2011)

[5] J. Hale and H. Koak, *Dynamics and Bifurcations*, Springer, New York, Heidelberg, Berlin, (1991).

[6] P. Holmes and D. Whitey, *Bifurcations of one-and two-dimensional maps*, Phil. Trans. R. Soc. Lond. 311, 43-102 (1984).

[7] G. A. Petri, *Communications with Automata*, Technical Report RADC-TR-65-377, New York University (1966).

[8] J. Rodriguez-Lopez, S. Romaguera and O. Valero, *Denotational semantics for programming languages, balanced quasi-metrics and fixed points*, International Journal of Computer Mathematics, 85:3, 623 - 630 (2008).

[9] S. Romaguera and M. Sanchis, *Applications of utility functions defined on quasi-metric spaces*, J. Math. Anal. Appl., 283, 219–235 (2003).

[10] P. Stefan, *A theorem of Sarkovskii on the existence of periodic orbits of continuous endomorphisms of the real line*, Commun. Math. Phys. 54, 237-248 (1977).

[11] S. Wigginns, *Introduction to Applied Nonlinear Systems and Chaos*, Springer, New York (1990).

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Real dynamics for damped Newton's method applied to cubic polynomials

J. M. Gutiérrez[1] **and A. A. Magreñán**[1]

[1] *Department of Mathematics and Computation, University of La Rioja, Spain*

emails: jmguti@unirioja.es, alberto.magrenan@gmail.com

Abstract

We study the dynamics of damped Newton's methods applied to some degree three polynomials. We consider a damping factor in the real line. We use in our study two different techniques that allow us to demonstrate our results: the Lyapunov exponents and the Feigenbaum diagrams. The conclusions drawn are that the inclusion of the damping factor modifies significantly the dynamical behavior of the method.

Key words: nonlinear equation, damped Newton's method, real dynamics, bifurcation
MSC 2000: 65P20, 65P30, 65S05

1 Introduction

In many areas related to the Applied Sciences one confronts the problem of solving a nonlinear equation of the form $f(x) = 0$. The solutions of these equations can rarely be found in closed form. That is why most solution methods are iterative. There exist lots of iterative methods with different properties that allow us to solve this kind of equations, but the most well-known and used is the Newton's method, which has the following form:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geq 0. \tag{1}$$

The study of the dynamics of the iterative methods attracts the attention of many groups [1, 2, 3, 4, 5]. Our main goal in this paper is to study the real dynamics for a variant of Newton's method that it is called *relaxed Newton's method* (see [6]):

$$x_{n+1} = x_n - \lambda f'(x_n)^{-1} f(x_n), \quad n \geq 0, \quad \lambda \in \mathbb{R}. \tag{2}$$

 ISBN: 978-84-616-2723-3

It is well known that under adequate conjugations the dynamics of every cubic polynomial is equivalent to the dynamics of

$$q(x) = x^3 + \alpha x + \beta. \tag{3}$$

In this work, we are interested in polynomials with $\beta = 0$.

Let $N_{\lambda,q}$ be the iteration function of damped Newton's method applied to a cubic polynomial of the form (3), with $\beta = 0$, that is

$$N_{\lambda,q}(x) = \frac{x\left((3-\lambda)x^2 + \alpha(1-\lambda)\right)}{3x^2+\alpha}.$$

Then we can divide the study in the following cases:

1. If $\alpha = 0$, then
$$N_{\lambda,q}(x) = N_{\lambda,0}(x) = \frac{1}{3}x(3-\lambda),$$
which is the damped Newton's method applied to the polynomial $p_0(x) = x^3$.

2. If $\alpha > 0$, then
$$N_{\lambda,q}(x) = N_{\lambda,+}(x) = \frac{x\left((\lambda-3)x^2 + 1 - \lambda\right)}{3x^2+1},$$
which is the damped Newton's method applied to the polynomial $p_+(x) = x^3 + x$.

3. If $\alpha < 0$, then
$$N_{\lambda,q}(x) = N_{\lambda,-}(x) = \frac{x\left((3-\lambda)x^2 + \lambda - 1\right)}{3x^2-1},$$
which is the damped Newton's method applied to the polynomial $p_-(x) = x^3 - x$.

The paper is organized as follows. In Section 2 we present the dynamical study of the damped Newton's method applied to $p_0(x) = x^3$, in section 3 the dynamical study for $p_+(x) = x^3 + x$ and in section 4 for polynomial $p_-(x) = x^3 - x$.

2 Damped Newton's method applied to $p_0(x) = x^3$

The polynomial p_0 only has one triple root at $x = 0$. We have that (2) has the following form:

$$N_{\lambda,0}(x) = -\frac{1}{3}x(-3+\lambda)$$

which is a linear contraction for values of $\lambda \in (0, 6)$, so every point of the real line will converge to the fixed point $x = 0$. For $\lambda = 6$, method (2) is minus the identity so every point will constitute a 2-cycle. Finally for values of $\lambda \notin (0, 6]$, the iteration of every point except the fixed point will diverge to infinity.

 ISBN: 978-84-616-2723-3

3 Damped Newton's method applied to $p_+(x) = x^3 + x$

In this case, p_+ has only one real root at $x = 0$, and the method (2) has the following form:

$$N_{\lambda,p_+}(x) = \frac{x\left((3-\lambda)x^2 + 1 - \lambda\right)}{1 + 3x^2}.$$

Since $p'_+(x) = 1 + 3x^2 > 0$ for all x, it follows that p_+ has no critical points, hence $N_{\lambda,p_+}(x)$ has no vertical asymptotes. Moreover, we obtain that the multiplier asociated to the fixed point $x = 0$ is $\mu = |1 - \lambda|$ and so for values of $\lambda \in (0, 2)$ the fixed point is atractor (super-atractor if $\lambda = 1$). For values of $\lambda \in (0, 3)$ there exist 4 critical points and for $\lambda \in (3, 4]$ there exist only two.

The dynamical behavior of the damped Newton method applied to polynomial $p_+(x) = x^3 + x$, is the following:

- If $\lambda < 0$, then the iterations of every point different of $x = 0$ diverge to infinity.
- If $\lambda \in (0, 2]$, 0 is an atractor fixed point and the iteration of every point converge to it.
- If $\lambda \in (2, 6)$, 0 is a repulsor fixed point,and N_{λ,p_+} has an atractor 2-cycle which it corresponds to the solutions of $-2 + \lambda - 6x^2 + \lambda x^2 = 0$.
- If $\lambda \geq 6$, then the iterations of every point different of $x = 0$ diverge to infinity.

4 Damped Newton's method applied to $p_-(x) = x^3 - x$

In this case, p_- has three different real roots at $x = 0$, $x = 1$ and $x = -1$ and the method (2) has the following form:

$$N_{\lambda,p_-}(x) = \frac{x\left((3-\lambda)x^2 + \lambda - 1\right)}{-1 + 3x^2}.$$

Notice that $N_{\lambda,p_-}(x)$ has 2 vertical asymptotes at $x = \pm\frac{1}{\sqrt{3}}$. Moreover, the multiplier asociated to the every fixed point is $\mu = |1 - \lambda|$ and so for values fixed points are atractor for values of $\lambda \in (0, 2)$(super-atractor if $\lambda = 1$). For values of $\lambda \in (0, 1]$ there exist 4 critical points and for $\lambda \in (1, 3)$ there exist only two.

The dynamical behavior of the damped Newton method applied to polynomial $p_-(x) = x^3 + x$, is the following:

- If $\lambda < 0$, then the iterations of every points distinct to ehe fixed points diverge to infinity.

 ISBN: 978-84-616-2723-3

- If $\lambda \in (0, 2]$, 0,1 y -1 are atractor, and the iteration of every point converge to one of them.
- If $\lambda \in (2, 2.37064)$, the fixed points are repulsor. Moreover, N_{λ,p_-} has atractor 2-cycles which are the roots of $(\lambda-3)^2x^4-(\lambda^2-4\lambda+6)x^2+4=0$.
- If $\lambda \in (2.37064, 2.4423)$, the fixed points are repulsor. Moreover, N_{λ,p_-} has atractor cycles.
- If $\lambda \in (2.4423, 2.72221)$ the fixed points are repulsor. The dynamics are chaotic.
- If $\lambda \in (2.72221, 2.753)$, the fixed points are repulsor. Moreover, N_{λ,p_-} has an atractor 4-cycle.
- If $\lambda \in (2.753, 6)$, the fixed points are repulsor. The dynamics are chaotic.
- If $\lambda \geq 6$, then the iterations of every points distinct to ehe fixed points diverge to infinity.

Acknowledgements

This work has been partially supported by Spanish Ministry of Science and Innovation (project MTM2011-28636-C02-01).

References

[1] S. Amat, S. Busquier y Á. A. Magreñán, Reducing chaos and bifurcations in Newton-type methods, *Abstract and Applied Analysis*, to appear.

[2] S. Amat, S. Busquier and S. Plaza, Chaotic dynamics of a third-order Newton-type method, *J. Math. Anal. Appl.*, (2010), 366, 24–32.

[3] A. Cordero, J. R. Torregrosa and P. Vindel, Dynamics of a family of Chebyshev-Halley-type methods arXiv:1207.3685.

[4] J.M. Gutiérrez, M.A. Hernández and N. Romero, Dynamics of a new family of iterative processes for quadratic polynomials, *J. Comput. Appl. Math.*, (2010), 233, 2688-2695.

[5] G. Honorato, S. Plaza and N. Romero, Dynamics of a high-order family of iterative methods, *J. Complexity*, (2011), 27, 221-229.

[6] S. Plaza and N. Romero, Attracting cycles for the relaxed Newton's method, *J. Comput. Appl. Math.*, (2011), 235(10), 3238-3244.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

High Performance Option Pricing based on Spatially Adaptive Sparse Grids

Alexander Heinecke[1]

[1] *Department of Informatics, Technische Universität München*

emails: heinecke@in.tum.de

Abstract

In this talk, we present a parallel approach for numerically solving the Black-Scholes equation in order to price European and American basket options. Therefore, hardware-features of contemporary HPC computer architectures such as NUMA and hardware-threading are exploited by a hybrid parallelization using MPI and OpenMP. Our approach is based on a sparse grid discretization with finite elements and makes use of a sophisticated adaption. The resulting linear system is solved by a conjugate gradient method that uses a parallel operator for applying the system matrix implicitly. Several numerical examples as well as an analysis of the performance for different computer architectures are provided.

Key words: Computional Finance, Option Pricing, High Performance Computing, Sparse Grids

Extended Abstract

Typically, there is no closed form solution available when pricing basket options based on the model of Black and Scholes. Hence, for determining the price of a basket, one has to resort to numerical methods. Up to now solving these multi-dimensional option pricing problems, Monte Carlo (MC) methods are widespread as the favorite method. They are flexible, can be implemented in a straightforward way and handle multiple dimensions, but they exhibit rather slow convergence rates. Several approaches based on variance reduction techniques exist, improving the speed of convergence. Consider, for example, quasi MC methods, control variate techniques, adaptive MC, or multi-level MC. But in recent years, also PDE methods have been examined for financial problems reasoning the fast speed of convergence compared to MC techniques which theoretically allows to obtain higher

accuracies. Additionally, PDEs allow the fast computation of Greeks, that are needed for hedging tasks, as an additional benefit. Unfortunately, up to now, PDE methods usually can not compete against MC methods in three or more dimensions in terms of computing time: the computations are rather costly and the PDE approach suffers from the so-called *curse of dimensionality*, the exponential dependency on the dimensionality. Hence, common PDE approaches as the ones stated above are currently typically limited to two or, in the best case, three dimensions.

A possible solution to overcome this limitation are the so-called *sparse grids*. In [1, 2, 3] we presented a different approach: Here, the (multi-asset) Black-Scholes PDE, see Eq. (1), is discretized by finite elements on spatially adaptive sparse grids in order to optimally adopt to a given option that should be priced.

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} + \sum_{i=1}^{d} \mu_i S_i \frac{\partial V}{\partial S_i} - rV = 0. \tag{1}$$

We achieved sufficiently high accuracies for European and American options. Nevertheless, one has to mention that calculation times (apart from the accuracy) are one of the biggest issues when dealing with higher dimensional PDEs on sparse grids. In this paper we address this issue by applying several optimization techniques (such as the principal axis transformation on log-transformed equation of Eq. (1), refer to [4] details) and a proper parallelization which unleashes the power of modern compute clusters.

The transformed Black-Scholes PDE is given in Eq. (2a)

$$\frac{\partial u}{\partial \tau} - \frac{1}{2}\sum_{i=1}^{d} \lambda_i \frac{\partial^2 u}{\partial z_i^2} = 0 \tag{2a}$$

with the initial condition

$$u(\mathbf{z}, 0) = \max\left\{ K - \frac{1}{d}\sum_{i=1}^{d} \exp\left(\sum_{j=1}^{d} q_{ij} z_j\right),\ 0 \right\} \tag{2b}$$

The solution of the original Black-Scholes PDE (1) is then obtained via the inverse transformation

$$V(\mathbf{S}, t) = e^{-r\tau} \cdot u(\mathbf{z}, \tau) \tag{3}$$

with

$$z_i = \tau \cdot \hat{\mu}_i + \sum_{j=1}^{d} q_{ji} \log(S_j), \qquad \hat{\mu}_i = \sum_{j=1}^{d} \left(\mu_j - \frac{1}{2}\sigma_j^2\right) q_{ji} \tag{4}$$

and backward time $\tau = T - t$.

Since spatially adaptive sparse grids require non-standard algorithms that directly work on the grid's data, the parallelization of this algorithms is the main focus of this article:

 ISBN: 978-84-616-2723-3

we present solutions for systems with a shared memory (like normal desktop systems or workstations) and implementations that support compute clusters with a high-speed interconnect.

The talk will describe steps that are needed to implement a Black-Scholes PDE solver on spatially adaptive sparse grids. Starting with mathematical foundations such as choosing sufficient boundary conditions, determining and appropriate domain size, we will continue with deriving Eq. 2a, which is basically the well-known heat-equation that requires a fast Laplace-operator. Afterwards we deep-dive into the solver's structure and develop a parallelization scheme that exploits all levels of parallelism in the specific sparse grid algorithms and map them on current hardware platforms by respecting their technically opportunities and as well their limitations.

References

[1] Hans-Joachim Bungartz, Alexander Heinecke, Dirk Pflüger, and Stefanie Schraufstetter. Option pricing with a direct adaptive sparse grid approach. *Journal of Computational and Applied Mathematics*, 236(15):3741 —- 3750, October 2011. online Okt. 2011.

[2] Hans-Joachim Bungartz, Alexander Heinecke, Dirk Pflüger, and Stefanie Schraufstetter. Parallelizing a black-scholes solver based on finite elements and sparse grids. *Concurrency and Computation: Practice and Experience*, March 2012.

[3] Alexander Heinecke, Stefanie Schraufstetter, and Hans-Joachim Bungartz. A highly-parallel black-scholes solver based on adaptive sparse grids. *International Journal of Computer Mathematics*, 89(9):1212–1238, June 2012.

[4] Christoph Reisinger. *Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2004.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

New families of iterative methods with fourth and sixth order of convergence and their dynamics

José L. Hueso[1], Eulalia Martínez[2] and Carles Teruel[3]

[1] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politécnica de Valéncia*

[2] *Instituto Universitario de Matemática Pura y Aplicada, Universitat Politécnica de Valéncia*

[3] *Escuela Técnica Superior de Ingeniería de Telecomunicación, Universitat Politécnica de Valéncia*

emails: jlhueso@mat.upv.es, eumarti@mat.upv.es, cartefer@teleco.upv.es

Abstract

In this work, we present a new family of iterative methods for solving nonlinear systems that are optimal in the sense of Kung and Traub's conjecture for the unidimensional case. We generalize this family performing a new step in the iterative method getting a new family with order of convergence six. We study the efficiency of these families for the multidimensional case introducing a new term in the computational cost defined by Grau-Sánchez et al. A comparison with already known methods is done studying the dynamics of these methods in an example system.

Key words: nonlinear systems, iterative methods, convergence order, optimal methods, computational cost, efficiency, dynamics.

1 Introduction

Finding iterative methods with high order of convergence in order to approximate the solution of a nonlinear system $F(x) = 0$ is an active field in the numerical analysis. The range of applications where it is required to use a high level of numeric precision is increasing nowadays. Focusing on higher order iterative methods for the multidimensional case, we can mention some recently published works: [1, 2, 3, 4].

In this work we generalize the technique used in [5], obtaining a new family of iterative methods with fourth order of convergence. The procedures used in [9] and [6] for increasing the convergence order of an iterative method, that is, to perform another Newton's step avoiding the evaluation of the jacobian matrix in the new step in order to get the maximum efficiency, do not work for the optimal method introduced in [5], so we propose a new procedure to increase the order with a reasonable efficiency.

Obviously, performing a new step in an iterative method carries more function evaluations and so one has to check if the higher convergence order justifies the increase of the computational cost. For nonlinear system, a thorough study of cost and efficiency can be found in [8]. Nevertheless, we introduce a new term in this cost definition for taking into account matrix-vector operations that occur in some iterative methods such as the considered here.

Finally, we study the dynamics of these methods for a particular nonlinear system.

2 New families of iterative methods

Our aim is to develop high order methods for nonlinear systems, in line with the recently published method of order 4 by Sharma et al. [5]. First of all, we generalize this technique introducing a new term in their proposal, obtaining a new family of iterative methods of order 4.

That is, we consider the family of iterative methods given by:

$$\begin{aligned} y_n &= x_n - \theta\, \Gamma_{x_n} F(x_n) \\ H(x_n, y_n) &= \Gamma_{x_n} F'(y_n) \\ G_s(x_n, y_n) &= s_1 I + s_2 H(y_n, x_n) + s_3 H(x_n, y_n) + s_4 H(y_n, x_n)^2 \\ z_n &= x_n - G_s(x_n, y_n) \Gamma_{x_n} F(x_n) \end{aligned} \tag{1}$$

where $\Gamma_{x_n} = F'(x_n)^{-1}$, and θ, s_1, s_2, s_3, s_4 are constants that we determine in order to get a new family of 4th-order optimal methods. Notice that, in the unidimensional case, we evaluate just three functions, $F(x_n)$, $F'(x_n)$ and $F'(y_n)$ so the family will be optimal in the sense of Kung and Traub's conjecture, [7].

By using Taylor's developments adequately we can prove the following result that gives us the convergence order.

Theorem 1 *Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be a sufficiently Fréchet differentiable function in a neighborhood D, convex set containing α, that is a solution of the system $F(x) = 0$, whose jacobian matrix is continuous and nonsingular in D. Then, for an initial approximation sufficiently close to α, the family of methods defined by (1) has local order of convergence 4 for the following relations among the parameters: $s_1 = \frac{5-8s_2}{8}$, $s_3 = \frac{s_2}{3}$, $s_4 = \frac{9-8s_2}{24}$; $\forall s_2 \in \mathbb{R}$ and for $\theta = \frac{2}{3}$.*

 ISBN: 978-84-616-2723-3

The error equation obtained is as follows:

$$e_{n+1} = \frac{(64s_2 + 117)c_2^3 - 81c_1c_3c_2 + 9c_1^2c_4}{81c_1^3}\, e_n^4 + O\left(e_n^5\right)$$

where $c_k = \frac{F^{(k)}(\alpha)}{k!}, k \geq 1$

Now we are interested in improving the convergence order of this family of methods, so we propose performing a new step in this terms:

$$x_{n+1} = z_n - G_t(x_n, y_n)\Gamma_{y_n} F\left(z_n\right) \tag{2}$$

where for each value of s_2, we find relations among the constants t_1, t_2, t_3, t_4 providing a family of 6th-order methods according to the following:

Theorem 2 *Considering the same conditions as in Theorem 1, the biparametric family of 3-step methods (2) has local order of convergence 6 for this relation among the constants:* $t_2 = -\frac{3+8t_1}{8}$, $t_3 = \frac{15-8t_1}{24}$, $t_4 = \frac{9+4t_1}{12}$ *;* $\forall (s_2, t_1) \in \mathbb{R}^2$. *The vectorial error difference equation can be written as:*

$$e_{n+1} = \frac{c_3\left(-(64s_2 + 117)c_2^3 + 81c_1c_3c_2 - 9c_1^2c_4\right)}{81c_1^4}\, e_n^6 + O\left(e_n^7\right)$$

where $c_k = \frac{F^{(k)}(\alpha)}{k!}, k \geq 1$

3 Computational efficiency

In order to compare the different methods we have to study their efficiency. We use the efficiency index introduced in [4], given by $E = \rho^{1/C}$, where ρ is the order of convergence and C is the computational cost per iteration. For a system of n nonlinear equations in n unknowns, C is obtained by:

$$C\left(\mu_0, \mu_1, n\right) = \mu_0 a_0 n + \mu_1 a_1 n^2 + P\left(n\right)$$

where a_0 and a_1 represent the number of evaluations of $F\left(x\right)$ and $F'\left(x\right)$ respectively, $P\left(n\right)$ is the number of products per iteration and μ_0 and μ_1 are the ratios between products and evaluations required to express the value of $C\left(\mu_0, \mu_1, n\right)$ in terms of products.

Apparently, the best methods of the family defined by (1) from the point of view of computational efficiency are obtained for $a_2 = \frac{9}{8}$ and $a_2 = 0$. The first one is the method proposed in [5], that we denote by $\mathrm{M1}_4$. The second one is a new method, denoted by $\mathrm{M2}_4$.

 ISBN: 978-84-616-2723-3

We point out that for each particular method of the fourth order family, performing the new step given in (2), we obtain a different family of sixth order methods. For the comparisons, starting from $M1_4$ nd $M2_4$ we choose for the new step value $t_1 = -\frac{9}{4}$ in both cases, and so, we obtain two new methods denoted by $M1_6$ and $M2_6$ respectively. We summarize in Table 1 the four methods considered in the numerical experiments.

Method	s_0	s_1	s_3	s_4	t_1	t_2	t_3	t_4
$M1_4$	$-1/2$	$9/8$	$3/8$	0	-	-	-	-
$M2_4$	$5/8$	0	0	$3/8$	-	-	-	-
$M1_6$	$-1/2$	$9/8$	$3/8$	0	$-9/4$	$15/8$	$11/8$	0
$M2_6$	$5/8$	0	0	$3/8$	$-9/4$	$15/8$	$11/8$	0

Table 1: Values of coefficient for differents methods

We will express the computational cost per iteration with the same notation as in [4], where p_0 denotes the number of scalar products per iteration, p_1 the number of complete resolutions of the linear system (LU decomposition and resolution of two triangular systems) and p_2 the number of resolutions of two linear systems when LU decomposition is computed in another step in the same iteration.

Nevertheless, we need to introduce a new factor p_3 that is the number matrix by vector products per iteration. This adds a new term in the expression of the total number of products:

$$p(n) = n/6(2p_1n^2 + (3p_1(k+1) + 6p_2))n + 6p_0 + p_1(3k-5) + 6p_2(k-1) + 6p_3n)$$

where it is supposed that a quotient is equivalent to k products.

Table 2 compares the computational cost of the analyzed methods.

Method	a_0	a_1	p_0	p_1	p_2	p_3	$C(\mu_0, \mu_1, n)$
M_2	1	1	0	1	0	0	$1/6n(-5 + 6\mu_0 + 3n + 6\mu_1 n + 2n^2 + 3k(1+n))$
$M1_4$	1	2	4	2	1	1	$1/3n(4 + 3\mu_0 + 9n + 6\mu_1 n + 2n^2 + 3k(2+n))$
$M2_4$	1	2	3	2	1	1	$1/3n(1 + 3\mu_0 + 9n + 6\mu_1 n + 2n^2 + 3k(2+n))$
$M1_6$	2	2	7	2	3	2	$1/3n(7 + 6\mu_0 + 18n + 6\mu_1 n + 2n^2 + 3k(4+n))$
$M2_6$	2	2	6	2	4	2	$1/3n(1 + 6\mu_0 + 21n + 6\mu_1 n + 2n^2 + 3k(5+n))$

Table 2: Computational cost for the different methods

 ISBN: 978-84-616-2723-3

	M_2	$M1_4$	$M2_4$	$M1_6$	$M2_6$
$n = 2$	1.05846	1.03818	1.0404	1.03116	1.03011
$n = 4$	1.01292	1.00933	1.00959	1.00833	1.00789
$n = 6$	1.00491	1.00377	1.00383	1.00356	1.00336
$n = 9$	1.00177	1.00143	1.00145	1.00143	1.00135
$n = 12$	1.00083	1.0007	1.0007	1.00072	1.00069
$n = 15$	1.00045	1.00039	1.00039	1.00042	1.0004
$n = 18$	1.00028	1.00024	1.00024	1.00026	1.00025

Table 3: Efficiency indexes for different values of n for $\mu_0 = 1.7$ and $\mu_1 = 0.7$

	M_2	$M1_4$	$M2_4$	$M1_6$	$M2_6$
$n = 2$	1.02123	1.02377	1.02462	1.01808	1.01772
$n = 4$	1.0071	1.00703	1.00717	1.00591	1.00569
$n = 6$	1.00329	1.00309	1.00313	1.00279	1.00266
$n = 8$	1.00179	1.00164	1.00166	1.00156	1.00148
$n = 12$	1.00069	1.00063	1.00064	1.00064	1.00061
$n = 16$	1.00034	1.00031	1.00031	1.00033	1.00031
$n = 20$	1.00019	1.00017	1.00017	1.00019	1.00018

Table 4: Efficiency indexes for different values of n for a $\mu_0 = 11.5$ and $\mu_1 = 1$

4 Dynamics of the methods

In this section we study the dynamics of the iterative methods $M1_4$, $M2_4$, $M1_6$ and $M2_6$ when applied to the solution of a system of quadratic equations, representing the intersection of two hyperbolas in $\mathbb{R}^2$ and compare them with the dynamics of Newton method. We show that the methods are generally convergent and depict their attraction basins. The chosen example present four simple real roots. When there are less roots or multiple roots, the convergence order is lower, as expected, and even the convergence fails in certain regions of the plane.

Let us first recall some dynamical concepts. Consider a Frechet differentiable function $G : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. For $x \in \mathbb{R}^n$, we define the orbit of x as the set $x, G(x), G^2(x), \ldots, G^p(x), \ldots$. A point x_f is a fixed point of G if $G(x_f) = x_f$. A periodic point x_r of period m is such that $G^m(x_r) = x_r$ where m is the smallest such integer. A fixed point x_f is called attracting if $\|J_G(x_f)\| < 1$, repelling if $\|J_G(x_f)\| > 1$, and neutral if $\|J_G(x_f)\| = 1$. If $J_G(x_f) = 0$, the

 ISBN: 978-84-616-2723-3

point x_f is superattracting. Let x_{af} be an attracting fixed point of the function G. The basin of attraction of x_{af} is the set of points whose orbits tend to this fixed point

$$\mathcal{A}(x_{af}) = \{x \in \mathbb{R}^n : G^p(x) \longrightarrow x_{af} \text{ for } p \longrightarrow \infty\}$$

The dynamics of Newton's method and higher order iterative methods has been widely studied ([10, 11, 12, 13]). In these references the method is applied to simple polynomial equations in the complex domain. Our purpose here is to show the aspect of the basins of attraction of the above mentioned methods applied to a system of nonlinear equations in $\mathbb{R}^2$. The pictures for the complex case are nicer, but we are mainly interested in the behavior of the methods for solving systems of nonlinear equations in the real n-dimensional space.

For the comparisons, we consider the following quadratic system representing the intersection of two hyperbolas,

$$\left.\begin{aligned}(x-3)^2 - 16y^2 &= 1\\ x^2 - y^2 &= 1\end{aligned}\right\}.$$

In this system the axes of one hyperbola are parallel to the asymptotes of the other. One intersection is near the barycenter of the other three.

For the comparisons, we have run the methods iterating with tolerance 10^{-6} performing a maximum of 100 iterations. The starting points form a uniform grid of 512×512 in a rectangle of the real plane. The attraction basins have been colored according to the corresponding fixed point.

Figures 1 and 2 show, respectively, the attraction basins and the number of iterations for Newton's method. Figures 3, 4, 5 and 6 show the attraction basins of methods $M1_4$, $M2_4$, $M1_6$, and $M2_6$, respectively.

Observe that the complexity of the basins increases with the order, but the convergence regions cover almost all the plane. Methods $M2_4$ and $M2_6$ have slightly more complex basins than their counterparts $M1_4$ and $M1_6$. The four roots are superattractive for all the analyzed methods. In a further study we will consider the existence of periodic orbits and the convergence in case of double or missing roots.

5 Conclusions

As it can be observed in Tables 3 and 4, Newtons method, M_2, maintains higher efficiency index than the other methods. Our method $M2_4$ always gets better indexes than $M1_4$ due to the fewer number of operations. However, $M2_6$ does not reach the efficiency of $M1_6$. Although the 4th-order methods are good for systems with a reduced number of equations, as more complex a systems is, more advantages we get using the methods of order six. So, the 4th-order methods are as good as the 6th-order ones for systems between nine to

 ISBN: 978-84-616-2723-3

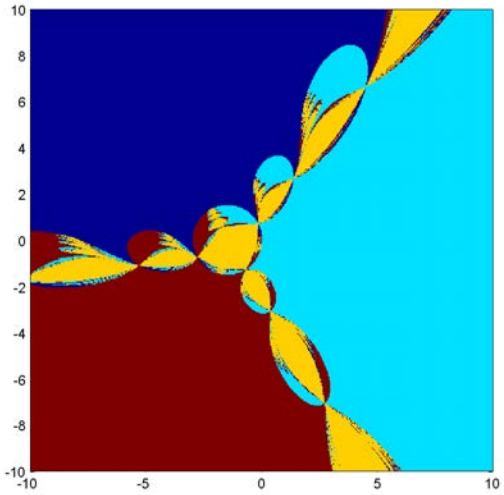


Figure 1: Attraction basins for Newton's method

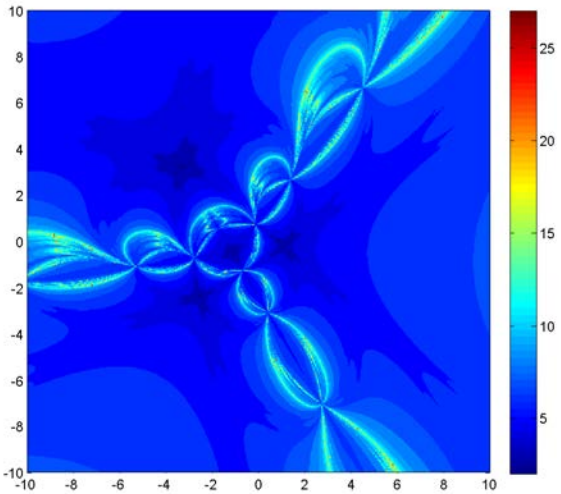


Figure 2: Iteration count for Newton's method

 ISBN: 978-84-616-2723-3

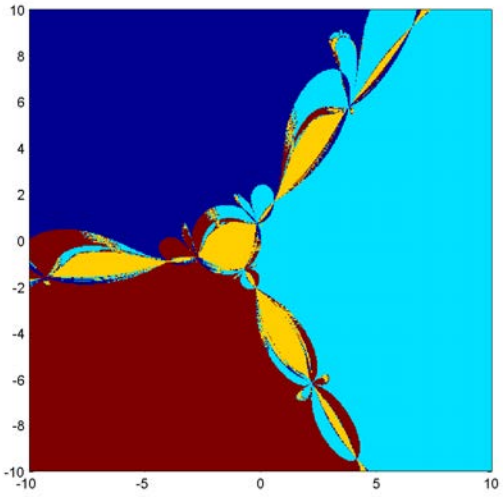


Figure 3: Attraction basins for method $M1_4$

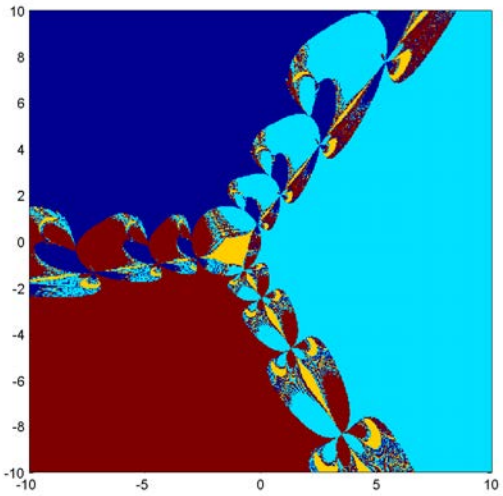


Figure 4: Attraction basins for method $M2_4$

 ISBN: 978-84-616-2723-3

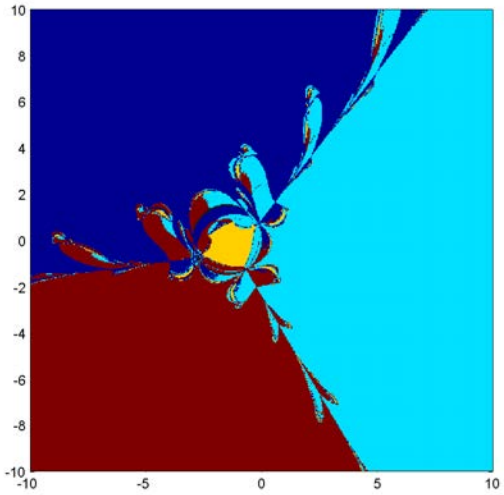


Figure 5: Attraction basins for method $M1_6$

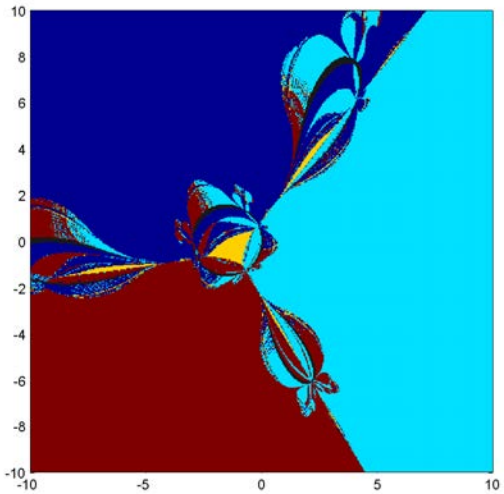


Figure 6: Attraction basins for method $M2_6$

 ISBN: 978-84-616-2723-3

twelve equations. From this point on, these last methods exceed lower order methods. In particular $M1_6$ goes closer than the others to the efficiency index of Newton's method. The dynamical experiment shows that the global convergence properties are not worsened by the increase of the order of the method.

Acknowledgments

This research has been supported by Ministerio de Ciencia e Innovación MTM2011-28636-C02-02 and by Vicerrectorado de Investigación Universitat Politècnica de València PAID-SP-2012-0498.

References

[1] G. H. Nedzhibov, *A family of multi-point iterative methods for solving systems of nonlinear equations*, Journal of Computational and Applied Mathematics, **222** (2008) 244-250.

[2] M. T. Darvishi, A. Barati, *A fourth-order method from quadrature formulae to solve systems of nonlinear equations*, Applied Mathematics and Computation, **188** (2007) 257-261.

[3] 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.

[4] S. Amat, S. Busquier, A. Grau, M. Grau-Sánchez, *Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications* Applied Mathematics and Computations, **219**, (2013), 7954-7963

[5] J. R. Sharma, R. K. Guha, R. Sharma, *An efficient fourth order weighted-Newton method for systems of nonlinear equations*, Numerical Algorithms, **62**, (2013), 307-323

[6] A. Cordero, J. L. Hueso, E. Martínez, J. R. Torregrosa, *Increasing the convergence order of an iterative method for nonlinear systems*, Applied Mathematics Letters, **24**, (2011), 2082-2086

[7] H. T. Kung, J. F. Traub, *Optimal order of one-point and multi-point iteration*, Journal of the Association for Computing Machinery, **21**, (1974) 643-651.

[8] M. Grau-Sánchex, M. Noguera, J. M. Gutiérrez, *On some computational orders of convergence*, Applied Mathematics Letters, **23**, (2010), 472-478

 ISBN: 978-84-616-2723-3

[9] X. Wang, J. Kou, Y. Li, *Modified Jarratt method with sixth-order convergence*, Applied Mathetmatics Letters, **22**, (2009), 1798-1802

[10] S. Mayer, D. Schleicher, *Immediate and virtual basins of Newtons method for entire functions*, Ann. Inst. Fourier **56** (2) (2006) 325336.

[11] S. Amat, S. Busquier, S. Plaza, *A construction of attracting periodic orbits for some classical third-order iterative methods*, Journal of Computational and Applied Mathematics **189** (2006) 2233.

[12] F. Chicharro, A. Cordero, J. M. Gutiérrez, J. R. Torregrosa, *Complex dynamics of derivative-free methods for nonlinear equations*, Applied Mathematics and Computation **219** (2013) 70237035.

[13] J. M. Gutiérrez, M. A. Hernández, N. Romero, *Dynamics of a new family of iterative processes for quadratic polynomials*, J. Comput. Appl. Math. **233** (2010) 26882695.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

GPU-accelerated uniform sampling of implicit surfaces

Masayuki Iwasaki[1], Susumu Nakata[2] and Satoshi Tanaka[2]

[1] *Graduate School of Information Science and Engineering, Ritsumeikan University, Japan*

[2] *College of Information Science and Engineering, Ritsumeikan University, Japan*

emails: is006084@ed.ritsumei.ac.jp, snakata@is.ritsumei.ac.jp,
stanaka@media.ritsumei.ac.jp

Abstract

An implicit surface sampling algorithm accelerated using parallel computation on Graphics Processing Unit (GPU) is presented in this paper. The sampling algorithm generates a set of surface points distributing uniformly in density and in distance. The proposed method is based on repulsive particle simulation that can be computed in parallel. In our numerical test, the repulsive particle simulation was accelerated up to 40 times over a single-thread CPU implementation.

Key words: GPU, implicit surfaces, repulsive particle system

1 Introduction

The purpose of this research is to develop a fast computational method for generating densely and uniformly distributed points on implicit surfaces. The sampling technique is required in many practical situations. For example, the boundary node method, a tool for solving boundary value problems, requires a set of surface points and the location of the points affects the accuracy of the solution. The point based rendering technique used in computer graphics or scientific visualization also requires a set of points densely distributed on surfaces and uniformity of the points determines the quality of the visualization.

Implicit surface sampling techniques have been well studied in computer graphics. Witkin and Heckbert [1] proposed a method for uniform sampling of implicit surfaces. In this method, the uniformity is achieved by applying a repulsive force model to the particles. Some improvements were made by Meyer et al. [2] for adaptive sampling and stable computation. In the sampling methods based on repulsive particles, initial points on surfaces are

ISBN: 978-84-616-2723-3

required. The unifrom sampling method proposed by Kojima et al. [3] adopted the sampling technique called the stochastic sampling method (SSM [4]) for automatic generation of initial point sets on implicit surfaces. In the SSM, a sequence of random surface points with a guarantee of convergence to a point cloud with uniform density can be stochastically generated and is suitable for an initial state of the repulsive particle simulation.

In this paper, we propose a GPU-accelerated uniform sampling method for implicit surfaces. The input of the process is a set of surface points given using a point generation method such as the SSM and we provide an effective parallel algorithm for iterative scheme of repulsive simulation. The algorithm is designed so that the local interactions of particles can be efficiently computed using GPU cores in parallel and, as a result, the motion of millions of repulsive particles can be performed with smaller cost than that on CPU.

2 Particle diffusion method for implicit surfaces

The particle diffusion method is a technique to generate a set of uniformly distributed surface points starting from a set of initial points. The uniformity of the points is achieved by assuming a repulsive force at all the given points.

Let us assume that an implicit surface defined by $F(\boldsymbol{x}) = 0$ and a set of surface points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are given, where N is the total number of the given points. In the particle diffusion method, the energy of a particle $\boldsymbol{x}_i$ derived by the other particles is defined by

$$E_i = \frac{1}{2} \sum_{j=1}^{N, j \neq i} E(\|\boldsymbol{r}_{ij}\|),$$

where E_i is the energy at the i-th particle, the notation $\|\cdot\|$ represent the Euclidean norm in three-dimensional space, $\boldsymbol{r}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and $E(t)$ a compactly supported function. Let $\boldsymbol{f}_{ij}$ be the repulsive force at the i-th particle as a result of the energy derived from the j-th particle. The force is defined as

$$\boldsymbol{f}_{ij} = -\frac{\boldsymbol{r}_{ij}}{\|\boldsymbol{r}_{ij}\|} \cdot \frac{d}{dt} E(\|\boldsymbol{r}_{ij}\|).$$

As a result, the force at the i-th particle imposed by its neighboring particles is determined as the total of the particle forces as

$$\boldsymbol{f}_i = P_i \sum_{j=1}^{N, j \neq i} \boldsymbol{f}_{ij}, \tag{1}$$

where P_i is the projection matrix that maps a three-dimensional vector onto the tangent plane at the i-th particle. See [4] for detail.

 ISBN: 978-84-616-2723-3

The motion of each particle is determined by the force (1). In the simulation, the locations of the particles are updated with a small time interval and the final locations are obtained by repeating this process. Note that, at each time step, the updated location of each particle deviates from the surface with a small distance because the motion of each particle is limited to the tangent plane. In our implementation, the gap is corrected at each time step using Newton's method.

3 Parallel computation of particle diffusion method on GPU

In each iterative step of the repulsive particle simulation, the following three processes are required: generation of a look-up table, computation of repulsive forces at all the particles and update of their locations with error correction. Note that all the coordinates of initial particles need to be sent to GPU memory in advance of the simulation and no other major data transfer is required during the iteration.

In the first stage, construction of the look-up table is required for efficient detection of neighboring particles of an arbitrary point in three-dimensional space. The table can be efficiently constructed using the combination of cell id of a uniform grid and particle id. In this process, sorting is required and the algorithm called the fast radix sort enables us fast sorting on GPU in parallel. In the second stage, the computation of the repulsive forces of all the particles can be performed in parallel. In our implementation, the computation of the force at one particle is assigned to one core on the GPU. In the final stage, locations of all the particles are updated in parallel according to the particles forces obtained in the previous stage. The error correction process using Newton's method is also performed in parallel after the update process.

The results of a test problem is summarized in Table 1 and an example of the result of uniform sampling with 10^5 points starting from the initial point cloud in Figure 1 is shown in Figure 2. The initial particles is generated using SSM and the number of iterations for uniform sampling is 10 for all tests. All the CPU computations are performed on an Intel Core i5 2500, 3.3 GHz processor (a single core is used throughout the numerical test), and the GPU computations are performed on an NVIDIA GeForce GTX 460 (336 SPs in total).

4 Conclusion

The uniform sampling based on repulsive particle system has successfully accelerated by parallel computation on GPU. The key of the effective parallelization is that the neighboring particle detection can be performed efficiently using a look-up table and that the repulsive force can be computed in parallel using many cores on GPU. The uniform sampling technique can be applied to practical problems such as continuum mechanics and visualization of surfaces.

 ISBN: 978-84-616-2723-3

Table 1: Computational time for repulsive particle simulation

num points	CPU [sec]	GPU [sec]	speed-up
100,000	3.9	2.8	1.3
310,000	30.1	4.3	7.0
1,000,000	222.6	14.7	15.1
3,160,000	1901.7	42.5	44.7

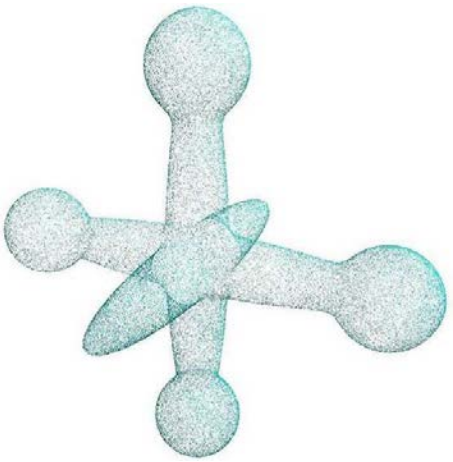


Figure 1: Original implicit surfaces model

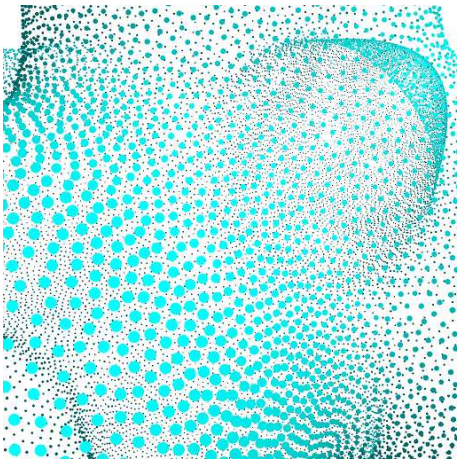


Figure 2: Result of uniform sampling

References

[1] A. P. Witkin and P. S. Heckbert, *Using particles to sample and control implicit surfaces*, Proc. SIGGRPAH '94, 1994

[2] M. D. Meyer, P. Georgel and R. T. Whitaker, *Robust particle for curvature dependent sampling of implicit surfaces*, Proc. Shape Modeling International 2005, 2005.

[3] K. Kojima, M. Oka, A. Shibata, S. Nakata and S. Tanaka, *Polygonization of high-density and large-scale point data based on a repulsive-force particle system on implicit surface*, Transactions of Visualization Society of Japan **27** (2007) 77–83.

[4] S. Tanaka, A. Morisaki, S. Nakata, Y. Fukuda and H. Yamamoto, *Sampling implicit surfaces based on stochastic differential equations with converging constraint*, Comput. Graph. **24** (2000) 419–431.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

GPU accelerated 4D-CT reconstruction using higher order PDE regularization in spatial and temporal domains

Daniil Kazantsev[1], William R.B. Lionheart[2], Philip J. Withers[1] and Peter D. Lee[1]

[1] *The Manchester X-Ray Imaging Facility, School of Materials, The University of Manchester, Manchester, M13 9PL, UK*

[2] *School of Mathematics, The University of Manchester, M13 9PL, UK*

emails: daniil.kazantsev@manchester.ac.uk, bill.lionheart@manchester.ac.uk, p.j.withers@manchester.ac.uk, peter.lee@manchester.ac.uk

Abstract

A variety of regularization techniques were used for iterative reconstruction in computed tomography. Generally, additional penalties on solution are imposed in the spatial domain only, however when data is undersampled and contaminated with noise the use of temporal information can be advantageous. In this paper we present a 4D spatial-temporal regularization based on fourth order partial differential equations (PDE's). Using higher order equations can be beneficial to obtain more smooth yet sharp on edges reconstructions and avoid piecewise-constant solutions related to lower order penalties, such as total variation (TV). To make reconstruction time feasible we implemented our method on graphical processor units (GPUs) using a shared memory approach. To show advantage of the technique presented we test it quantitatively using a computational 4D phantom. Numerical results show that the spatial-temporal approach presented yields images with sharper features, lower noise and bias compared to other techniques.

Key words: 4D reconstruction, spatial-temporal penalties, higher order PDE's, splitting methods, GPU acceleration

1 Introduction

The presence of noise in tomographic measurements (e.g. due to short exposure time) and the limited number of projections acquired (e.g. fast imaging to avoid blurring due to

sample motion) can lead to a low signal-to-noise ratio (SNR) of reconstructed images which are difficult to analyse. To increase the SNR and resolution of images it is advisable to use iterative reconstruction techniques [1] over analytical methods, such as filtered back-projection (FBP) or Fourier direct inversion [2].

Dealing with underdetermined ill-posed problems in tomography it is necessary to impose additional information on the solution (e.g. smoothness) to ensure well-posedness of the iterative algorithm [3]. Penalties based on partial differential equations (PDE), such as total variation (TV) [4] and anisotropic diffusion (AD) [5] are successful in dealing with noise while leaving edges intact. However, considering lower order equations leads to an undesirable "cartoon" effect where images recovered as piecewise-constant valued regions. One can avoid the staircase effect in reconstruction by using higher order PDE's (e.g. fourth order).

In situations where series (time frames) of projection data are available it is beneficial to use temporal information in addition to the spatial constraints [7],[9],[10],[11]. It is crucial to emphasize that the motion model is usually problem specific, therefore various model assumptions lead to different 4D reconstruction algorithms (a good overview can be found in [8]). Adding information using only adjacent time phases is the most "safe" method since it minimizes the risk of smoothing over quite different phases [9],[10]. This *local* approach can be used in experiments where significant motion is involved during scan. However this might impose its own limitations on reconstructed quality of images (see Discussion).

To impose sparsity in spatial and temporal domains simultaneously the TV penalty is widely used in 4D iterative reconstruction [10]. Instead of the conventional TV penalty which is usually linked with an undesirable staircase effect, in this work we use a higher order diffusivity term [6] for spatial and temporal regularization.

Calculating higher order derivatives generally means longer computational time since larger voxel stencils are considered. One can overcome this problem by employing parallel computing. Here we used GPUs to accelerate our algorithm by implementing a CUDA based code with an efficient shared memory model.

We use a splitting technique [12] to minimize a cost function alternatingly switching between minimization of several sub-problems [10]. The first step is performed with a conjugate gradient least squares (CGLS) optimization algorithm [13]; the second step is a minimization problem of the fourth order diffusivity functional in the spatial domain and the third step is optimization using the same penalty function in temporal domain.

Numerical experiments were performed using a synthetic 4D phantom to demonstrate visually and quantitatively the advantage of the proposed spatial-temporal penalty (CGLSST). We compared CGLSST with CGLS algorithm, CGLS with spatial regularization (CGLSS), and FBP.

 ISBN: 978-84-616-2723-3

2 METHOD

2.1 Parallel beam tomography model in 4D

A discrete representation of the attenuation to be reconstructed can be written as a system of linear equations:

$$b_j = \sum_{i=1}^{\mathrm{N}} a_{ji} x_i + \delta_j, \tag{1}$$

where $b_j, j = 1, \ldots, \mathrm{M}$ is the measured projection data (sinogram), $x_i, i = 1, \ldots, \mathrm{N}$ is the discrete distribution of attenuation coefficient to be reconstructed and δ_j is the noise component in b_j measurements. Weights $a_{ji} \in [0, 1]$ are forming the sparse system matrix $A : \mathbb{R}^{\mathrm{N}} \to \mathbb{R}^{\mathrm{M}}$.

Writing equation (1) in a matrix-vector form and adding the temporal dimension:

$$\boldsymbol{b}_k = A\boldsymbol{x}_k + \boldsymbol{\delta}_k, \quad k = 1, 2, \ldots, K \tag{2}$$

where K is a total number of 3D time frames or phases.

In our case the system of equations (2) is underdetermined ($\mathrm{M} \ll \mathrm{N}$) and the system matrix A is ill-conditioned. Here we aim to reconstruct iteratively the unknown set of images $\boldsymbol{x}_k$ while adding regularizing penalties in spatial and temporal domains.

2.2 Main structure of the 4D reconstruction algorithm

Lets define a vector consisting all image time frames as $X = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_K^T)^T$ and similarly the measured projection vectors as $B = (\boldsymbol{b}_1^T, \boldsymbol{b}_2^T, \ldots, \boldsymbol{b}_K^T)^T$.

The reconstruction problem is the minimization of the following cost function:

$$X = \min_X \|AX - B\|_2^2 + \beta_1 R_s(X) + \beta_2 R_t(X), \tag{3}$$

where the first term is related to the least squares data fidelity, R_s is a penalty in the spatial domain and R_t in temporal with corresponding regularization parameters $\beta_{1,2}$.

Using splitting ideas [12] one can write the optimization problem (3) as a nested algorithm [10]:

 ISBN: 978-84-616-2723-3

Algorithm 1 A nested method for 4D spatial-temporal reconstruction

$$\textbf{Step 1:} \quad \boldsymbol{v}_k^n = \boldsymbol{x}_k^n - \gamma A^T(A\boldsymbol{x}_k^n - \boldsymbol{b}_k) \tag{4}$$

$$\textbf{Step 2:} \quad \boldsymbol{t}_k^n = \min_{\boldsymbol{x}_k} \|\boldsymbol{x}_k^n - \boldsymbol{v}_k^n\|_2^2 + \beta_1 \|f(|\nabla^2 \boldsymbol{x}_k^n|)\| \tag{5}$$

$$\textbf{Step 3:} \quad \boldsymbol{x}_k^{n+1} = \min_{\boldsymbol{x}_k} \|\boldsymbol{x}_k^n - \boldsymbol{t}_k^n\|_2^2 + \beta_2(\|f(|\nabla^2(\boldsymbol{x}_k^n - \boldsymbol{x}_{k-1}^n)|)\| + \|f(|\nabla^2(\boldsymbol{x}_k^n - \boldsymbol{x}_{k+1}^n)|)\|) \tag{6}$$

$$\textbf{Step 4:} \quad \text{Repeat steps 1-3 until } \|\boldsymbol{x}_k^n - \boldsymbol{x}_k^{n-1}\|_2^2 < \epsilon_{tolerance} \tag{7}$$

Here step 1 is a gradient descent minimization of the least squares data term and for convergence acceleration it will be replaced by the CGLS algorithm [13]. Steps 2 and 3 are separate optimization problems which can be solved by gradient descent method. The proof of convergence for nested algorithms Lagrangian type is given in [12], here we will focus on the nature of penalties in steps 2-3.

2.3 Fourth order diffusion filter

Variational penalties based on L_1 norm minimization, such as TV [4], are used in image reconstruction with optimization of the following cost function:

$$x = \min_x \|Ax - b\|_2^2 + \beta_1 \|\nabla x\|_1, \tag{8}$$

where $\|\nabla x\|_1 = \sqrt{|x_x|^2 + |x_y|^2 + |x_z|^2}$. While promoting sparsity in image space the second term is successfully deals with noise in undersampled data. However, considering lower order derivatives in the Euler-Lagrange equation of (8) the resulting solution is prone to be piecewise constant, which usually is not favourable.

To impose spatial and temporal constraints in (3), we use a fourth order diffusion penalty [6] which successfully compromises between smooth variations in the flat regions and sharp edges. The cartoon effect is eliminated resulting in more visually pleasant smooth images with sharp edges.

Initializing $u_0 = \boldsymbol{v}_k^n$, $u^{m=0} = \boldsymbol{x}_k^n$ and applying Euler-Lagrange equation to minimization problem in the step 2 of Algorithm 1 the gradient descent iterations can be written as:

$$u^{m+1} = u^m + \tau \left((u^m - u_0) - \beta_1(\nabla^2(c(\|\nabla u^m\|)^2 u_{\eta\eta}^m + c(\|\nabla u^m\|)(u_{\xi_1\xi_1}^m + u_{\xi_2\xi_2}^m))\right), \tag{9}$$

where τ is a small time step constant, the diffusivity function $c \in (0, 1]$ is defined by Perona and Malik [5] as: $c(\|\nabla u\|) = \sigma^2/(\sigma^2 + \|\nabla u\|^2)$, σ is an edge preserving parameter, η is a direction of the gradient (normal) and $\xi_{1,2}$ are tangential directions. The strength of smoothing in the normal direction is suppressed by the square factor. Equation (9) is solved

 ISBN: 978-84-616-2723-3

explicitly (small time step τ and large number of iterations m) using Euler approximation for discretization in 3D space, the details for 2D case can be found in [6].

The temporal information is embedded in the step 3 of Algorithm 1 by using adjacent time frames. It is the same type of gradient descent algorithm as (9) with a different initialization.

3 Numerical Experiments

In this section we present numerical results of reconstruction using a synthetic 4D phantom.

3.1 Synthetic 4D phantom reconstruction

We use the following quantitative measures: given the true image u and reconstructed image u^* the criterion signal to noise ratio (SNR) in the region of interest (ROI) is:

$$SNR(u,u^*)_{ROI} = 10\log\left(\frac{\|u^*-\overline{u}^*\|_2}{\|u^*-u\|_2}\right), \tag{10}$$

where $\overline{u}^*$ is a mean of the reconstructed ROI of image u^*.

For accuracy measure the normalized root-mean-square error (NRMSE) used:

$$NRMSE(u,u^*)_{ROI} = \frac{\|u^*-u\|_2}{\|u\|_2}, \tag{11}$$

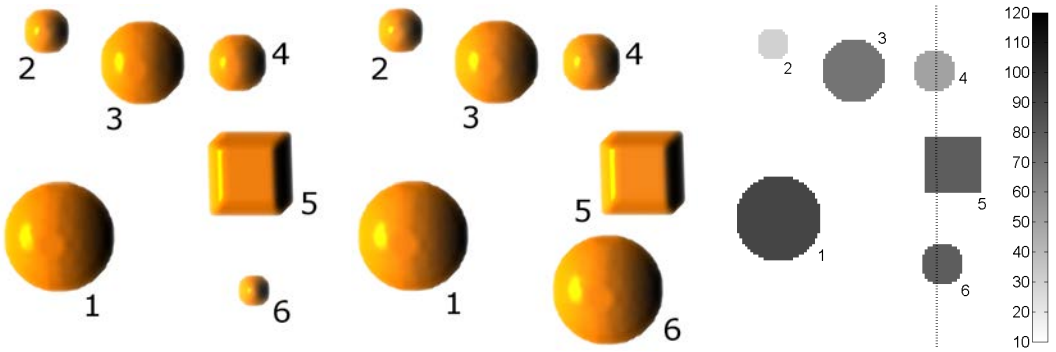


Figure 1: 3D phantom $128 \times 128 \times 128$ which consists of 6 objects, static spheres No. 1-4 and dynamic objects No. 5-6 (Obj. No. 5 is shifting and No.6 is expanding); Left: the 1st time frame, Centre: the 7th (final) phase, Right: The middle slice $z = 64$ of the 3rd time frame.

In Fig. 1, the volumetric synthetic phantom ($128\times128\times128$ voxels) is used for numerical experiments. The phantom comprises four static objects (they remain stationary during

 ISBN: 978-84-616-2723-3

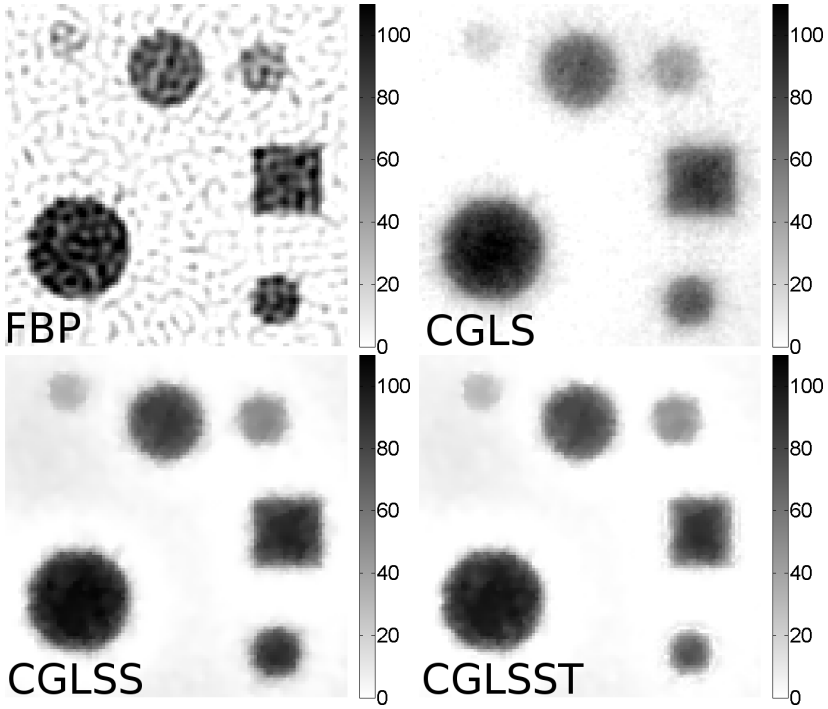


Figure 2: 2D slices of 3D phantom (time frame 3) (see Fig. 1, right) reconstructed with: FBP, CGLS (3 iterations), CGLS (4 iterations) with spatial smoothing R_s (30 iterations of (9) in step 2 of Algorithm 1) and CGLS (4 iterations) with spatial R_s and temporal smoothing R_t (30 iterations in step 3 of Algorithm 1). The gray map scale of intensities is given.

the experiment), namely objects No. 1-4 and dynamic objects No.5-6. The objects have different intensity values. We test our method by introducing seven time frames ($K = 7$) during which object No. 5 is shifting and No. 6 is expanding (see 1 left and centre).

After forward projection of x_k phantoms using 190 detectors and 160 acquisition angles in $[0, \pi]$ angular interval, δ_k realizations of Poisson noise was applied to the data. Using noisy sinograms b_k, all phantoms were reconstructed using FBP method with Shepp-Logan filter [2], CGLS [13], CGLS with spatial penalty R_s (9) and CGLS with proposed temporal-spatial reconstruction technique (CGLSST).

The iteration processes for CGLS, CGLSS and CGLSST were stopped when the minimum of NRMSE and maximum of SNR was reached for each method. The reconstructions in Fig.2 are shown for the optimal NRMSE-SNR values. The NRMSE-SNR values were calculated for ROIs belong to static and dynamic objects and presented in Fig. 3. As expected, for dynamic objects (see 3, right) the NRMSE-SNR values are slightly worse than

 ISBN: 978-84-616-2723-3

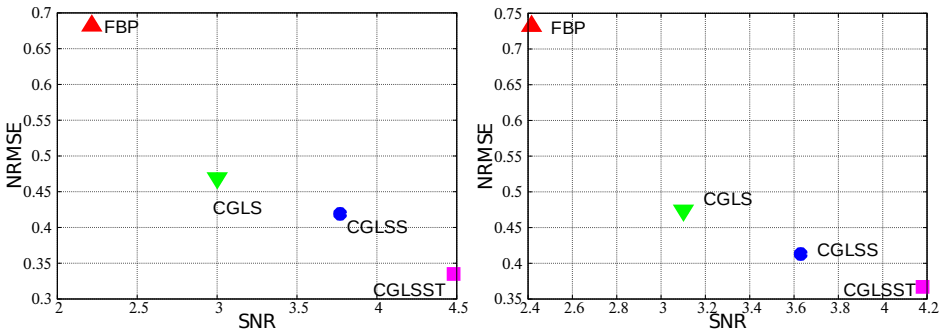


Figure 3: NRMSE-SNR plots for 4 methods, left: ROI's are static objects No. 1-4, right: ROI's are dynamic objects No. 5-6.

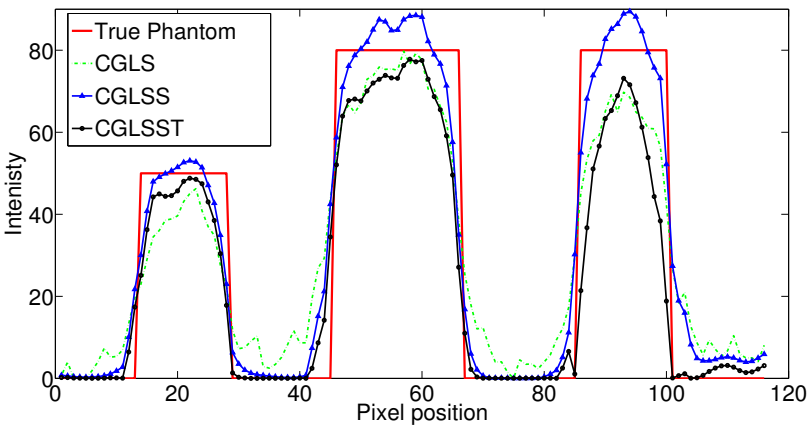


Figure 4: 1D plot across the reconstructed volume (3rd time frame) (see dotted line in the Fig. 1, right). Edges are more emphasized when using CGLSST method.

for static ones, but nevertheless objects No. 5-6 have more defined edges with combined spatial-temporal reconstruction (see Fig. 2).

The 1D plot (see Fig. 4) through the reconstructed 3rd time frame (see dotted line in the Fig. 1, right) shows more emphasized edges for static and dynamic objects. Moroever the curve for the CGLSST is the closest one to the profile of exact phantom (the NRMSE equals to 0.335 for CGLSST and 0.419 for CGLSS).

The images reconstructed with FBP (see Fig.2) suffer from the strong noise (low SNR) and high error (see Fig.3). CGLS reconstruction gives significant enhancement of NRMSE and SNR, however noise level is still high. Using spatial regularization with CGLS leads to a better noise reduction, however the edges of objects are slightly blurred. The bias value is high for CGLSS (the intensity level is overestimated according to the plot in Fig.

 ISBN: 978-84-616-2723-3

4). Temporal regularization strongly increases NRMSE-SNR characteristics of static and dynamic objects, edges are better emphasized with CGLSST.

The CUDA implementation on GPU using shared memory gives feasible time of reconstruction using spatio-temporal penalties. Reconstruction of 7 volumes takes less than one minute with CGLSST. Further optimization of the code is possible and currently in progress.

4 Discussion

Using adjacent time frames in the step 3 of Algorithm 1 leads to better bias-noise characteristics. However, when motion is known to be insignificant the use of all time frames can be more beneficial.

To appreciate the benefits of the higher order penalties to the lower ones we aim to test our method in the future applied to smooth objects (e.g. Gaussian) and also real data.

5 Conclusions

In this paper we present preliminary results of spatial-temporal regularization technique which based on the fourth order diffusivity model. Applying this method results in better resolution and reduced noise of reconstructed images. As such this method has potential for application in the visualisation and quantification of dynamic processes.

Acknowledgements

This work has been supported by the Engineering and Physical Sciences Research Council under grants EP/J010456/1 and EP/I02249X/1. Travel was supported by the European Commission under the 7th Framework Programme through the "Research Infrastructures" action of the "Capacities" Programme, NMI3-II Grant No. 283883, Project No.20120553

References

[1] C. L. Byrne, *Applied Iterative Methods*, Natick, MA: Peters, 2008.

[2] A. Kyrieleis, V. Titarenko, M. Ibson, T. Connoley, P.J. Withers, *Region-of-interest tomography using filtered back-projection: assessing the practical limits*, J. Micros. **241:1** (2011) 69–82.

 ISBN: 978-84-616-2723-3

[3] V. Titarenko, R. Bradley, C. Martin, P.J. Withers, S. Titarenko, *Regularization methods for inverse problems in x-ray tomography*, In Proc., SPIE 7804, 78040Z. (2010).

[4] L.I. Rudin, S. Osher, E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D. **60** (1992) 259–268.

[5] P. Perona, J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Analysis **12** (1990) 629–639.

[6] M. R. Hajiaboli, *An Anisotropic Fourth-Order Diffusion Filter for Image Noise Removal*, International Journal of Computer Vision **92:12** (2011) 177–191.

[7] D. S. Lalush, B. M. W. Tsui, *Block-iterative techniques for fast 4D reconstruction using a priori motion models in gated cardiac SPECT*, Phys Med. Biol. **43** (1998) 875–886.

[8] A. Rahmim, J. Tang, H. Zaidi, *Four-dimensional (4D) image reconstruction strategies in dynamic PET: Beyond conventional independent frame reconstruction*, Med. Phys. **36** (2009)3654–3670.

[9] X. Jia, Y. Lou, B. Dong, Z. Tian, S. Jiang, *4D computed tomography reconstruction from few-projection data via temporal non-local regularization*, Lect. Not. Comput. Sci. **6361** (2010) 143–150.

[10] H. Wu, A. Maier, R. Fahrig, J. Hornegger, *Spatial-temporal total variation regularization (STTVR) for 4D-CT reconstruction*, Proc. SPIE 8313 (2012) 237–240.

[11] W.M. Thompson, W.R.B. Lionheart, E.J. Morton, *Real-Time imaging with a high speed X-Ray CT system*, 6th International Symposium on Process Tomography (2012).

[12] P. L. Combettes, J. C. Pesquet, *Proximal splitting methods in signal processing*, Eds. New York: Springer-Verlag (2010).

[13] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 2006.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Riccati Transformation Method for Solving Constrained Dynamic Stochastic Optimal Allocation Problem

Soňa Kilianová[1] **and Daniel Ševčovič**[1]

[1] *Dept. Applied Mathematics & Statistics, Comenius University*
842 48 Bratislava, Slovakia

emails: kilianova@fmph.uniba.sk, sevcovic@fmph.uniba.sk

Abstract

In this paper we present our recent results on application of the Riccati transformation for solving the evolutionary Hamilton-Jacobi-Bellman equation arising from the stochastic dynamic optimal allocation problem. It turns out that the fully nonlinear Hamilton-Jacobi-Bellman equation governing evolution of the value function can be transformed into a quasi-linear parabolic equation. Its diffusion function is obtained as a value function of certain parametric convex optimization problem. A solution is then constructed by means of an implicit iterative finite volume numerical approximation scheme. As an application we present results of computing optimal strategies for a portfolio investment problem.

Key words: Hamilton–Jacobi–Bellman equation, Riccati transformation, quasi-linear parabolic equation, finite volume approximation scheme
MSC 2000: 34E05 70H20 91B70 90C15 91B16

1 Introduction

The goal of this paper is to investigate a novel method based on the Riccati transformation for solving a time dependent Hamilton-Jacobi-Bellman equation arising from a stochastic dynamic optimal allocation problem on a finite time horizon. Our motivation arises from a dynamic stochastic optimization problem in which the purpose is to maximize the conditional expected value

$$\max_{\boldsymbol{\theta}|_{[0,T)}} \mathbb{E}\left[U(X_T^{\boldsymbol{\theta}}) \,\middle|\, X_0^{\boldsymbol{\theta}} = x_0\right], \tag{1}$$

 ISBN: 978-84-616-2723-3

of the terminal utility $U(X_T^{\boldsymbol{\theta}})$ of a portfolio. Here $\{X_t^{\boldsymbol{\theta}}\}$ is an Itō's stochastic process on the finite time horizon $[0,T]$, $U:\mathbb{R}\to\mathbb{R}$ is a given terminal utility function and x_0 a given initial state condition of $\{X_t^{\boldsymbol{\theta}}\}$ at $t=0$. The function $\boldsymbol{\theta}:\mathbb{R}\times[0,T)\to\mathbb{R}^n$ mapping $(x,t)\mapsto\boldsymbol{\theta}(x,t)$ represents an unknown control function governing the underlying stochastic process $\{X_t^{\boldsymbol{\theta}}\}_{t\geq 0}$. Here $\boldsymbol{\theta}|_{[t,T)}$ for $0\leq t<T$ denotes the restriction of the control function $\boldsymbol{\theta}$ to the time interval $[t,T)$. We assume that $X_t^{\boldsymbol{\theta}}$ is driven by the stochastic differential equation

$$\mathrm{d}X_t^{\boldsymbol{\theta}}=\left\{\varepsilon e^{-X_t}+r+\mu(\boldsymbol{\theta})-\sigma(\boldsymbol{\theta})^2/2\right\}\mathrm{d}t+\sigma(\boldsymbol{\theta})\mathrm{d}W_t, \tag{2}$$

where W_t denotes the standard Brownian motion and the functions $\mu(\boldsymbol{\theta})$ and $\sigma(\boldsymbol{\theta})$ are the drift and volatility functions depending on the control function $\boldsymbol{\theta}$. The parameter $\varepsilon\geq 0$ represents a constant inflow rate of property to the system whereas $r\geq 0$ is the interest rate. Throughout the paper we shall assume that the control parameter $\boldsymbol{\theta}\in\mathcal{S}^n$ belongs to the compact simplex

$$\mathcal{S}^n=\{\boldsymbol{\theta}\in\mathbb{R}^n\mid\boldsymbol{\theta}\geq\mathbf{0},\mathbf{1}^T\boldsymbol{\theta}=1\}\subset\mathbb{R}^n, \tag{3}$$

where $\mathbf{1}=(1,\cdots,1)^T\in\mathbb{R}^n$. It should be noted that the process $\{X_t^{\boldsymbol{\theta}}\}$ is a logarithmic transformation of a stochastic process $\{Y_t^{\tilde{\boldsymbol{\theta}}}\}_{t\geq 0}$ driven by the SDE:

$$\mathrm{d}Y_t^{\tilde{\boldsymbol{\theta}}}=\left\{\varepsilon+[r+\mu(\tilde{\boldsymbol{\theta}})]Y_t^{\tilde{\boldsymbol{\theta}}})\right\}\mathrm{d}t+\sigma(\tilde{\boldsymbol{\theta}})Y_t^{\tilde{\boldsymbol{\theta}}}\mathrm{d}W_t, \tag{4}$$

where $\tilde{\boldsymbol{\theta}}(y,t)=\boldsymbol{\theta}(x,t)$ with $x=\ln y$.

As a typical example leading to the stochastic dynamic optimization problem (1) in which the underlying stochastic process satisfies SDE (2) one can consider a problem of dynamic portfolio optimization in which the assets are labeled as $i=1,\cdots,n$, and associated with the price processes $\{Y_t^i\}_{t\geq 0}$, each of them following a geometric Brownian motion

$$\frac{\mathrm{d}Y_t^i}{Y_t^i}=\mu_i\mathrm{d}t+\sum_{j=1}^n\bar{\sigma}_{ij}\mathrm{d}W_t^j,$$

(cf. Merton [12, 13], Browne [4], Bielecki and Pliska [3]. The value of a portfolio with weights $\tilde{\boldsymbol{\theta}}=\tilde{\boldsymbol{\theta}}(y,t)$ is denoted by $Y_t^{\tilde{\boldsymbol{\theta}}}$. We have $\mu(\boldsymbol{\theta})=\boldsymbol{\mu}^T\boldsymbol{\theta}$ and $\sigma(\boldsymbol{\theta})^2=\boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta}$ with $\boldsymbol{\mu}=(\mu_1,\cdots,\mu_n)^T$ and $\boldsymbol{\Sigma}$ is a positive definite symmetric covariance matrix, $\boldsymbol{\Sigma}=\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{\Sigma}}^T$ where $\bar{\boldsymbol{\Sigma}}=(\bar{\sigma}_{ij})$. It can be shown that $\{Y_t^{\tilde{\boldsymbol{\theta}}}\}_{t\geq 0}$ satisfies (4) with $\varepsilon=r=0$. The assumption $\boldsymbol{\theta}\in\mathcal{S}^n$ corresponds to the situation in which borrowing of assets is not allowed, i.e. $\theta_i\geq 0$ and $\sum_{i=1}^n\theta_i=1$. A function $U(x)$ represents a given terminal utility function representing investor's risk preferences.

 ISBN: 978-84-616-2723-3

2 Hamilton-Jacobi-Bellman Equation and Method of Riccati transformation

It is known from the theory of stochastic dynamic programming that the so-called value function

$$V(x,t) := \sup_{\boldsymbol{\theta}|_{[t,T)}} \mathbb{E}\left[U(X_T^{\boldsymbol{\theta}})|X_t^{\boldsymbol{\theta}} = x\right] \tag{5}$$

subject to the terminal condition $V(x,T) := U(x)$ can be used for solving the stochastic dynamic optimization problem (1) (cf. Bertsekas [2]). If the process $X_t^{\boldsymbol{\theta}}$ is driven by (2), then the value function $V = V(x,t)$ satisfies the Hamilton-Jacobi-Bellman (HJB) equation

$$\partial_t V + \max_{\boldsymbol{\theta}\in\mathcal{S}^n}\left\{\left(\varepsilon e^{-x} + r + \mu(\boldsymbol{\theta}) - \frac{1}{2}\sigma(\boldsymbol{\theta})^2\right)\partial_x V + \frac{1}{2}\sigma(\boldsymbol{\theta})^2\partial_x^2 V\right\} = 0\,, \tag{6}$$

for all $x \in \mathbb{R},\ t \in [0,T)$ subject to the terminal condition $V(x,T) := U(x)$ (see e.g. Macová and Ševčovič [11] or Ishimura and Ševčovič [6]).

Following the methodology of the Riccati transformation studied by Ishimura *et al.* [1, 5, 7] and further analyzed by Ishimura and Ševčovič [6], we introduce the following Riccati like transformation:

$$\varphi(x,t) = 1 - \frac{\partial_x^2 V(x,t)}{\partial_x V(x,t)}. \tag{7}$$

According to [8, Theorem 3.2], the transformed function φ is a solution to a Cauchy problem for the following quasi-linear parabolic equation

$$\begin{aligned} &\partial_t\varphi + \partial_x^2\alpha(\varphi) + \partial_x[(\varepsilon e^{-x} + r)\varphi + (1-\varphi)\alpha(\varphi)] = 0, \quad x \in \mathbb{R}, t \in [0,T), \\ &\varphi(x,T) = 1 - U''(x)/U'(x), \quad x \in \mathbb{R}, \end{aligned} \tag{8}$$

where the diffusion function $\alpha(\varphi)$ is obtained as the value function of the parametric non-linear constrained optimization problem.

$$\alpha(\varphi) = \min_{\boldsymbol{\theta}\in\mathcal{S}^n}\{-\mu(\boldsymbol{\theta}) + \frac{\varphi}{2}\sigma(\boldsymbol{\theta})^2\}\,. \tag{9}$$

In our application the problem (9) is a convex quadratic programming problem with $\mu(\boldsymbol{\theta}) := \boldsymbol{\mu}^T\boldsymbol{\theta}$ and $\sigma(\boldsymbol{\theta})^2 := \boldsymbol{\theta}^T\boldsymbol{\Sigma}\boldsymbol{\theta}$ where $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma}$ is a positive definite $n \times n$ matrix.

Unfortunately, the value function $\alpha(\varphi)$ need not be sufficiently smooth. Indeed, according to [8, Theorem 4.1] $\alpha \in C^{1,1}(\mathbb{R}^+)$, i.e. its derivative is Lipschitz continuous only. Moreover, with regard to [8] there are concrete market data examples of German DAX 30 stock index for which the value function can have a finite number of discontinuities in the second derivative of α.

 ISBN: 978-84-616-2723-3

Applying the methodology of Schauder estimates we were able to prove the following result on existence and smoothness of classical solutions to (8) belonging to the parabolic Hölder spaces $H^{2+\lambda,1+\lambda/2}(\mathbb{R}\times[0,T])$ for some $0<\lambda<1$. The detailed proof can be found in the recent paper [8] by the authors.

Theorem 2.1 *Suppose that* $\mathbf{\Sigma}$ *is positive definite,* $\boldsymbol{\mu}\in\mathbb{R}^n, \varepsilon, r\geq 0$, *and the optimal value function* $\alpha(\varphi)$ *is given by (9). Assume that the terminal condition* $\varphi(x,T)=1-U''(x)/U'(x)$, $x\in\mathbb{R}$, *is positive and uniformly bounded for* $x\in\mathbb{R}$ *and belongs to the Hölder space* $H^{2+\lambda}(\mathbb{R})$ *for some* $0<\lambda<1/2$. *Then there exists a unique classical solution* $\varphi(x,t)$ *to the backward quasi-linear parabolic equation (8) satisfying the terminal condition* $\varphi(x,T)$. *The function* $t\mapsto\partial_t\varphi(x,t)$ *is* $\lambda/2$*-Hölder continuous for all* $x\in\mathbb{R}$ *whereas* $x\mapsto\partial_x\varphi(x,t)$ *is Lipschitz continuous for all* $t\in[0,T]$. *Moreover,* $\alpha(\varphi(.,.))\in H^{2+\lambda,1+\lambda/2}(\mathbb{R}\times[0,T])$.

3 Application to portfolio optimization

In [8], the authors proposed an iterative numerical approximation scheme for solving the Cauchy problem for the quasi-linear parabolic equation (8). We followed the method of a finite volume approximation scheme (cf. LeVeque [10]) combined with a nonlinear equation iterative solver proposed by Mikula and Kútik in [9]. The scheme has been tested with semi-explicit traveling wave solutions (see [8, Sections 6,7]) and it turned out that the scheme is of the second experimental order of convergence. We furthermore applied the scheme to a practical example in which our goal was to optimize a portfolio consisting of $n=30$ assets forming the German DAX 30 Index. The regular contribution to the portfolio was set to $\varepsilon=1$ and $r=0$. As far as the utility function is concerned, we considered the constant absolute risk aversion (CARA) utility function of the form $U(x)=-\frac{1}{a-1}\exp(-(a-1)x)$ with a coefficient of the absolute risk aversion $a=9$. In terms of the transformed variable $x=\ln y$ the CARA utility function corresponds to the constant relative risk aversion (CRRA) function $\widetilde{U}(y)=-\frac{1}{a-1}y^{-a+1}$. We considered the finite time horizon $T=10$.

Using the finite volume approximation scheme we constructed a numerical solution $\varphi(x,t)$ to the quasilinear parabolic equation (8). Then, by solving the parametric quadratic programming problem (9) for $\varphi=\varphi(x,t)$ we found optimal response strategies $\boldsymbol{\theta}$ as a function of the logarithmic level of property x and time t. Results of numerical calculation are shown at Fig. 1.

It turned out shows that there are only a few relevant assets out of the set of thirty assets entering the DAX 30 Index. The figure reveals the highest portion of Merck stocks for the early period of saving and for low account values y. It is indeed reasonable to invest in an asset with the highest expected return, although with the highest volatility, when the account value is low, in early times of saving. Evident fast decrement of the Merck company weight can be observed for increasing account value. It should be noted that Fresenius Medical company has the lowest volatility out of the considered five assets (and

 ISBN: 978-84-616-2723-3

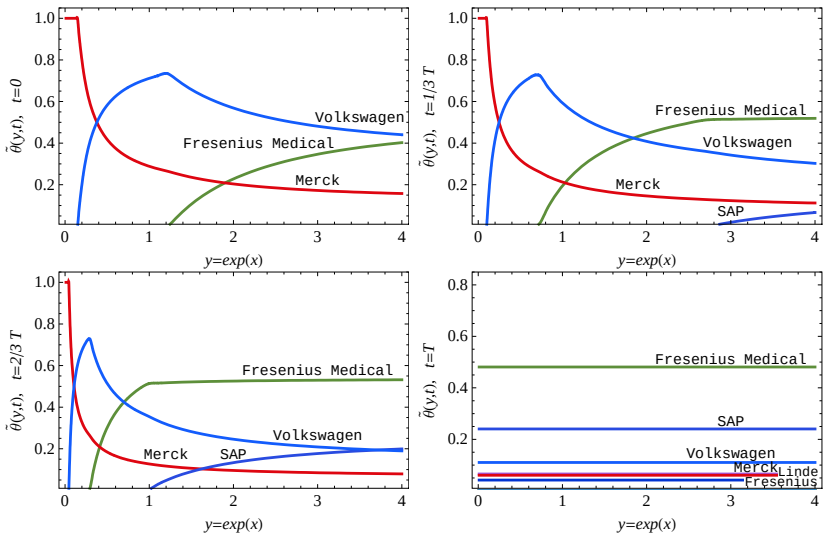


Figure 1: Nonzero components $\tilde{\boldsymbol{\theta}}_i, i \in \{1, \cdots, n\}$ of tptimal response strategy vector $\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}(y,t) = \boldsymbol{\theta}(\ln y, t)$ for the DAX 30 index portfolio optimization, for time instances $t = 0$, $t = T/3$, $t = 2T/3$ and $t = T$ where $T = 10$ Source: [8].

third lowest out of all thirty assets) and third best mean return, which is reflected in its major representation in the portfolio.

Acknowledgments

This work has been partially supported by VEGA grants 1/2429/12 and 1/0747/12.

References

[1] R. Abe and N. Ishimura: Existence of solutions for the nonlinear partial differential equation arising in the optimal investment problem, *Proc. Japan Acad., Ser. A.* **84** (2008), 11–14.

[2] D.P. Bertsekas: *Dynamic Programming and Stochastic Control*. Academic Press, 1976.

[3] T.R. Bielecki, S.R. Pliska and S.J. Sheu: Risk Sensitive Portfolio Management with Cox–Ingersoll–Ross Interest Rates: the HJB Equation. *SIAM J. Control and Optimization*, **44**(5) (2005), 1811–1843.

[4] S. Browne: Risk-Constrained Dynamic Active Portfolio Management. *Management Science*, **46**(9) (1995), 1188–1199.

[5] N. Ishimura and S. Maneenop: Traveling wave solutions to the nonlinear evolution equation for the risk preference, *JSIAM Letters* **3** (2011), 25–28.

[6] N. Ishimura and D. Ševčovič: On traveling wave solutions to a Hamilton-Jacobi-Bellman equation with inequality constraints, *Japan Journal of Industrial and Applied Mathematics* **30**(1) (2013), 51–67.

[7] N. Ishimura and N. Nakamura: Risk preference under stochastic environment. In: BMEI 2011 - Proceedings 2011 International Conference on Business Management and Electronic Information, Vol. 1, 2011, Article number 5917024, 668–670.

[8] S. Kilianová, M. and D. Ševčovič: Transformation Method for Solving Hamilton-Jacobi-Bellman Equation for Constrained Dynamic Stochastic Optimal Allocation Problem, *submitted*, 2013.

[9] P. Kútik and K. Mikula: Finite Volume Schemes for Solving Nonlinear Partial Differential Equations in Financial Mathematics. In: Finite Volumes for Complex Applications VI Problems & Perspectives, Springer Proceedings in Mathematics, 2011, Vol. 4(1), 643–651.

[10] R. LeVeque: *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[11] Z. Macová and D. Ševčovič: Weakly nonlinear analysis of the Hamilton-Jacobi-Bellman equation arising from pension saving management, *International Journal of Numerical Analysis and Modeling* **7**(4) (2010), 619–638.

[12] R.C. Merton: Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case. *Rev. Econom. Statist.* **51** (1969), 247–257.

[13] R.C. Merton: Optimal consumption and portfolio rules in a continuous time model, *J. Econ. Theory* **3** (1971), 373–413.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Computational Comparison of Various FEM Adaptivity Approaches

Lukáš Korous[1], Pavel Kůs[1] and Pavel Karban[1]

[1] *Faculty of Electrical Engineering, University of West Bohemia*

emails: korous@rice.zcu.cz, pkus@rice.zcu.cz, karban@kte.zcu.cz

Abstract

Mesh adaptivity algorithms for the Finite Element Method are crucial for solving problems with no, or only little a-priori information about its solution as well as for solution of any problem where saving memory and CPU time is important.

These algorithms can significantly influence the results in terms of quality and resolution. But CPU time and memory consumption are not less important when it comes to solving real problems for engineering or any other purposes.

- On one hand one can get unnecessarily well resolved results (below the level of detail that the engineer can utilize) and pay for it by waiting unbearable amount of time.
- On the other hand by decreasing the mesh resolution to obtain the results faster and cheaper, one can lose substantial details of the solution at hand.

A lot of work has been done in the development of robust adaptive strategies ([**?**], [**?**]) and also in comparing their performance from the mathematical point of view ([**?**], [**?**]), but as soon as an adaptive implementation is able to give the user the solution he is looking for with a sufficient resolution, a natural question about their speed arises.

In this work, we focus on the speed aspects of some FEM adaptive algorithms, and not only the CPU time - as it is a very implementation-specific quantity to compare. We compare the following attributes:

- Steps the adaptive algorithm must make in order to achieve the error threshold.

ISBN: 978-84-616-2723-3

- Number of unknowns reached by the adaptive algorithm (the smaller the algebraic problems, the better).
- Cumulative number of unknowns of all systems solved in the process (even when the final number of unknowns is small, when this quantity is big, the CPU time will suffer).
- Capabilities of the algorithm to perform faster by caching of any sort, in this sense, how many local stiffness matrices and rhs vectors can be reused (both for matrix/vector assembling and the linear system solution) compared to their total number.
- Cumulative direct solver factorization size, used memory and flops needed.
- Measurement how well a particular adaptive strategy followed the prescribed error threshold (by not dropping unnecessarily much below it, etc.).
- Error estimate and exact error (where appropriate) comparison.
- Nonlinear solver performance on adapted meshes.

We compare h-adaptivity, p-adaptivity, and many kinds of hp-adaptivity with various polynomial orders, with various settings and strategies how to perform refinements. The test examples are taken from ([**?**]), and from the Hermes library ([**?**]) examples collection. Data are collected from a large number of problem and algorithm settings to obtain a general view of the algorithms' performance.

All this work was done within the framework of the open source Finite Element library Hermes2D, for problems of various size and complexity. The underlying data structures and algorithms are described to support the conclusions.

Key words: FEM, hp-FEM, adaptivity, algorithm performance, memory consumption, CPU time, finite element method

Acknowledgments

This work was supported by the European Regional Development Fund and Ministry of Education, Youth and Sports of the Czech Republic (project No. CZ.1.05/2.1.00/03.0094: Regional Innovation Center for Electrical Engineering - RICE) and by the project P102/11/0498 (Grant Agency of the Czech Republic).

References

[1] W. Mitchell, *A Collection of 2D Elliptic Problems for Testing Adaptive Algorithms*, NISTIR 7668, (2010).

 ISBN: 978-84-616-2723-3

[2] W. Mitchell, *A Survey of hp-Adaptive Strategies for Elliptic Partial Differential Equations*, Annals of the European Academy of Sciences, (2011).

[3] P. Solin, D. Andrs, J. Cerveny, M. Simko, *PDE-Independent Adaptive hp-FEM Based on Hierarchic Extension of Finite Element Spaces*, J. Comput. Appl. Math. **233** (2010) 3086-3094.

[4] P. Solin, J. Cerveny, I. Dolezel, *Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM*, Math Comput Simul **77** 117132.

[5] P. Solin, K. Segeth, I. Dolezel, *Higher-order finite element methods*, Chapman & Hall/CRC Press (2003).

[6] hp-FEM group, *Hermes2D library*, http://www.hpfem.org/hermes.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

On Synergy of Totally Connected Flows on Chainmails

V.V. Kozlov[1], A.P. Buslaev[2] and A.G. Tatashev[3]

[1] *Steklov Mathematical Institute, Russian Academy of Sciences*

[2] *Department of Higher Mathematics, Moscow Automobile and Road State Technical University*

[3] *Department of Mathematical Cybernetics and Information Technologies, Moscow Technical University of Communications and Informatics*

emails: kozlov@pras.ru, apal2006@yandex.ru, a-tatashev@rambler.ru

Abstract

In present paper, *the movement of clusters on networks of some symmetrical structures* (**chainmails**) is considered. The network consists of rings, and each ring has a common point (vertex) with each of two or four neighboring rings. There is a single cluster on each ring. Each cluster moves in the same direction except in the case of situations when two clusters compete for the common point of the rings. In the latter case the loser cluster is waiting for the intersection. We consider the problem *of finding sufficient conditions of that a finite time instance exists since that all the clusters move freely with maximum possible velocity (***synergy***), and also conditions of that this effect does not occur for finite time.*

Key words: Cluster models, synergy, chainmails, monotonic random walks

1 Introduction

On of the possible approaches in flow modeling is to represent the movement in form of clusters, i.e., fragments of uniform distributed particles, which move with a velocity depending on the flow density and interacting according special rules.

As we suppose, *models of this type have been introduced in [1 – 3],* where the movement of clusters on a straight line and a circle have also been investigated.

In present paper, we consider *the movement of clusters on networks of some symmetrical structures* (**chainmails**). The network consists of rings, and each ring has a common point (vertex) with each of two neighboring rings.

In common case, the model is reduced to a system of ordinary differential equation with non-linear hand, and with dimension (the number of equations and parameters) on graphs, though, in the case of classic Greenschields, considered in our paper, the right hand is linear. This fact does not help to investigate the problem, because support is too complicated.

There is a single cluster, which moves in the same direction except in the case of situations when two clusters compete for the common point of the rings. In the latter case the loser cluster is waiting for the intersection, and its state varies in accordance with a designated scenario.

The problem *of finding sufficient conditions of that a finite time instance exists since that all the clusters move freely with maximum possible velocity (***synergy***), and also conditions of that this effect occurs for no finite time.*

2. Model description

Let us describe the model. There are some set of circles (rings) of radius $r = 1$. These circles form a network of a given structure (**chainmail**). There is a cluster on each ring. The density, the coordinates of boundaries, and the velocity of movement characterize the cluster. No common point of two rings can be located within two clusters simultaneously.

Suppose the cluster density cannot be more than a fixed value (for example, $\rho_{max} = 1$.) At present time each boundary (front and rear) of the cluster moves with velocity $v_0 = f(\rho_0)$, which is a function of the cluster density, or does not move. The function f can be any smooth function $[0, 1] \to R$, $f(0) = 1$, $f(1) = 0$, e.g., $f(x) = 1 - x$, and the cluster state is defined as a step function of density in accordance with one of the following rules.

(C_{ld}) *Cluster model with locally distributed information.* In this case there two possible values of density function. The densities depend on the velocity of the cluster boundaries velocity. The density function is uniquely determined by the law of mass conservation. All changes occur at the junction of a congestion and natural flow state.

If the front cluster boundary O coincides with the vertex, through that the neighboring cluster is going, and the cluster density ρ_0 is less than 1, then the front boundary of the considered cluster continues be located at the vertex, the rear boundary moves with velocity v_0, Fig. 1., and the congestion boundary x varies as

$$\dot{x} = -\frac{v_0\rho_0}{1-\rho_0}$$

since the point O.

Let l_0 be the length of a cluster, moving freely; ρ_0 is the density; $v_0 = f(\rho_0)$ is the velocity; $l(t)$ is the cluster length at time t; $x(t)$ is the length of the cluster component of density $\rho_0;$, $y(t)$ is the length of the cluster component of density 1. Then the following

 ISBN: 978-84-616-2723-3

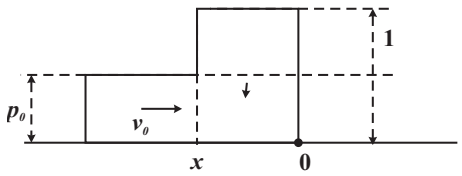


Figure 1: A cluster, decelerating at a cross-roads, C_{ld}

equality is true

$$x(t)\rho_0 + y(t) \times 1 = l_0\rho_0,$$

and the cluster simultaneous velocity is

$$v(t) = \frac{x(t)\rho_0 \times v_0 + y(t) \times 1 \times 0}{l_0\rho_0} = v_0\frac{x(t)}{l_0}.$$

Therefore velocity of the cluster, moving on the ring, is

$$v(t) = \frac{\rho_0 v_0 \frac{l_0\rho_0 - y(t)}{\rho_0} + 0}{l_0\rho_0} = v_0(1 - \frac{y(t) \times 1}{l_0 \times \rho_0}). \quad (1)$$

The average velocity v_c of a cluster, moving on the chainmail, is the cluster velocity averaged over all rings.

The average v_c^* over time is defined as

$$v_c^* = \lim_{t\to\infty} \frac{v_c(t)}{t},$$

if this limit exists.

If the cluster intersection has released, then the congestion, i.e. the decelerating cluster of density 1, begin to come nearer to the operating mode with the initial parameters, and the congestion is moving awhile in the direction opposite to the direction of cluster movement, Fig.2.

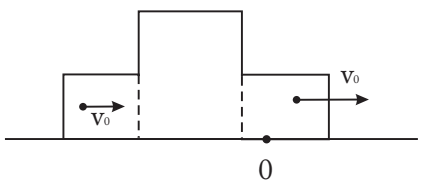


Figure 2: cluster, leaving the wait state, C_{ld}

 ISBN: 978-84-616-2723-3

(C_{ud}) *Cluster model with uniformly distributed information).* All particles react instantly to changing of movement conditions at the front boundary of the cluster. In this case the cluster velocity is constant and can have one of two values v_0 and 0.

Common mode of moving through the cross-roads is that *first comes — first moves, and the other is waiting.* In the case of simultaneous arriving to the cross-roads there two scenarios of cluster behavior.

(I_p) - *Priority mode.* It is given which of to adjacent rings is the priority ring. If the boundaries of two clusters come simultaneously to some vertex, then the cluster of the priority ring goes through the vertex.

(I_s)- *Random mode.* For each pair of clusters coming to the vertex, each cluster is chosen equiprobable.

3. Two rings necklace

Suppose the network consist of two rings, and there is a common point. Let us introduce the coordinate system for each circle, and origins of these systems is in the common point (vertex), Fig. 3. Suppose the initial length of the ith cluster is equal to $0 < l_i < 2\pi$, $i = 1, 2$. The densities of both the clusters are the same, and, therefore, these clusters are moving with the same velocity v_0.

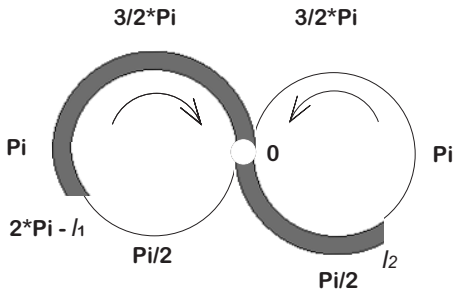


Figure 3: Two rings with a common point – necklace

Theorem 1. Suppose $C = C_{ld} \vee C_{ud}$, $I = I_p \vee I_s$.

1) If $l_1 + l_2 < 2\pi$, *then, for any initial conditions, a finite instant after that both the clusters move with constant velocity* v_0 *(synergy), and, therefore,*

$$v_c^* = v_0.$$

 ISBN: 978-84-616-2723-3

2) If $l_1 + l_2 > 2\pi$, then the movement is never synchronized, and the average velocity is

$$v_c^* = \frac{2\pi}{l_1 + l_2} v_0. \tag{2}$$

Proof.

1) Suppose $0 < l_1 + l_2 < 2\pi$. Some initial conditions exist such that each cluster comes to the vertex ever when the other cluster is not going through the vertex, i.e., at initial time the front boundary of the cluster 1 and the rear boundary of the cluster 2 are located at the vertex. In the case of this initial conditions, if one of the cluster comes to the vertex, and, at the same instant, the other cluster is going through the vertex, then since the instant at that the rear boundary of the moving cluster comes through the vertex, both the clusters ever come to the vertex when it is free. The first statement of Theorem 1 has been proved.

2) Suppose $l_1 + l_2 > 2\pi$.

Suppose the movement becomes synchronized. Then the portion of time during that the cluster i is going through the vertex is equal to $l_i/2\pi$, $i = 1, 2$. However, it is impossible because $l_i/2\pi + l_i/2\pi > 1$.

Suppose that one of the clusters comes to the vertex, while the other cluster is going through the vertex. Suppose, for example, the cluster 2 comes to the vertex. The front boundary of the cluster 2 begins to move as soon as the rear boundary of the cluster 1 is going through the vertex. In that instant, the remaining distance that the front boundary has to go to the vertex is $2\pi - l_2$. In the time interval of duration $(2\pi - l_1)/v_0$ the front boundary of the cluster 1 comes to the vertex, Fig. 5, but the rear vertex of the cluster 1 reaches the vertex in time interval of duration $(l_1 + l_2 - 2\pi)/v_0$, Fig. 4, after that the front boundary of the cluster 1 for the time interval of the same duration the front boundary of the cluster 1 is not moving. After that time intervals of duration $2\pi/v_0$ when the cluster moves with velocity v_0, and time intervals of duration $(l_1 + l_2 - 2\pi)/v_0$ when the front boundary of cluster does not move and is waiting at the vertex. The total duration of two such subsequent intervals is equal to $(l_1 + l_2)/v_0$. For this time the front boundary of the cluster 1 covers the distance equal to the circle length, i.e, the distance equal to 2π. Therefore, the average velocity of the cluster 1 is calculated with formula (1). Similarly, it is proved that the velocity of the cluster 2 is calculated also with formula (1).

Theorem 1 has been proved.

4. Clusters on a closed 2-necklace

Suppose there is a network consists of two circles, which have two common points (vertices).

On both the circles, coordinates of one of the vertex are equal to 0, and coordinates of the other vertex are equal to π, Fig. 4. Suppose the initial cluster i length equals l_i,

 ISBN: 978-84-616-2723-3

$0 < l_i < 2\pi$, $i = 1, 2$. Both the clusters have the same density, and, therefore, the clusters move with the same velocity v_0.

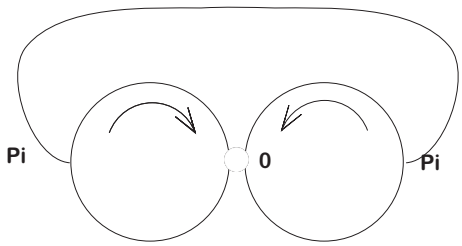


Figure 4: Closed necklace of two rings

The following Theorem is true that is analogous to Theorem 1.

Theorem 2. Suppose $C = C_{ld} \vee C_{ud}$, $I = I_p \vee I_s$.

1) If $l_1 + l_2 < 2\pi$, *then, for any initial conditions there exists a time instant after that the both clusters ever move with velocity* v_0, *and, therefore,* $v = v_0$.

2) If $l_1 + l_2 > 2\pi$, *then the movement cannot be synchronized and the average velocity is calculated with formula (2), if* $\min(l_1, l_2) < \pi$. . *3) If* $\min(l_1, l_2) > \pi$, *then the movement stops for a finite time,* $v \equiv 0$.

Proof. The proof of Theorem is similar to the proof of Theorem 2.

5. Flows on a $2N$-necklace

Let us consider a generalization of the model that was considered in Section 4. Suppose there are $2N$ rings. Each ring has common points (vertices) with two adjacent, Fig. 5. Each vertex has the same coordinates on the both rings. These coordinates are equal to 0 and π. Neighboring clusters have indices difference of that is equal to 1 (except neighboring rings 1 and $2N$, which have also a common point). By l_i denote the initial length of the cluster i, $i = 1, \ldots, 2N$.

Theorem 3. Suppose $C = C_{ud}$, $I = I_s$.

1) If $l_i = l < \pi$, $i = 1, \ldots, 2N$, *then, for any initial conditions there exists a time instant after that the both clusters move always with velocity* v_0, *and therefore, this is a case of flow synergy.*

2) If the total length of two adjacent rings is more than 2π, *then, for any initial con-*

 ISBN: 978-84-616-2723-3

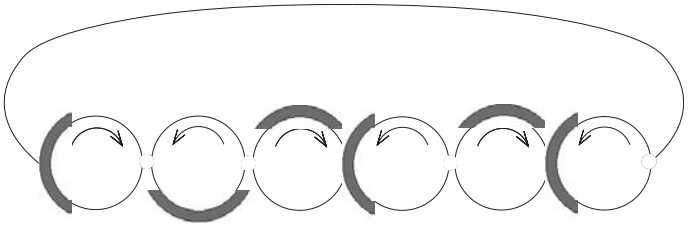


Figure 5: Closed $2N$-necklace

ditions, the average velocity v_c^* of the clusters is less than v_0, i.e., the movement cannot be synchronized.

3) If $\min(l_1, \dots, l_{2N}) > \pi$, then the movement stops for a finite time, $v \equiv 0$.

Suppose that $\theta_k(x,t) = 1$, if the point x on the ring k at time t is located within the cluster; $\theta_i(x,t) = 0$, otherwise;

$$I_k(t) = \int_0^{2\pi} \sum_1^{2N} \theta_i(x,t)\theta_{i+1}(x,t)\,dx,\ \ k = 1, \dots, 2N-1,$$

$$I_{2N}(t) = \int_0^{2\pi} \sum_1^{2N} \theta_{2N}(x,t)\theta_1(x,t)\,dx,$$

$$I(t) = \sum_{k=1}^{2N} I_k.$$

The movement is synchronized, if, since some time instant, $I(t) = 0$..

Lemma 1. Suppose $C = C_{ud}$. If $l_i < \pi$ for any $i = 1, \dots 2N$, then $I(t)$ is the non-increasing function of time.

Proof. The item $I_k(t)$, $k = 2, \dots, 2N-1$, in the sum $I(t) = \sum\limits_{k=1}^{2N} I_k$ can increase at time instant t, if at this instant at least one of the clusters k and $k+1$ is not moving. Suppose, for example, it is the cluster k. However, the item I_{k-1} decreases with the same velocity as increases the item I_k, and the sum I is non-increasing. The cases $k = 1$ and $k = 2N$ are considered similarly.

Lemma 1 has been proved.

Lemma 2 Suppose $C = C_{ud}$. If $l_i > \pi$ for any $i = 1, \dots 2N$, then the movement stops for a finite time.

 ISBN: 978-84-616-2723-3

Proof. There are $2N$ clusters and $2N$ vertices. We have $l_i > \pi$, $i = 1, \dots 2N$. Hence any cluster covers at least a vertex, and no cluster can cover more than one vertex. The front boundary of no cluster can enter a vertex, because such a cluster covered two vertices, and this is impossible.

Lemma 2 has been proved.

6. Flows on chainmail of a torus

Consider a network that has two-dimensional structure 2×2. There are four rings: $(1,1)$, $(1,2)$, $(2,1)$, and $(2,2)$. Points of each ring are characterized with a coordinate on the circle. Each ring has common points (vertices) with four rings. Hence the structure of the network is a torus structure. The location of vertices is shown in Fig. 6. The clusters have the same mass l, $(1, 2, \frac{\pi}{2})$;

The rings $(1,1)$ and $(2,2)$ are priority rings.

Theorem 4 *Suppose $C = C_{ud}$, $I = I_s$.*

1) If $l < \frac{\pi}{2}$, then, for any initial conditions, synergy occurs after some finite time interval.

2) If $\frac{\pi}{2} < l < \pi$, then, for any density, there exist both initial conditions for that the movement is synchronized and initial conditions for that the movement cannot be synchronized.

If $l > \pi$, then the flow stops for a finite time.

The movement is synchronized, for example, if at initial time, on the rings $(1,1)$ and $(2,2)$ the front boundaries are located at the vertices with coordinate 0, and on the rings $(1,2)$ and $(2,1)$ the front boundaries are located at the vertices with coordinate π.

The movement is synchronized, if

$$x_{11} = 0,\ x_{21} = \frac{\pi}{2},\ x_{22} = \pi, x_{12} = \frac{3\pi}{2},$$

where x_{ij} is the coordinate of the front boundary of cluster on the ring (i,j), $i,j = 1,2$.

In Fig. 6–10, the networks states are shown at some times, for $l = \frac{11}{18}\pi$, $\rho = \frac{1}{2}$, and the velocity of free movement $v_0 = \frac{2\pi}{360}$.

 ISBN: 978-84-616-2723-3

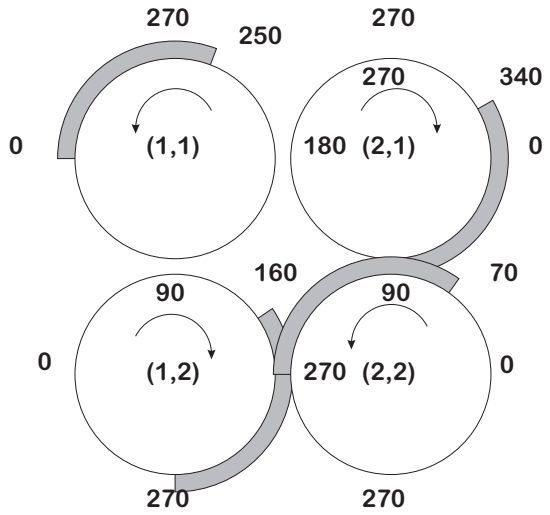


Figure 6: Network state at $t = 0$

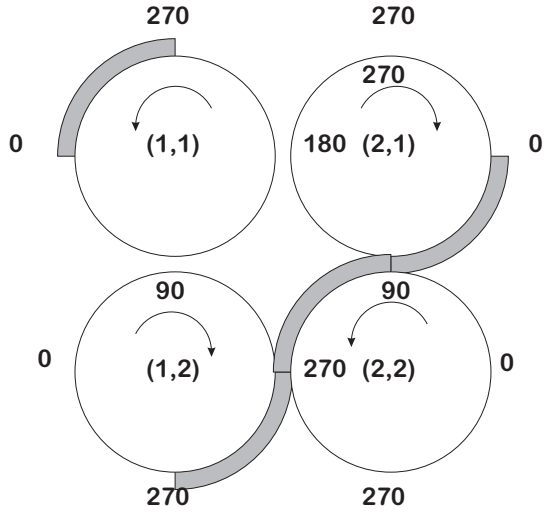


Figure 7: Network state at $t = 20$

In this case the average velocity of the cluster movement on each ring is equal to

$$v = \frac{2\pi}{2l + \pi} v_0.$$

 ISBN: 978-84-616-2723-3

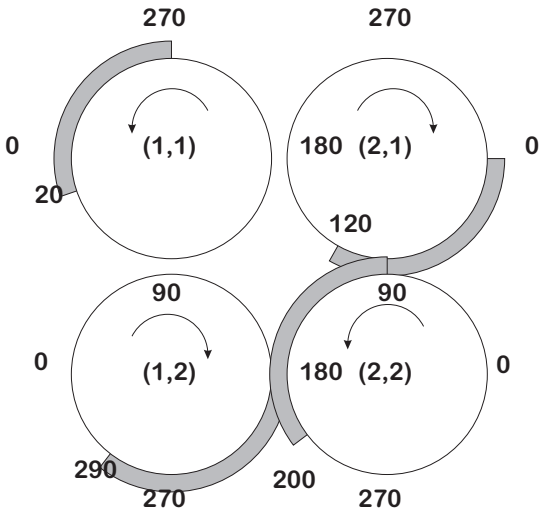


Figure 8: Network state at $t = 40$

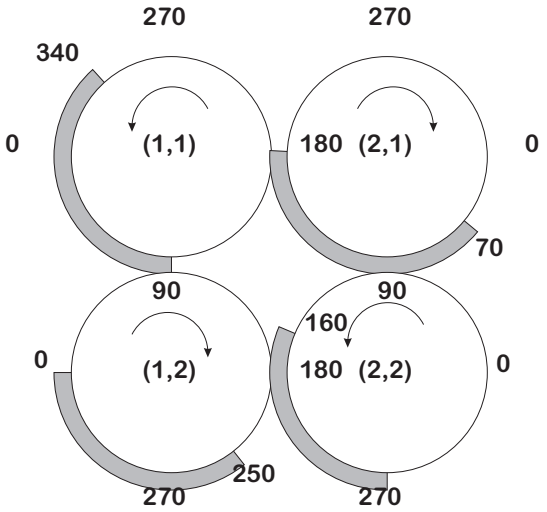


Figure 9: Network state at $t = 110$

7. Flows without synergy

Consider a chain of four rings connected at points with coordinates 0 and π, Fig. 9.

On each ring a cluster is moving of length π. If two clusters come to the vertex simul-

 ISBN: 978-84-616-2723-3

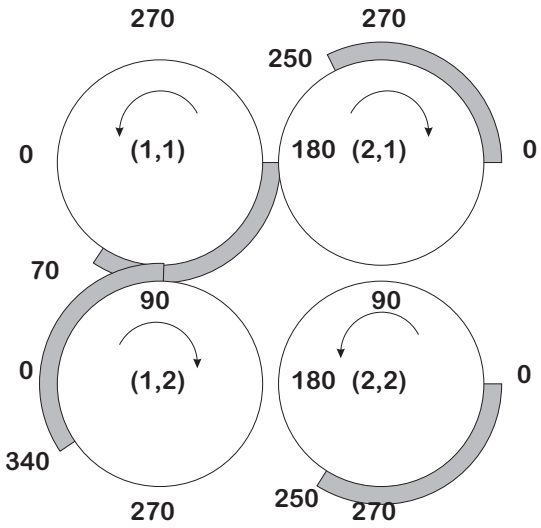


Figure 10: Network state at $t = 200$

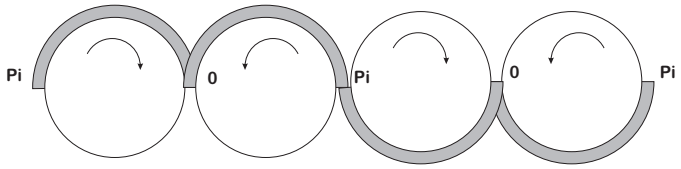


Figure 11: Location of clusters at the initial time instant

taneously, then the cluster located to the right has the priority. This corresponds to case in that, for each vertex, the priority cluster is assigned. The velocity of free cluster movement equals v_0.

Suppose, at the initial time instant, the front boundaries of clusters 1 and 2 are located at vertices with coordinates 0, and the front boundaries of clusters 3 and 4 are located at vertices with coordinates π.

Then for time interval of the length π/v_0, the clusters 2, 3 and 4 go through a distance π, and the cluster 1 is not moving, Fig. 10.

In time interval $(\pi/v_0, 2\pi/v_0)$, all clusters are moving save the cluster 3.

At time $t = 2\pi/v_0$ the front boundaries of clusters 1 and 4 are located at vertices with coordinates π, and the front boundaries of clusters 2 and 3 are located at vertices with coordinates 0, Fig. 11. The situation is repeated that is similar to initial on. A similar situation will be repeated in time intervals of duration $2\pi/v_0$. The movement is not synchronized.

 ISBN: 978-84-616-2723-3

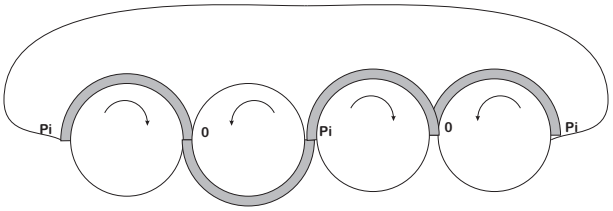


Figure 12: Location of clusters for $t = \pi/v_0$

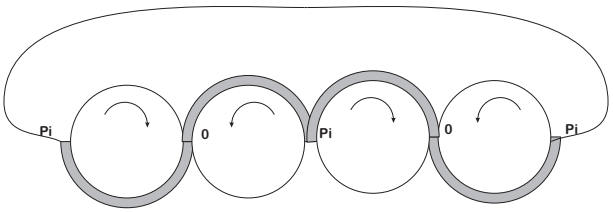


Figure 13: Location of clusters for $t = 2\pi/v_0$

8. Simplified stochastic model of flows on a necklace

Suppose there are $2N$ rings. Each ring has common points (vertices) with two adjacent rings. Neighboring clusters have indices difference of that is equal to 1. The adjacent rings 1 and $2N$ have also a common point. The vertices on the same ring are located at opposite edges of diameter, Fig. 12. The rings are numerated to the right.

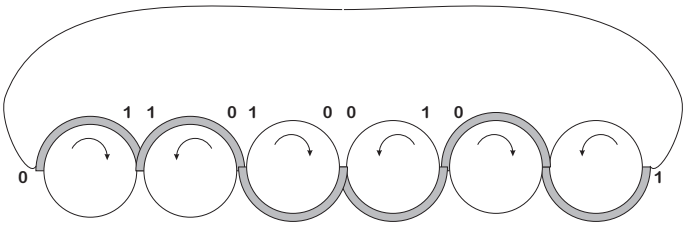


Figure 14: Clusters on a necklace

On each ring clusters move. The length of each cluster is equal to a half circle. At each discrete time instant (tact) each cluster occupies the upper part of its circle or the lower one. We say that the ring with an odd index is at the state 1, if the cluster occupies the upper part of the ring, and is at the state 2, if the cluster occupies the lower part of the ring. The ring with an even index is at the state 1, if the cluster occupies the lower part of the ring, and is at the state 2, if the cluster occupies the upper part of the ring. The front boundary of cluster with an odd number is located to right, if the cluster occupies

 ISBN: 978-84-616-2723-3

the upper part of the ring, and to left, if the cluster occupies the lower part of the ring. The front boundary of cluster with an even number is located to left, if the cluster occupies the upper part of the ring, and to right, if the cluster occupies the lower part of the ring. Clusters moves clockwise on the rings with odd numbers and counter-clockwise on the rings with even numbers. At each tact, each cluster changes occupying part save the case when the front boundary of the given cluster is located at the same vertex as the front boundary of the neighboring cluster. In the latter case each of two clusters comes to the other part of the circle with probability 1/2 and with the same probability the cluster does not move at given tact.

Theorem 5. For the time interval with a time interval with a finite expectation all rings come or to the state 1, or to the state 2, after that no conflicts occur and the clusters move freely and are in turns at states 1 and 2.

Proof. Let us describe the clusters movement process by a Markov chain [5], states of which correspond to configurations of clusters on the network of wings. There are 2^{2N} the chain states. By $(i_1, \ldots, i_{2N})$ denote the chain state for that the ring j is at state i_j, $i_j = 1, 2$; $j = 1, \ldots, N$. In accordance with rules of movement, each cluster always changes its location at every step save cases when this ring is at the state 1 and the ring to right is at the state 2, or when the given ring is at the state 2 and the ring to left is at the state 2, Fig. 13.

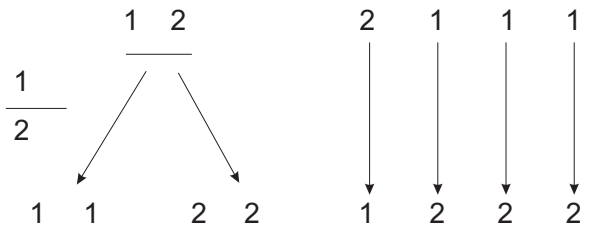


Figure 15: Rules of clusters movement

This means that, if, at some time, the chain comes to the state $E_1 = (1, 1, \ldots, 1)$ or to the state $E_2 = (2, 2, \ldots, 2)$, then, after this, the chain will be by turns at each of these states. If each cluster of the chain is in turns at odd steps at the state E_1, and at even steps at the state E_2, then we say that the chain is at the mode 1. If each cluster of the chain is in turns at even steps at the state E_1, and at odd steps at the state E_2, then we say that the chain is at the mode 2. Let us prove that the chain come from any state come to the mode 1 no more for N steps. We say that a ring, at an odd time instant is at the state corresponding to the mode 1 (corresponds to the mode 1), if this ring, at this instant, is at the state 1. We say that a ring, at an even time instant is at the state corresponding

 ISBN: 978-84-616-2723-3

to the mode 1, if this ring, at this instant, is at the state 2. Suppose the chain is at the state k rings are at the states do not correspond to the mode 1. If the chain is at the state neither E_1 nor E_2, then there is at least a pair of adjacent rings such that the left ring is at state 1 and the right ring is at state 2. One ring of such the pair correspond to the mode 1, and the other ring does not correspond to the mode 1. With probability 1/2, the ring corresponding to the mode 1, moves at the present step, and the other ring does not move after that both the clusters will correspond to the mode 1. With a probability, which is not less than $1/2^{2k}$, at each of next k steps corresponding to the mode 1 will increase unless all rings correspond to the mode 1. Since $k \leq 2N$, then, with probability no less $1/2^{3N}$, the chain, from any state, for no more than N steps, comes to the mode 1. Similarly, it can be proved that, with probability no less $1/2^{3N}$, the chain, from any state, for no more than N steps, comes to the mode 2. Hence, with probability no less $1/2^{3N-1}$, the chain, from any state, for no more than N steps, comes to the mode 1 or to the mode 2. Thus the chain, from any state, comes to mode 1 or to the mode 2 for the time interval expectation of that is no more thanNd, where d is the expectation of a random value distributed geometrically with the parameter $p = 1/2^{3N-1}$. This expectation equals 2^{3N-1}.

Theorem 5 Has been proved.

Acknowledgements

This work has been supported by grants of RFBR No. 12-01-00974-a and No. 11-07-00622-a.

References

[1] Kozlov V.V., Buslaev A.P., *Metropolis Traffic Modeling: from intelligent monitoring trough physical representation to mathematical problems*, Proc. of Int. Conf. CMMSE V.1, (2012) 750 –756.

[2] Bugaev A.S., Buslaev A.P., Kozlov V.V., Yashina M.V. *Distributed Problems of Monitoring and Modern Approaches to Traffic Modelling*, 14-th International IEEE Conference on Intelligent Transportation Systems (ITSC 2011), Washington, USA, 5-7.10.2011. DOI:10.1109/ITSC. 20116082805. (2011), 447 - 481

[3] Buslaev A.P., Tatashev A.G., and Yashina M.V. *Cluster flow models and properties of appropriate dynamic systems.* Journal of Applied and Functional (JAFA), (2013), Vol. 8, No. 1, 54–76.

[4] Kozlov V.V., Buslaev A.P., Tatashev A.G. *Monotonic random walks and clusters flows on networks. Models and applications.* Lambert Academic Publishing, 2013.

[5] Karlin S. *A first course in stochastic processes.* New York and London, 1968.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Efficient Implementation of Newton Solver for the Finite Element Method

Pavel Kůs[1], Lukáš Korous[1] and Pavel Karban[1]

[1] *Department of Theory of Electrical Engineering, University of West Bohemia in Pilsen*

emails: pkus@rice.zcu.cz, korous@rice.zcu.cz, karban@kte.zcu.cz

Abstract

The use of Newton method is a well-established option for solving nonlinear partial differential equations. The theory of its convergence is, however, rather complicated and the results are restricted to simple academic examples. The matter is even more complicated due to necessity of use of damping, attempts to save computation time by using the same Jacobian for several successive steps, or other variants of the algorithm. In the case of solution of complicated problems arising in the engineering practice, one usually have to fine-tune the method for each problem individually in order to ensure convergence and avoid extensively long calculation. The goal of this contribution is to develop wide range of variants of the Newton method, test them on several real-world engineering problems and compare their performance. Resulting algorithms, which were implemented in the frame of our general-purpose finite element software, allow us to automatically select appropriate settings for the given problem.

Key words: Nonlinear partial differential equations, Newton solver, higher-order finite element method, numerical software

1 Introduction

In this work, we would like to address the issue of effective implementation of the Newton method, used for solution of partial differential equations. It is a part of ongoing effort to create universal software for solution of nonlinear coupled partial differential equations, which would use the most advanced approaches towards the finite element method, but, in the same time, would be easily usable for engineers performing complicated real-world calculations.

2 Hermes and Agros2D software

Hermes (see, e.g., [3], [5], [6]) is a C++ library for rapid development of adaptive *hp*-FEM and *hp*-DG solvers, with emphasis on nonlinear, time-dependent, multi-physics problems. It implements several unique features, that can speed-up the calculation significantly. It has been shown, that *hp*-adaptivity can lead to exponential convergence. The use of arbitrary-level hanging nodes significantly reduces the need of unnecessary refinements during the adaptivity process. Multi-mesh approach allows us to use separate mesh for each field, respecting its specific requirements, but creating one discrete problem. Other key features, such as curvilinear elements or dynamical meshes, are described on the web page of the Hermes project [7].

Agros2D (see, e.g., [4], [8]) is a multi-platform engineering software for solution of nonlinear coupled problems from large variety of engineering practice. Its goal is to allow the use of advanced algorithms implemented in the Hermes library in a convenient way. It brings advanced GUI, with which it is simple to develop complicated models including interaction of several physical fields. As it uses Hermes library as its computational core, it brings all its unique features, such as automatic *hp*-adaptivity with arbitrary-level hanging nodes, multi-mesh assembling, curvilinear elements, etc. Agros2D comprises convenient pre-processor for geometry definition, meshing tools, FEM calculation (using the Hermes library) and advanced post-processor, defining large variety of point and integral physical quantities for each individual physical field.

3 Selected nonlinear problems

Our work is focused at solution of nonlinear coupled problems, comprising interaction of more physical fields. In this work, we focus on the study of three nonlinear problems arising in three different areas, which may be then used to formulate coupled problems.

3.1 Temperature distribution

Problems of temperature distribution T are described by equation

$$-\nabla \cdot (\lambda \nabla T) + \rho c_{\mathrm{p}} \Big(\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \Big) = Q,$$

where thermal conductivity λ, specific heat c_{p} and density ρ may be considered nonlinear.

 ISBN: 978-84-616-2723-3

3.2 Magnetic field

The distribution of magnetic field will be calculated in terms of vector magnetic potential A, given by equation

$$\nabla \times \Big(\frac{1}{\mu}(\nabla \times A - B_{\mathrm{r}})\Big) - \sigma \mathbf{v} \times \nabla \times A = J_{\mathrm{ext}},$$

where the permeability μ exhibits a very strong nonlinear dependence on magnetic induction $B = \nabla \times A$ and causes convergence problems.

3.3 Incompressible flow

Equations governing incompressible flow are as follows:

$$\begin{aligned} \rho \mathbf{v} \cdot \nabla \mathbf{v} &= -\nabla p + \mu \triangle \mathbf{v} + \mathbf{f} \\ \nabla \cdot \mathbf{v} &= 0 \end{aligned}$$

Those equations are nonlinear by its nature (the term on the left-hand side of the first equation) and also its coefficients have to be considered nonlinear in some cases.

In the case of coupled problems, the situation is even more complicated due to mutual dependence of nonlinear parameters on individual fields. Such problems can be solved in monolithic formulation using the Hermes multi-mesh capability.

4 Newton method

Various approaches towards the Newton method can be found in a large number of sources (see, e.g., [1], [2]). The main focus of this paper is to implement different variants (including the use of automatic or fixed damping factor, different strategies of reusing of the Jacobian for several successive steps, different stopping criteria, etc.), to compare their performance in terms of convergence and CPU time and to identify optimal parameters for each type of equation. The basic algorithm can be written as follows:

```
initial step
while stopping criterion not reached
  recalculate jacobian
  solve the linear problem
  while condition on residual not satisfied
    update damping factor
    calculate residual
  end
  while condition on jacobian reuse satisfied
```

 ISBN: 978-84-616-2723-3

```
    solve the linear problem
    calculate residual
end
```

In the presentation, extensive comparisons of different choices of stopping criterion, conditions influencing the calculation of damping factor and strategies for Jacobian reuse and other parameters will be provided. The comparisons will be done for all mentioned equations with the aim to allow more effective calculations.

Acknowledgements

This work was supported by the European Regional Development Fund and Ministry of Education, Youth and Sports of the Czech Republic (project No. CZ.1.05/2.1.00/03.0094: Regional Innovation Center for Electrical Engineering - RICE) and by the project P102/11/0498 (Grant Agency of the Czech Republic).

References

[1] P. Wriggers, *Nonlinear Finite Element Methods*, Springer-Verlag, Berlin Heidelberg, 2010.

[2] P. Deuflhard, *Newton Methods for Nonlinear Problems*, Springer Series in Computational Mathematics, **35** (2004).

[3] P. Solin, K. Segeth and I. Dolezel, *Higher-Order Finite Element Methods*, Chapman & Hall/CRC Press, Boca Raton, 2004.

[4] P. Karban, F. Mach, P. Kus, D. Panek and I. Dolezel, *Numerical solution of coupled problems using code Agros2D*, Computing In press (2013) DOI 10.1007/s00607-013-0294-4.

[5] P. Solin, J. Cerveny, I. Dolezel, *Arbitrary-Level Hanging Nodes and Automatic Adaptivity in the hp-FEM*, Math. Comput. Simul. **77** (2008), 117–132.

[6] P. Solin, L. Korous, *Adaptive Higher-Order Finite Element Methods for Transient PDE Problems Based on Embedded Higher-Order Implicit Runge-Kutta Methods*, J. Comput. Physics, accepted (2011).

[7] http://www.hpfem.org/hermes

[8] http://www.agros2d.org

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

A Stabilized Finite Volume Numerical Scheme for Solving the Partial Differential Equation in the Heston Model

Pavol Kútik[1] **and Karol Mikula**[1]

[1] *Department of Mathematics and Descriptive Geometry, Slovak University of Technology*

emails: pavol.kutik@gmail.com, karol.mikula@gmail.com

Abstract

This article is aimed to provide a stable numerical scheme, based on the finite volume method discretization, for solving the partial differential equation arising in the Heston model. In order to build a scheme which does not violate the discrete minimum-maximum principle a diamond–cell approximation of the gradient is used and a splitting into an inflow/implicit and an outflow/explicit part is applied. By the use of appropriate weights, a sufficiently large set of the outflow fluxes are transfered to the corresponding inflow fluxes of the neighbouring finite volumes. We illustrate the stability and accuracy of the scheme on two numerical experiments, the former one with a European binary option and the latter one with a European call option.

Key words: Finite volume method, Heston model, Stabilized scheme

1 Introduction

Since its introduction in 1973 the Black-Scholes formula (cf. [1]) has remained the most widely used application for pricing of European call options. However, as Hull and White (cf. [8]), Heston (cf. [9]) and many others point out, some of the underlying assumptions which determine the stock price dynamics, have to be modified in order to replicate the option prices more accurately. Particularly, Heston relaxes the constant variance assumption and allows it to follows the so-called mean reverting square root process originally proposed by Cox, Ingersoll and Ross in [3]. Hence the couple of stochastic differential equations which govern the price evolution looks as

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t, \tag{1}$$

$$dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v_t} dZ_t \tag{2}$$

ISBN: 978-84-616-2723-3

where $\{S_t\}_{t\geq 0}$ and $\{v_t\}_{t\geq 0}$ are stochastic processes for the underlying stock and the variance of the underlying, respectively. Processes $\{W_t\}_{t\geq 0}$ and $\{Z_t\}_{t\geq 0}$ are Wiener stochastic processes mutually correlated by $\mathbb{E}[dW_t dZ_t] = \rho dt$. The list of other parameters reads as follows:

(i) ρ is the correlation parameter,

(ii) κ is the reversion speed,

(iii) σ is the volatility of variance,

(iv) θ denotes the long-term variance,

(v) μ represents the drift of the process for the stock.

Modeling the asset price by means of (1)-(2) allows to capture important skewness effect of the spot returns that arise from the mutual correlation $|\rho| > 0$ between the two Wiener processes. Now, if we follow the hedging procedure of a synthetic portfolio, omit the subscripts t indicating time dependence in S_t and v_t and assume zero market price of risk we can write the governing partial differential equation as follows:

$$\begin{aligned}&\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \rho\sigma vS\frac{\partial^2 V}{\partial S \partial v}\\ &+\frac{1}{2}\sigma^2 v\frac{\partial^2 V}{\partial v^2} + \kappa(\theta - v)\frac{\partial V}{\partial v} - rV = 0\end{aligned} \tag{3}$$

where $r > 0$ is the risk-free interest rate. What is more, the Heston model provides a closed-form solution (cf. [9]) for a European call option which we later take as a benchmark when testing the accuracy of our scheme.

2 Numerical Schemes

In order to obtain an efficient numerical scheme we apply the finite volume discretization method to the linear advection-diffusion-reaction equation of the form (3). Before we do so, it is convenient to transform the governing equation into a form with the diffusion and advection term in divergent form. To this end we first use standard substitutions $x = \ln\left(\frac{S}{E}\right), y = v, \tau = T - t$ and $u(x, y, \tau) = V(S, v, t)$ and rewrite equation (3) in a compact form as follows

$$\frac{\partial u}{\partial t} + \vec{A}\cdot\nabla u = \nabla\cdot(\mathbf{B}\nabla u) - ru, \tag{4}$$

where

$$\mathbf{B} = \frac{1}{2}y\begin{pmatrix} 1 & \rho\sigma \\ \rho\sigma & \sigma^2 \end{pmatrix} \tag{5}$$

 ISBN: 978-84-616-2723-3

and

$$\vec{A} = -\begin{pmatrix} r - \frac{1}{2}y - \frac{1}{2}\rho\sigma \\ \kappa(\theta - y) - \frac{1}{2}\sigma^2 \end{pmatrix}. \tag{6}$$

If we shrink the computational domain, for implementation purposes, to $\Omega \approx (X_l, X_r) \times (0, Y)$ and denote the uniform lengths of each rectangular finite volume in the x- and y-direction by

$$\begin{aligned} h_x &= \frac{X_r - X_l}{N_x}, \\ h_y &= \frac{Y}{N_y} \end{aligned}$$

we can define the admissible mesh $\mathcal{T}_h$ in the sense of [5] of the domain $\Omega \subset \mathbb{R}^2$ by

$$p_{ij} = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}), \text{ for } i = 1, \ldots, N_x \text{ and } j = 1, \ldots, N_y. \tag{7}$$

The boundaries of each finite volume (cell) p_{ij} are defined by the set of points

$$\begin{aligned} &x_{\frac{1}{2}} = X_l, \; x_{i+\frac{1}{2}} = x_{i-\frac{1}{2}} + h_x, \text{ for } i = 1, \ldots, N_x, \\ &y_{\frac{1}{2}} = 0, \; y_{j+\frac{1}{2}} = y_{j-\frac{1}{2}} + h_y, \text{ for } j = 1, \ldots, N_y. \end{aligned}$$

Concerning the time discretization, we use uniform discrete time step $\kappa = \frac{T}{N_{ts}}$ in order to discretize the time interval $I = (0, T)$.

Before integrating equation (4) let us rewrite also the advection term into conservative and nonconservative part as follows:

$$\vec{A} \cdot \nabla u = \nabla \cdot (\vec{A}u) - (\nabla \cdot \vec{A})u. \tag{8}$$

Inserting this identity into (4) and integrating it over a finite volume p we obtain its following integral form:

$$\int_p \frac{\partial u}{\partial \tau} \mathrm{d}x + \int_p [\nabla \cdot (\vec{A}u) - (\nabla \cdot \vec{A})u] \mathrm{d}x = \int_p \nabla \cdot (\mathbf{B}\nabla u) \mathrm{d}x - \int_p ru \mathrm{d}x. \tag{9}$$

On the terms $\nabla \cdot (\vec{A}u)$ and $\nabla \cdot (\mathbf{B}\nabla u)$ we can apply Green's theorem to obtain

$$\begin{aligned} &\int_p \frac{\partial u}{\partial \tau} \mathrm{d}x + \sum_{q \in N(p)} \int_{\sigma_{pq}} \vec{A}u \cdot \mathbf{n}_{pq} \, \mathrm{d}\gamma - \int_p (\nabla \cdot \vec{A})u \mathrm{d}x = \\ &\sum_{q \in N(p)} \int_{\sigma_{pq}} \mathbf{B}\nabla u \cdot \mathbf{n}_{pq} \, \mathrm{d}\gamma - \int_p ru \mathrm{d}x \end{aligned} \tag{10}$$

where σ_{pq} represents a mutual edge between finite volumes p and q. Furthermore, the symbol $\mathbf{n}_{pq}$ denotes the unit outer normal vector relative to the finite volume p and $N(p)$

 ISBN: 978-84-616-2723-3

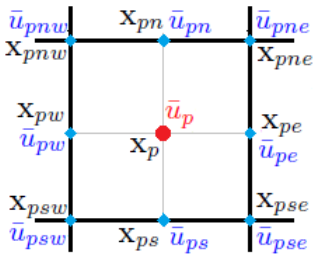


Figure 1: A detail of the finite volume p and corresponding adjacent points with representative solution values $\bar{u}_{pq}$.

is the set of all cells which have a common edge with the cell p, i.e. $N(p) = \{e, w, n, s\}$ (see Figure 1).

Let us denote by v_{pq} the approximation of the averaged exact advection velocity on the face σ_{pq} in the inward normal direction to the finite volume p, i.e.

$$v_{pq} \approx -\frac{1}{m(\sigma_{pq})} \int_{\sigma_{pq}} \vec{A} \cdot \mathbf{n}_{pq} \, \mathrm{d}\gamma \tag{11}$$

where $m(\sigma_{pq})$ denotes the measure of a mutual face between cells p and q. Hence for each edge we can write $v_{pq} = -\vec{A}_{pq} \cdot \mathbf{n}_{pq}$ where $\vec{A}_{pq}$ denotes the mean value of the advection vector along σ_{pq}. Particularly, for each edge σ_{pq}, the velocity approximation v_{pq} can be formulated as

$$v_{pn} = -a^2_{pn}, \;\; v_{pw} = a^1_{pw}, \;\; v_{pe} = -a^1_{pe}, \;\; v_{ps} = a^2_{ps} \tag{12}$$

where a^1_{pq}, a^2_{pq} are the mean scalar elements of the advection velocity vector $\vec{A} = \begin{pmatrix} a^1 \\ a^2 \end{pmatrix}$ evaluated along the edge σ_{pq}.

In order to approximate the exact gradient ∇u along the edge σ_{pq} let us introduce a discrete diamond-cell gradient operator $\nabla^{DC} = (\nabla^{DC}_x, \nabla^{DC}_y)$ which we define for each edge $\sigma_{pq}, \;\; q \in N(p)$ as follows (cf. [4]):

$$\nabla^{DC}_{pe}(u) = \begin{pmatrix} \frac{\bar{u}_e - \bar{u}_p}{h_x} \\ \frac{\bar{u}_{pne} - \bar{u}_{pse}}{h_y} \end{pmatrix}, \;\; \nabla^{DC}_{pw}(u) = \begin{pmatrix} \frac{\bar{u}_p - \bar{u}_w}{h_x} \\ \frac{\bar{u}_{pnw} - \bar{u}_{psw}}{h_y} \end{pmatrix}, \tag{13}$$

$$\nabla^{DC}_{pn}(u) = \begin{pmatrix} \frac{\bar{u}_{pne} - \bar{u}_{pnw}}{h_x} \\ \frac{\bar{u}_n - \bar{u}_p}{h_y} \end{pmatrix}, \;\; \nabla^{DC}_{ps}(u) = \begin{pmatrix} \frac{\bar{u}_{pse} - \bar{u}_{psw}}{h_x} \\ \frac{\bar{u}_p - \bar{u}_s}{h_y} \end{pmatrix} \tag{14}$$

where $\bar{u}_{pq}, \;\; q \in N(p)$ denotes a representative value on the edge σ_{pq}. Furthermore, by defining $N'(p) \backslash N(p)$ as a set which contains all cells q having a common point with the

 ISBN: 978-84-616-2723-3

finite volume p, i.e. $N'(p)\backslash N(p) = \{ne, nw, se, sw\}$ (see Figure 1), we can define $\bar{u}_{pq},\ q \in N'(p)\backslash N(p)$ as a representative value in the vertex $\mathbf{x}_{pq},\ q \in N'(p)\backslash N(p)$. Analogically, $\bar{u}_p$ denotes a representative value in the finite volume p. In what follows we assume the representative points to fulfill the most natural choice for reconstructions

$$\bar{u}_p^m = u_p^m, \qquad \bar{u}_{pq}^m = \frac{u_p^m + u_q^m}{2}, \quad \text{if } q \in N(p), \tag{15}$$

$$\bar{u}_{pne}^m = \frac{u_p^m + u_e^m + u_{ne}^m + u_n^m}{4}, \qquad \bar{u}_{pnw}^m = \frac{u_p^m + u_n^m + u_{nw}^m + u_w^m}{4}, \tag{16}$$

$$\bar{u}_{psw}^m = \frac{u_p^m + u_w^m + u_{sw}^m + u_s^m}{4}, \qquad \bar{u}_{pse}^m = \frac{u_p^m + u_s^m + u_{se}^m + u_e^m}{4} \tag{17}$$

where m denotes the old time layer $m = n-1$ or the new time layer $m = n$. Using the following approximations for the advection terms

$$\int_{\sigma_{pq}} \vec{A}u \cdot \mathbf{n}_{pq}\,\mathrm{d}\gamma \approx \bar{u}_{pq} \int_{\sigma_{pq}} \vec{A} \cdot \mathbf{n}_{pq}\,\mathrm{d}\gamma, \tag{18}$$

$$\int_p (\nabla \cdot \vec{A})u\mathrm{d}x \approx \bar{u}_p \int_{\sigma_{pq}} \vec{A} \cdot \mathbf{n}_{pq}\,\mathrm{d}\gamma \tag{19}$$

and inserting the approximate advection velocity (12) and the diamond-cell approximation of the gradient (13)-(14) into the equation (10) we obtain

$$\int_p \frac{\partial u}{\partial \tau}\mathrm{d}x + \sum_{q \in N(p)} v_{pq}\,(\bar{u}_p - \bar{u}_{pq})\,m(\sigma_{pq}) =$$
$$\sum_{q \in N(p)} \begin{pmatrix} b_{pq}^{11}\nabla_{x,pq}^{DC}(u) + b_{pq}^{12}\nabla_{y,pq}^{DC}(u) \\ b_{pq}^{21}\nabla_{x,pq}^{DC}(u) + b_{pq}^{22}\nabla_{y,pq}^{DC}(u) \end{pmatrix} \cdot \mathbf{n}_{pq}\,m(\sigma_{pq}) - \bar{u}_p \int_p r\mathrm{d}x. \tag{20}$$

To simplify the notation of the numerical schemes to come we introduce advection-associated coefficients $a_{pq}, q \in N(p)$ and diffusion-associated coefficients $b_{pq}, q \in N'(p)$ which we define as follows:

$$a_{pe} = -\frac{1}{2}h_y a_{pe}^1, \quad a_{pw} = \frac{1}{2}h_y a_{pw}^1, \quad a_{pn} = -\frac{1}{2}h_x a_{pn}^2, \quad a_{ps} = \frac{1}{2}h_x a_{ps}^2, \tag{21}$$

$$b_{pe} = \frac{h_y}{h_x}b_{pe}^{11} + \frac{b_{pn}^{21}}{4} - \frac{b_{ps}^{21}}{4}, \quad b_{pw} = \frac{h_y}{h_x}b_{pw}^{11} - \frac{b_{pn}^{21}}{4} + \frac{b_{ps}^{21}}{4}, \tag{22}$$

$$b_{pn} = \frac{h_x}{h_y}b_{pn}^{22} + \frac{b_{pe}^{12}}{4} - \frac{b_{pw}^{12}}{4}, \quad b_{ps} = \frac{h_x}{h_y}b_{ps}^{22} - \frac{b_{pe}^{12}}{4} + \frac{b_{pw}^{12}}{4}, \tag{23}$$

$$b_{pne} = \frac{b_{pe}^{12}}{4} + \frac{b_{pn}^{21}}{4}, \quad b_{psw} = \frac{b_{pw}^{12}}{4} + \frac{b_{ps}^{21}}{4}, \tag{24}$$

$$b_{pnw} = -\frac{b_{pw}^{12}}{4} - \frac{b_{pn}^{21}}{4}, \quad b_{pse} = -\frac{b_{pe}^{12}}{4} - \frac{b_{ps}^{21}}{4}. \tag{25}$$

 ISBN: 978-84-616-2723-3

Now, if we insert the reconstructions(15)-(17) into (20) with $m = n$ and replacing the time derivative $\frac{\partial u}{\partial \tau}$ by a backward difference $\frac{u_p^n - u_p^{n-1}}{k}$ in a representative point $\mathbf{x}_p$ we obtain the *basic diamond-cell-based fully-implicit scheme*:

$$(1 + kr)u_p^n + \frac{k}{m(p)} \sum_{q \in N'(p)} c_{pq} \left(u_p^n - u_q^n\right) = u_p^{n-1} \tag{26}$$

where

$$\begin{aligned} c_{pq} &= a_{pq} + b_{pq}, \quad && q \in N(p), \\ c_{pq} &= b_{pq}, && q \in N'(p) \backslash N(p). \end{aligned}$$

and $m(p) = h_x h_y$ is the measure of any finite volume p. Following the ideas in [2, 4] we are able to prove that for $h_x > 0, h_y > 0$ sufficiently small an unique solution of the scheme (26) does exist and it is conditionally stable in $L_2(I, \Omega)$ norm. Similarly, it may be also possible to prove the convergence of the discrete numerical solution to the weak solution of the Heston model (3) accompanied by some appropriate boundary conditions. However, since for some $h_x > 0, h_y > 0$ the solution delivered by the numerical scheme (26) may not always exist and if it does it is only conditionally stable our further goal is to enhance these properties.

In order to obtain a scheme which is always solvable our goal is to construct a non-singular system matrix. We can take advantage of the Levy-Desplanques theorem (cf. [7]) which states that a strictly diagonally dominant matrix is regular. Such matrix can be built by splitting the coefficients c_{pq} into two groups as follows

$$c_{pq}^{in} = \max(c_{pq}, 0) \quad \text{and} \quad c_{pq}^{out} = \min(c_{pq}, 0) \tag{27}$$

where c_{pq}^{in} denote the so-called *inflow* coefficients and c_{pq}^{out} the so-called *outflow* coefficients, cf. [6]. Now, taking all numerical fluxes $c_{pq}(u_p^m - u_q^m)$ associated with the inflow coefficients, i.e. $c_{pq} > 0$, implicitly (from the time layer $m = n$) and all numerical fluxes $c_{pq}(u_p^m - u_q^m)$ associated with the outflow coefficients, i.e. $c_{pq} < 0$, explicitly (from the time layer $m = n - 1$) leads to the *split scheme*:

$$\begin{aligned} (1 + kr)u_p^n + \frac{k}{m(p)} \sum_{q \in N'(p)} c_{pq}^{in}(u_p^n - u_q^n) = \\ u_p^{n-1} - \frac{k}{m(p)} \sum_{q \in N'(p)} c_{pq}^{out}(u_p^{n-1} - u_q^{n-1}). \end{aligned} \tag{28}$$

The system matrix of scheme (28) is a M-matrix which not only is diagonally dominant but also keeps the numerical solution on the new time layer in the range of the right-hand side vector. However, if the magnitude of the outflow coefficients c_{pq}^{out} (actually corresponding to backward diffusion) in the right-hand side vector is too significant, spurious oscillations

 ISBN: 978-84-616-2723-3

in the solution may occur. Hence the final goal is to propose a scheme which would ensure that the discrete minimum-maximum principle is not violated. The *stabilized scheme* can be formulated in the following form:

$$(1+\kappa r)u_p^n + \frac{\kappa}{m(p)} \sum_{q\in N'(p)} \mathcal{C}_{pq}^{in}(u_p^n - u_q^n) =$$
$$u_p^{n-1} - \frac{\kappa}{m(p)} \sum_{q\in N'(p)} \theta_{pq}^{out} c_{pq}^{out}(u_p^{n-1} - u_q^{n-1}) \tag{29}$$

where outflow weighting factors $\theta_{pq}^{out} \in [0,1]$ and corrected inflows $\mathcal{C}_{pq}^{in}$ were used. We define the coefficients θ_{pq}^{out} as

$$\theta_{pq}^{out} = \min\left(1, \frac{m(p)(u_p^{max,n-1} - u_p^{n-1})}{\kappa n_p^{out} c_{pq}^{out}(u_q^{n-1} - u_p^{n-1})}\right), \text{ if } c_{pq}^{out}(u_q^{n-1} - u_p^{n-1}) > 0, \tag{30}$$

$$\theta_{pq}^{out} = \min\left(1, \frac{m(p)(u_p^{min,n-1} - u_p^{n-1})}{\kappa n_p^{out} c_{pq}^{out}(u_q^{n-1} - u_p^{n-1})}\right), \text{ if } c_{pq}^{out}(u_q^{n-1} - u_p^{n-1}) < 0, \tag{31}$$

$$\theta_{pq}^{out} = 1, \qquad \text{if } c_{pq}^{out}(u_q^{n-1} - u_p^{n-1}) = 0 \tag{32}$$

where the symbol n_p^{out} is defined as the number of nonzero outflows from the finite volume p to all its neighbours. Symbols $u_p^{max,n-1}$ and $u_p^{min,n-1}$ denote the upper and lower bound for an arbitrary right-hand side element $p \in \mathcal{T}_h$. By using definitions (30)-(32) we have reduced the outflow coefficients in the scheme (29) by the factor $(1-\theta_{pq}^{out})c_{pq}^{out}$ which must be added to the inflows of the neighbors. To this end we define

$$\mathcal{C}_{qp}^{in} = c_{qp}^{in} - (1-\theta_{pq}^{out})c_{pq}^{out}. \tag{33}$$

which determines the new inflow coefficients in the stabilized scheme (29).

3 Numerical Experiments

We have performed two types of numerical experiments. The first one is related to the overall accuracy respectively convergence and the second one to the discrete minimum-maximum principle. In the former case we have exploited the fact that we are provided with a quasi closed-form solution for a European call option in the Heston model. We shall thus use it as a benchmark for the numerical test. Table 1 lists all parameters and variable ranges used in this section. Furthermore, a quadratic coupling between the time step κ and the space mesh sizes $h_x,\ h_y$ have been chosen, i.e. $\kappa = h_x h_y$. Errors $e_{N_x,N_y}^{N_{ts}}$ are estimated in the $L_2(I,\Omega)$ numerical norm and the experimental order of convergence is defined as

$$\mathrm{EOC}_{\kappa\sim h_x h_y} = \log_2 \frac{||e_{N_x,N_y}^{N_{ts}}||}{||e_{2N_x,2N_y}^{4N_{ts}}||}.$$

 ISBN: 978-84-616-2723-3

Table 1: Variable ranges and parameter values used for the computation of the numerical solution for the Heston model.

Parameter	Value	Parameter	Value
x	$[-7, 3]$	S	$[0.1, 2008]$
y	$[0, 1]$	v	$[0, 1]$
τ	$[0, 0.05]$	t	$[0, 0.05]$
E_1	100	E_2	120
ρ	-0.5	κ	5
θ	0.07	σ	0.5
r	0.1	T	0.05

We have applied the basic scheme (26) and the stabilized scheme (29) on the evolution of the European call option initial profile described by $V(S, v, T) = \max(S - E, 0)$ with $E = E_1$. Since the call option is a continuously differentiable function $\forall \tau > 0$ and does not contain large gradients we can expect that not much stabilization is needed during the computation. This fact is clearly visible in Table 2 where the basic scheme only slightly outperforms its stabilized version regarding both accuracy and convergence. Nevertheless, for this solution profile the stabilized scheme seems to exhibit second order convergence in space as expected.

Table 2: Errors in $L_2(I, \Omega)$ norm and EOCs of the schemes (26) and (29) compared with the exact solution of a European call option.

N_x	N_y	N_{ts}	Basic	EOC	S^{2}IIOE	EOC
20	10	1	$1.341\ 10^{-3}$	-	$1.390\ 10^{-3}$	-
40	20	4	$5.491\ 10^{-4}$	1.29	$5.387\ 10^{-4}$	1.37
80	40	16	$1.952\ 10^{-4}$	1.49	$2.105\ 10^{-4}$	1.36
160	80	64	$6.118\ 10^{-5}$	1.67	$6.943\ 10^{-5}$	1.60
320	160	256	$1.729\ 10^{-5}$	1.82	$1.993\ 10^{-5}$	1.80

The second experiment is related to the discrete minimum-maximum principle. We have investigated the evolution of an European-style binary option under the Heston model whose payoff is defined as follows (cf. Figure 2):

$$V(S, v, T) = 1, \text{ if } S \in [E_1, E_2]$$
$$V(S, v, T) = 0, \text{ otherwise.} \tag{34}$$

For the analysis of the violation of the global minimum-maximum principle we can exploit

 ISBN: 978-84-616-2723-3

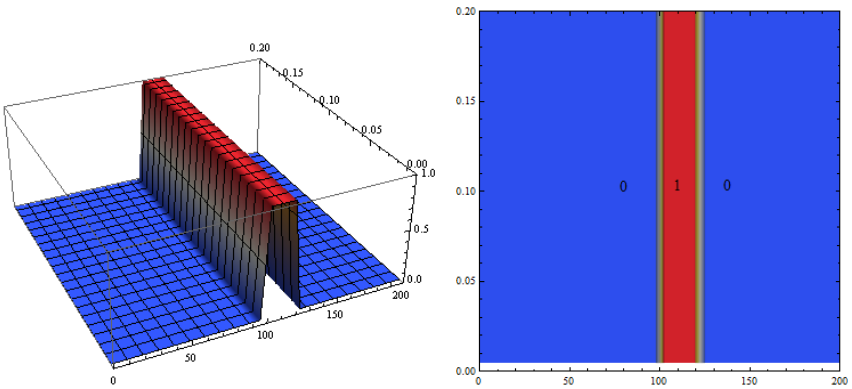


Figure 2: Payoff function $V(S, v, T)$ for a binary option with strike prices $E_1 = 100$ and $E_2 = 120$: 3D view (left) and contour plot (right).

the non-arbitrage principle which implies that the price of such a product can never reach negative values and it can never exceed its payoff maximum $V(S, v, t) \leq 1, \ \forall t \in [0, T)$. In this example the interest rate has been put $r = 0.3$. The space discretization has been chosen as follows: $h_x = 0.05$, $h_y = 0.01$, i.e. $N_x = 200$, $N_y = 100$. The coupling between time and space stepping has been set to $\kappa = h_x h_y$, i.e. $N_{ts} = 100$.

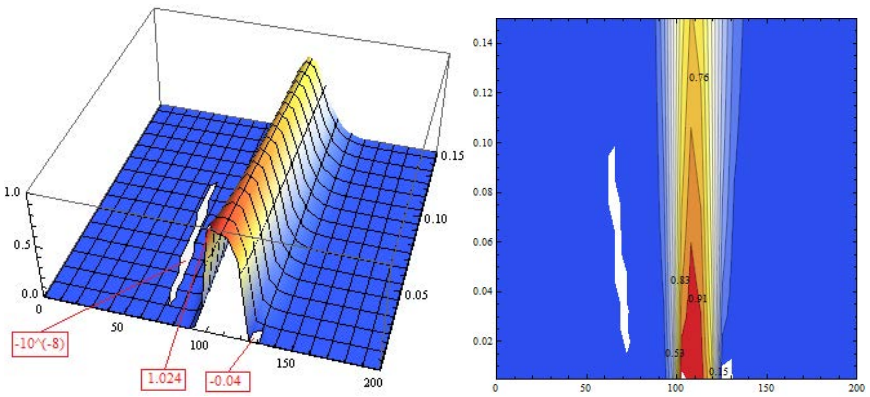


Figure 3: Numerical solution computed by the basic diamond-cell scheme (26) for a binary option with $r = 0.3$ at $T = 0.05$: 3D view (left) and contour plot (right).

In Figures 3 - 4 numerical solution profiles in time $t = 0$ are presented. Clearly, the unstabilized basic scheme exhibit oscillations – on both sides of the main mass ($S < 100$

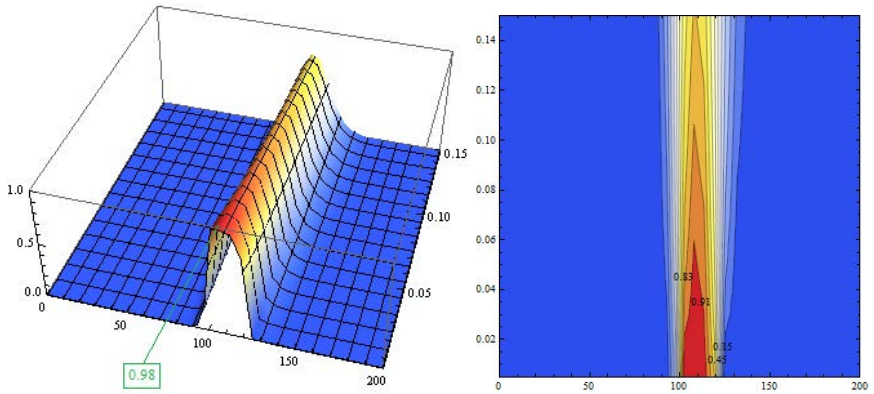


Figure 4: Numerical solution computed by the stabilized scheme (29) for a binary option with $r = 0.3$ at $T = 0.05$: 3D view (left) and contour plot (right).

and $S > 120$) and on both ends of the original range $[0, 1]$. It is the consequence of the backward diffusion inherent in the basic scheme applied on a solution profile containing large gradients. The global extremes are exceeded by an error of order 10^{-2}. On the other hand, the stabilized scheme gives results which behave according to the discrete minimum-maximum principle.

4 Conclusion

We have introduced three numerical schemes based on the finite volume discretization for solving an advection-diffusion-reaction partial differential equation arising in the Heston model. In order to obtain the basic scheme we have replaced the diffusion fluxes by means of the diamond-cell approximation and all reconstructed values have been taken implicitly. Secondly, we have split the flux coefficients into inflow and outflow group. Taking all inflows implicitly and all outflows explicitly has led to a system M-matrix and favourable solvability properties of the split scheme. Thanks to an adjustment of weights of certain coefficients in the split scheme we have been able to propose a stabilized scheme which does not violate the discrete minimum-maximum principle and is second order accurate in space. Finally, we have shown on numerical experiments that all these theoretical properties hold.

Acknowledgements

This work was supported by the grant APVV–0184–10.

References

[1] F. Black, M. Scholes, *The pricing of options and corporate liabilities*, The Journal of Political Economy, 81. **3** (1973) pp. 637-654.

[2] Y. Coudiere, J. P. Vila, P. Villedieu, *Convergence rate of a finite volume scheme for a two-dimensional convection-diffusion problem*, M2AN Math. Model. Numer. Anal., 33. (1999) pp. 493–516.

[3] J. C. Cox, J. E. Ingersoll, S. A. Ross, *A Theory of the Term Structure of Interest Rates*, Econometrica, 53. (1985) pp. 385–408.

[4] O. Drblíková, K. Mikula, *Convergence analysis of finite volume scheme for nonlinear tensor anisotropic diffusion in image processing*, SIAM Journal on Numerical Analysis, 46. **1** (2007) pp. 37–60.

[5] R. Eymard, T. Gallouët, R. Herbin, *Finite Volume Methods in Handbook for Numerical Analysis, 7*, Elsevier, Amsterdam, 2000.

[6] A. Handlovičová, P. Kútik, K. Mikula, *Stabilized semi-implicit finite volume scheme for parabolic tensor diffusion equations*, Proceedings of the 19th Conference on Scientific Computing ALGORITMY 2012. (2012) pp. 438–477.

[7] R. A. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[8] J. C. Hull, A. White, *The Pricing of Options on Assets with Stochastic Volatilities*, Journal of Finance, 42. (1987) pp. 281–300.

[9] S. L. Heston, *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies, 6. **2** (1993) pp. 327–343.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Population dynamics of a predator-prey system with recruitment and capture on both species

Lilia M. Ladino[1], Edison I. Sabogal[1] and Jose C. Valverde[2]

[1] *Department of Sciences, University of the Llanos, Colombia*

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

emails: lladino@unillanos.edu.co, esabogal@unillanos.edu.co, jose.valverde@uclm.es

Abstract

This paper provides a mathematical model for a predator-prey system, with recruitment and capture on both species, and analyzes its qualitative dynamics. The model is formulated considering a population growth based on a general form of *recruitment* and *predator functional response*, as well as the *capture* on both preys and predators at a rate proportional to their populations. In this sense, it is proved that the dynamics and bifurcations are determined by a two-dimensional threshold parameter $\mathcal{R} = (m_1, m_2)$ with $m_1, m_2 > 0$. Finally, some numerical simulations, varying the parameters values m_1 and m_2, show different scenarios about the evolution of the system and allow to validate the model.

1 Introduction

Dynamics of predator-prey systems continue being of interest to both applied mathematicians and ecologists due to its universal existence and importance [2]. Predator-prey interaction is one of the basic interspecies relations for ecological and social models [24]. Mathematical models to describe this type of interactions have their origin in Lotka [17] and Volterra [23], who, in the 1920s, formulated a similar model, but independently, which is known as Lotka-Volterra model in honor to them [12]. Since then, a logistic type growth has been usually assumed for the prey species in the models, while a linear mortality rate for the predator species [24].

Some time later, a *predator functional response* began to be incorporated, i.e., it was assumed that the rate of predation depends on the density of the population of victims [12].

Some models with these features, known as Rosenzweig-MacArthur models [20], are of the form

$$\begin{cases} \dot{x} = xf(x) - xy\phi(x) \\ \dot{y} = cxy\phi(x) - ey, \end{cases} \tag{1}$$

where: $xf(x)$ represents the growth rate of the prey species in the absence of predators; the term $xy\phi(x)$ is called *predator functional response*; $x\phi(x)$ is the number of preys consumed per predator in a unit time; the constant c is the conversion efficiency of preys into predators (usually $0 < c < 1$ [24]); the term $cxy\phi(x)$ is named the *predator numerical response*; and the constant e is the predator mortality rate [4].

So far, some conventional forms for the predator functional response that have been used (see [4, 5, 11, 14, 15, 18, 22, 24]) are:

$x\phi(x) = \frac{\alpha x}{M}, \quad 0 \leq x \leq M, \quad x\phi(x) = \alpha, \quad x > M, \quad (\alpha, M > 0)$ [Holling type I]

$x\phi(x) = \frac{\alpha x}{x+A}, \qquad (\alpha, A > 0)$ [Holling type II]

$x\phi(x) = \frac{\alpha x^2}{x^2+A^2}, \qquad (\alpha, A > 0)$ [Holling type III]

$x\phi(x) = a(1 - e^{-cx})$ [Ivlev type]

$x\phi(x) = ax^q, \qquad (q < 1)$ [Rosenzweig type]

The harvesting and capture in mathematical models of population dynamics have been treated by authors as Brauer and Castillo-Chavéz [4], Britton [5], Murray [18] or Isaza and Campos [12]. In particular, for predator-prey systems, authors have only studied the capture on predators. However, Brauer and Castillo-Chavéz suggest studying the capture also on preys, since this could be appropriate for an examination of the extent to which one can control a population by tampering with its food supply [4].

When studying populations that are fishery resources, researchers are generally interested only in the exploitable population, since it is the part of the total population that is visible to fishing and fisheries research [7]. For this reason, it is necessary to know about the different changes of state of the fish concerning to its biological life cycle and its exploitation. The life cycle of the fishes develops to the extent that each individual goes through stages of egg, larva, juvenile and adult. In the early stages of life, the fish cannot be captured by those engaged in fishing activities, either because they are too small or because they are outside the fishing areas [7]. But, as the fish grows, the conditions are modified until a change (in the size, location and/or habits of the new fish) which provokes that the fish can be detected and caught by the existing fishing methods for the first time [7]. The

 ISBN: 978-84-616-2723-3

number of individuals that arrive in the fishing area for this first time is called *recruitment* to the exploitable phase [6]. These individuals grow, spawn (once or several times) and die. The variations in their abundances are mainly due to predation and environmental factors (winds, currents, temperature, salinity, etc.) [6]. In these non exploitable phases, the mortality is not directly caused by fishing [6] and is usually very high, particularly at the end of the larvae phase [9]. This results in a small percentage of survivors until the recruitment [6]. In [16], the authors set out an approach to these kinds of models, providing one that considers recruitment in a two-stage species.

The relationship between stock S and recruitment R in fish populations has been subject of many studies (see [8, 9, 10, 13, 21]). Among the models that have been developed to fit stock-recruitment curves to data sets are the well-known Beverton-Holt [3] and Ricker [19] curves, namely:

$$R = \frac{\alpha S}{S+\beta}, \qquad (\alpha, \beta > 0) \qquad \text{[Beverton-Holt]}$$

$$R = \delta S e^{-\lambda S}, \qquad (\delta, \lambda > 0) \qquad \text{[Ricker]}$$

In this sense, the main purpose of this work is to provide and analyze a mathematical model for the dynamics of a predator-prey system, which is formulated considering a population growth based on a general form of *recruitment*, a fairly general *predator functional response* and *capture* on both predators and preys at a rate proportional to their populations. Thus, the results obtained cover the casuistry originated by the different types of functional response and recruitment stated before.

So, we set out a continuous mathematical model for the dynamics of a predator-prey system with recruitment and capture on both species, which can be modeled by the following system of nonlinear differential equations,

$$\begin{cases} \dot{x}(t) = x(t)f(x(t), y(t)) = x(t)[r(x(t)) - y(t)\phi(x(t)) - m_1] \\ \dot{y}(t) = y(t)g(x(t), y(t)) = y(t)[s(y(t)) + cx(t)\phi(x(t)) - m_2] \end{cases} \tag{2}$$

All the parameters in this model are non-negative, in order to have a biological significance. Additionally, the following properties are satisfied,

- $\forall x \geq 0, \quad r(x) > 0, \ r'(x) < 0$ and $\lim_{x\to\infty} r(x) = 0$,
- $\forall y \geq 0, \quad s(y) > 0, \ s'(y) < 0$ and $\lim_{y\to\infty} s(y) = 0$,
- $[xr(x)]' \geq 0$ and $[ys(y)]' \geq 0$,
- $\forall x \geq 0, \quad \phi(x) > 0, \ \phi'(x) \leq 0$ and $[x\phi(x)]' \geq 0$, being $x\phi(x)$ bounded as $x \to \infty$ [4, 15].

 ISBN: 978-84-616-2723-3

The proposed model constitutes a new perspective for mathematical modeling of biological populations, because it incorporates a growth rate based on the *recruitment*, a fairly general *predator functional response* and the *capture* of both predators and preys, as suggested by Brauer and Castillo [4] at a rate proportional to their populations.

The mathematical results achieved have great biological interest, since they allow to predict when the predator and prey populations tend to be stable or, on the contrary, when they tend to disappear, either both or only one of them.

These theoretical results have been validated by numerical simulations which have been executed by means of a software develop by the authors [1], using statistical data of some fish stocks of the genus *Prochilodus* and *Pseudoplatystoma*, that interact as prey and predator, respectively, in the Orinoco basin in Colombia.

References

[1] A.M. Atehortua, L.M. Ladino, J.C. Valverde. Dsamala toolbox: software for the analysis and simulation of discrete, continuous and stochastic dynamical systems, Rev. Ing. Invest. 32 (2012), 51–57.

[2] A.A. Berryman, The origins and evolution of predator-prey theory, Ecology 73 (1992) 1530–1535.

[3] R.J.H. Beverton, S.J. Holt, On the dynamics of exploited fish populations, Fish. Invest. (Ser. 2) 19 (1957) 1–533.

[4] F. Brauer, C. Castillo-Chávez, Mathematical models in population biology and epidemiology, Texts in Applied Mathematics, 40, Springer, New York, 2001.

[5] N.F. Britton, Essential mathematical biology, Springer Undergraduate Mathematics Series, Springer-Verlag, United States of America, 2003.

[6] E.L. Cadima, Fish Stock Assessment Manual, FAO Fisheries Technical Paper 393, Rome, 2003.

[7] J. Csirke, Introducción a la dinámica de poblaciones de peces, Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO), Roma, 1989.

[8] D.H. Cushing, The study of stock and recruitment. In: J.A. Gul- land. Fish population dynamics: the implications for management. John Wiley, New York (1988) 105–128.

[9] D.H. Cushing, Towards a science of recruitment in fish populations. In: Excelence in Ecology, Book 7, Ecology Institut, Oldendorf/Scuhe, Germany, 1996.

 ISBN: 978-84-616-2723-3

[10] R. Hilborn, C.J. Walters, Quantitative fisheries stock assessment: choice, dynamics, and uncertainty, Chapman and Hall, United States of America, 1992.

[11] C.S. Holling, The functional response of predators to prey density and its role in mimicry and population regulation, Mem. Entomol. Soc. Canada 45 (1965) 1–60.

[12] J.F. Isaza, D. Campos, Ecología: una mirada desde los sistemas dinámicos, Editorial Pontificia Universidad Javeriana, Colombia, 2006.

[13] T.C. ILES, A Review of stock-recruitment relationships with reference to flatfish populations, Netherlands Journal of Sea Research 32 (3/4) (1994) 399–420.

[14] V.S. Ivlev, Experimental Ecology of the Feeding of Fishes, Yale University Press: New Haven, 1961.

[15] N.D. Kazarinoff, P. Van Den Driessche, A model predator-prey system with functional response. Mathematical Biosciences 39 (1978) 125–134.

[16] L.M. Ladino, J.C. Valverde, Populations dynamics of a two-stage species with recruitment, Mathematical Methods in the Applied Sciences, 36(6):722–729, 2013.

[17] A.J. Lotka, Elements of physical biology, Williams and Wilkins, Baltimore, 1925.

[18] J.D. Murray, Mathematical Biology I. An Introduction, Springer, Berlin, 2002.

[19] W.E. Ricker, Stock and recruitment, J. Fish. Res. Bd Can. 11 (1954) 559-623.

[20] M.L. Rosenzweig, R.H. MacArthur, Graphical representation and stability conditions of predator-prey interactions, Amer. Naturalist 97 (1963) 209–223.

[21] B.J. Rothschild, Dynamics of marine fish populations. Harvard University Press, Cambridge MA, (1986) 1–277.

[22] P. Turchin, Complex population dynamics: a theoretical/empirical synthesis, Princeton University Press, 2003.

[23] V. Volterra, Fluctuations in the abundance of species, considered mathematically. Nature, (1926) 118–558.

[24] J. Wang, J. Shi, J. Wei, Predator-prey system with strong Allee effect in prey, J. Math. Biol. 62 (2011) 291-331.

[25] R. Welcomme, River fisheries, FAO Fisheries Technical Paper 262 (1985) 1–130.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Radial basis function methods in computational finance

Elisabeth Larsson[1], Sônia M. Gomes[2], Alfa Heryudono[3] and Ali Safdari-Vaighani[4]

[1] *Department of Information Technology, Uppsala University, Sweden*

[2] *Department of Applied Mathematics, Universidade Estadual de Campinas, Brazil*

[3] *Department of Mathematics, University of Massachusetts, Dartmouth, MA, USA*

[4] *Department of Mathematics and Statistics, Allameh Tabatabai University, Tehran, Iran*

emails: elisabeth.larsson@it.uu.se, soniag@ime.unicamp.br, aheryudono@umassd.edu, alisafdari87@gmail.com

Abstract

Radial basis function (RBF) based approximation methods for numerical solution of partial differential equations are interesting due to their potentially spectral accuracy and due to being meshfree. This could be especially beneficial for high dimensional problems, where meshing is non-trivial. In this work, we present different RBF approaches and evaluate them on a multi-asset option pricing problem. The conclusion is that the properties of the problem need to be taken into account in the solution method in order to have an approach that is viable for higher dimensions. Furthermore, we suggest to use an RBF based partition of unity approach in order to introduce locality and reduce the computational cost.

Key words: radial basis function, option pricing, partition of unity, collocation
MSC 2000: 65M70

1 Introduction

Radial basis function (RBF) based methods [4] have become quite popular for pricing of financial derivatives based on partial differential equation (PDE), or in the case of jump diffusion partial-integro differential equation (PIDE) formulations of the pricing problem. One of the main arguments is that RBF methods are easy to use in high-dimensions, i.e.,

for several underlying assets. The methods work only with scattered node points and do not require meshing. Furthermore, the basic mathematical formulation is the same in any number of dimensions. In an RBF method, the RBF approximation $s(\underline{x},t)$ to the value $u(\underline{x},t)$ of the financial derivative is typically of the form

$$s(\underline{x},t)=\sum_{j=1}^{N}\lambda_j(t)\phi(\varepsilon\|\underline{x}-\underline{x}_j\|)\equiv\sum_{j=1}^{N}\lambda_j(t)\phi_j(\underline{x}), \tag{1}$$

where $\phi(r)$ is a (conditionally) positive definite RBF, ε is the shape parameter, which makes the RBF more flat as it goes to zero and more peaked as it goes to infinity, and x_j are scattered node points that act as the center points for the RBFs. The coefficients λ_j can be determined through collocation with equations and boundary conditions.

As can be seen from (1), the RBF approximation yields a continuous representation of the solution function. This allows explicit evaluation of derivatives of the approximation, which is an advantage in finance where the partial derivatives $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$, denoted by Δ and Γ, are needed for hedging purposes.

In the following sections, we will describe some different approaches, comment on their strengths and weaknesses, provide some relevant citations, and show numerical experiments to demonstrate the performance. Finally, we will reach to what we think is currently the most promising approach, radial basis function partition of unity methods, and present some preliminary results for these.

2 The model problems used for demonstrations

We will use the simplest possible option pricing problem to test the numerical approaches. We consider this to be a European basket call option, priced using the multi-dimensional Black-Scholes equation. Any added features like jump diffusion, stochastic volatility and exoticity may need special treatment by the numerical methods, but this is not an issue that we are pursuing in this paper.

The d-dimensional Black-Scholes equation for an option on d underlying assets is defined on $\mathbb{R}^d_+$. For computational purposes, we define a computational domain $\Omega\subset\mathbb{R}^d_+$. Furthermore, we define $\Gamma\subset\partial\Omega$ as the part of the boundary of the computational domain where we impose boundary conditions. After transformation of the time-variable [17] and scaling of the spatial variables as in [13], we can write the Black-Sholes equation as the following initial-boundary value problem

$$\frac{\partial u}{\partial t}(\underline{x},t)=\mathcal{L}u(\underline{x},t), \qquad \underline{x}\in\Omega, \qquad t>0, \tag{2}$$

$$u(\underline{x},t)=g(\underline{x},t), \qquad \underline{x}\in\Gamma, \qquad t>0, \tag{3}$$

$$u(\underline{x},0)=\Phi(\underline{x}), \qquad \underline{x}\in\Omega, \tag{4}$$

 ISBN: 978-84-616-2723-3

where $u(\underline{x},t)$ is the value of the option, $\underline{x} \in \mathbb{R}^d_+$ contains the scaled values of the d assets, and t is the time left to the exercise time T of the option. The spatial operator has the form

$$\mathcal{L}u(\underline{x},t) = r\sum_{i=1}^{d} x_i \frac{\partial u}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{d} \left[\sigma\sigma^T\right]_{ij} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j} - ru,$$

where r is the risk free interest rate and σ is the volatility matrix. For our numerical examples, we use the contract function

$$\Phi(\underline{x}) = \max(0, \frac{1}{d}\sum_{i=1}^{d} x_i - K), \tag{5}$$

where, in our present case, the exercise price K is always equal to 1 due to scaling. The boundary conditions are linked to the contract function [19]. At the near-field boundary, consisting of the origin $\underline{x} = \underline{0}$, we use

$$g(\underline{x},t) = 0, \tag{6}$$

and at the far-field boundary, here defined as the part of the boundary where $\frac{1}{d}\sum_{i=1}^{d} x_i \geq 4K$, we impose

$$g(\underline{x},t) = \frac{1}{d}\sum_{i=1}^{d} x_i - K\exp(-rt). \tag{7}$$

3 Discretization in time and approximation in space

Let the time interval $[0,T]$ be divided into M steps of length $k^n = t^n - t^{n-1}$, $n = 1,\ldots,M$, and let the approximate solution at the discrete times t^n be denoted by

$$v^n(\underline{x}) \approx u(\underline{x},t^n).$$

In the majority of the numerical experiments we discretize the PDE problem (2–4) in time using the unconditionally stable, second-order accurate, implicit BDF-2 method [11, p. 401], resulting in

$$v^1(\underline{x}) - k^1\mathcal{L}v^1(\underline{x}) = v^0(\underline{x}), \qquad \underline{x} \in \Omega, \tag{8}$$

$$v^n(\underline{x}) - \beta_0^n\mathcal{L}v^n(\underline{x}) = \beta_1^n v^{n-1}(\underline{x}) - \beta_2^n v^{n-2}(\underline{x}), \qquad \underline{x} \in \Omega, \quad n = 2,\ldots,M, \tag{9}$$

$$v^n(\underline{x}) = g(\underline{x},t^n), \qquad \underline{x} \in \Gamma, \quad n = 1,\ldots,M, \tag{10}$$

$$v^0(\underline{x}) = \Phi(\underline{x}). \qquad \underline{x} \in \Omega, \tag{11}$$

 ISBN: 978-84-616-2723-3

The details of how we choose the coefficients β_i are described in [13]. For the approximation in space, we use (1) in its time discrete form, evaluated at the node points to get

$$v^n(\underline{x}_i) = \sum_{j=1}^{n} \lambda_j^n \phi_j(\underline{x}_i), \quad i = 1, \ldots, N, \tag{12}$$

corresponding to the linear system

$$\underline{v}^n = A\underline{\lambda}^n, \tag{13}$$

where $\underline{v}^n = (v^n(\underline{x}_1), \ldots, v^n(\underline{x}_N))^T$, $A_{ij} = \phi_j(\underline{x}_i)$, and $\underline{\lambda}^n = (\lambda_n(\underline{x}_1), \ldots, \lambda^n(\underline{x}_N))^T$. In a similar fashion, we get

$$\mathcal{L}\underline{v}^n = B\underline{\lambda}^n, \tag{14}$$

where $B_{ij} = \mathcal{L}\phi_j(\underline{x}_i)$. Combining the two, we get

$$\mathcal{L}\underline{v}^n = BA^{-1}\underline{v}^n, \tag{15}$$

allowing us to work with nodal values as unknowns. Note that we can easily exchange the set of evaluation points $\{\underline{x}_i\}_{i=1}^N$ in the matrix B for some other set of points $\underline{x} \in \Omega$ to compute solution values or derivatives at arbitrary locations.

4 Numerical results for different RBF approaches

We use two different radial basis functions for the numerical experiments. The multiquadric RBF, which is conditionally positive definite, but nevertheless guarantees a non-singular interpolation matrix for distinct nodes and $\varepsilon > 0$,

$$\phi(r) = \sqrt{1 + \varepsilon^2 r^2},$$

and the Gaussian RBF, which is positive definite,

$$\phi(r) = \exp(-\varepsilon^2 r^2).$$

The scaled exercise price $K = 1$, and the exercise time used is $T = 1$ year. In the volatility matrix, we set $\sigma_{ii} = 0.3$ and $\sigma_{ij} = 0.05$, $i \neq j$. The risk free interest rate is set to $r = 0.05$. As computational domain we use $\Omega = \mathbb{R}_+^d \setminus \{\underline{x} \,|\, \frac{1}{d}\sum x_i > 4K\}$, which results in the interval [0 4] in 1-D, a triangle with corners in(0, 0), (0, 8), and (8, 0) in 2-D, and higher order simplexes in more dimensions. This is possible because the RBF method is meshfree, and it leads significant savings in the computational cost compared with solving over a hypercube. This approach was used in [19] and [13]. In [13], we also showed that there is no loss of accuracy from this truncation of the domain.

Figure 1 shows examples of node layouts used in the numerical experiments in the 2-D case. The uniform and Chebyshev nodes can be generated for any number of dimensions

 ISBN: 978-84-616-2723-3

and are based on barycentric coordinates within the simplex [14]. In the Chebyshev case, the nodes are clustered in a Chebyshev fashion towards each boundary. The adapted nodes are more dense in the region of interest, and take the strike location into account.

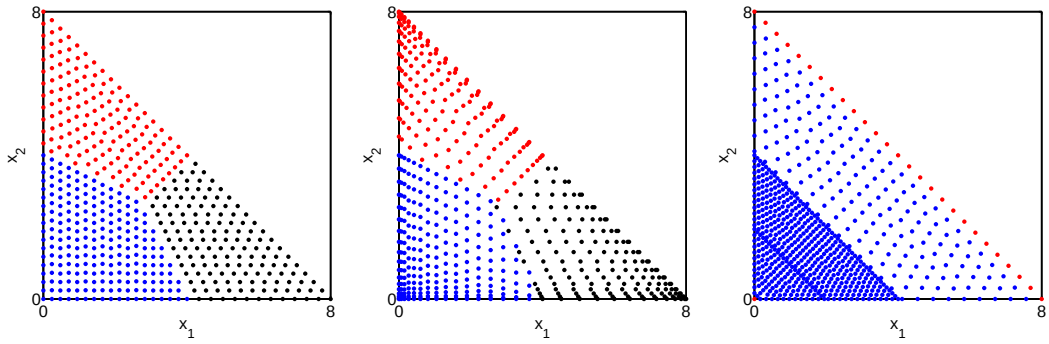


Figure 1: Examples of the *uniform* nodes, *Chebyshev* nodes, and *adapted* nodes that are used in our 2-D experiments.

In the following, the one-dimensional problem will be used as a starting point for quantitative investigation of different methods (due to the reasonable computational cost), the two-dimensional problem will be used for testing if the 1-D results carry over, and the potential for solving higher dimensional problems is discussed.

As asset prices are typically given with four or five digits of accuracy, we consider $\tau = 1 \cdot 10^{-4}$ to be a reasonable target accuracy for the solution of the option pricing problem. Of course, the desired accuracy significantly influences how large or how high-dimensional problems we can solve, so with a lower accuracy the projections become brighter. In the following subsections, we discuss different approaches in detail.

4.1 Global collocation using uniform node layouts

The most straightforward approach of an RBF method to option pricing is to use (1) directly on a set of uniformly distributed nodes. This was done for European and American options in one dimension by Hon et al. in [12, 23], and for one and two dimensions by Fasshauer et al. [6] and Marcozzi et al. [16]. RBF methods have also been applied to other types of options and contracts such as a digital option [5], a currency option [2], and a credit default swap [10], as well as to problems with jump diffusion [1, 9, 22]. In all cases, the methods work well.

Figure 2 shows the results for different values of N as a function of ε. It should be noted that due to ill-conditioning that grows with increasing N and decreasing ε, the errors blow up and are not shown for the lower left corner of the figure. The target accuracy is reached the first time for $N = 46$ node points. The solutions are shown to the right and

the errors are about equally large at the boundaries and at the region of interest near the exercise price $K = 1$. It should also be noted that the results are sensitive to the placement of nodes near the strike discontinuity (see also [19]). Therefore, the number of node points have been chosen in the most favourable way.

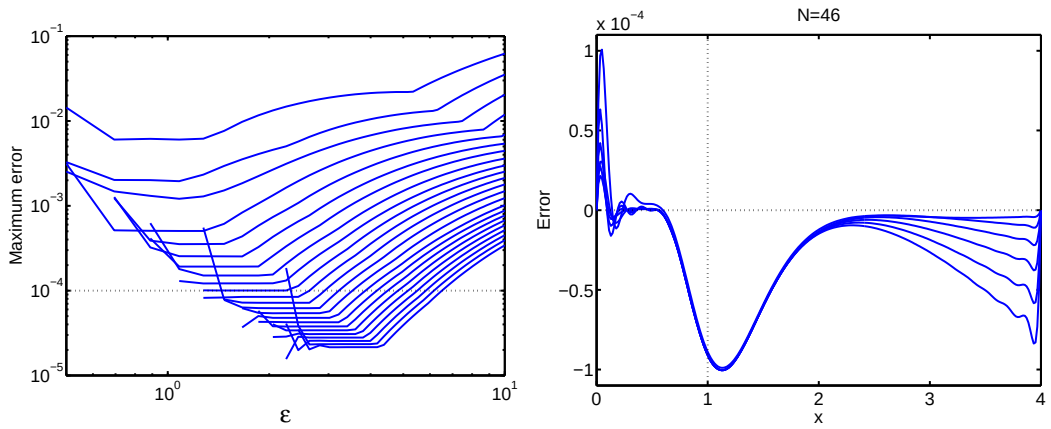


Figure 2: Left: The maximum error in the solution to the one-dimensional European option pricing problem as a function of ε for different values of N. From top to bottom, $N = 4j+2$, $j = 2, \ldots, 24$. The dotted line shows the target error. Right: The solution error as a function of the scaled asset price for different values of $\varepsilon \in (1.25,\ 2.25)$ for $N = 46$.

Now, consider the same problem in two dimensions. We can try to estimate how many points we are likely to need to achieve the same accuracy. It is reasonable to assume that we want the same node distance along the diagonal $x_1 = x_2$ in the two-dimensional case as we had along the interval in the one-dimensional case. This line is $\sqrt{2}$ times longer than the interval, leading to $\sqrt{2}N_{1\mathrm{D}}$ nodes. We can use this as a measure of the number of points per dimension in the 2-D case. We can apply similar arguments in higher dimensions, leading to the special and general cases

$$N_{2\mathrm{D}} \propto \frac{(\sqrt{2}N_{1\mathrm{D}})^2}{2} = N_{1\mathrm{D}}^2, \qquad N_{\mathrm{dD}} \propto \frac{(\sqrt{d}N_{1\mathrm{D}})^d}{d!}, \tag{16}$$

where the factorial in the denominator is due to the ratio of the simplex to the hypercube. With 46 points in one dimension, this indicates that we need around 2100 points in two dimensions. The best solution we could come up with, without an extensive search of the parameter space, was for $\varepsilon = 1.5$ and $N = 2939$ with a maximum error $E = 5.2 \cdot 10^{-3}$. The error, displayed in Figure 5, is largest in the strike region. In the three-dimensional case, formula (16) indicates over 28 000 nodes. Since we need to solve a dense linear system of this size, this becomes very expensive, both in terms of computational cost and memory requirements. In four dimensions, we judge it to be unfeasible in practice.

 ISBN: 978-84-616-2723-3

4.2 Global collocation using adapted node layouts

As discussed in the previous subsection, the problem is sensitive to the placement of nodes near the strike region. Furthermore, the errors are large in this region, which is where we want to know the solution. Typically, options are traded with exercise prices in the vicinity of the current asset value.

By employing an adapted node layout, we aim to reduce the error in the region of interest, while possibly sacrificing accuracy in other parts of the domain. Therefore, we introduce a different error measure, the financial error [19], defined by

$$E_f = \max_{\underline{x} \in \Omega_K} |s(\underline{x}, T) - u(\underline{x}, T)|,$$

where $\Omega_K = \{\underline{x} \,|\, K - \frac{2}{3}K \leq \frac{1}{d}\sum_{i=1}^{d} x_i \leq K + \frac{2}{3}K\}$. We use the type of adapted node layout shown in Figure 1 and perform the same experiment as for the uniform nodes, but using the financial error measure. The results are shown in Figure 3. We can see that the errors in the strike region are much smaller than in the rest of the domain, and we reach the target accuracy already at $N = 17$ points. If we again use formula (16) we now need around 300 points in 2-D, 4000 points in 3-D, and 56 000 points in 4-D. This means that 3-D is definitely accessible, while 4-D might be stretching it a bit, but could be done with a lower target accuracy.

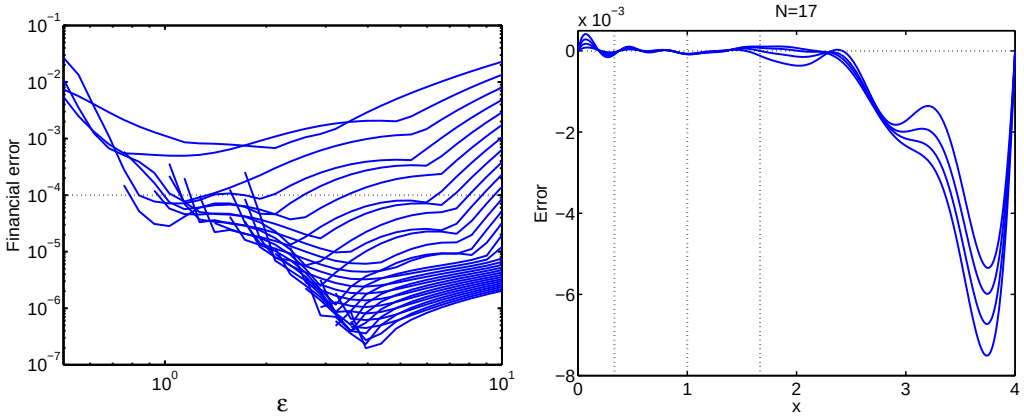


Figure 3: Left: The financial error for the adapted nodes as a function of ε for different values of N. From top to bottom, $N = 3j + 2$, $j = 3, \ldots, 33$. The dotted line shows the target error. Right: The solution error as a function of the scaled asset price for different values of $\varepsilon \in (1.03, 1.41)$ for $N = 17$.

Actual experiments in 2-D show that we can reach the target accuracy. However, because of the special node layout, we cannot hit the target exactly. At $N = 599$ points, we get a financial error $E_f = 5.1 \cdot 10^{-5}$. The result is shown in Figure 5.

 ISBN: 978-84-616-2723-3

The approach with adapted nodes is compared with the adaptive finite difference method from [18] in [19]. With our target tolerance, the adapted node RBF method is about 40 times faster in 1-D and 30 times faster in 2-D.

In [20], it is shown why using exponentially converging methods on uniform nodes must lead to exponential ill-conditioning. This issue can be overcome by clustering the nodes toward the boundaries, which was done successfully in [7]. However, for the option pricing problems, it results in reducing the errors at the boundaries while increasing them in the strike region, and is hence not an effective approach. Another example of an approach with node adaption that seems to work well is given in [1], where the adaptive residual subsampling method of [3] is used. There, the shape parameter is scaled individually for each RBF, proportionally to the inverse of the local node distance. This was suggested in [8], based on a heuristic exploration of optimal node locations and shape parameter values for some test problems. Here, we have employed this strategy for scaling the shape parameter in the adapted node approach. The results are shown in Figure 4. The error does become smaller outside of the strike region and the general behaviour of the error is somewhat improved, but there are no dramatic changes.

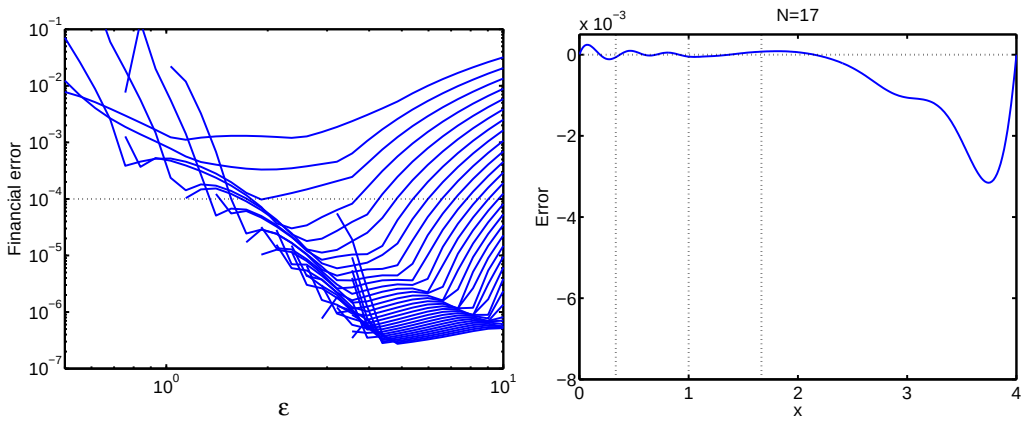


Figure 4: Results for adapted nodes and individually scaled shape parameters.

4.3 Least squares and multi-level approximations

Using a least squares approach, i.e., using more evaluation points than node points, leads to a better approximation of the non-smooth initial condition in terms of capturing the low frequencies compared with using pure collocation. By using uniform nodes and a least squares approximation we reach the target accuracy with $N = 26$ nodes. Solving the least squares problem is more expensive than solving the collocation system, but in [14], we show

 ISBN: 978-84-616-2723-3

that a least squares method is more effective than collocation in terms of the total work for a given accuracy.

An issue regarding the solutions to the option pricing problems that we have not mentioned so far is the transition from non-smooth to smooth. The non-smooth initial condition is best approximated using a large shape parameter, but the smooth solution at the final time fares better with a small shape parameter. The solution we propose in [14] is to use a multilevel approach with different shape parameters at different levels. This results in a method that is quite robust with respect to the choice of method parameters, that has a quite uniform error distribution and has a comparatively small error over the whole time interval.

Figure 5 shows how the least squares multilevel method compares with the other approaches. The number of nodes at the fine level (which determines the computational cost) is comparable to the adapted case, so the potential for solving higher dimensional problems is the same. However, the new method is more robust and the overall error behaviour is better. Therefore, we consider this to be the most promising approach so far.

5 The RBF partition of unity method

The main obstacle to using the different versions of global RBF methods is computational cost. An attractive compromise between high order and locality is offered by RBF based partition of unity (RBF-PU) methods. We have developed a method of this type in the manuscript [15]. This first paper deals with time-independent PDEs, and we are able to show theoretical results of the types below for the RBF-PU approximant.

$$\|s(\underline{x}) - u(\underline{x})\|_{W^2_\infty(\Omega)} \leq C \max_j C_j \rho_j^{m-\frac{d}{2}-\alpha} \|u\|_{\mathcal{N}(\Omega_j)}, \tag{17}$$

$$\|s(\underline{x}) - u(\underline{x})\|_{W^2_\infty(\Omega)} \leq C e^{\gamma \log(h)/\sqrt{h}} \max_j \|u\|_{\mathcal{N}(\Omega_j)}, \tag{18}$$

where Ω_j are the partitions that cover Ω, ρ_j is the radius of the partition Ω_j, h is the local node distance, and α is the degree of the PDE operator. An example of adapted partitions and nodes is shown in Figure 5. The meaning of the two estimates is the following

(i) If we fix the number of nodes/partition, we get algebraic convergence in ρ.

(ii) If we fix the partitions, we get spectral convergence in the local node distance.

A numerical demonstration of the theoretical results is given in Figure 6. Note that the target accuracy can be reached without using RBF-QR [7]. This is relevant for higher dimensions since RBF-QR is currently only available in up to three dimensions. The system matrix of the RBF-PU method is sparse, which allows us to solve very large systems of equations. We are currently working on a parallel iterative solver for these systems.

 ISBN: 978-84-616-2723-3

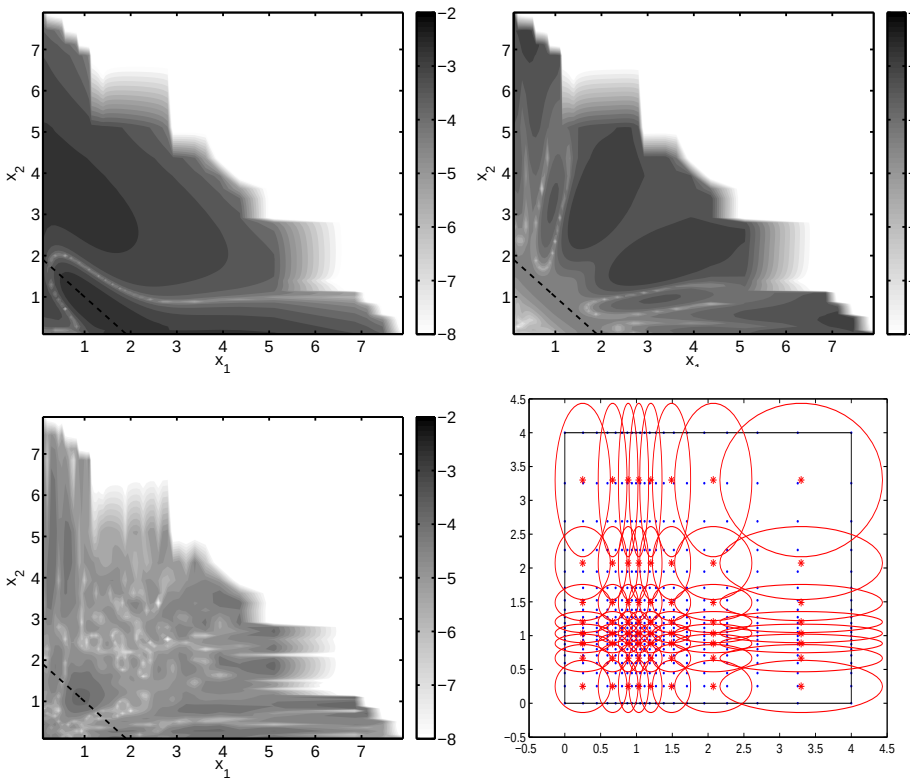


Figure 5: Top left: Collocation on $N = 2939$ uniform nodes for $\varepsilon = 1.5$. The dashed line indicates the location of the strike discontinuity. Top right: Collocation with $N = 599$ adapted nodes. Bottom left: The least squares multilevel method with $N_f = 592$ nodes at the fine level and $N_c = 96$ at the coarse level for $\varepsilon_f = 2$ and $\varepsilon_c = 0.1$. Bottom right: Partitions and nodes for the RBF-PU method.

In [21] the RBF-PU method is applied to a convection-diffusion problem and an American option pricing problem with promising results. The node and partition layout used for the American option pricing problem is shown in Figure 5.

References

[1] R. T. L. Chan, *Numerical analysis of American and European options under Lévy processes by meshless methods*, PhD thesis, Birckbeck, University of London, 2011.

[2] S. Choi and M. D. Marcozzi, *A numerical approach to American currency option valuation*, The Journal of Derivatives, 9 (2001), pp. 19–29.

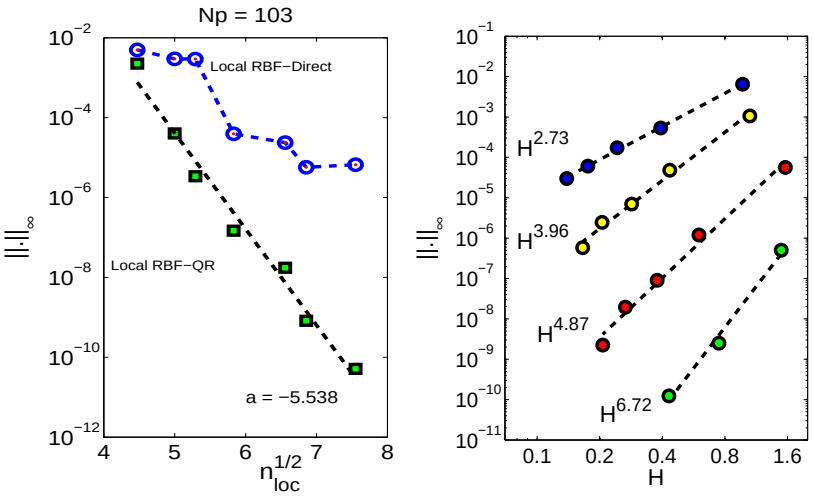


Figure 6: Left: Spectral convergence in the local node distance $h \propto \sqrt{n_{\text{loc}}}$. Right: Algebraic convergence in the partition size $H \propto \rho_j$ for increasing numbers of local nodes.

[3] T. A. Driscoll and A. R. H. Heryudono, *Adaptive residual subsampling methods for radial basis function interpolation and collocation problems*, Comput. Math. Appl., 53 (2007), pp. 927–939.

[4] G. E. Fasshauer, *Meshfree approximation methods with MATLAB*, vol. 6 of Interdisciplinary Mathematical Sciences, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

[5] G. E. Fasshauer, A. Q. M. Khaliq, and D. A. Voss, *A parallel time stepping approach using meshfree approximations for pricing options with non-smooth payoffs*, in Third World Congress of the Bachelier Finance Society, Chicago, IL, USA., July 2004.

[6] ——, *Using meshfree approximation for multi-asset american option problems*, J. of Chin. Inst. Eng., 27 (2004), pp. 563–571.

[7] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., 33 (2011), pp. 869–892.

[8] B. Fornberg and J. Zuev, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, Comput. Math. Appl., 54 (2007), pp. 379–398.

[9] A. Golbabai, D. Ahmadian, and M. Milev, *Radial basis functions with application to finance: American put option under jump diffusion*, Mathematical and Computer Modelling, 55 (2012), pp. 1354–1362.

 ISBN: 978-84-616-2723-3

[10] A. Guarin, X. Liu, and W. L. Ng, *Enhancing credit default swap valuation with meshfree methods*, European Journal of Operational Research, 214 (2011), pp. 805–813.

[11] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations. I*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1993. Nonstiff problems.

[12] Y. C. Hon and X.-Z. Mao, *A radial basis function method for solving options pricing model*, Journal of Financial Engineering, 8 (1999), pp. 1–24.

[13] E. Larsson, K. Åhlander, and A. Hall, *Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform*, J. Comput. Appl. Math., 222 (2008), pp. 175–192.

[14] E. Larsson and S. M. Gomes, *A least squares multi-level radial basis method with applications in finance.* manuscript in preparation, 2013.

[15] E. Larsson and A. Heryudono, *A partition of unity radial basis function collocation method for partial differential equations.* manuscript in preparation, 2013.

[16] M. D. Marcozzi, S. Choi, and C. S. Chen, *On the use of boundary conditions for variational formulations arising in financial mathematics*, Appl. Math. Comput., 124 (2001), pp. 197–214.

[17] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space–time adaptive FD-method*, Comput. Vis. Sci., 10 (2007), pp. 173–183.

[18] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci., 10 (2007), pp. 173–183.

[19] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson, *Improved radial basis function methods for multi-dimensional option pricing*, J. Comput. Appl. Math., 222 (2008), pp. 82–93.

[20] R. B. Platte, L. N. Trefethen, and A. B. J. Kuijlaars, *Impossibility of fast stable approximation of analytic functions from equispaced samples*, SIAM Rev., 53 (2011), pp. 308–318.

[21] A. Safdari-Vaighani, A. Heryudono, and E. Larsson, *A radial basis function partition of unity collocation method for convection diffusion equations.* manuscript in preparation, 2013.

[22] A. A. E. F. Saib, D. Y. Tangman, and M. Bhuruth, *A new radial basis functions method for pricing American options under Merton's jump-diffusion model*, Int. J. Comput. Math., 89 (2012), pp. 1164–1185.

[23] Z. Wu and Y. C. Hon, *Convergence error estimate in solving free boundary diffusion problem by radial basis functions method*, Engrg. Anal. with Bound. Elem., 27 (2003), pp. 73–79.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

An application of generalized centro-invertible matrices

Leila Lebtahi[1], Óscar Romero[2] and Néstor Thome[1]

[1] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València. E–46022 Valencia, Spain.*

[2] *Departamento de Comunicaciones, Universitat Politècnica de València. E–46022 Valencia, Spain.*

emails: leilebep@mat.upv.es, oromero@dcom.upv.es, njthome@mat.upv.es

Abstract

In this paper, we consider generalized centro-invertible matrices, which are an extension of centro-invertible matrices introduced by R.S. Wikramaratna in [The centro-invertible matrix: A new type of matrix arising in pseudo-random number generation, Linear Algebra and its Applications, 434, 1, 2011, 144–151]. An application to image encryption is given by means of the design of algorithms for encrypting and decrypting based on generalized centro-invertible matrices.

Key words: involutory matrix, centro-symmetric matrix, encryption

1 Introduction

An involutory matrix is a matrix that is its own inverse. They appear in a wide range of different topics such as: computing elementary matrices (permutation matrices), the signature matrices, an orthogonal matrix which is also symmetric, reflection against a plane, classification of finite simple groups, taking the transpose in a matrix ring, etc.

On the other hand, in cryptography, it was suggested by Hill [1] to use an involutory matrix as a key while encrypting with the Hill Cipher. The Hill's idea was to use the same matrix for encrypting and decrypting avoiding the computation of an inverse matrix. Therefore, the Hill cipher's keyspace consists of all matrices of a given size that are invertible over the ring $\mathbb{Z}_m$ of the integers modulo m. The number of such matrices was computed in [2]. The authors also compared this number with the total number of matrices and the number of involutory matrices (for a given size and modulus).

 ISBN: 978-84-616-2723-3

Let J be the square matrix with ones on the cross diagonal and zeros elsewhere; J is often called the centro-symmetric permutation matrix. This matrix J allows us to introduce the centro-invertible matrices as those matrices X such that its inverse coincides with the rotation of all the elements of the matrix through 180 degrees about the mid-point of the matrix, that is JXJ [4]. The author studied these matrices computing the total number of them by means of a bijection with the involutory matrices of the same size. As an application of this type of matrices, an algorithm to generate uniformly distributed pseudo-random numbers was developed in [3].

In what follows we consider a special type of matrix, namely generalized centro-invertible matrices. They are an extension of the centro-invertible matrices where an involutory matrix R is used instead of the centro-symmetric permutation matrix J. Specifically, let R be an $n \times n$ integer involutory matrix. An integer $n \times n$ matrix A is called generalized centro-invertible matrix if satisfies $RAR = A^{-1}$.

In this paper algorithms for encrypting and decrypting based on generalized centro-invertible matrices are presented.

2 An application

Next we present algorithms for encrypting by computing previously an involutory matrix R and then a generalized centro-invertible matrix A.

Let $\mathbb{Z}_{256}^{n\times n}$ be the set of $n \times n$ matrices with coefficients in $\mathbb{Z}_{256}$. The part of the image to be encrypted is subdivided in $X_1, \ldots, X_t$ where X_i denotes $n \times n$ sub-images of the original one. The encryption algorithm can be applied to every sub-images X_i or to a subset of them. Consider the encryption function $e_A : \mathbb{Z}_{256}^{n\times n} \to \mathbb{Z}_{256}^{n\times n}$ defined by

$$e_A(X_i) = AX_i (\mathrm{mod}\,(256)) \text{ for } X_i \in \mathbb{Z}_{256}^{n\times n}.$$

Algorithm for encrypting

Inputs: Sub-image X_i and the size n of A and R.

Outputs: Encrypted sub-image Y_i.

Step 1 Generate $n \times n$ random integer matrices T_R, T_A and Q_R, Q_A such that T_R, T_A are lower triangular and Q_R, Q_A are upper triangular, all of them with 1's in the main diagonal.

Step 2 Compute $P_R = T_R Q_R$ and $P_A = T_A Q_A$.

Step 3 Choose an arbitrary integer $r(R)$ such that $1 \leq r(R) \leq n-1$ and set

$$R = P_R \begin{bmatrix} I_{r(R)} & O \\ O & -I_{n-r(R)} \end{bmatrix} P_R^{-1}.$$

 ISBN: 978-84-616-2723-3

Step 4 Choose an arbitrary integer $r(A)$ such that $1 \leq r(A) \leq n-1$ and set

$$A = RP_A \begin{bmatrix} I_{r(A)} & O \\ O & -I_{n-r(A)} \end{bmatrix} P_A^{-1}.$$

Step 5 $Y_i = AX_i (\mathrm{mod}\,(256))$.

End

In order to decrypt we proceed as follows. Let us now consider the decryption function $d_A : \mathbb{Z}_{256}^{n\times n} \to \mathbb{Z}_{256}^{n\times n}$ defined by

$$d_A(Y) = (RAR)Y (\mathrm{mod}\,(256)) \text{ for } Y \in \mathbb{Z}_{256}^{n\times n}.$$

We first restrict the function d_A to the set $\mathcal{Y} = \{Y_1, \ldots, Y_t\}$ where Y_i are the encrypted sub-images by means of A. Then, for every $i = 1, \ldots, t$, we get $d_A(Y_i) = (RAR)Y_i(\mathrm{mod}\,(256)) = A^{-1}Y_i(\mathrm{mod}\,(256)) = X_i$. Thus, the decrypted sub-images coincide with the matrices X_i.

This reasoning allows us to design an algorithm for decryption. The inputs are the encrypted sub-images Y_i and the key matrices R and A obtained in the encryption algorithm. And the output will be the decrypted sub-images.

3 Numerical example

Our algorithms can easily be implemented on a computer. We have used the MATLAB R2010b package.

First, we obtain an involutory matrix $R \in \mathbb{Z}^{12\times 12}$ and a random integer generalized centro-invertible matrix $A \in \mathbb{Z}^{12\times 12}$. Figure 1 (a) shows the original image partially encrypted via the algorithm for encrypting. In this example, the algorithm has been applied to the sub-images $X_1, \ldots X_{193}$ grouped as indicated in Figure 2 where the corresponding mesh can be observed. Figure 1 (b) shows the decrypted image.

We can conclude that our encryption method provides a large number of keys because a large number of matrices A and R can be chosen for encrypting and decrypting. This large quantity of matrices is due to the randomness of the selection of these matrices as the algorithm for encrypting shows. It is important to remark that our decrypting algorithm does not compute inverse matrices.

Acknowledgements

This work has been partially supported by the Ministry of Education of Spain (Grant DGI MTM2010-18228).

 ISBN: 978-84-616-2723-3

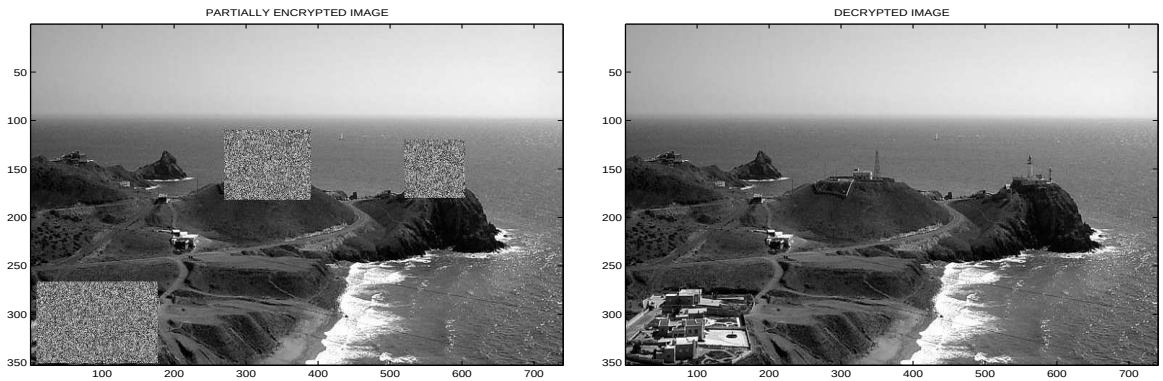


Figure 1: (a) Partially encrypted image and (b) decrypted image

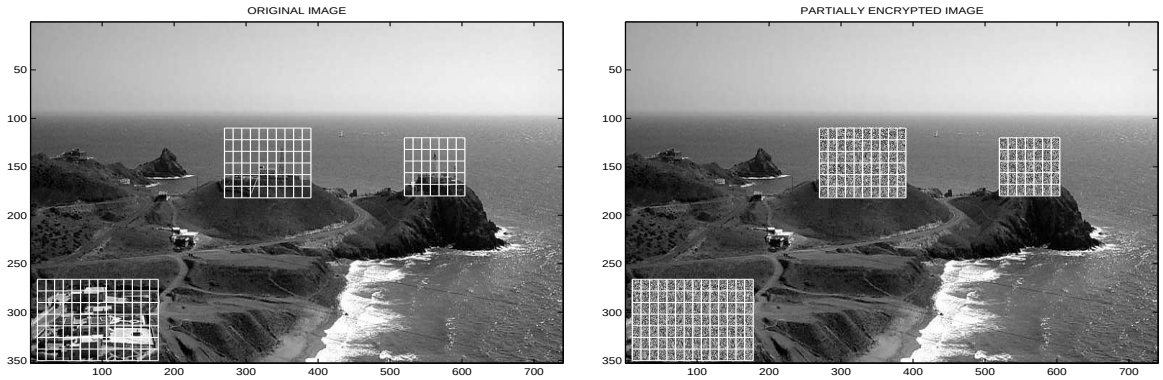


Figure 2: Mesh for the encrypting to the sub-images $X_1, \ldots, X_{193}$

References

[1] L. S. Hill, *Cryptography in an algebraic alphabet*, Amer. Math. Monthly, **36** (6) (1929) 306–312.

[2] J. Overbey, W. Traves and J. Wojdylo, *On the keyspace of the Hill Cipher*, Cryptologia, **29** (1), (2005) 59–72.

[3] R. S. Wikramaratna, *The additive congruential random number generator - a special case of a multiple recursive generator*, Linear Algebra Appl., **216** (1) (2008) 371–387.

[4] R. S. Wikramaratna, *The centro-invertible matrix: A new type of matrix arising in pseudo-random number generation*, Linear Algebra Appl., **434** (1) (2011) 144–151.

 ISBN: 978-84-616-2723-3

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

Further accuracy analysis of a mesh refinement method using 2D lid-driven cavity flows

Zhenquan Li[1]

[1] *School of Computing and Mathematics, Charles Sturt University*

emails: jali@csu.edu.au

Abstract

Lid-driven cavity flows have been widely investigated and accurate results have been achieved as benchmarks for testing the accuracy of computational methods. This paper verifies the accuracy of a mesh refinement method using 2D lid-driven flows. The accuracy is shown by comparing the coordinates of centres of primary and secondary vortices located by the mesh refinement method with the corresponding benchmark results. The accuracy verification shows that the mesh refinement method provides refined meshes that all centres of primary and secondary vortices are contained in refined grids based on the numerical solutions of Navier-Stokes equations solved by finite volume method. The well known SIMPLE algorithm is employed for pressure-velocity coupling. The accuracy of the numerical solutions is shown by comparing the profiles of horizontal and vertical components of velocity fields with the corresponding benchmarks and also streamlines. The mesh refinement method verified in this paper can be applied to find the accurate numerical solutions of any mathematical models containing continuity equations for incompressible fluid or steady state fluid flows.

Key words: finite volume method, lid-driven cavity flow, mesh refinement
MSC 2000: AMS codes 35Q35, 70-08, 68U20

1 Introduction

Meshing is the process of breaking up a physical domain into smaller sub-domains (called grids or elements or cells) in order to evaluate the numerical solutions of differential equations. Adaptive mesh refinement is a computational technique to improve the accuracy of numerical solutions of differential equations by starting the calculations on a coarse basic mesh (initial mesh) and then refining this mesh where less accuracy may occur locally.

There are a large number of publications on adaptive mesh refinements and their applications. Some refinement methods use a refinement criterion which is based on local truncation errors (e.g. Almgren *et al.* [1]; Bell *et al.* [3]). Other common methods include h-refinement(e.g. Lohner [18]; Speares & Berzins [21]), p-refinement (e.g. Bell *et al.* [3]; Zienkiewicz *et al.* [23]) or r-refinement (e.g. Miller & Miller [19]; Mosher [20]), with different combinations of these also possible (e.g. Capon & Jimack [4]; Demkowicz *et al.* [5]). The overall aim of these adaptive algorithms is to allow a balance to be obtained between accuracy and computational efficiency. The h-refinement is a method where meshes are refined and/or coarsened to achieve a prescribed accuracy and efficiency. The p-refinement is a method where the accuracy orders are assigned to elements to achieve exponential convergence rates and r-refinement is a method where elements are moved and redistributed to track evolving non-uniformities. In summary, all these mesh refinement methods are proposed based on the quantitative considerations of numerical solutions of differential equations.

We introduced adaptive mesh refinement methods from a different point of view for 2D velocity fields (Li [12]) and for 3D velocity fields (Li [11]) based on a theorem in qualitative theory of differential equations (Theorem 1.14, page 18, Ye *et al.* [22]). The theorem indicates that a divergence free vector field has no limit cycles or one sided limit cycles, that is, the trajectories (or streamlines) of divergence free vector fields are closed curves in bounded domains (singular points are streamlines). Identification of accurate locations of singular points and asymptotic lines (planes), and drawing closed streamlines are some of the accuracy measures for computational methods. The accuracy of the adaptive mesh refinement methods for the numerical velocity fields obtained by taking the vectors of the analytical velocity fields at nodes of meshes has been verified with examples by locating the singular points and asymptotic lines for two-dimensions [12]; the singular points and asymptotic plane for three-dimensions [11]; and drawing closed streamlines (Li [10]; Li & Mallinson [13]) using the refined meshes with a pre-specified number of refinements of the initial meshes. The Lebesgue measure of the set of the grids on which the mesh refinement criteria are satisfied tends to zero as the number of mesh refinements tends to infinity from the examples. However, it is impossible to achieve such numerical velocity fields in practice. The sensitivity analysis of the 2D adaptive mesh refinement for achieving the same above results for the numerical velocity fields obtained by solving mathematical models numerically is considered using 2D lid-driven cavity flows (Li & Lal [16]). The accuracy of the 2D adaptive mesh refinement method is investigated using coarse meshes [17].

This paper establishes the accuracy of the 2D mesh refinement method using 2D lid-driven cavity flows, a different finite volume method from the one used for sensitivity analysis [16] and finer meshes than those used before [17]. A comparison of the accuracy between the second order colocated finite volume method (GSFV) with a splitting method for time discretization [7] and a finite volume method with SIMPLE algorithm [8] has been done

 ISBN:978-84-615-5392-1

[15]. Our programs for these two finite volume methods use different stop conditions but the same number of grids. The comparison shows that our implementation for the finite volume method with SIMPLE algorithm provides more accurate outputs. We report the results from the refined meshes using the numerical solutions of Navier–Stokes equations obtained from the latter finite volume method.

2 Algorithm of mesh refinement and finite volume method

In this section, we summarize the mesh refinement method based on the law of mass conservation (Li [12]) and the finite volume method used (Ferziger & Peric [8]).

Assume that $\mathbf{V}_l = \mathbf{AX} + \mathbf{B}$ is a vector field obtained by linearly interpolating the vectors at the three vertexes of a triangle, where

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_1' \\ b_2' \end{pmatrix}$$

are constant matrices and vertical vector respectively, and $\mathbf{X} = (x_1, x_2)^T$. $\mathbf{V}_l$ is unique if the area of the triangle is not zero [13]. Mass conservation for an incompressible fluid or steady flows means that

$$\nabla \cdot \mathbf{V}_l = \text{trace}(\mathbf{A}) = 0. \tag{1}$$

Let f be a scalar function depending only on spatial variables. We assume that $f\mathbf{V}_l$ satisfies Equation (1) and then calculate the expressions of f. Li [12] gives the expressions of f for the four different Jacobian forms of coefficient matrix $\mathbf{A}$ in Table 1. The conditions MC (MC is the abbreviation of mass conservation) are the functions f in Table 1 not equaling zero or infinity at any point on the triangular domains when $f\mathbf{V}_l$ satisfies Equation (1) on these triangular domains.

We take that $\mathbf{V}_l$ does not satisfy mass conservation (1) or f does not satisfy the conditions MC as the criteria for the adaptive mesh refinement. Such adaptive mesh refinement method can be used to both triangular and quadrilateral grids. We have applied the adaptive mesh refinement method to quadrilateral grids for analytical velocity fields [11, 12]. In this paper we also consider quadrilateral grids. Li [12] presents the algorithm of the 2D mesh refinement.

In Section 3, we use the finite volume method with SIMPLE algorithm for pressure-velocity coupling to evaluate numerical velocity fields [8]. This finite volume method has different arrangement for pressure-velocity from the finite volume method we used before [7].

In this paper, we subdivide a quadrilateral by connecting the mid-points of the two opposite sides of a quadrilateral and the threshold number T=1, i.e., we subdivide a grid once only for testing the accuracy of the refinement method.

3 Accuracy analysis by comparisons with benchmarks

We take the results obtained by using a mesh with 601×601 uniform grids, stream function and vorticity as the benchmarks (Erturk *et al.* [6]). We consider the accuracy of the mesh refinement method in the following two aspects:

- the variations of the refined meshes according to the comparison of horizontal profile of the numerical velocity fields with the corresponding benchmarks (omitted the vertical profile due to the limit of the paper length but the final results are the same).
- inclusion of the centres of vortices located in the benchmarks in the refined mesh.

3.1 Variation of refined meshes

We consider the refined meshes for 2D lid-driven cavity flows for different mesh sizes and Reynolds number Re = 1000 and 2500, respectively. We show horizontal profiles at $x = 0.5$ as the corresponding benchmarks are known, streamlines of $\mathbf{V}_l$, and refined meshes. The streamlines are generated by *Matlab* build-in function *streamline*. The *streamline* generates streamlines from vector data so the numerical velocity fields are accurate if the streamlines are closed. A grid is said to be a refined grid if a cross is drawn inside.

One of the possible comparisons is the adaptive mesh refinement which refines everywhere that solution gradients are large (Henderson [9], 293-299). The refinement criteria enforce

$$\|\nabla u^{(k)}\| \leq \epsilon \|u^h\|_1$$

everywhere in the mesh, where $\|\cdot\|$ is the L_2 norm, $\|\cdot\|_1$ is the H^1 norm, ϵ is the discretization tolerance, u^h is finite-dimensional approximation for u, and k in $\|\nabla u^{(k)}\|$ is the number of subdomains. Figure 5.7 of [9] shows the refined meshes for $\epsilon = 10^{-3}$, 10^{-4}, 10^{-5}, and 10^{-6} for lid-driven cavity flow at Re = 1000. Even though there might be some relations between the refined meshes and the vorticity field as ϵ decreases, no one provides any information on the pattern of the flow field such as locations of the centres of vortices and separation curves of the regions (e.g., primary and secondary vortex regions).

3.1.1 Re = 1000

We show the figures for Re = 1000 generated from a mesh with 99×99 uniform grids. From Figure 1, the profile of the horizontal component u of the numerical velocity field at $x = 0.5$ shows a slight difference with the corresponding benchmark. The horizontal and vertical profiles reflect the local accuracy of the numerical velocity field. The streamlines in Figure 2 provide the global accuracy of the numerical velocity field. The streamlines in Figure 2 are not closed (spiral lines) so we conclude that the velocity field $\mathbf{V}_l$ does not satisfy Equation (1) [22, 6] or f does not satisfy the condition MC on some grids in the

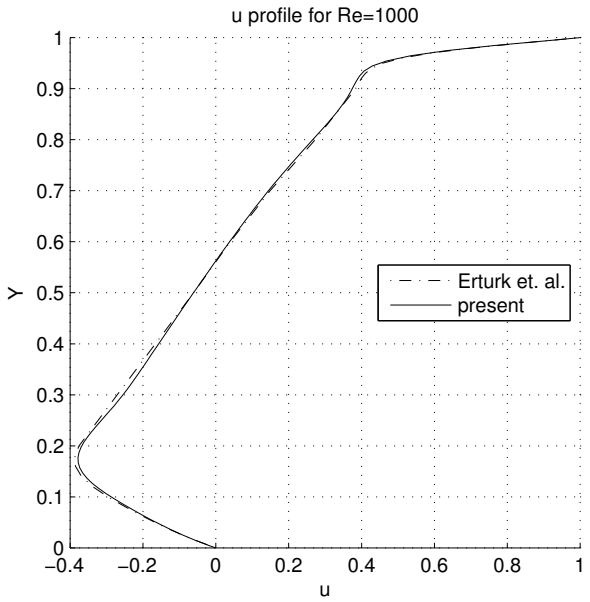


Figure 1: Horizontal profile of velocity field at $x = 0.5$ for mesh size 99×99.

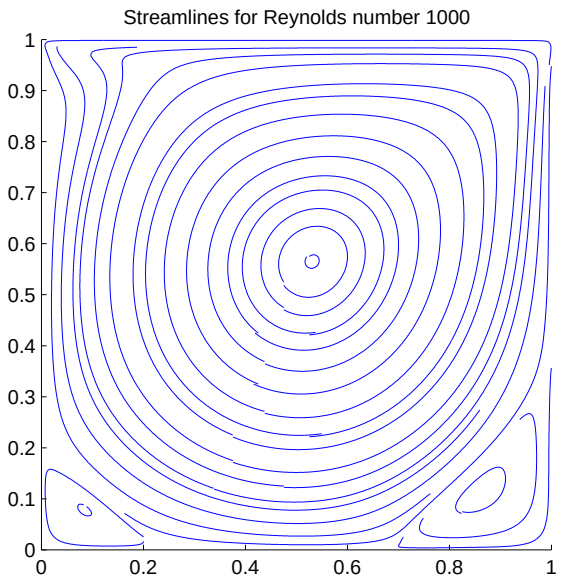


Figure 2: Streamlines for mesh size 99×99.

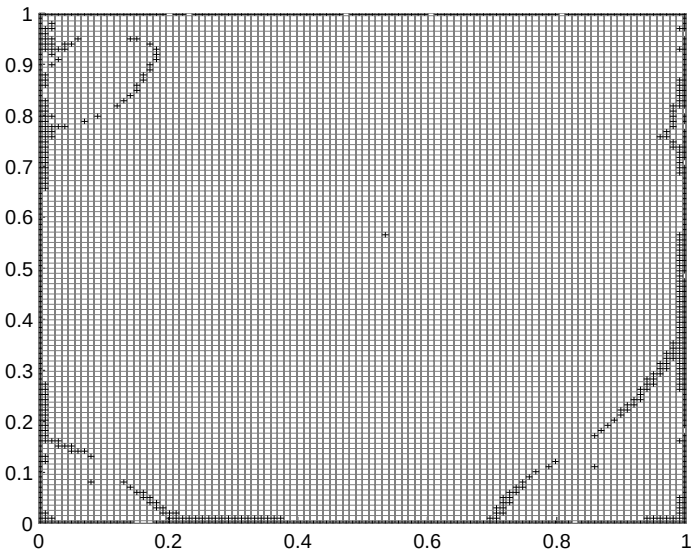


Figure 3: Refined mesh for Re = 1000 with mesh size 99×99.

regions. Figure 3 shows the refined mesh. There are three isolated grids in the refined mesh: one in the primary region and two in the secondary regions. The refined grid in the primary regions contains the centre of primary vortex, and the isolated refined grid on the bottom left side contains the centre of the bottom left secondary vortex, and the isolated refined grid on the bottom right side contains the centre of the bottom right secondary vortex (refer to Table 1). Even though the centres of tertiary vortices are included in some of the refined grids, we can not identify them in the refined mesh. Further mesh refinement is needed for more information on this matter.

3.1.2 Re = 2500

We show the figures for Re = 2500 generated from a mesh with 121×121 uniform grids. From Figure 4, the difference between the horizontal profile u of the numerical velocity field at $x = 0.5$ and the corresponding benchmark is small. The streamlines in the primary vortex region in Figure 5 are almost closed with very small errors. If the errors come from the process of generating of streamline, we conclude that the velocity field $\mathbf{V}_l$ satisfies Equation (1) or f satisfies the condition MC and there is no refinement in the region. If the errors come from the numerical velocity field, there are refinements in the region. There is no refinement in the primary vortex region in Figure 6 so the errors shown in Figure 5

 ISBN:978-84-615-5392-1

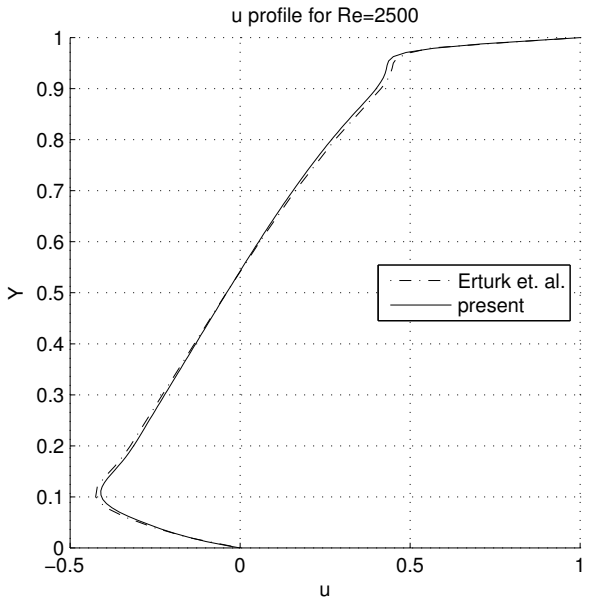


Figure 4: Horizontal profile of velocity field at $x = 0.5$ for mesh size 121×121.

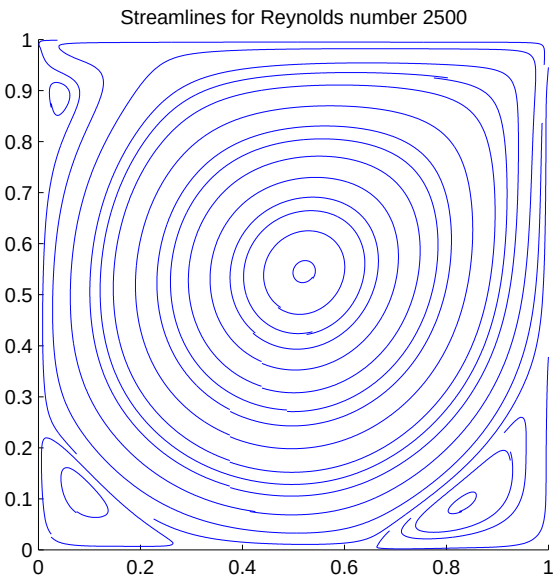


Figure 5: Streamlines for mesh size 121×121.

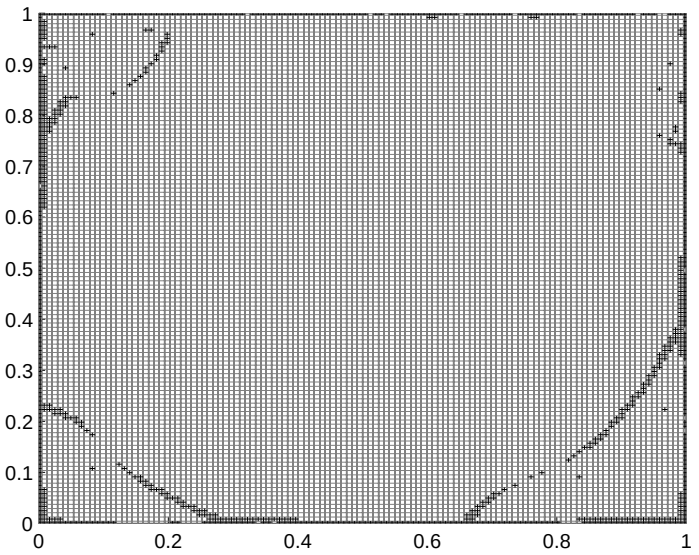


Figure 6: Refined mesh for Re = 2500 with mesh size 121×121.

come from the generation of the streamlines. The difference between the coordinates of the centre of primary vortex from the benchmark and linearly interpolated velocity field $\mathbf{V}_l$ is shown in Table 1. Even though the centre of primary vortex is not shown in Figure 6 due to the linearly interpolated velocity field $\mathbf{V}_l$ satisfying Equation 1 or $f\mathbf{V}_l$ satisfying condition MC in the primary region, the centre of an extra tertiary vortex is shown in the bottom right corner. This result is different from the case using 85×85 uniform grids [17]. The three isolated refined grids in the two bottom corners contain the centres of two secondary vortices (refer to Table 1) and the isolated grid located at the top of the bottom right corner comes from the error of the numerical velocity field. There are no such isolated grids for analytical velocity fields [11, 12].

3.1.3 Vortex centre locations

This subsection shows the comparison of the centres of vortices between the benchmarks and the corresponding estimates obtained in this paper.

Table 1 presents that coordinates of centres of vortices in the benchmark (blue) and the corresponding coordinates for Re = 1000 and 2500 from the linearly interpolated velocity fields $\mathbf{V}_l$. In this table, the abbreviations BR, BL and TL refer to bottom right, bottom left and top left corners of the cavity, respectively. The numbers following these abbreviations

Table 1: Locations of the centre of vortices

Vortex type	Reynolds numbers	
	Re = 1000	Re = 2500
Primary vortex	(0.5316,0.5659) (0.5300,0.5650)	(0.5202,0.5446) (0.5200,0.5433)
BR1	(0.8634,0.1128) (0.8633,0.1117)	(0.8318,0.0910) (0.8350,0.0917)
BL1	(0.0839,0.0779) (0.0833,0.0783)	(0.0845,0.1108) (0.0850,0.1100)
BR2	- (0.9917,0.0067)	(0.9851,0.0056) (0.9900,0.0100)
BL2	(0.0075,0.0075)) (0.0050,0.0050)	(0.0090,0.0083) (0.0067,0.0067)
TL1	- -	(0.0441,0.8904) (0.0433,0.8900)

refer to the vortices that appear in the flow, which are numbered according to size (e.g. BR1 refers to bottom right secondary vortex, and BR2 refers to bottom right tertiary vortex, etc.).

3.2 Refined grids containing centres of vortices

We take Re = 2500 as an example to verify if the centres of vortices are contained in some refined grids of the refined mesh except the centre of the primary vortex. If the centres of vortices are included in refined grids, further refinements of the mesh will provide more accurate estimate locations of the centres. The following figures show the sub plots of bottom left and bottom right corners of the refined mesh for Re = 2500. The red dots are the centres given in the benchmark [6]. We conclude that the centres are contained in refined grids in these enlarged sub plots clearly. The top left corner is the same as bottom left and bottom right corners. Figure 7 shows the refined mesh and the centres of vortices in region [0 0.3]×[0 0.3]. Figure 8 shows the refined mesh and the centres of vortices in region [0.6 1]×[0 0.4]. The centre of the secondary vortex is almost located at the centre of the refined grid.

4 Discussions

We considered the accuracy of the 2D adaptive mesh refinement method using two cases of 2D lid-driven cavity flows and finer meshes. We use horizontal profile of velocity fields

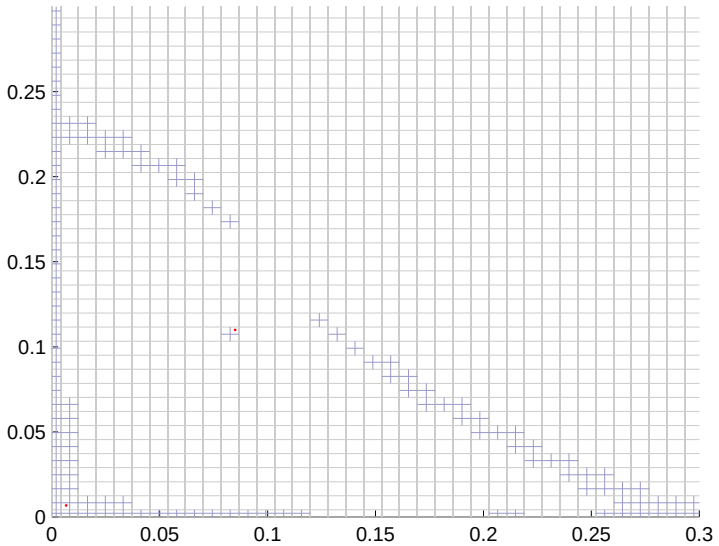


Figure 7: Sub plot of bottom left corn of refined mesh for Re = 2500.

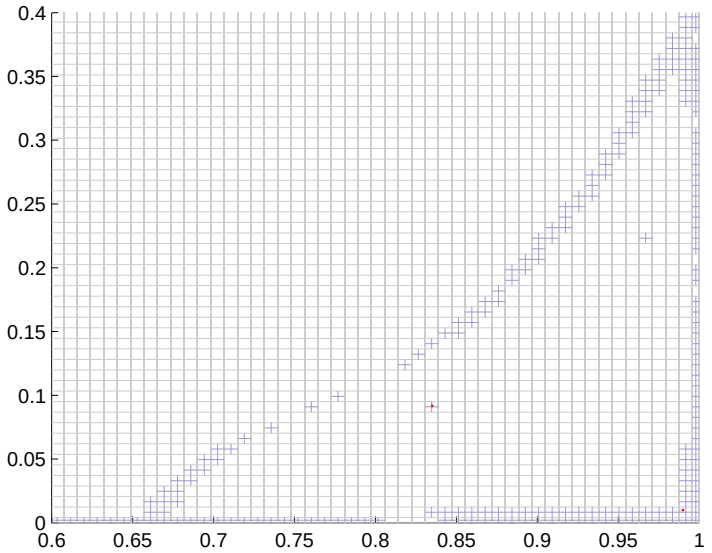


Figure 8: Sub plot of bottom right corn of refined mesh for Re = 2500.

 ISBN:978-84-615-5392-1

at $x = 0.5$ and the streamlines generated by *Matlab* built-in function *streamline* for the accuracy of the numerical velocity fields. We then consider whether the refined meshes can locate the centres of vortices. Besides the centre of primary vortex for Re = 2500 which has been estimated accurately, the other centres of vortices locate in the refined grids in the refined meshes. Further refinement for the refined meshes provide more accurate estimates for location of the centres of the vortices.

References

[1] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell and M. L. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations*, J. Comput. Phys. **142** (1998) 1–46.

[2] B. F. Armaly, F. Durst, J. C. F. Pereira and B. Schonung, *Experimental and theoretical investigation of backward-facing step flow*, J. Fluid Mech. **127** (1983) 473–496.

[3] J. Bell, M. Berger, J. Saltzman and M. Welcome, *Three-dimensional adaptive mesh refinement for hyperbolic conservation laws*, SIAM J. of Scient. Comput. **15** (1994) 127–138.

[4] P. J. Capon and P. K. Jimack, *An adaptive finite element method for the compressible Navier–Stokes equations*, in M. J. Baines and K. W. Morton, editors, Numerical Methods for Fluid Dynamics, 5. OUP, 1995.

[5] L. Demkowicz, J. T. Oden, W. Rachwicz and O. Hardy, *An h-p Taylor–Galerkin finite element method for the compressible Euler equations*, Comput. Methods Appl. Mech. Eng. **88** (1991) 363–396.

[6] E. Erturk, T. C. Corke and C. Gökcöl, *Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers*, Int. J. Numer. Methods Fluids **48** (2005) 747–774.

[7] S. Faure, J. Laminie and R. Temam, *Colocated finite volume schemes for fluid flows*, Commun. Comput. Phys. **4** (2008) 1–25.

[8] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*, Springer, 3ed, 2002.

[9] R. D. Henderson, *Adaptive spectral element methods for turbulence and transition*, in T. J. Barth and H. Deconinck, editors, High-Order Methods for Computational Physics, Springer-Verlag, Berlin Heidelberg, 1999.

[10] Z. Li, *A mass conservative streamline tracking method for two dimensional CFD velocity fields*, J. Flow Visual. Image Processing **9** (2002) 75–87.

[11] Z. Li, *An adaptive three-dimensional mesh refinement method based on the law of mass conservation*, J. Flow Visual. Image Processing **14** (2007) 375–395.

[12] Z. Li, *An adaptive two-dimensional mesh refinement method based on the law of mass conservation*, J. Flow Visual. Image Processing **15** (2008) 17–33.

[13] Z. Li and G. Mallinson, *Mass conservative fluid flow visualisation for CFD velocity fields*, KSME Int. J. **15** (2001) 1794–1800.

[14] Z. Li and R. Lal, *An application of a mesh refinement method based on the law of mass conservation*, Proc. of 2010 Int. conf. on Comput. Inf. Sci. (2010) 226–229 IEEE CPS.

[15] Z. Li, *An application of a mesh refinement to lid-driven cavity flow*, Proc. of FLUCOME 2011 paper no. 241.

[16] Z. Li and R. Lal, *Sensitivity analysis of a mesh refinement method using the numerical solutions of 2D steady incompressible driven cavity flow*, submitted for publication.

[17] Z. Li, *Accuracy analysis of an adaptive mesh refinement method using benchmarks of 2D steady incompressible lid-driven cavity flows*, submitted for publication.

[18] R. Lohner, *An adaptive finite element scheme for transient problems in CFD*, Comput. Methods Appl. Mech. Eng. **61** (1987) 323–338.

[19] K. Miller and R. Miller, *Moving finite elements, Part I*, SIAM J. Numer. Anal. **18** (1981) 1019–1032.

[20] M. C. Mosher, *A variable node finite element method*, J. of Comput. Phys. **57** (1985) 157–187.

[21] W. Speares and M. Berzins, *A 3-D unstructured mesh adaptation algorithm for time-dependent shock dominated problems*, Int. J. Numer. Methods Fluids **25** (1997) 81–104.

[22] Y. Ye and Others, *Theory of limit cycles*, American Mathematical Society Press, 1986.

[23] O. C. Zienkiewicz, D. W. Kelly and J. P. Gago, *The hierarchical concept in finite element analysis*, Comput. Struct. **16** (1983) 53–65.

 ISBN:978-84-615-5392-1

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

High-Order Energy-Conserved Splitting FDTD Scheme for Solving Maxwell's Equations

Dong Liang[1] **and Qiang Yuan**[1]

[1] *Department of Mathematics and Statistics, York University*

emails: dliang@mathstat.yorku.ca,

Abstract

In this work, we will present our new high-order energy-conserved splitting FDTD scheme for solving Maxwell's equations. The proposed scheme has the significant properties that are energy-conserved, unconditionally stable, non-dissipative, high-order accurate, and computationally efficient. We prove that the scheme satisfies energy conversations and is unconditionally stable. We analyze theoretically the convergence of the scheme by using the energy method and obtain the optimal spatial fourth-order error estimates in the discrete L_2-norm for the approximations of the electric and magnetic fields. Further, the divergence-free convergence is also analyzed and the error estimate of the approximation of divergence-free is obtained. Numerical experiments show that the proposed scheme preserves energy conservations and has high-order accuracy, which confirm our theoretical results.

Key words: Energy-conserved, S-FDTD, High order, Maxwell's equations
MSC 2000: 65M10, 65M15, 65N10, 65N15

1 Introduction

In computational electromagnetics, a very popular numerical method for solving Maxwell's equations is the finite-difference time-domain (FDTD) scheme, which was first introduced by Yee in [1] and was further developed in [2, 3], etc. The scheme has been widely applied to simulate transient electromagnetic wave propagations in a broad range of practical problems with perfectly electric conducting boundary conditions or absorbing boundary conditions. For problems requiring long-time integration and problems of wave propagations over longer distances, it has led to the development of high-order FDTD schemes which produce smaller dispersion or phase errors for a given mesh resolution. The fourth-order explicit schemes

were developed for solving Maxwell's equations in [4, 5], etc. However, some developed high-order FDTD schemes are conditionally stable and require large computational memory and huge computational cost. On the other hand, during the propagation of electromagnetic waves in lossless media without sources, the electromagnetic energy keeps constant for all time, which explains the physical feature of conservation of electromgnemtic energy in long term behavior.It thus is significantly important to physically keep the invariance of energy in time, for developing efficient numerical schemes in computation of Maxwell's equations and specially in a long term computation of electromagnetic fields. Based on the Yee's grid and splitting technique, [6] first proposed second-order energy-conserved splitting FDTD schemes. It was proved both theoretically and numerically that the EC-S-FDTD I&II schemes are energy-conserved and unconditionally stable and the EC-FDTDII scheme is of second order convergence in both time and space steps. Thus, it is very important and challenging to develop high-order splitting FDTD schemes which provide discrete energy conservations, unconditional stability, non-dissipativity, and higher-order accuracy.

2 Problems

The Maxwell's equations in an isotropic, homogeneous and lossless medium are

$$-\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t}, \tag{1}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t}, \tag{2}$$

where $\mathbf{E}$ and $\mathbf{H}$ are electric and magnetic fields; $\mathbf{D}$ and $\mathbf{B}$ are the electric displacement and magnetic flux density, $\mathbf{D} = \epsilon\mathbf{E}$ and $\mathbf{B} = \mu\mathbf{H}$. (1) is Faraday's Law and (2) is Ampere's Law. In the absence of electric charge, the electric displacement and magnetic flux density satisfy divergence-free conditions (Gauss's Law)

$$\nabla \cdot \mathbf{B} = 0, \tag{3}$$

$$\nabla \cdot \mathbf{D} = 0, \tag{4}$$

where ϵ is the electric permittivity and μ is the magnetic permeability. The speed of the electromagnetic wave is $c = \frac{1}{\sqrt{\epsilon\mu}}$.

For the simplicity of notations, we shall focus on the two-dimensional transverse electric (**TE**) problems in a lossless medium and without sources and charges, where the electric field is a plane vector while the magnetic field is a scalar. Let the domain $\Omega = [0, a] \times [0, b]$ and the time period $T > 0$. The electric and magnetic fields are $\mathbf{E} = (E_x(x, y, t), E_y(x, y, t))$ and H_z=$H_z(x, y, t)$. We consider the perfectly electric conducting (PEC) boundary condition:

$$(\mathbf{E}, 0) \times (\mathbf{n}, 0) = 0, \quad \text{on } (0, \mathbf{T}] \times \partial\Omega, \tag{5}$$

 ISBN: 978-84-616-2723-3

where $\mathbf{n}$ is the outward normal vector on the boundary. The initial conditions are given as

$$\mathbf{E}(x,y,0)=\mathbf{E}_0(x,y)=(E_{x0}(x,y),E_{y0}(x,y)) \text{ and } H_z(x,y,0)=H_{z0}(x,y). \tag{6}$$

3 The high-order EC-S-FDTD scheme and Resutls

We define the spatial fourth-order difference operator to $\frac{\partial}{\partial x}E_y$ for the strict interior nodes by a linear combination of two central differences, one with a spatial step and the other with three spatial steps above, as

$$\Lambda_x {E_y}^n_{i+\frac{1}{2},j+\frac{1}{2}}=\frac{1}{8}(9\delta_x-\delta_{2,x}){E_y}^n_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{7}$$

for $i=1,2,\cdots,I-2$ and $j=0,1,\cdots,J-1$. However, when we treat the near boundary nodes with $i=0$ and $i=I-1$, the function values in the definition of $\delta_{2,x}{E_y}^n_{i+\frac{1}{2},j+\frac{1}{2}}$ will go out the domain where $E_{y_{-1,j+\frac{1}{2}}}$ and $E_{y_{I+1,j+\frac{1}{2}}}$ are not defined. We propose the spatial fourth-order difference operator $\widetilde{\delta}_{2,x}E_y$ for the near boundary node with $i=0$ by

$$\widetilde{\delta}_{2,x}{E_y}^n_{\frac{1}{2},j+\frac{1}{2}}=\frac{E^n_{y_{1,j+\frac{1}{2}}}+E^n_{y_{2,j+\frac{1}{2}}}-2E^n_{y_{0,j+\frac{1}{2}}}}{3\Delta x}. \tag{8}$$

Thus, we can define the difference operators to approximate $\frac{\partial}{\partial x}E_y$ for the near boundary node with $i=0$ by

$$\widetilde{\Lambda}_x {E_y}^n_{\frac{1}{2},j+\frac{1}{2}} \quad = \quad \frac{1}{8}(9\delta_x-\widetilde{\delta}_{2,x}){E_y}^n_{\frac{1}{2},j+\frac{1}{2}}, \tag{9}$$

for $j=0,1,\cdots,J-1$.

Based on the proposed high-order difference operators on the strict interior node and the near boundary nodes, we proposed the high-order energy-conserved splitting FDTD scheme.

In this work, we develop and analyze the spatial high-order energy-conserved splitting FDTD scheme. One important issue is to construct the numerical boundary difference schemes to be energy conservative and high-order relative to the interior difference schemes. It is because the high-order difference operators often have a large spatial stencil which cannot be used in the near boundary nodes. The one-sided differences and extrapolation/interpolation numerical boundary schemes normally break the property of energy conservations near the boundary. The proposed scheme in this work has the significant properties that are energy-conserved, unconditionally stable, non-dissipative, high-order accurate, and computationally efficient. We prove that the scheme satisfies energy conversations and is unconditionally stable. We analyze theoretically the convergence of the scheme by using the energy method and obtain the optimal-order error estimates in the

 ISBN: 978-84-616-2723-3

discrete L_2-norm for the approximations of the electric and magnetic fields. Further, the divergence-free convergence is analyzed and we obtain the error estimate of the approximation of divergence-free. Numerical experiments show that the proposed scheme preserves energy conservations and has fourth-order accuracy in space.

Acknowledgements

This work was supported by Natural Sciences and Engineering Research Council of Canada.

References

[1] Yee, K.S., *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*, IEEE Trans. Antennas and Propagation, 14(1966), 302-307.

[2] Taflove, A. and Hagness, S., *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Second Ed., Artech House, Boston, MA, 2000.

[3] Monk, P. and Suli, E., A convergence analysis of Yee's scheme on nonuniform grids, SIAM J. Numer. Anal., 31 (1994), 393-412.

[4] Turkel, E., and Yefet, A., On the construction of a high order difference scheme for complex domains in a Cartesian grid, Appl. Numer. Math., 33 (2000), 113-124.

[5] Yefet, A. and Petropoulos, P.G., A non-dissipative staggered fourth-order accurate explicit finite difference scheme for the time-domain Maxwell's equation, J. Comput. Phys., 168 (2001), 286-315.

[6] Chen, W., Li, X. and Liang, D., *Energy-conserved splitting FDTD methods for Maxwell's equations*, Numer. Math., 108 (2008), 445-485.

[7] Namiki, T., *A new FDTD algorithm based on alternating direction implicit method*, IEEE Trans. Microw. Theory Tech., 47 (2003-2007), 1999.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Tracing traintors via elliptic curves

M.A. Lodroman[1] and J.A. Lopez-Ramos[1]

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

emails: antonela_lodroman@yahoo.com, jlopez@ual.es

Abstract

We introduce a protocol based on blind signatures using elliptic curves that allows to trace those users, in any service, that share their legitimate licence to access a service, with other people, avoiding the abuse or the unauthorized use of the legal licences. Blind signature are very useful to provide the users anonymity and the signers privacy. The scheme uses the inherent advantage of elliptic curve cryptosystem in terms of smaller key size and lower computational resources.

Key words: elliptic curve, blind signature
MSC 2000: AMS codes (optional)

1 Introduction

A digital signature provides proof of authenticity that a transaction originated from a particular sender but also reveals the identity of the individual in the process. Blind signatures provide the same authentication but do so in a non-identifiable or 'blind' manner. The recipient is assured of the fact that a transmission is authentic and reliable, without knowing who actually sent it.

Besides the property of blindness, a blind signature scheme must satisfy an additional requirement: unlinkability. This property refers to the fact that the signer cannot trace the requester of a blind signature after the corresponding message-signature pair has been published.

The blind digital signature is a digital signature protocol created by David Chaum in [4].

One of the goals of the blind signature schemes is to preserve the anonymity in transactions, and therefore are used in applications where the sender privacy is important. This includes various "digital cash" [4, 2] schemes and voting protocols [3].

Another situation where anonymity of users is desirable is the use of online services, nowadays widely used thanks to the popularization of platforms of video and/or audion on demmand. These services make use of what it is known as multicast secure schemes to distribute both the encrypted contents and the secret key to access them. The schemes that are becoming more popular are those with a computational approach due to the low requiremets of storing at the client part and because of their good properties concerning self-synchronism. Some specific examples of such multicast secure schemes based on computational approaches are presented in [5][6] and [7]. However the problem of identifying those users that abuse on the licences , i. e. those that share the private information that allow the access to the private contents is still and open problem.

Our aim is the use of a scheme based on blind signatures with elliptic curves to detect those users that share their licence or key in an illegal way in order to third people access some service in a fraudulently. Our idea is that as it is made in the case of double spending with digital cash, protocol that also uses blind signatures, traintors, i.e. those users that share their licence or key, will be used when a double access is detected. The protocol also allows anonymity to any user.

2 The protocol

The situation is as follows: A Server provides an online service, e.g. TV, audio streaming, antivirus, etc., that will be accessed only by using a determined key that it is also broadcasted in an encrypted way; the user recovers the key by using a private key that was previously distributed individually at the moment of subscription together with a unique licence corresponding to that individual key; everytime a user demands access to the service, he contacts an agent that verifies the validity of the licence and who will detect a fraudulent use in that case.

Thus in this protocol there exist three different parts:

1. A Server - that will provide the service as well as the session key to access it.
2. An Agent - that will check the validity of a licence shown by the user requesting the service.
3. The User.

2.1 Set up

We are assuming that the service provided by the Server makes use of a session key K_S.

Let E be an elliptic curve and P a point in E generating a subgroup of order a prime p and let q be a prime such that $q|p-1$. Then the Server computes a point G multiple of P of order q.

 ISBN: 978-84-616-2723-3

The Server computes

$$G_1 = k_1\, G \quad \text{and} \quad G_2 = k_2\, G$$

for k_1 and k_2 secret random numbers and it makes public

$$\{p, q, G, G_1, G_2\}.$$

The Server now chooses a secret identity x, computes and makes public

$$\{Q_0 = x\, G \quad Q_1 = x\, G_1 \quad Q_2 = x\, G_2\}.$$

We will also consider three different hash functions H_1, H_2 and H_3 whose inputs are 1, 5 and 4 integers and that outputs each one a unique integer modulus q.

2.2 The protocol

1. When a user access for the first time a session on the Server, he chooses a random secret integer u and sends $I = u\, G_1$ to the Server.

2. The Server stores I, together with some information that identifies this user and sends a private ticket T, used to recover the session key K_S and that will allow to access the service.

3. The Server chooses w, computes and sends back the triple

$$\{z' = x(I + G_2), \quad h = wG, \quad k = w(I + G_2)\}$$

4. The User chooses seven random secret numbers

$$(s, x_1, x_2, n_1, n_2, n_3, n_4)$$

and computes:

- $A = s(I + G_2)$
- $B = x_1 G_1 + x_2 G_2$
- $z = sz'$
- $a = n_1 h + n_2 G$
- $b = s n_1 k + n_2 A$
- $a' = n_3 h + n_4 G$
- $c \equiv_q n_1^{-1} H_2(A_{x_0}, B_{x_0}, z, a, b)$

 ISBN: 978-84-616-2723-3

- $d \equiv_q n_3^{-1} H_2(A_{x_0}, B_{x_0}, z, a, H_1(T))$, where $H_1(T)$ denotes the hash value of the ticket T and $A_{x_0},\ B_{x_0}$ denote the first coordinate of the points A and B respectively.

5. The User sends to the Server the pair $\{c, d\}$.

6. The Server computes and sends back the pair

$$\{c_1 \equiv_q cx + w, \quad d_1 = dx + w\}$$

7. The User computes $r \equiv_q n_1 c_1 + n_2$ and $r' \equiv_q n_3 d_1 + n_4$.

   The licence that corresponds to the ticket T will be

$$L = \{A, B, z, a, b, r, a', r'\}$$

Theorem 2.1 *The following equalities hold for the licence $L = \{A, B, z, a, b, r, a', r'\}$ corresponding to the ticket T:*

$$rG = a + Q_0 H_2(A_{x_0}, B_{x_0}, z, a, b),$$

$$rA = b + zH_2(A_{x_0}, B_{x_0}, z, a, b),$$

$$r'G = a' + Q_0 H_2(A_{x_0}, B_{x_0}, z, a, H_1(T))$$

Proof. Firstly we recall that $r' \equiv_q n_3 d_1 + n_4$ and so $r' \equiv_q n_3 dx + n_3 w + n_4$. Then $d \equiv_q n_3^{-1} H_2(A_{x_0}, B_{x_0}, z, a, H_1(T))$.

$a' + Q_0 H_2(A_{x_0}, B_{x_0}, z, a, H_1(T)) = n_3 h + n_4 G + xGH_2(A_{x_0}, B_{x_0}, z, a, H_1(T)) = n_3 wG + n_4 G + xGH_2(A_{x_0}, B_{x_0}, z, a, H_1(T)) = G(n_3 w + n_4 + xdn_3) = (n_3 dx + n_3 w + n_4)G = r'G$

Analogously we get that $rG = a + Q_0 H_2(A_{x_0}, B_{x_0}, z, a, b)$

We check that $b + zH_2(A_{x_0}, B_{x_0}, z, a, b) = rA$ since $H_2(A_{x_0}, B_{x_0}, z, a, b) \equiv_q n_1 c$ and $c_1 \equiv_q w + cx$. □

2.3 Accessing the contents

1. When the User wants to access the service, the Agent in charge of validating the licence requires him the pair $(h(T), L)$.

2. The Agent verifies equalities of Theorem 2.1. In that case (if the theorem holds) the Agent computes the hash value $H_3(A_{x_0}, B_{x_0}, H_1(T), t)$ where t denotes a timestamp, which is sent to the User.

 ISBN: 978-84-616-2723-3

3. The User computes

$$s_1 = H_3(A_{x_0}, B_{x_0}, H_1(T), t)us + x_1 \quad \text{and} \quad s_2 = H_3(A_{x_0}, B_{x_0}, H_1(T), t)s + x_2$$

where u is his own private information and s, x_1 y x_2 are the random numbers generated on step 4.

4. The Agent verifies that $s_1G_1 + s_2G_2 = A + BH_3(A_{x_0}, B_{x_0}, H_1(T), t)$ and in that case the Agent sends to the Server the pair $(h(T), L)$ together with the triple $(s_1, s_2, H_3(A_{x_0}, B_{x_0}, H_1(T), t))$.

5. The Server sends K_S to the User, encrypted using his private ticket T and the information concerning the licence will be stored while the user is still using the service.

Theorem 2.2 *With the above notation, the following equality holds*

$$s_1G_1 + s_2G_2 = AH_3(A_{x_0}, B_{x_0}, H_1(T), t) + B$$

Proof: Since $I = uG_1$, $A = s(I + G_2)$ and $B = x_1G_1 + x_2G_2$, we get
$s_1G_1 + s_2G_2 = (H_3(A_{x_0}, B_{x_0}, H_1(T), t)us + x_1)G_1 + (H_3(A_{x_0}, B_{x_0}, H_1(T), t)s + x_2)G_2 =$
$H_3(A_{x_0}, B_{x_0}, H_1(T), t)usG_1 + x_1G_1 + H_3(A_{x_0}, B_{x_0}, H_1(T), t)sG_2 + x_2G_2 =$
$H_3(A_{x_0}, B_{x_0}, H_1(T), t)sI + H_3(A_{x_0}, B_{x_0}, H_1(T), t)sG_2 + x_1G_1 + x_2G_2 =$
$H_3(A_{x_0}, B_{x_0}, H_1(T), t)s(I + G_2) + B = AH_3(A_{x_0}, B_{x_0}, H_1(T), t) + B$ □

Proposition 2.3 *The above protocol avoids the use of a determined licence by two different users simultaneously, protects and avoids the use of a stolen licence and the reuse of a recorded message previously utilised to get a determined service.*

Proof: Assume that a legal user shares all his private information, including the ticket, the licence and every number generated throughout the protocol, and the copy is used to authenticate $(H_1(T), L)$ along with a different triple $(s_1', s_2', H_3(A_{x_0}, B_{x_0}, H_1(T), t')$ while someone else is logged in the system using the same $(H_1(T), L)$ along with $(s_1, s_2, H_3(A_{x_0}, B_{x_0}, H_1(T), t)$. Then the Server will detect that there is a double use of the same licence and by computing $u \equiv_q (s_1 - s_1')(s_2 - s_2')^{-1}$, the Server will identify the traitor given by $I = uG_1$. Now considering that computing u from I involves solving the elliptic logarithm, it is highly probable that the user identified by I has share his private information to someone else.

Suppose now that an outsider records or steals a licence $(H_3(T), L)$ in order to get the service. Then the forger should know the private information u to generate s_1 and s_2. We note that generating s_1 and s_2 such that verify Theorem 2.2 involves solving the elliptic logarithm

$$s_1G_1 = AH_3(A_{x_0}, B_{x_0}, H_1(T), t) + B - s_2G_2$$

□

 ISBN: 978-84-616-2723-3

Proposition 2.4 *The above protocol provides anonymity to the users.*

Proof. This follows again by observing that identifying user $I = uG_1$ requires solving an elliptic logarithm even in case an attacker knows every number generated on step 4 for the given licence $(H_1(T), L)$. □

Acknowledgements

This work has been supported by the Spanish Ministry of Science and Innovation (TEC2009-13763-C02-02) and Junta de Andalucía (FQM 0211).

References

[1] Kin-Ching Chan and S.-H.G. Chan. Key management approaches to offer data confidentiality for secure multicast. *Network, IEEE, 17(5):30 - 39, sept. - oct. 2003.*

[2] D. Chaum. Blind signatures for untraceable payments. *Advances in Cryptology Proceedings of Crypto 82, page 199-203. (1983).*

[3] D.Chaum. Secret-ballot receipts: True voter-verifiable elections. *Security & Privacy, IEEE, 2(1):1 38 - 47, Jan.-Feb. 2004.*

[4] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. *Proceedings on Advances in Cryptology, Springer-Verlag, New York, (1990), 319327.*

[5] G.-H. Chiou and W.-T Chen. Secure broadcasting using the secure lock. *Software Engineering, IEEE Transactions on, 15(8):929 - 934, aug 1989.*

[6] Baofeng Liu, Wenjun Zhang and Tianpu Jiang. A scalable key distribution scheme for conditional access system in digital pay-tv system. *Consumer Electronics, IEEE Transactions on, 50(2):632 - 637, may 2004.*

[7] J.A.M. Naranjo, N. Antequera, L.G. Casado and J.A. López Ramos. A suite of algorithms for key distribution and authentication in centralized secure multicast environments. *Journal of Computational and Applied Mathematics, 236(12):3042 - 3051, 2012.*

[8] S. Zhu and S. Jajodia. Scalable group key management for secure multicast: A taxonomy and new directions. *Network Security - Springer, 29:57 - 75, 2010.*

[9] A. Kiayias and M. Yung. The vector-ballot e-voting approach. *In FC 2004 , volume 3110 of LNCS , pages 7289. Springer-Verlag, 2004.*

 ISBN: 978-84-616-2723-3

[10] A. Fujioka, T. Okamoto and K. Ohta: A Practical Secret Voting Scheme for Large Scale Elections. *ASIACRYPT, 1992*

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Analyzing GOP-based parallel strategies with the HEVC encoder

Otoniel López-Granado[1], Manuel P. Malumbres[1], Hector Migallón[1] and Pablo Piñol[1]

[1] *Department of Physics and Computer Architecture, Miguel Hernández University*

emails: otoniel@umh.es, mels@umh.es, hmigallon@umh.es, pablop@umh.es

Abstract

The HEVC is the very last video coding standard that significantly increases the computing demands to encode video to reach the limits on compression efficiency. Our interest is centered on applying parallel processing techniques to HEVC encoder in order to significantly reduce the computational power demands without disturbing the coding performance behavior. So, we propose several parallelization approaches to the HEVC encoder which are well suited to multicore architectures. Our proposals use OpenMP programming paradigm working at a coarse grain level parallelization we call GOP-based level. GOP-based approaches encode simultaneously several groups of consecutive frames. Depending on how these GOPs are conformed and distributed it is critical to obtain good parallel performance, taking also into account the level of coding performance degradation.

Key words: Parallel algorithms, video coding, HEVC, multicore, performance

1 Introduction

The new High Efficiency Video Coding (HEVC) standard has been recently developed by the Joint Collaborative Team on Video Coding (JCT-VC) which was established by the ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). This new standard will replace the current H.264/AVC [1] standard in order to deal with nowadays and future multimedia market trends. 4K definition video content is a nowadays fact and 8K definition video will not last to become a reality too. Furthermore, the new standard supports high quality color depth at 8 and 10 bit. The HEVC standard

ISBN: 978-84-616-2723-3

aims to provide a doubling in coding efficiency with respect to the H.264/AVC High profile, delivering the same video quality at half the bit rate.

Regarding complexity, HEVC decoder does not appear to be significantly different from the H.264/AVC one [2]. However, HEVC encoder is expected to be several times more complex than H.264/AVC encoder and will be a hot research topic in years to come. At the time of developing this work, the current version of the reference software, called the HEVC test model (HM), is HM 10.0 which corresponds to the HEVC text specification draft 10 [3]. A good overview of HEVC standard can be found in [4].

We can find in the literature several works about complexity analysis and parallelization strategies for the emerging HEVC standard as in [5] [6] [7]. Most of the available HEVC parallelization proposals are focused in the decoding side, looking for the most appropriate parallel optimizations at the decoder that provide real-time decoding of High-Definition (HD) and Ultra-High-Definition (UHD) video contents.

Currently, there are few works focused at the HEVC encoder. In [8] authors propose a fine-grain parallel optimization in the motion estimation module of the HEVC encoder allowing to perform the motion vector prediction in all prediction units (PUs) available at the Coding Unit (CU) at the same time. In [9] authors propose a parallelization inside the Intra prediction module that consist on removing data dependencies among subblocks of a CU, obtaining interesting speed-up results.

In this paper we will analyze the available parallel strategies in the HEVC standard and its viability over the HM reference software. Furthermore, we present a parallelization alternative for the HEVC encoder which is specially suited for low delay encoding profiles. Our proposal works at Group Of Pictures (GOP) processing level, following different parallel GOP-based strategies and analyzing the overall behavior in terms of complexity reduction and coding performance.

The remainder of this paper is organized as follows: In Section 2 an overview of the available profiles in HEVC and common condition test are presented. Section 3 provides an overview of the high-level parallelism strategies proposed in the HEVC standard. Section 4 presents the GOP-based parallel alternatives we propose for the low delay application profile, while in Section 5 a comparison between the proposed parallel alternatives is presented. Finally, in Section 6 some conclusions and future work are discussed.

2 HEVC profiles

In [10] the JCT-VC defines the common test conditions and software reference configurations to be used for HEVC experiments. In that paper it can be found a series of settings in order to evaluate HEVC video codec and to compare the different contributions made to it.

A total of 24 video sequences are specified, arranged in 6 classes. Also the Quantization Parameter (QP) and the set of configuration files for the encoding process are detailed.

Using these common conditions, makes easier to perform comparisons between innovative proposals. JCT-VC also provides a spreadsheet to calculate Rate-Distortion (RD) curves and the percentage of gain in bit rate, by using Bjontegaard-Delta (BD) measurements [11].

Classes from A to E include natural video sequences at diverse frame sizes. Class F comprises sequences than contain synthetic video in part of them or in its whole. Two of the sequences in class A have a bit depth of 10 bits and the rest of the sequences have a bit depth of 8 bits. The frame rate of the sequences ranges from 20 to 60 fps.

It is indicated to code and decode test sequences to compare them with the anchors at four QP values: 22, 27, 32, 37. For each of these points, bit rate and Peak Signal-to-Noise Ratio (PSNR) are obtained. With these values, RD-curves can be drawn and by using cubic interpolation or piecewise cubic interpolation, BD rate differences can be computed.

Configuration files are provided within reference software package [12]. There are 8 different test conditions which are a combination of 2 bit depths: Main (8 bits) and Main10 (10 bits) with 4 coding modes: All Intra (AI), Random Access (RA), Low-Delay B (LB), and Low-Delay P (LP). When a sequence has a bit depth of 8 bits and we choose to code it with one of the Main10 modes the coder converts each source sample value by multiplying it by 4. If a sequence has a bit depth of 10 bits and we choose to code it with one of the Main modes the coder converts every source sample by adding it 2 and dividing it by 4 in order to clip it to the [0, 255] range.

In All Intra mode every frame is coded as an I-frame i.e. it is coded without any motion estimation/compensation. So each frame is independent from the other frames in the sequence. This mode gets lower compression rates (compared to the other 3 modes) because P-frames and B-frames can usually obtain better compression rates than I-frames at the same quality level. On the other hand, the coding process for All Intra mode is faster than for the other 3 modes because no time is wasted in motion estimation. Every frame is coded in rendering order. Applications that require a fast encoding process and are not concerned about limited bandwidth or storage capacity, fit perfectly in this coding mode.

Random Access mode combines I-frames and B-frames. A B-frame is a frame that uses motion estimation/compensation in order to achieve good compression rates. Each block of a B-frame can use up to 2 reference frames, so in the coding process 2 lists of reference pictures are maintained. The GOP (Group Of Pictures) size used is 8. Reference frames can be located earlier or later than the frame we are currently coding. So, in this mode, coding (and decoding) order is not the same as rendering order. So as to allow navigating along the coded sequence (pointing to a certain moment) or to allow functions like fast forward, an I-frame is inserted periodically. Depending on the frame rate of each sequence the intra refresh period varies. We have a value of 16 for 20 fps, 24 for 24 fps, 32 for 30 fps, 48 for 50 fps, and 64 for 60 fps. The intra period is a multiple of 8 (the size of the GOP) which inserts an I-frame approximately every second. Applications that do not have time constraints (when coding a video sequence) and need features like the aforementioned fast

forward, are the target applications of this coding mode.

Low-Delay modes (LP and LB) code each frame in rendering order. First an I-frame is inserted in the coded bit stream and then only P-frames (or B-frames) are used for the rest of the sequence. GOP size is 4. All the reference pictures are located earlier than the current frame. This two modes achieve better compression performance than AI mode and do not suffer from the delay that RA mode introduces. Applications like video-conference which have bandwidth and time constraints can benefit from low delay modes.

3 HEVC high-level parallelism strategies

High-level parallel strategies may be classified in a hierarchical scheme depending on the desired parallel grain size. This classification should carefully be applied taking into account the available parallel hardware resources in order to perform the most adequate and efficient implementation. So, we define from coarser to finer grain parallelism levels: GOP, tile, slice, and wavefront. When designing a HEVC parallel version we first analyze the available hardware where the parallel encoder will run, in order to determine which parallelism levels are the most appropriate.

The coarsest parallelization level, GOP-based, is based on breaking the whole video sequence in GOPs in such a way that the processing of each GOP is completely independent from the other GOPs. In general, this approach will be the one that best parallel efficiency should provide. However, depending on the way we define the GOPs structure and remove the inter-GOP dependencies, the coding performance may be affected.

Tiles are used to split a picture horizontally and vertically into multiple sub pictures. By using Tiles, prediction dependencies are broken just at Tile boundaries. Consecutive tiles are represented in raster scan order. The scan order of Coding Tree Blocks (CTBs) remains a raster scan. When splitting a picture horizontally, tiles may be used to reduce line buffer sizes in an encoder, as it operates on regions narrower than a full picture. Tiles also permit the composition of a picture from multiple rectangular sources that are encoded independently.

Slices follow the same concept as in H.264/AVC allowing a picture to be partitioned into groups of consecutive Coding Tree Units (CTUs) in raster scan order, each for transmission in a separate network adaptation layer unit that may be parsed and decoded independently, except for optional inter slice filtering. There is a break in prediction dependences at slices boundaries, which causes a loss in coding efficiency. The use of slices is more concerned with error resilience or maximum transmission unit size matching than a parallel coding technique, although it has undoubtedly been exploited for this purpose in the past.

Wavefronts split a picture into CTU rows, where each CTU row may be processed by a different thread. Dependences between rows are maintained except for the CABAC [13] context state, which is reinitialized at the beginning of each CTU row. To improve

the compression efficiency, rather than performing a normal CABAC reinitialization, the context state is inherited from the second CTU of the previous row.

All high-level parallelization tools become more useful with image sizes growing beyond HD for both encoder and decoder. At small image sizes where real-time decoding in a single-threaded manner is possible, the overhead associated with parallelization might be too high for there to be any meaningful benefit. For large image sizes it might be useful to enforce a minimum number of picture partitions to guarantee a minimum level of parallelism for the decoder.

Current HM reference software does not directly support most of the high-level parallelism approaches mainly due to its implementation design. In the next section we will present several GOP-based parallelization approaches that may be implemented in cluster-based or multicore-based hardware architectures.

4 Parallel algorithms

In previous sections we have reviewed the main features of the HEVC video compression standard. We have parallelized the HEVC reference software using LB and AI modes, both of them combined with Main profile (bit depth of 8 bits). This two modes are useful for applications that have time constraints, so we think they can benefit from parallelization strategies. Obviously, this work can be easily extended to use Low-Delay P mode. But this is not true for Random-Access mode due to the way it uses reference frames. In particular, Random-Access mode uses both past and future frames as reference pictures so dependencies between frames are tighter than in the two evaluated modes.

The developed parallel algorithms are designed at GOP-based parallelization level. First of all, note that in AI mode the GOP size is 1, because all frames are computed as I-frames (no reference frames are used at all). In LB mode the GOP size used is 4, however this value could be changed. On the other hand, we have considered synchronous algorithms where the synchronization process is performed after the GOP computation. We have developed four parallel approaches:

- Option I: (LB) in this option we sequentially assign each GOP to one process in the parallel execution, so processes will encode isolated GOPs.

- Option II: (LB) in this approach we divide the sequence in as many parts as the number of parallel processes, so that each process will encode a block of adjacent GOPs.

- Option III: (LB) similar to Option II, except that each process begins the coding by inserting an I-frame.

 ISBN: 978-84-616-2723-3

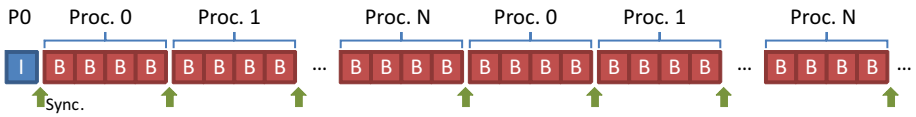


Figure 1: Option I: Parallel distribution.

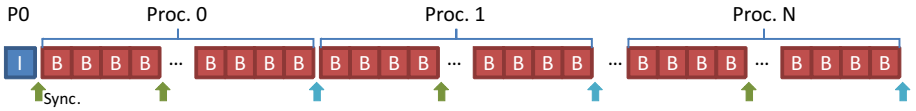


Figure 2: Option II: Parallel distribution.

- Option IV: (AI) similar to Option I where each GOP is sequentially assigned to a process, but here a GOP consists of only one I-frame.

Figure 1 shows the parallel distribution performed when Option I is used. Note that the synchronization processes are located after each GOP. The root process (Proc. 0 or P0) computes the first frame as an I-frame. After that, all processes encode a GOP of 4 B-frames. All processes, except the root process, encode their first B-frame without reference pictures, therefore the number of bits needed to encode this first B-frame is similar to the number of bits needed to encode an I-frame.

In order to increase the performance of the proposed parallel algorithms, each process will have its own working buffers. This fact changes the real pattern of the reference pictures used. For instance, in sequential processing the second B-frame of a GOP uses frames *-1 -2 -6 -10* as reference pictures (*-1* means the previous frame, and so on). As the GOP size is 4, frame *-2* points to the last frame of the previous GOP (the frame two positions before the current frame in the original video sequence). In parallel processing, as we assign isolated GOPs to each process, the previous GOP is not the previous adjacent GOP in the original video sequence and therefore frame *-2* will not point to the frame two positions before the current frame. If, for instance, the number of processes is 6, then the previous GOP for this process will be located in the video sequence 6 GOPs away from the current GOP. So for the second B-frame of a GOP, the reference picture *-2* will point to frame *-22 (-2-(6-1)x4=-22)* in the original video sequence. We can conclude that both parallel and sequential algorithms will produce different bit streams. We will analyze, in Section 5, the impact of this fact in terms of PSNR and bit rate.

In Figure 2 we can see a representation of the Option II parallel distribution. As in Option I, the synchronization processes are located after each GOP. Then, all processes, except the root process, encode their first B-frame without reference pictures. Note that the root process encodes the first I-frame. In this case the reference pictures are not significantly disturbed, because each process works with a group of adjacent GOPs. In the previous

 ISBN: 978-84-616-2723-3

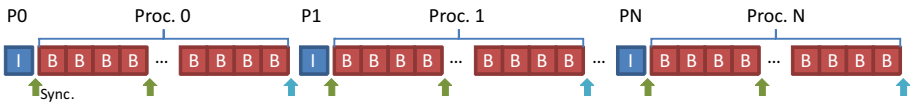


Figure 3: Option III: Parallel distribution.

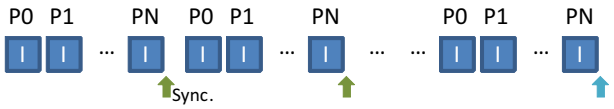


Figure 4: Option IV: Parallel distribution.

example, for the second frame of the GOP, the pattern is only altered for the first three GOPs. From this point onward all reference pictures needed are available in the private working buffer of each process.

Figure 3 shows the parallel distribution of Option III, where the parallel structure is similar to Option II. Here each process starts the encoding procedure computing the first frame as an I-frame. In this case, the parallel and sequential executions can be exactly the same if in the sequential execution we perform a slight change in the standard configuration.

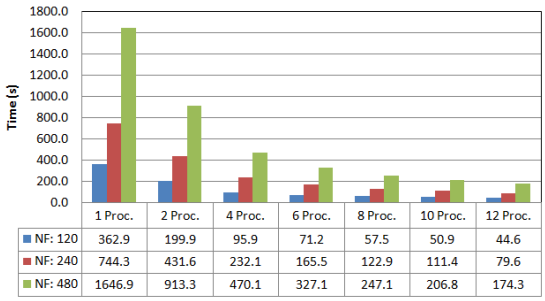
In order to get the same bit stream with both the parallel and sequential algorithms we must change the *IntraPeriod* parameter according to the number of processes of the parallel execution. Table 1 shows the value of the *IntraPeriod* parameter when we compute 240 and 480 frames. Moreover the I-frame included must be an IDR (Instantaneous Decoding Refresh), so we set the *DecodingRefreshType* parameter equal to 2.

Finally, in Figure 4 the parallel distribution for Option IV is shown. Note that the parallel structure is similar to the parallel structure of Option I, but the GOP always consists of one I-frame. Moreover the I-frames are IDRs, therefore there are no differences between the parallel and the sequential execution.

Number of Processes	240 frames	480 frames
2	120	240
4	60	120
6	40	80
8	*30*	60
10	24	48
12	20	40

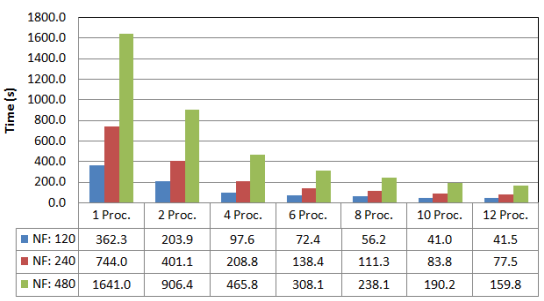
Table 1: Option III: *IntraPeriod* parameter to match sequential and parallel execution.

 ISBN: 978-84-616-2723-3



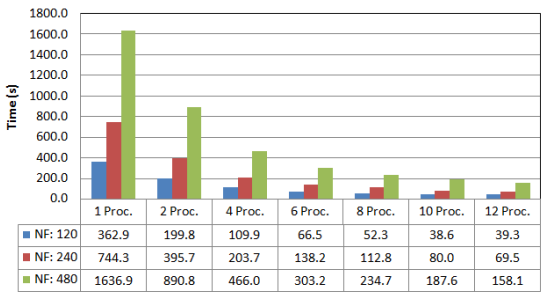
	1 Proc.	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	362.9	199.9	95.9	71.2	57.5	50.9	44.6
NF: 240	744.3	431.6	232.1	165.5	122.9	111.4	79.6
NF: 480	1646.9	913.3	470.1	327.1	247.1	206.8	174.3

(a) Option I.



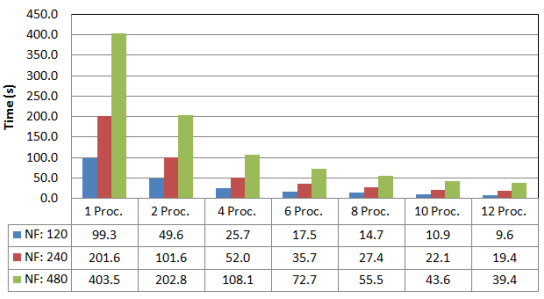
	1 Proc.	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	362.3	203.9	97.6	72.4	56.2	41.0	41.5
NF: 240	744.0	401.1	208.8	138.4	111.3	83.8	77.5
NF: 480	1641.0	906.4	465.8	308.1	238.1	190.2	159.8

(b) Option II.



	1 Proc.	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	362.9	199.8	109.9	66.5	52.3	38.6	39.3
NF: 240	744.3	395.7	203.7	138.2	112.8	80.0	69.5
NF: 480	1636.9	890.8	466.0	303.2	234.7	187.6	158.1

(c) Option III.



	1 Proc.	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	99.3	49.6	25.7	17.5	14.7	10.9	9.6
NF: 240	201.6	101.6	52.0	35.7	27.4	22.1	19.4
NF: 480	403.5	202.8	108.1	72.7	55.5	43.6	39.4

(d) Option IV.

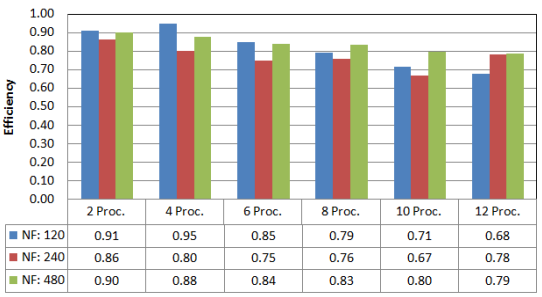
Figure 5: Computational times for the parallel algorithms. 120, 240 and 480 frames.

5 Numerical experiments

We will analyze the parallel algorithms described in Section 4, in terms of parallel performance, PSNR and bit rate. First of all, we have used the OpenMP [14] programming paradigm.

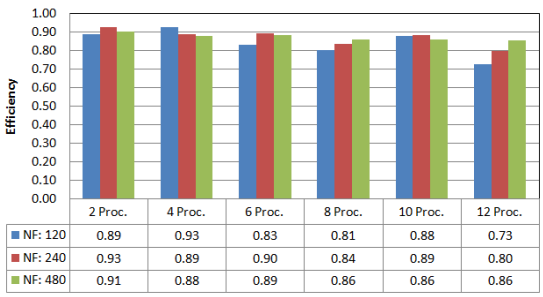
The multicore platform used is a HP Proliant SL390 G7 with two Intel Xeon X5660, each CPU with six cores at 2.8 GHz, therefore the experiments reported use up to 12 processes. The testing video sequence used is *BasketballPass_416x240.yuv*, and disposes of 500 frames at $50Hz$ with a frame size equal to $416x240$ pixels. We have run the parallel algorithms encoding 120, 240 and 480 frames and low delay mode. Note that the low delay profile sets the GOP size equal to 4, computes only the first frame as I-frame and the value of quantization parameter (QP) is equal to 32.

In Figure 5 we present the computation times for Option I, Option II, Option III, and Option IV parallel algorithms. The results show good parallel behavior in all cases. On the other hand, when using just 1 process, all the proposed algorithms show the same timings than the ones obtained with the sequential version. Respect to Option IV, the reference



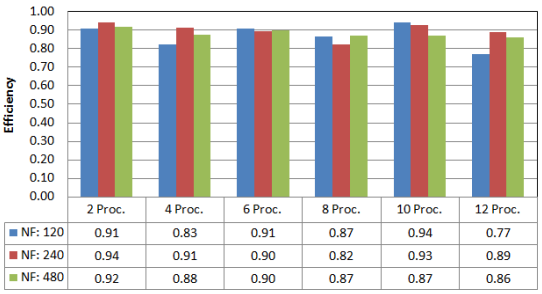
	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	0.91	0.95	0.85	0.79	0.71	0.68
NF: 240	0.86	0.80	0.75	0.76	0.67	0.78
NF: 480	0.90	0.88	0.84	0.83	0.80	0.79

(a) Option I.



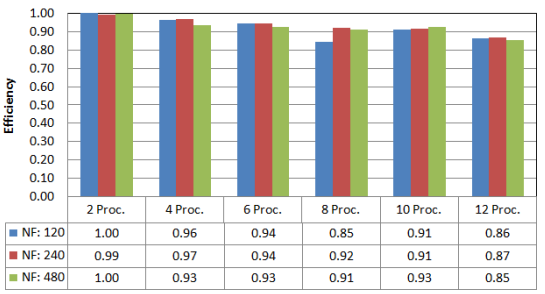
	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	0.89	0.93	0.83	0.81	0.88	0.73
NF: 240	0.93	0.89	0.90	0.84	0.89	0.80
NF: 480	0.91	0.88	0.89	0.86	0.86	0.86

(b) Option II.



	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	0.91	0.83	0.91	0.87	0.94	0.77
NF: 240	0.94	0.91	0.90	0.82	0.93	0.89
NF: 480	0.92	0.88	0.90	0.87	0.87	0.86

(c) Option III.



	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
NF: 120	1.00	0.96	0.94	0.85	0.91	0.86
NF: 240	0.99	0.97	0.94	0.92	0.91	0.87
NF: 480	1.00	0.93	0.93	0.91	0.93	0.85

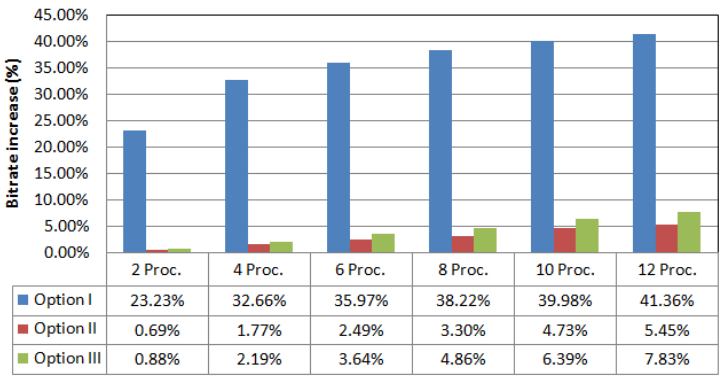
(d) Option IV.

Figure 6: Efficiency for the parallel algorithms. 120, 240 and 480 frames.

sequential execution is not the same, as we are using AI mode configuration.

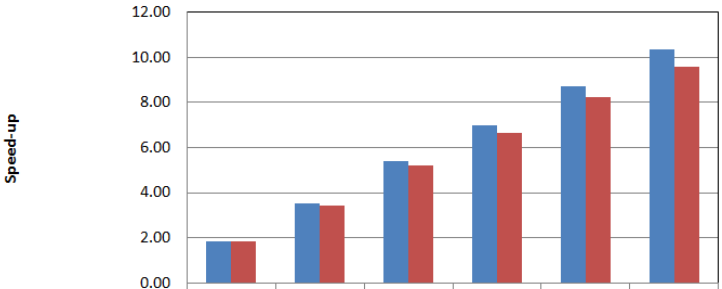
Figure 6 shows the efficiency associated to the results shown in Figure 5. This figure confirms the good behavior of the proposed parallel algorithms, obtaining nearly ideal efficiencies in some cases, and above 0.85 in most of the performed experiments. We want to remark that Option IV obtains an average efficiency greater than 0.95.

As it was said in Section 4, the parallel versions do not provide the same results than the ones produced by the sequential algorithm. So, in Figure 7 we show how parallel versions modify the sequential version bit rate. It is important to remark that Figure 7 shows results for Option I, II and III, but not for Option IV, because in the last case the parallel and sequential versions exhibit the same bit rate. Furthermore, we can observe that the bit rate increase introduced by Option I algorithm is not acceptable. This algorithm drastically changes the structure of the reference pictures, and as a consequence it causes the large bit rate increase shown in Figure 7. In all cases the bit rate increase becomes larger as the number of processes does. Note that the first frames of each process are encoded without reference frames, lineally increasing the bit rate as the number of processes increases. Finally, the bit rate increase is greater in Option III, because the initial I-frame



	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
Option I	23.23%	32.66%	35.97%	38.22%	39.98%	41.36%
Option II	0.69%	1.77%	2.49%	3.30%	4.73%	5.45%
Option III	0.88%	2.19%	3.64%	4.86%	6.39%	7.83%

Figure 7: Percentage of bit rate increase for the parallel algorithms. 480 frames.



	2 Proc.	4 Proc.	6 Proc.	8 Proc.	10 Proc.	12 Proc.
Speed-up NON_EQUIV	1.84	3.51	5.40	6.97	8.73	10.36
Speed-up EQUIV	1.82	3.44	5.19	6.65	8.22	9.60

Figure 8: Speed-up for Option III parallel algorithms. 480 frames.

is encoded with higher quality than the initial B-frame in Option II algorithm, as specified in the low delay profile configuration.

Table 2 shows the PSNR data, i.e. a quality measurement, for the parallel algorithms II and III. We can observe that using Option II algorithm the quality of the encoded video decreases, although, in Figure 7, we have showed that the bit rate increases. In contrast, the bit rate increase for Option III algorithm showed in Figure 7 is compensated by a quality increase as can be seen in Table 2.

Finally, we modify the low delay profile configuration in order to obtain the same PSNR and bit rate results with both parallel and sequential versions of Option III algorithm. Figure 8 shows as *Speed-up NON_EQUIV* the speed-up when parallel and sequential algorithms obtain slightly different results, and as *Speed-up EQUIV* the speed-up when they provide equivalent executions. We can conclude that the proposed Option III algorithm obtains

Algorithms	1 Proc	2 Proc	4 Proc	6 Proc	8 Proc	10 Proc	12 Proc
Option II	33.23	33.23	33.20	33.18	33.17	33.16	33.15
Option III	33.23	**33.26**	**33.31**	**33.35**	**33.39**	**33.44**	**33.47**

Table 2: Luminance PSNRs (dB) for parallel algorithms. 480 frames.

good efficiencies in the range of 0.80 (worst case) to 0.91.

6 Conclusions

In this paper we have proposed several parallel algorithms of the HEVC video encoder. These algorithms are based on a coarser grain parallelization approach with the organization of video frames in GOPs and different GOP process allocation schemes. They are specially suited for multicore architectures. After implementing the algorithms in the HEVC software under low delay mode, some experiments were performed showing interesting results as (a) GOP organization determines the final coding performance, being the best approach the Option IV (AI mode) when comparing both sequential and parallel versions in terms of speed-up/efficiency, and (b) Although the Option III algorithm introduces a bit rate overhead as the number of processes increase, the overall parallel performance and the improvements in PSNR make it a good approach when LB coding mode is demanded.In general, all proposed versions attain high parallel efficiency results, showing that GOP-based parallelization approaches should be taken into account to reduce the HEVC video encoding complexity. As future work, we will explore hierarchical parallelization approaches combining GOP-based approaches with slice and wavefront parallelization levels.

Acknowledgements

This research was supported by the Spanish Ministry of Education and Science under grant TIN2011-27543-C03-03, the Spanish Ministry of Science and Innovation under grant TIN2011-26254 and Generalitat Valenciana under grant ACOMP/2013/003.

References

[1] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," *ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC) version 16, 2012.*

[2] J. Ohm, G. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including high efficiency video coding

(hevc)," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1669–1684, 2012.

[3] B. Bross, W. Han, J. Ohm, G. Sullivan, Y.-K. Wang, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 10," *Document JCTVC-L1003 of JCT-VC, Geneva*, January 2013.

[4] G. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1648 –1667, December 2012.

[5] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1685–1696, 2012.

[6] M. Alvarez-Mesa, C. Chi, B. Juurlink, V. George, and T. Schierl, "Parallel video decoding in the emerging HEVC standard," in *International Conference on Acoustics, Speech, and Signal Processing, Kyoto*, March 2012, pp. 1–17.

[7] E. Ayele and S.B.Dhok, "Review of proposed high efficiency video coding (HEVC) standard," *International Journal of Computer Applications*, vol. 59, no. 15, pp. 1–9, 2012.

[8] Q. Yu, L. Zhao, and S. Ma, "Parallel AMVP candidate list construction for HEVC," in *VCIP'12*, 2012, pp. 1–6.

[9] J. Jiang, B. Guo, W. Mo, and K. Fan, "Block-based parallel intra prediction scheme for HEVC," *Journal of Multimedia*, vol. 7, no. 4, pp. 289 –294, August 2012.

[10] F. Bossen, "Common test conditions and software reference configurations," Joint Collaborative Team on Video Coding, Geneva, Tech. Rep. JCTVC-L1100, January 2013.

[11] G. Bjontegaard, "Improvements of the BD-PSNR model," Video Coding Experts Group (VCEG), Berlin (Germany), Tech. Rep. VCEG-M33, July 2008.

[12] HEVC Reference Software, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-10.0/.

[13] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 620–636, 2003.

[14] "Openmp application program interface, version 3.1," *OpenMP Architecture Review Board. http://www.openmp.org*, 2011.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Algorithms to develop semi-analytical planetary theories using Sundman generalized anomalies as temporal variables with aid of a C++ Poisson series processor

José A. López Ortí[1,2], Vicente Agost Gómez[2] and Miguel Barreda Rochera[1,2]

[1] *Institute of Mathematics and Applications of Castellón, University Jaume I of Castellón. Spain*

[2] *Department of Mathematics, University Jaume I of Castellón. Spain*

emails: lopez@uji.es, agostv@uji.es, barreda@uji.es

Abstract

One of the main problems in celestial mechanics is the study of the motion of the planets in the solar system. To solve this problem numerical, semi-analytical and analytical methods can be used. This paper is focused on the study of semi-analytical theories, these theories involve the use of the Fourier Poisson series developments depending on entire combinations of the anomalies. Poisson series involve two problems one the slow convergence rate when the eccentricities or mutual inclination of orbirs are not small and the presence of small denominators in the integration process that can induce great inequalities. In this paper we show that the use of an appropriate anomaly in the genelalized Sundman family con improve the convergence rate and the value of the great inequalities.

Key words: Celestial Mechanics. Planetary Theories. Algorithms. Orbital Mechanics. Perturbation Theory. Computational Algebra.
MSC 2000: 70F05, 70F10, 70F15,70M20.

1 Introduction

One of the main problems in celestial mechanics is the study of the motion of the main bodies of the solar system. Its solutions are the so-called planetary theories. To obtain these solutions there are three main procedures: numerical methods, based on the integration by

the appropriate numerical methods of the differential equation of the motion; the analytical ones, based on the literal expansion of the second member of the Lagrange planetary equations [10], [14] and the semi-analytical theories, based on the use of numerical values for the amplitude coefficients of developments and literal values for the anomalies. The second and third ways are based on the integration of the differential equations through the solution of the well known two body problem and using the perturbation theory.

To integrate these differential equations using analytical or semi-analytical methods it is necessary to develop the second member of Lagrange planetary equations as Fourier series of the anomalies. Classical methods use the mean anomaly (or mean longitude) of the planets as temporal variables.

In the year 1977 Nacozy [13], in order to improve the performance of the numerical methods, introduces the intermediate anomaly as an extension of the Sundman transformation $dt = Cr^{\alpha}d\tau$ defined by $\Psi_{\alpha} = \Psi_{\alpha}(M)$, where $\Psi_{\alpha}(M)$ satisfies that $\Psi_{\alpha}(M)$ is a 2π periodic function of M, $\frac{d\Psi_{\alpha}}{dM} > 0$, $\Psi_{\alpha}(-M) = -\Psi_{\alpha}(M)$, $\Psi_{\alpha}(0) = 0$, $\Psi_{\alpha}(2\pi) = 2\pi$. This family includes the mean ($\alpha = 0$), eccentric ($\alpha = 1$), true ($\alpha = 2$) and intermediate ($\alpha = \frac{3}{2}$) anomalies.

The analytical properties of the generalized Sundman anomalies have been studied by Lopez [12], these proprerties include

$$\Psi_{\alpha} = M + \sum_{s=0}^{\infty} \Psi_s(e, \alpha) \sin\, sM \tag{1}$$

the Kepler equation obtained through the use of the Deprit algorithm [6].

$$M = \Psi_{\alpha} + \sum_{s=1}^{\infty} H_s(e,_{\alpha}) \sin\, s\Psi_{\alpha} \tag{2}$$

and the developments of the quantities r, $\sin V$ and $\cos V$ where r is the vector radius, and V the true anomaly.

2 Integration of the Lagrange equations

To integrate the Lagrange planetary equations by semi-analytical methods it is necessary to develop the second member of the planetary equations as Fourier series of the selected anomalies [1],[7],[3],[10]. To obtain these developments the main problem is to develop the inverse of the distance between two planets. For this purpose we can use the Kovalevsky iteration algorithm [9], [5]

$$\left(\frac{1}{\triangle}\right)_{k+1} = \frac{3}{2}\left(\frac{1}{\triangle}\right)_{k} - \frac{1}{2}\left(\frac{1}{\triangle}\right)_{k}^{3} \triangle^2 \tag{3}$$

 ISBN: 978-84-616-2723-3

where $\triangle$ is the distance between the two planets.

The quantity $\triangle^2$ can be computed using the developments of r, $\sin V$, $\cos V$ with great accuracy.

An appropriate first approximation for the inverse of the distance is given by

$$\frac{1}{\triangle_0} = \frac{1}{a'}\left[b_{1/2}^{(0)} + \sum_{j=1}^{\infty} b_{1/2}^{(j)}(\alpha)\cos j(\lambda - \lambda')\right] \tag{4}$$

where $b_s^{(j)}$ are the Laplace coefficients [14], and λ, λ' the mean longitudes of the planets.

The use of the Kovalewsky algorithm produces a development

$$\frac{1}{\triangle} = \sum_{k_1,k_2} [A_{k1,k_2}\cos(k_1\Psi_1 + k_2\Psi_2) + B_{k1,k_2}\sin(k_1\Psi_1 + k_2\Psi_2)] \tag{5}$$

and from them the osculating elements σ satisfies

$$\dot{\sigma} = \sum_{k_1,k_2} [S_{k1,k_2}\cos(k_1\Psi_1 + k_2\Psi_2) + R_{k1,k_2}\sin(k_1\Psi_1 + k_2\Psi_2)] \tag{6}$$

To integrate with respect of t each term of the series using the generalized Sundman anomalies Ψ_1, Ψ_2 of two planests as temporal variables we can proceed as [11].

To handle with this developments a new C++ Poisson series processor poisson.h has been developed. A Poisson series processor is a especialised software package to manage aritmethic, and functional operations with Poisson series [8],[4].

3 Concluding Remarks

The use of appropriate anomalies in the generalized Sundman family can improve the convergence rate of the Poisson series developments of the osculating elements of the bodies, especially if the eccentricities are not small.

In the case of the small denominators, the value of the induced great inequalities can be decreased using an aprorpiate value of the parameters α in the generalized Sundman family.

Acknowledgements

This research has been partially supported by Grant P1-1B2012-47 from Universidad Jaume I of Castellón.

References

[1] Abu-El-Ata, N.,Chapront, J. 1975. Développements analytiques de l'inverse de la distance en mécanique céleste. *Astron. Astrophys.*, **38**, 57-66.

[2] Broucke, R.,Smith, C. 1971. Expansion of the planetary disturbing function. *Celestial Mechanics.* **4**, 490–499

[3] Brower, D., Clemence, G.M. 1965. *Celestial Mechanics*, Ed Academic Press, New York.

[4] Brumberg, V.A. 1995. *Analytical Techniques of Celestial Mechanics*, Ed Springer-Verlag.

[5] Chapront J., Bretagnon P., Mehl, M. 1975. Une Formulaire pour le calcul des perturbations d'ordres élevés dans les problèmes planétaires. *Celestial Mechanics.* **11** 379–399

[6] Deprit, A.,A Note on Lagrange's Inversion Formula,*Celestial Mechanics.* **20** (1979) 325–327.

[7] Hagihara, Y. 1970. *Celestial Mechanics.* **vol 2.**, Ed MIT Press, Cambridge MA.

[8] Ivanova, T. 2001. A new echeloned Poisson series processor (EPSP). *Celestial Mechanics.* **80**, 167–176.

[9] Kovalevsky, J. 1967. *Introduction to Celestial Mechanics*, Ed D. Reidel Publishing Company, DoDrecht-Holland.

[10] Levallois, J.J., Kovalevsky, J. 1971. *Géodésie Générale Vol 4*, Ed Eyrolles, Paris.

[11] López,J.A.,Mártinez, M.J., Marco, F.J. 2008. A Formulation to Obtain Semi-analytical integration algorithms based on the use of several kinds of anomalies as temporal variable. *Planetary and Space Science. .* **56** 1862–1868.

[12] López, J.A., Agost, V., Barreda M. 2012. A Note on the Use of the Generalized Sundman Transformations as Temporal Variables in Celestial Mechanics. *International Journal of Computer Mathematics.* **89.**, 433–442.

[13] Nacozy, P. 1977, The intermediate anomaly. *Celestial Mechanics.* **16** 309–313.

[14] Tisserand, F. 1896. *Traité de Mécanique Céleste*, Ed Gauthier-Villars, Paris.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Computationally efficient algorithm for mesh refinement based on octrees and linked lists

M. López-Portugués[1], J. A. López-Fernández[1], D. Marful-Díaz[1], R. G. Ayestarán[1] and F. Las-Heras[1]

[1] *Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y de Sistemas, Universidad de Oviedo, Spain*

emails: mlopez@tsc.uniovi.es, jelofer@tsc.uniovi.es, UO179576@uniovi.es, rayestaran@tsc.uniovi.es, flasheras@tsc.uniovi.es

Abstract

In this work, we present an efficient algorithm that performs the refinement of 3D meshes composed of triangular facets. Each original facet is subdivided into four new ones and this process may be repeated iteratively as many times as necessary in order to obtain the desired level of refinement. The method is based on an octree spatial subdivision with Z-ordering (also known as Gray encoding) that facilitates the task of avoiding the insertion of duplicated vertices. The computational complexity of the developed solution is $O(N)$ in terms of time. Moreover, in order to reduce the memory usage (which may be a limiting factor), we use linked lists to store the vertices in each octree box. Finally, we show some results that demonstrate the low runtime and the moderate memory consumption of our solution. Thus, this method is ideal to produce the meshes composed of millions (or even billions) of facets that the most challenging scattering problems require.

Key words: mesh refinement, octrees, algorithmic efficiency

1 Introduction

In the field of acoustic and electromagnetic scattering, it is of great interest to analyze realistic objets in a wide range of frequencies. The increasing efficiency of the techniques that compute the scattered field [1, 2, 3] is widening its range of application to big size objects [4, 5].

By means of methods such as the *Boundary Elements Method* (BEM) [6] or the *Method of Moments* (MoM) [7] it is possible to numerically solve the scattering problem. In order to obtain accurate results, the BEM and MoM require between 6 and 10 basis functions (or elements) per linear wavelength. As a consequence, the analysis of big objects may require geometrical meshes with a huge number of facets that may need more than one billion unknowns for the biggest problems analyzed until the moment [5].

In this work, we present an algorithm whose goal is the efficient refinement of 3D surface meshes composed of triangular facets. The algorithm proceeds iteratively subdividing, in each iteration, every facet of the original mesh into four new ones until the desired level of refinement is gotten. The algorithm guarantees that no duplicated vertices are inserted on the new mesh. In this manner, the target geometry has a lower memory footprint and the possibility of BEM (MoM) numerical instabilities caused by repeated facets is eluded. The presented algorithm relies on the use of *octree* [8] structures to efficiently avoid the insertion of repeated nodes. Among different types of mesh subdivision algorithms, our algorithm pertains to the category of vertex insertion [9], in a similar manner as the *Butterfly* scheme presented in [10].

This paper is organized as follows. Section 2 presents a brief summary of the theory associated to tree structures in d dimensions, 2^d-trees. In Section 3, we explain the algorithm that we have developed for mesh subdivision. In Section 4, some results of the computational cost of the algorithm are shown. Finally, some conclusions are presented in Section 5.

2 2^d-trees theory review

In this Section, we describe the fundamental aspects related to the 2^d-tree structures [8] that result more relevant in the development of the presented algorithm. A 2^d-tree structure is a type of space subdivision[1] that may be used to group the facets of a mesh within that space. Although we apply the 2^d-tree structures to 3D problems, this review is developed for a general d. This does not complicate the algebra so much and it allows us to use 2D examples ($d = 2$, or *quadtree* structure) which eases its representation.

The first step consists on enclosing the geometry under study on a box, that is assigned the level 0. Afterwards, the 0 level box, noted *parent box*, is subdivided into 2^d equal size boxes whose sides have a length that is half of its parent box. All these boxes are assigned to level 1, and are called *children boxes* respect to the level 0 box. This process of subdivision may be repeated as many times as necessary (see recursive division on Figure 1). Two boxes are called *neighbors* when they share at least one point.

[1] Note that d is the dimension of the space.

 ISBN: 978-84-616-2723-3

2.1 2^d-tree numbering

In the level l of a 2^d-tree, there are 2^{dl} boxes, each of which is assigned an index $n = 0, 1, \ldots, 2^{dl} - 1$. Therefore, every box may be characterized by the pair (n, l) (also called *universal number* [8]). It is also possible to identify each box by another number that is more adequate for the calculations that must be performed. Every box is associated to a set of l numbers in the form $(N_1, N_2, \ldots, N_l)$, being N_j the index of the box at j level. This set is known as *string number* [8]. Finally, the string number may be mapped to the universal number by means of the following expression:

$$n = (2^d)^{l-1} N_1 + (2^d)^{l-2} N_2 + \ldots + (2^d) N_{l-1} + N_l \ . \tag{1}$$

In Figure 1, a quadtree numbering example is represented. According to equation (2.1), the red box has the string number $(2, 3, 1)$ and universal number $(45, 3)$.

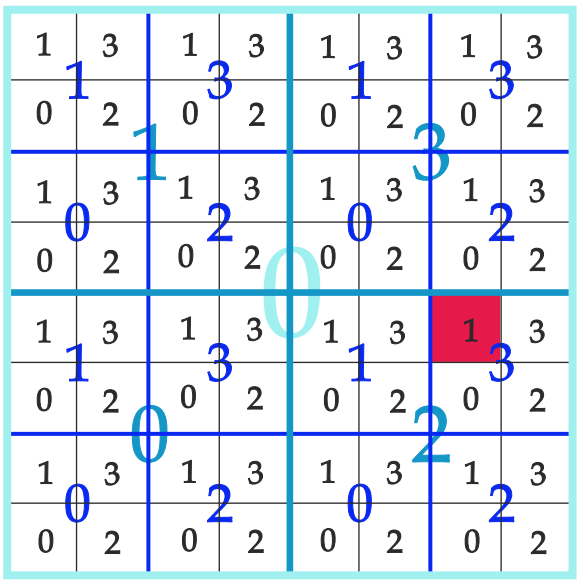


Figure 1: Hierarchical numbering for a quadtree.

2.2 2^d-tree generation

During the process of the 2^d-tree generation, it is necessary to map the 0 level box to a reference box (coordinate normalization). In this manner, if the coordinates of any point of

 ISBN: 978-84-616-2723-3

the geometry are given by $r = (x_1, x_2, \ldots, x_d)$, the *normalized coordinates* become:

$$\tilde{r} \;=\; (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d) \;=\; 2^{L_{tree}} \left(\frac{r - r_{min}}{D} \right), \tag{2}$$

where D is the biggest geometry size, L_{tree} is the highest 2^d-tree subdivision level, and r_{min} is the vertex coordinate of 0 level box that is closest to the origin of coordinates.

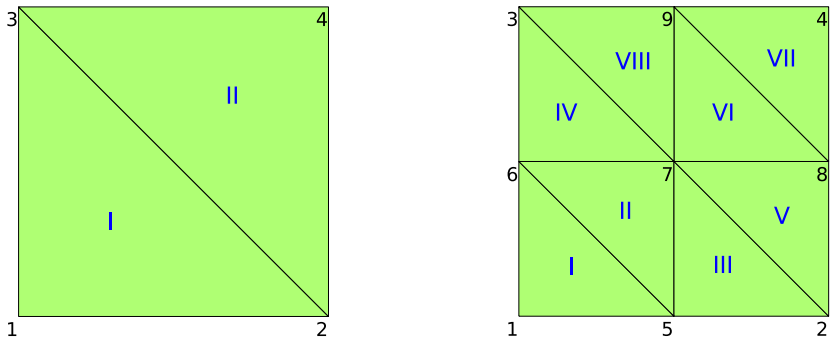
Finally, each point of the geometry may be represented by the string number of the group it pertains to, just by applying an interleaving process [8] to the binary representation of the normalized coordinates.

3 Efficient mesh refinement algorithm

The method presented in this work operates in an iterative manner in order to refine meshes. A 3D mesh, composed of triangular facets, inputs the algorithm and, throughout the iterative execution, it produces a new set of meshes. After each iteration, every original triangle is divided into four new ones that are obtained by connecting the midpoints of each side, as shown in Figure 2. Thus, the number of facets that form each new mesh may be calculated as follows:

$$f_{out} = 4^i \cdot f_{in}, \tag{3}$$

where i is the iteration number, f_{in} is the number of facets of the original input mesh, and f_{out} is the number of facets of the output mesh in the ith iteration.



(a) Iteration 0. (b) Iteration 1.

Figure 2: Simplified example. Arabic numerals are used to identify vertices whereas roman numerals are used to enumerate facets.(a) Input mesh. (b) Output mesh.

At the beginning of every iteration, two different files are read. The first file is the facets file, which defines every triangle by means of three vertex indices. The second file

 ISBN: 978-84-616-2723-3

is the vertices file, which contains the three coordinates of each vertex. Table 1 shows the facets and vertices files corresponding to the geometry of Figure 2(b).

Table 1: facets and vertices files in the iteration 1 (related to Figure 2)

	F (facets file)		
f_1:	v_1	v_5	v_6
f_2:	v_5	v_7	v_6
f_3:	v_5	v_2	v_7
f_4:	v_6	v_7	v_3
f_5:	v_2	v_8	v_7
f_6:	v_7	v_8	v_9
f_7:	v_8	v_4	v_9
f_8:	v_7	v_9	v_3

	V (vertices file)		
v_1:	x_1	y_1	z_1
v_2:	x_2	y_2	z_2
v_3:	x_3	y_3	z_3
v_4:	x_4	y_4	z_4
v_5:	x_5	y_5	z_5
v_6:	x_6	y_6	z_6
v_7:	x_7	y_7	z_7
v_8:	x_8	y_8	z_8
v_9:	x_9	y_9	z_9

Once the files that define the mesh have been loaded, the refinement process begins. For every facet of the previous iteration, its three original vertices are preserved. In addition, three new vertices are created. These new vertices are the midpoints of each side in the original facet (see Figure 2). Then, the six vertices related to the previous facet produce four new facets that are created by connecting the midpoints of each side (see Figure 2). Thus, the new vertices file will contain the previous vertices and the new ones, but the new facets file will only contain the new facets (see Figure 2 and Table 1).

All the process described in the above paragraph would be straightforward if there were no duplicated vertices. But, in a closed surface, each new vertex belongs to two different facets. For example, in Figure 2(b) it is showed that the vertex 7 belongs to the facets I and II of the previous iteration (Figure 2(a)). In order to avoid duplicated vertices, we use two different data structures: octrees and linked lists. Every time a new vertex (a midpoint of a previous facet side) is going to be included in the vertices file, the algorithm checks whether it is new or duplicated. First, the octree box that encloses the new vertex is calculated. Then, the algorithm accesses to the linked list associated to that box (using the box number as an index). Thus, the search for a duplicate is confined to a single octree box instead of the whole mesh. If the vertex was present in the list, that would mean the new vertex is a duplicate and it would be discarded. If the vertex was not in the list, then the new vertex is inserted in the list and it is also written in the vertices file. Once the algorithm has the indices of the correct vertices, the new facets are generated and then written to the facets file.

In order to achieve a time cost of $O(N)$ per iteration (where N is the number of facets of the current mesh), the search for a duplicated vertex in its associated list must be of

$O(1)$. This requirement is satisfied whenever the maximum number of vertices per list is constant. Because of this, the octree box size must be carefully chosen. Furthermore, the side of the octree boxes is halved (i.e, the octree level is increased) every iteration in order to keep constant the maximum number of vertices per list.

The use of an array of linked lists to store the vertices in each octree box allows to keep a low memory footprint, since only non-empty boxes take up space. Moreover, since the vertices are added to the list as needed, we avoid the usage of preallocated memory prior to knowing the exact number of vertices per list.

4 Results

To obtain the results presented in this section, we used a workstation that consists of 1 CPU (Intel Core i7-3820 with 4 cores at 3.6 GHz) and 64 GB of RAM. The source code is written in C and was compiled using Intel icc 12.1. It is also worth noting that the implementation is fully sequential and based on single-precision arithmetic.

The first example is a mesh of a sphere with 1 000 988 triangular facets. The input mesh is refined 5 times in order to obtain a set of meshes, each one with 4x the number of facets of the previous one. Table 2 shows the runtime and the memory consumption for this example. It should be noted that the runtime is the wall time (including the refinement process and I/O). The algorithm is able to generate a mesh with a billion facets in about 5 minutes and using only 28 GB of RAM.

Table 2: Runtime and memory consumption using a sphere.

Input: mesh with 1 000 988 triangular facets ($\varnothing$ 4 m sphere)				
Iteration	facets	Octree boxes	Runtime [s]	RAM usage [GB]
1	4 003 952	2^{18}	1.2	0.1
2	16 015 808	2^{21}	4.6	0.4
3	64 063 232	2^{24}	19.7	1.7
4	256 252 928	2^{27}	77.1	7.1
5	1 025 011 712	2^{30}	317.3	28.0

The second example is a mesh of a full-scale Airbus A380 airplane with 1 009 392 triangular facets. The input mesh is also refined 5 times in order to obtain a set of high resolution meshes. As shown in Table 3, in this case the algorithm is able to generate a mesh with a billion facets in about 8 minutes and using only 26 GB of RAM.

 ISBN: 978-84-616-2723-3

Table 3: Runtime and memory consumption for an Airbus A380 airplane.

Input: mesh with 1 009 392 triangular facets (full-scale airplane)				
Iteration	facets	Octree boxes	Runtime [s]	RAM usage [GB]
1	4 037 568	2^{18}	5.4	0.1
2	16 150 272	2^{21}	10.9	0.4
3	64 601 088	2^{24}	34.3	1.7
4	258 404 352	2^{27}	124.3	6.6
5	1 033 617 408	2^{30}	484.2	26.0

5 Conclusions

In this work, we present an algorithm based on the octree theory that efficiently performs the refinement of 3D meshes. The developed solution has a time complexity of $O(N)$ and enables obtaining meshes with a billion facets within minutes using a workstation. Additionally, its iterative operation is ideal for scattering problems, since the algorithm produces sets of meshes of the same object, which allows the analysis of the object at different frequencies. As a result, the developed solution has a great applicability on the field of acoustic and electromagnetic scattering.

Acknowledgements

This work has been partially supported by the European Union under COST action IC1102 (VISTA); by "Ministerio de Ciencia e Innovación" from Spain/FEDER under the projects TEC2011- 24492/TEC (iScat) and CONSOLIDER CSD2008-00068 (TeraSense); by "Gobierno del Principado de Asturias" (PCTI)/FEDER-FSE under the projects IPT-2011-0951-390000 (Tecnigraf), EQUIP08-06, FC09-COF09-12, EQUIP10-31, and grant BP11-166.

References

[1] M. López-Portugués, J. A. López-Fernández, J. Menéndez-Canal, A. Rodríguez-Campa and J. Ranilla, *Acoustic scattering solver based on single level FMM for multi-GPU systems*, J. Parallel Distrib. Comput. **72** (2012) 1057–1064.

[2] M. López-Portugués, J. A. López-Fernández, J. Ranilla, R. G. Ayestarán and F. Las-Heras, *Parallelization of the FMM on distributed-memory GPGPU systems for acoustic scattering prediction*, J. Supercomput. **64** (2013) 17–27.

[3] J. M. Taboada, M. G. Araújo, J. M. Bértolo, L. Landesa, F. Obelleiro and J. L. Rodríguez, *MLFMA-FFT parallel algorithm for the solution of large-scale problems in electromagnetics*, Prog. Electromagn. Res. **105** (2010) 15–30.

[4] M. G. Araújo, J. M. Taboada, F. Obelleiro, J. M. Bértolo, L. Landesa, J. Rivero and J. L. Rodríguez, *Supercomputer aware approach for the solution of challenging electromagnetic problems*, Prog. Electromagn. Res. **101** (2010) 241–256.

[5] J. M. Taboada, M. G. Araújo, F. Obelleiro, J. L. Rodríguez and L. Landesa, *MLFMA-FFT parallel algorithm for the solution of extremely large problems in electromagnetics*, Proc. IEEE (Special issue on Large Scale Electromagnetic Computation for Modeling and Applications) **101** (2013) 350–363.

[6] T. W. Wu, *Boundary Element Acoustics*, WIT Presss, Southampton, 2000.

[7] R. Harrington, *Matrix Methods for Field Problems*, Proc. IEEE **55** (1967) 136–149.

[8] N. A. Gumerov, R. Duraiswami and E. A. Borovikov, *Data structures, optimal choice of parameters, and complexity results for generalized multilevel fast multipole methods in d dimensions*, Technical Reports from UMIACS (2003).

[9] D. Zorin and P. Schröder, *Subdivision for Modeling and Animation*, SIGGRAPH 99 Course Notes, Los Angeles, 1999.

[10] N. Dyn, D. Levin, and J. A. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*, ACM Transactions on Graphics **9** (1990) 160-169.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Aircraft noise scattering computation using GPUs

M. López-Portugués[1], J. A. López-Fernández[1], José Ranilla[2] and R. G. Ayestarán[1]

[1] *Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y de Sistemas, Universidad de Oviedo, Spain*

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

emails: mlopez@tsc.uniovi.es, jelofer@tsc.uniovi.es, ranilla@uniovi.es, rayestaran@tsc.uniovi.es

Abstract

In this work, we present an efficient tool, implemented in CUDA, that computes the scattered noise by an object over whose surface the pressure distribution and its normal derivative are known. The method essentially implements a Matrix-Vector Product where the matrix elements are calculated on the fly in order to keep a low memory footprint. Our implementation is tested using two different GPU architectures (Fermi and Kepler) and we achieve a reduction of an order of magnitude in the runtime compared to our reference OpenMP codes. As a result, the use of the presented implementation together with an efficient computation of the acoustic field over the obstacle surface enables a powerful tool for noise control applications.

Key words: heterogeneous, acoustic scattering, MVP, CUDA, OpenMP

1 Introduction

Nowadays, acoustic scattering is a topic of major concern for the industry. Scattering prediction applications are, among others, the simulation of the acoustic behavior of industrial products, such as aircrafts. Scattering simulations allow to predict the acoustic behavior of a product in early stages of the design phase in contrast to the traditional try and error prototyping methodology.

The solution of the scattering problem may be carried out in two steps: i) computation of the acoustic field (pressure and its normal derivative) over the obstacle surface, and

ii) calculation of the pressure at any point outside of the obstacle from the previously calculated acoustic field on its surface.

The *Boundary Elements Method* (BEM) [1] provides an accurate numerical formulation of this type of problems. Nonetheless, the BEM solution of the scattering problem may be very expensive from a computational viewpoint. The first step requires to solve a linear system with N equations and N unknowns. Efficient solvers based on spectral representations on the κ-space —*Fast Multipole Method* (FMM) [2] and its *Multilevel* counterpart [3]— may reduce the solution complexity up to $O(N \log(N))$ per iteration. In addition, the computation of the noise scattered by the aircraft over a surrounding wide region may be very computationally demanding, since it requires a *Matrix-Vector Product* (MVP) where the matrix size is proportional to the discretization of the obstacle times the noise observation points.

For its part, *hardware accelerators* may help in the reduction of the run time of demanding problems. Especially, the *Graphics Processing Units* (GPUs) may provide large speedups in those compute-intensive algorithms that have a high degree of parallelism [4]. Since similar MVPs are proved to be prone to parallelization [5, 6], the second step in the solution of the scattering problem seems to fulfill the requirements to exploit the massively parallel architecture of modern GPUs.

This work actually complements the ones presented by the authors in [7, 8] where an efficient GPU solver for computing the acoustic field over the obstacle surface (the above mentioned first step) was presented. BEM is used to model numerically the physical problem.

2 Developed solution using CUDA

In this work, we tackle a MVP whose computational complexity is of $O(NM)$ in terms of time, where N is the number of source points and M is the number of observation points. Moreover, when analyzing the noise generated by a real-scale object —such as an aircraft— on a plane with a meaningful area, M and N may be on the order of millions. Thus, the $N \times M$ matrix that relates the source points and the observation points may be huge. Because of this, the MVP should not be performed in the usual way, but calculating the matrix elements on the fly in order to minimize the memory footprint.

Prior to develop the CUDA [9] algorithm, we chose to begin with a simpler parallel approach. The algorithm that we have used as a starting point is based on the OpenMP *parallel for* construct [10]. In this algorithm, each thread only deals with the vector and a single matrix row at a time, thus calculating a single element of the solution vector (scattered pressure on the observation points). In this manner, we have a parallel algorithm that is suitable for shared memory systems and does not require large amounts of RAM.

Nevertheless, the number of parallel tasks of the above-mentioned solution is not suffi-

 ISBN: 978-84-616-2723-3

cient for many-core architectures like GPUs. Thus, the chosen approach to deal with GPUs had to be different, in order to turn the original MVP into thousands of sub-problems that may be computed concurrently.

The calculation of the MVP is accomplished in such a way that matrix rows are assigned to CUDA *thread blocks* [9] and the elements within a row are assigned to those threads within the same block. Moreover, the assignment is performed in a cyclic fashion, since M and N may be much larger than the *grid size* and the *block size* [9], respectively.

First, each thread computes its matrix element on the fly —thus avoiding the storage of the matrix in global memory. After that, each thread multiplies its matrix element and its vector element and stores the result in a register. Finally, in order to obtain the MVP, all the threads within the same block add its partial results by performing a reduction in shared memory [9].

Table 1 shows the kernel execution parameters when compiling for *Fermi* and *Kepler* architectures. The *occupancy* has been calculated using the occupancy calculator that is part of the CUDA toolkit and has been verified by means of the NVIDIA Visual Profiler. The parameters shown below are the ones that delivered the best performance throughout our tests.

Table 1: CUDA kernel execution parameters for both Fermi and Kepler architectures.

Common parameters	
Registers per thread	26
Shared memory per block	2 KB
Grid size $*$ block size	$8192 * 256$
Shared memory : L1 cache ratio	16 KB : 48 KB
Fermi architecture	
Compute capability	2.1
Multiprocessor occupancy	67 %
Kepler architecture	
Compute capability	3.0
Multiprocessor occupancy	100 %

3 Results

In order to obtain the results presented in this work, we used two workstations with different GPU architectures. The first workstation consists of 1 CPU (Intel Core i5-2500T with

 ISBN: 978-84-616-2723-3

4 cores at 2.3 GHz), 16 GB of RAM, and 1 NVIDIA Fermi GPUs (GTX 560 with 336 cores at 1.62 GHz). The second workstation consists of 1 CPU (Intel Core i7-3820 with 4 cores / 8 threads at 3.6 GHz), 64 GB of RAM, and 1 NVIDIA Kepler GPU (GTX 680 with 1536 cores at 1.07 GHz). The source code is written in C and CUDA C, and was compiled using Intel icc 12.1 and NVIDIA CUDA compilation tools 4.2, respectively. It is also worth noting that single-precision arithmetic was used in all cases.

In this section, we analyze the noise generated by a real-scale aircraft model (Airbus A300 series) at a frequency of 1 kHz. The mesh that is used to represent the aircraft consists of 1009392 triangular facets ($N = 1009392$) and has been generated to model the geometry using approximately six elements per linear wavelength. First, we calculated the acoustic field over the aircraft surface by means of the tool presented in [8]. Then, we calculated the acoustic field on a plane at $z = 0$ m (beneath the aircraft) by using the tool presented in this work. The dimensions of the observation plane are 80.07 m $\times$ 80.07 m, with a resolution of four elements per linear wavelength ($M = 889249$). Figure 1 shows the magnitude of the total pressure on the observation plane when the engines (noise sources) are placed beneath the wing.

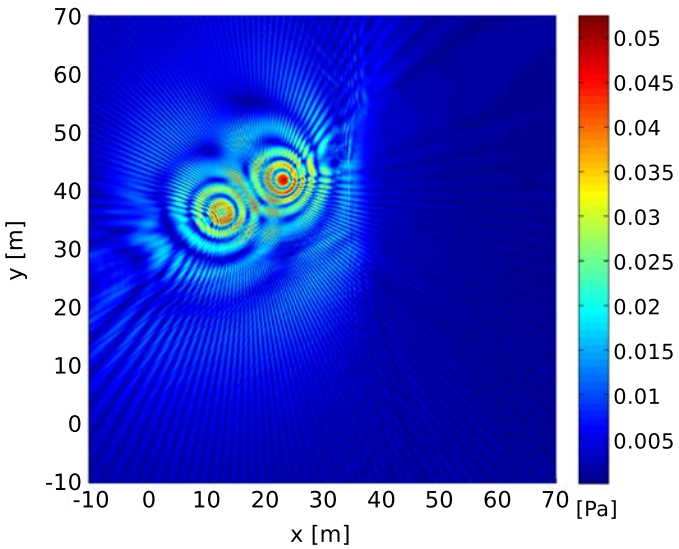


Figure 1: Magnitude of the total pressure on a plane at $z = 0$ m (beneath the aircraft). Engines placed beneath the wing. Frequency: 1 kHz.

Table 2 shows a comparison between the OpenMP implementation and the CUDA implementation using the first workstation (Fermi architecture). It should be noted that the incident field calculation is not much demanding, so we decided to perform this computation

Table 2: CPU implementation (OpenMP) vs GPU implementation (CUDA) using the first workstation.

Workstation with Fermi GPU		
Problem size: $N = 1009392$, $M = 889249$		
OpenMP	Incident field calculation time	0.03 s
	Scattered field calculation time	2069.96 s
	Overall runtime	2070.63 s
CUDA	Incident field calculation time	0.03 s
	Scattered field calculation time	308.09 s
	Overall runtime	309.62 s

using the CPU. On the other hand, it is noticeable that the scattered field calculation —the MVP whose computational complexity is of $O(NM)$— is much more demanding. Using a mid-range Fermi GPU (GTX 560), the CUDA implementation achieves a speedup of 6.7x when compared to the OpenMP parallel algorithm.

Table 3: CPU implementation (OpenMP) vs GPU implementation (CUDA) using the second workstation.

Workstation with Kepler GPU		
Problem size: $N = 1009392$, $M = 889249$		
OpenMP	Incident field calculation time	0.05 s
	Scattered field calculation time	1162.60 s
	Overall runtime	1163.86 s
CUDA	Incident field calculation time	0.05 s
	Scattered field calculation time	112.56 s
	Overall runtime	113.52 s

Table 3 also shows a comparison between the OpenMP implementation and the CUDA implementation, but this time making use of the second workstation (Kepler architecture). Using a NVIDIA GTX 680, the CUDA solution is 10.3 times faster than the OpenMP solution.

 ISBN: 978-84-616-2723-3

It is also worth noting the advantage of calculating the matrix elements on the fly. For this problem, the storage of the matrix that relates the source points and the observation points would require at least 6.5 TB of RAM (assuming 8-byte complex numbers are used).

4 Conclusions

In this work, a method for computing the scattered acoustic pressure using GPUs is presented. We test our implementation using two different GPU architectures. Using a mid-range Fermi GPU, the presented solution achieves a speedup of 6.7x when compared to our own parallel algorithm for CPUs. In addition, the CUDA implementation allows reducing the runtime in an order of magnitude with a Kepler card compared to the reference OpenMP implementation. As a result, we get a tool for acoustic-scattering prediction that clearly surpasses the performance of our parallel algorithm for CPUs, thus allowing the accomplishment of accurate and more efficient analyses in noise control applications.

Acknowledgements

This work has been partially supported by the European Union under COST action IC1102 (VISTA); by "Ministerio de Ciencia e Innovación" from Spain/FEDER under the projects TEC2011- 24492/TEC (iScat) and CONSOLIDER CSD2008-00068 (TeraSense); by "Ministerio de Economía y Competitividad" from Spain under project TEC2012-38142-C04-04; and by "Gobierno del Principado de Asturias" (PCTI)/FEDER-FSE under the projects IPT-2011-0951- 390000 (Tecnigraf), EQUIP08-06, FC09-COF09-12, EQUIP10-31, and grant BP11-166.

References

[1] T. W. Wu, *Boundary Element Acoustics: Fundamentals and Computer Codes (Advances in Boundary Elements)*, WIT Press, Southampton, 2000.

[2] V. Rokhlin, *Diagonal Forms of Translation Operators for the Helmholtz Equation in Three Dimensions*, Appl. Comput. Harmon. A., **1** (1993) 82–93.

[3] J. Song and W. Chew, *Multilevel Fast-Multipole Algorithm for Solving Combined Field Integral Equations of Electromagnetic Scattering*, Microw. Opt. Techn. Let. **10** (1995) 14–19.

[4] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone and J. C. Phillips, *GPU Computing*, P. IEEE **96** (2008) 879–899.

 ISBN: 978-84-616-2723-3

[5] J. A. López-Fernández, M. López-Portugués, Y. Álvarez, C. García, D. Martínez-Álvarez and F. Las-Heras, *Fast antenna characterization using the sources reconstruction method on graphics processors*, Prog. Electromagn. Res. **126** (2012) 185–201.

[6] M. López-Portugués, Y. Álvarez, J. A. López-Fernández, C. García, R. G. Ayestarán and F. Las-Heras, *A multi-gpu sources reconstruction method for imaging applications*, Prog. Electromagn. Res. **136** (2013) 703–724.

[7] M. López-Portugués, J. A. López-Fernández, J. Menéndez-Canal, A. Rodríguez-Campa and J. Ranilla, *Acoustic scattering solver based on single level FMM for multi-GPU systems*, J. Parallel Distrib. Comput. **72** (2012) 1057–1064.

[8] M. López-Portugués, J. A. López-Fernández, J. Ranilla, R. G. Ayestarán and F. Las-Heras, *Parallelization of the FMM on distributed-memory GPGPU systems for acoustic scattering prediction*, J. Supercomput. **64** (2013) 17–27.

[9] NVIDIA Corporation, *CUDA C Programming Guide (Design Guide)*, available online at: `http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf` (2012).

[10] The OpenMP ARB, *OpenMP*, available online at: `http://www.openmp.org/` (2004).

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

DyRM: A Dynamic Roofline Model Based on Runtime Information

O.G. Lorenzo[1], T.F. Pena[1], J.C. Cabaleiro[1], J.C. Pichel[1] and F.F. Rivera[1]

[1] *Centro de Investigación en Tecnoloxías da Información, CITIUS, Univ. of Santiago de Compostela, Santiago de Compostela, Spain*

emails: oscar.garcia@usc.es, tf.pena@usc.es, jc.cabaleiro@usc.es, juancarlos.pichel@usc.es, ff.rivera@usc.es

Abstract

Modern systems present complex memory hierarchies and heterogeneity among cores and processors. As a consequence, efficient programming is challenging. An easy-to-understand performance model, offering guidelines and information about the behaviour of a code, may be useful to alleviate these issue. In this paper, we present a new model, the Dynamic Roofline Model, an extension of the well known Berkeley Roofline Model. The aim is to take into consideration the complexities of multicore and heterogeneous systems, to understand their influence in the performance of a code when it is executed in a particular system. A set of tools to obtain and represent the model have been implemented. Different views are displayed by the tool that can be used to extract the main features of the code. Results of studying the NAS Parallel Benchmarks for OpenMP with these tools using the Dynamic Roofline Model are presented.

Key words: Roofline Model, Performance, Hardware Counters, PEBS, NPB.

1 Introduction

For a parallel code to be correctly and efficiently executed, its programming must be careful. Taking into account architectural features, particularly the behaviour of memory accesses, is critical to improve locality among accesses and affinity between data and processors. Performance bottlenecks can be identified by collecting data related to how an application or system performs. This collection is known as *performance monitoring*. Characterising the nature and cause of the bottlenecks using this information allows us to understand reasons why a program behaves in a particular way. Some performance issues in which this

information is important are, among others, data locality or load balancing. Characterising them may help lead to a performance improvement [1].

In order to help programmers to understand the performance of their codes in a particular system various performance models have been proposed. In particular Berkeley Roofline Model [2] (RM) offers a nice balance between simplicity and descriptivism. Nevertheless, its own simplicity might hide some performance bottlenecks present in modern architectures. In this paper, an extension to the RM is presented. The RM is extended taking different measurements during the life of an application, in order to show the evolution of different phases on the execution. We call this model the Dynamic Roofline Model (DyRM), to highlight the fact that it shows the evolution in time of a kernel, in a per thread basis. This has special importance in multicore and heterogeneous systems, since it shows clearly differences in the execution in each core. To obtain this model we have developed a tool that takes advantage of the hardware counters present in modern processors. The data it collects is used in by a second tool to render various figures.

Nowadays, performance monitoring counters, also known as *hardware counters*, are powerful monitoring mechanisms included in the Performance Monitoring Unit [3] of most of modern microprocessors. Their use is gaining popularity as an analysis and validation tool. Their effect in the monitored program is virtually imperceptible and their precision has noticeably increased recently thanks to the new *Precise Event-Based Sampling* (PEBS) [4] features.

To show the benefits of our extended model an study of the NAS Parallel Benchmark Suite for OpenMP (NPB-OMP) [5] was carried out. The NPB-OMP is a set of kernels and pseudo-applications designed to test shared memory systems in general.

The rest of the paper is organised as follows: Section 2 introduces the RM and the proposed extension, the DyRM. Section 3 introduces the PEBS hardware counters present in modern Intel processors, and the usage we made of them in the performance tools. In Section 4 the results of our analysis of the NPB-OMP3.3 benchmark suite are presented. Finally, the conclusions of this paper are drawn in Section 5.

2 The Dynamic Roofline Model

The RM [2] is an easy-to-understand model, offering performance guidelines and information about the behaviour of a program. It offers insight on how to improve the performance of software and hardware. Stochastic analytical models and statistical performance models can accurately predict program performance on multiprocessors but rarely provide insight into how to improve the performance of programs, compilers, and computers and can be difficult to use by non experts [6] [7] [8].

The RM uses a simple bound and bottleneck analysis approach, where the influence of the system bottleneck is highlighted and quantified. In modern systems the main bottleneck is often the connexion between processor and memory. This is why the RM relates

 ISBN: 978-84-616-2723-3

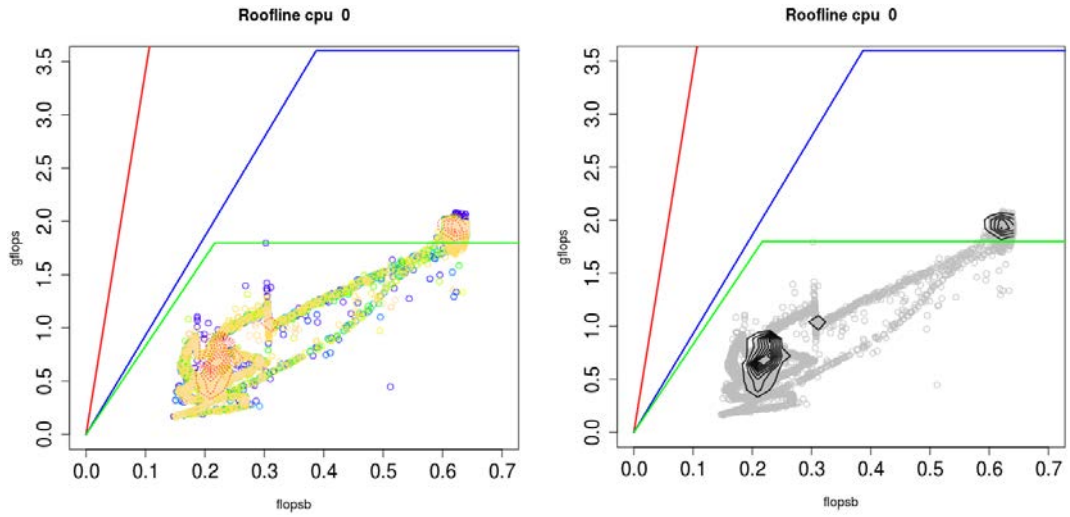
processor performance to off-chip memory traffic. It uses the term operational intensity to mean operations per byte of DRAM traffic (measured in $Flops/Byte$). Note that, it measures traffic between the caches and memory rather than between the processor and the caches. Thus, operational intensity predicts the DRAM bandwidth needed by a kernel on a particular computer. The RM ties together floating-point performance (measured in $GFlops/sec$), operational intensity, and memory performance in a 2D graph.

The RM allows the definition of performance limits. A horizontal line showing peak floating-point performance of the computer can be drawn. The actual floating point performance of a particular kernel can be no higher than this horizontal line, since this line is the hardware limit. A second line that bounds the maximum floating-point performance that the memory system of the computer can support for a given operational intensity can be plotted. Its slope would correspond to the peak memory bandwidth. The two lines would intersect at the point of peak computational performance and peak memory bandwidth. Thus we get a upper limit for performance, or roof. This way, if a kernel operational intensity is below the slanted part of the roof it means its performance is memory-bound. If it is otherwise below the flat part, it would be compute-bound.

The RM gives a simple representation of a program performance in a particular system. Nonetheless, in some cases it may be misleading. Consider an example program which goes through two phases of execution. One of them might be close to the maximum $GFlops/sec$ and operational intensity of the machine, while the other might be performing poorly. The RM would place the program performance at a single point of the figure, perhaps between the performance of both phases, that would not be entirely representative of the program real behaviour. In another example, consider an heterogeneous system. While the RM would give a performance point for the entire system, thus hiding the heterogeneity, differences inside the system would mean threads would have to be studied separately. Situations like these justify the proposal of a DyRM, that provides information at regular intervals of an execution, in a thread by thread basis.

The DyRM is essentially the equivalent of dividing in time slices the execution of a code and getting one RM for each one, then combining them in just one graph. This way a more detailed view of the performance during the entire life of the code, showing its evolution and behaviour, is obtained. With a DyRM different execution phases or behaviours can be detected. To show the kernel evolution in time we colour each point in the graph sequentially, using a colour gradient. To better show the phases, a two dimensional density estimation of the points in the DyRM can be performed. Such an estimation allows to readily find clusters, the zones in the model where the code spends more time, which allows to easily identify performance bottlenecks. The resulting groups can be highlighted in the DyRM, and changing colouring a Density Graph can be shown, giving a better view of these. Using both graphs DyRM combines the simplicity of the RM with a detailed view of a program execution.

 ISBN: 978-84-616-2723-3



(a) Dynamic Roofline sp.B (b) Density Graph sp.B

Figure 1: Examples of Dynamic Roofline Models for NPB benchmark sp.B.

An example is shown in Figure 1. Here we display the DyRM of an application running in an Intel Xeon E5-2603. In these models we have drawn three roofs, representing a processor core maximum performance. The topmost roof represents the peak $GFlops/sec$ using SIMD instructions and its theoretical maximum memory bandwidth [9], this roof reaches 14.4 $GFlops/sec$ and it is cut from the Figure. The middle roof represents the maximum $GFlops/sec$ without SIMD instructions, considering one multiply and add operations per cycle and the maximum memory bandwidth given by the STREAM benchmark [10]. The lower roof represent the $GFlops/sec$ considering only one floating operation per cycle and the worst memory bandwidth given by the STREAM benchmark. In Figure 1(a) we show how the application remains mostly under the lower roof during its life. Each point is coloured according to the time it was taken. In Figure 1(b) we show the clusters where the application spend more of its execution time, mainly two. One of these clusters exceeds the lower roof, meaning it makes more than one flop per cycle, by combining add and multiply operations. The other cluster is below the slanted side of the roof, so it is memory-bound and would not benefit greatly from using SIMD operations.

3 DyRM Tools for Data Capture

3.1 Intel PEBS

PEBS [4] is an advanced sampling feature of the Intel Core-based processors in which the processor is directly recording samples into a designated memory region. Each sample

 ISBN: 978-84-616-2723-3

contains the machine state of the processor at the time a hardware counter reaches a set goal. The precision of PEBS comes from the fact that the instruction pointer recorded in each sample is at most one instruction away from where the counter actually overflowed. The skid is minimised compared to regular interrupted instruction pointer. Another key advantage of PEBS is that is minimizes the overhead because the Linux kernel is only involved when the PEBS buffer fills up, i.e., there is no interruption until a number of samples are available. A constraint of PEBS is that it works only with certain events.

In the modern Intel processors, starting with the Nehalem architecture, the PEBS record format allows for detailed information about memory accesses. When sampling memory operations the virtual address of the operation data is recorded. For load operations the latency in which the data is served is also recorded (in cycles), as well as information detailing the memory level from where the data was read.

In order to interact with hardware counters we use the `linux perf_events` interface. This interface provides means to interact with the Linux kernel, via a system call, so hardware counters can be programmed and read. This interface is compatible with PEBS, in this case the sampling buffer is stored in kernel space and can be read in user space once it has overflowed. The Linux `perf_events` interface expands the PEBS format, recording for each sample data like the `pid` and `tid` of the active process and timing information. By using this interface we ensure compatibility across numerous systems, as long as they are Linux and Intel based.

3.2 Data Capture

A tool to obtain and process performance information of a complete shared memory system has been implemented. It uses the Linux `perf_events` interface to read the hardware counters in each CPU to characterise the performance of the whole system. Then the data are processed in an R environment[11].

To obtain a DyRM two kinds of data are needed. We need data from the floating point operations performed in each core. In modern Intel Sandy/IvyBridge processors [12] this means recording up to ten different hardware events (variations of FP_COMP_OPS_EXE and SIMD_FP_256). If packed floating point instruction are ignored it can be enough to measure just two events FP_COMP_OPS_EXE: SSE_SCALAR_DOUBLE and FP_COMP_OPS_EXE: SSE_FP_SCALAR_SINGLE, since FP_COMP_OPS_EXE:X87 is usually negligible. To measure memory traffic between the caches and main memory for each core the number of cache lines read from main memory (event OFFCORE_REQUEST:ALL_DATA_READ) are recorded. PEBS and a sampling buffer are used to capture the information. In every sample floating point operations data and data memory reads are captured. The instruction count in each core is used to set the sampling period, so in each overflow information about instructions, floating point operations and data reads, is obtained. This way the sampling period can be easily set, independently, of the floating point load of the program.

 ISBN: 978-84-616-2723-3

The state of the hardware counters for each core is captured. If more that one process or thread is executed simultaneously in the same core, the data must be scaled, so each thread can be measured individually. Each sample has information about the `pid` and `tid` of the specific instruction captured. This, together with the timing information provided by the `perf_events` interface, allows us to scale the data in each core to approximate the values for each thread. In this way programming the hardware counters becomes an easy task and the complete execution of a program can be recorded, including Operative System processes needed, but still allowing for a detailed review of the performance.

4 Case Study

In this section we show performance results for some of the NPB3.3-OMP benchmarks [5], executed in a system with two Intel Xeon E5-2603 processors – 4 core per processor, 8 total – and 16 GB of RAM. All executions were carried out with 8 threads, and a DyRM for each thread was obtained. The compiler used was `gcc 4.6.3`. Since, for the purpose of this paper, a detailed analysis was not needed, and the resulting models are quite similar for all threads, only the one thread will be used to represent the benchmark. The benchmark we study are: CG (Conjugate Gradient), irregular memory access and communication; FT (Discrete 3D fast Fourier Transform), all-to-all communication; UA (Unstructured Adaptive mesh), dynamic and irregular memory access; and the solvers LU (Lower-Upper Gauss-Seidel), BT (Block Tri-diagonal), and SP (Scalar Pentadiagonal). These kernel can use various sizes for input. The ones we use are Classes A, B, C, the standard test problems; 4X size increase going from one class to the next.

In our DyRM each point in the graph represents the performance of an application during a small slice of time. Each point can be coloured according to the actual time in which they are executed, to show the applications progression. Colours are selected using a gradient format. This way we can distinguish different phases in the execution of an application, and rate their performance separately. The Density Graphs show the performance spots where the program stays longer, allowing to see more clearly phases and to calculate a more general performance.

4.1 Overhead

The data capture application is very lightweight. As such, the overhead is mainly determined by the sampling rates: the higher the desired resolution, the larger the overhead. Most figures in this paper were obtained sampling each 10^8 instructions – about one sample per 5ms for most benchmarks – . When the execution time of the benchmark was larger than 50 seconds, the sampling period was heightened to $3x10^8$ instructions, thereby the file size of the resulting trace remains reasonable – under 10 MB – , and only a small effect in the resolution of the graphs is noticed. The overhead we obtained in this study,

 ISBN: 978-84-616-2723-3

Table 1: Data Capture Overhead.

	Size A		Size B		Size C	
Benchmark	Time(s)	Overhead(%)	Time(s)	Overhead(%)	Time(s)	Overhead(%)
bt	53.27	*0.23*	218.61	*0.30*	834.15	*0.02*
cg	0.70	1.07	36.07	3.19	96.46	*3.15*
ft	3.21	2.18	41.10	1.85	–	–
lu	48.67	2.95	197.41	*0.36*	754.40	*0.46*
ua	41.37	1.00	164.30	*0.77*	629.21	*1.99*
sp	54.36	*<0.01*	213.29	*0.56*	786.36	*0.27*

is shown in Table 1. In this table the execution times for the NPB benchmarks studied, compiled without any optimisation (column Time), and the overhead obtained with the same benchmarks running alongside the data capture program (column Overhead), are displayed. Note that, the overhead from the sampling and data capture program is low and, in many cases, inside the error of the measurement. In the cases where the sampling rate was set to $3x10^8$ (in italic in Table 1) the overhead usually remains below 0.80%.

4.2 Effect of Compiler Optimisations

To illustrate the use of the tool we analysed the effect of general optimisations in the model. In the next paragraphs we describe the NPB benchmarks compiled without optimisation and with an `O2` optimisation level. Due to space constraints only the most representative cases are shown. Note that, in order to better see the resulting model, the axes values in the next figures vary.

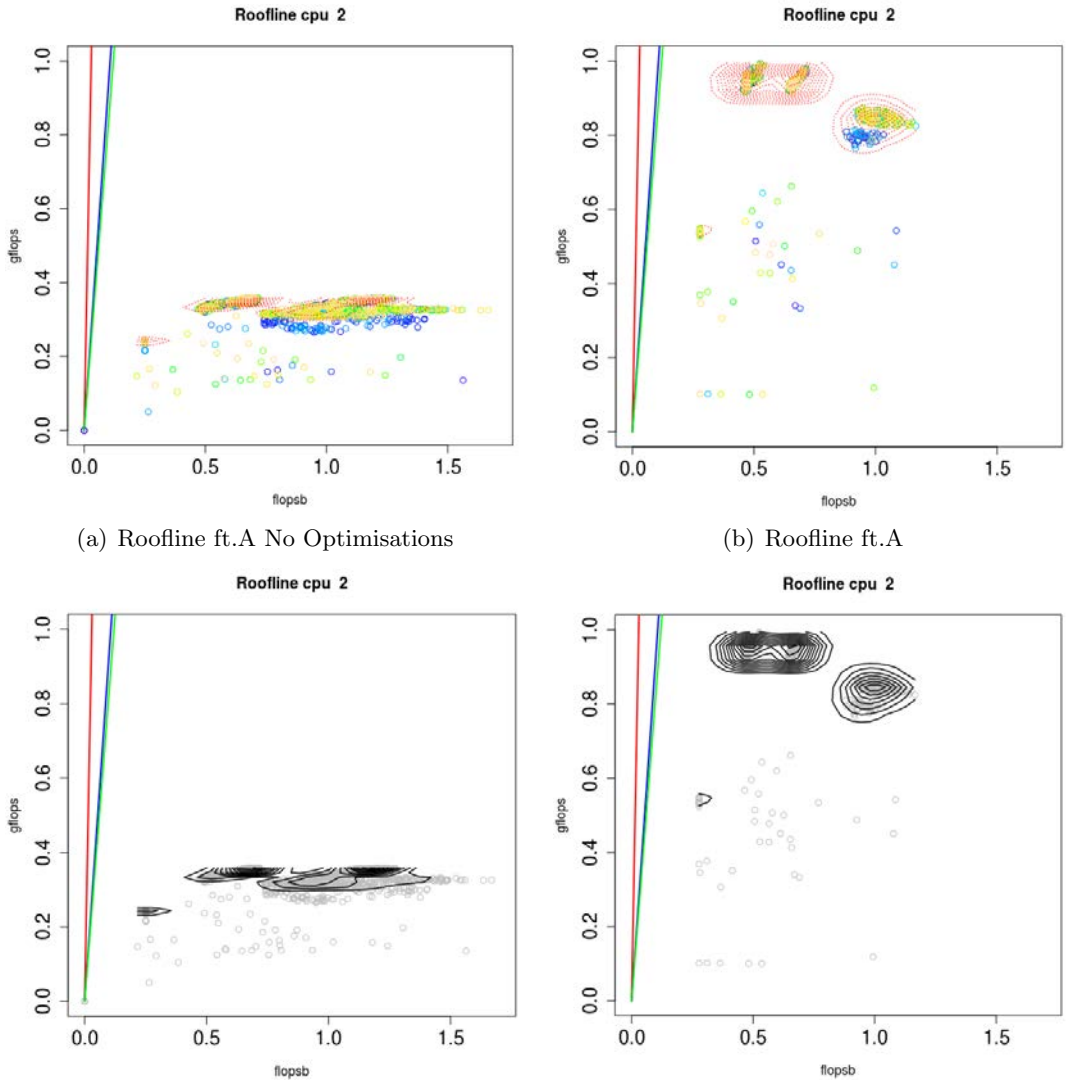
The FT benchmark shows several differentiated phases (Figure 2), one with an initialisation phase. Optimising the program improves the $GFlop/sec$ count, but the difference among the phases persists, which may indicate the need to manually optimise each one separately.

The UA benchmark (Figure 3) shows a strange pattern in the non optimised version, but the density graph shows that it is not very important. Once optimised, the pattern changes sightly, and the density graph seems to imply that the right side of the graph is more important.

4.3 Effect of the Problem Size

In this Section we show the effects in our model of different problem sizes in the NAS benchmarks. Only the most representative kernels are shown.

In the CG benchmark note that it has an initialisation phase, which shows a much lower



(a) Roofline ft.A No Optimisations

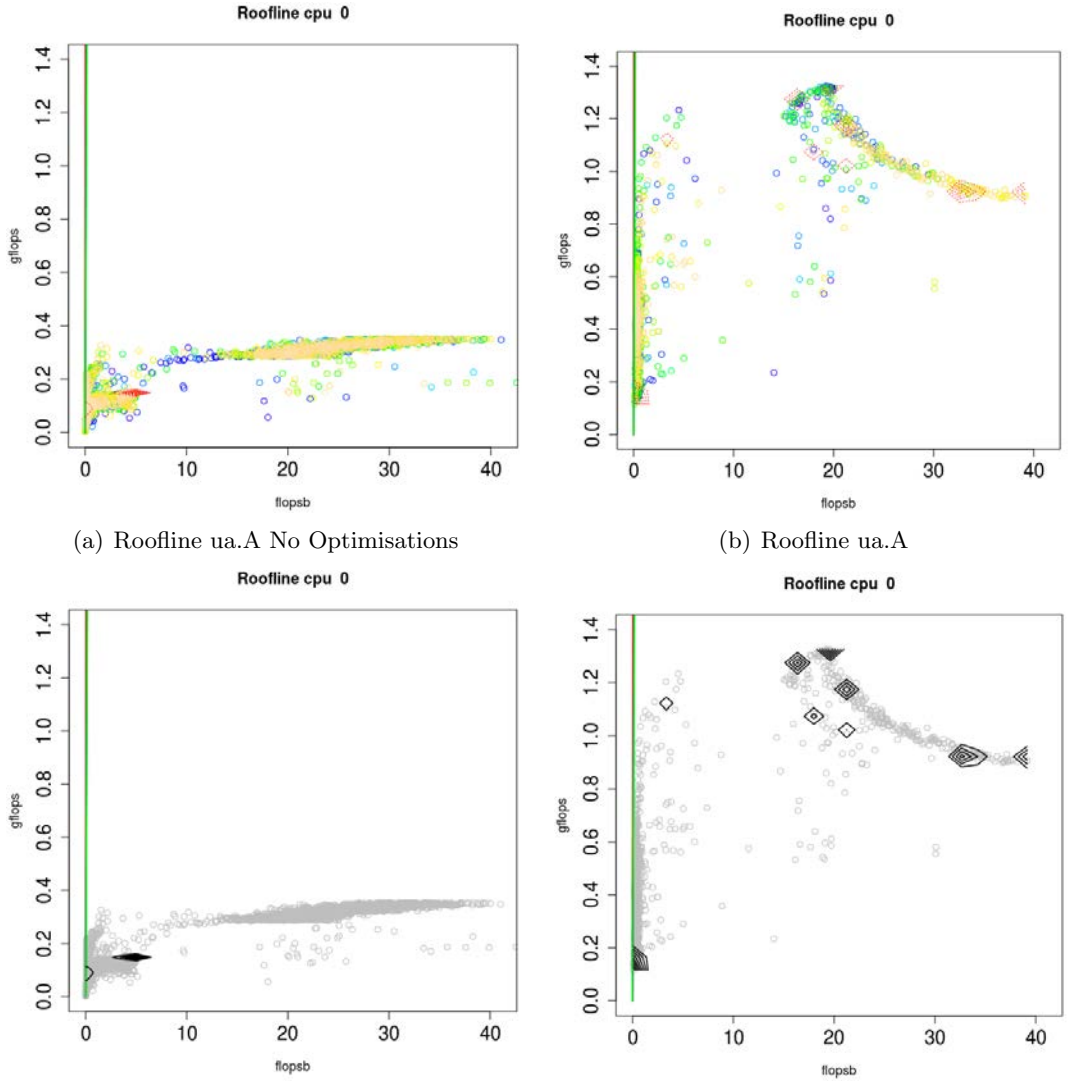
(b) Roofline ft.A

(c) Density. ft.A No Optimisations

(d) Density. ft.A

Figure 2: Roofline for ft.A benchmark.

 ISBN: 978-84-616-2723-3



(a) Roofline ua.A No Optimisations

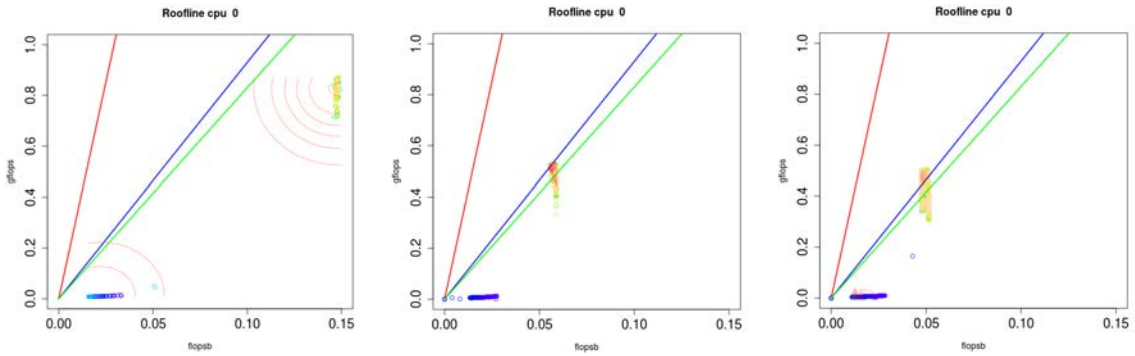
(b) Roofline ua.A

(c) Density. ua.A No Optimisations

(d) Density. ua.A

Figure 3: Roofline for ua.A benchmark.

 ISBN: 978-84-616-2723-3

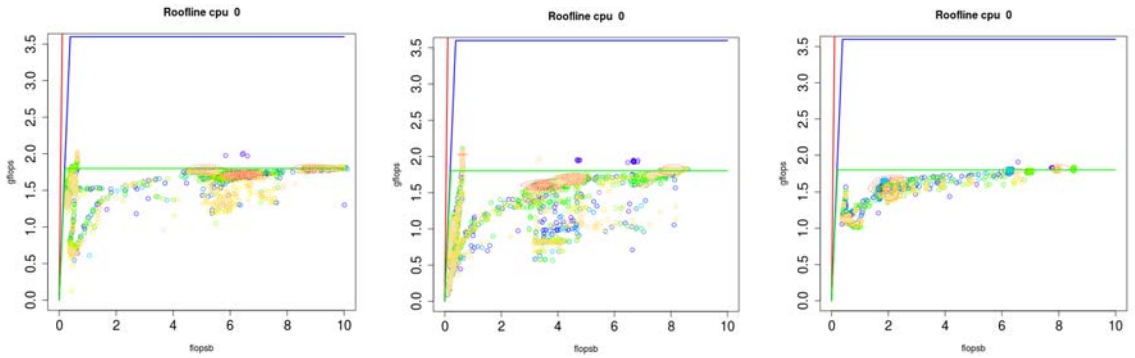


(a) Roofline cg.A (b) Roofline cg.B (c) Roofline cg.C

Figure 4: Roofline for CG benchmark. Sizes A, B and C.

performance. It can be seen in Figures 4(a), 4(b) and 4(c) in the lower left corner, with a blue colour which means it happens earlier in time. We observe it is a memory-bound benchmark. As problem size increases, both computational intensity and $GFlops/sec$ count decrease, and the initialisation phase still retains its importance. We see how it follows the roof, since it is memory-bound. In the case of size C, the performance overcomes the second roof, since it is not using SIMD instructions it probably means it achieves a better bandwidth than the STREAM benchmark.

In the BT benchmark (Figure 5)at least three phases can be seen, although, a seen in the density lines, one of them is of greater importance than the others, and it grows with



(a) Roofline bt.A (b) Roofline bt.B (c) Roofline bt.C

Figure 5: Roofline for BT benchmark. Sizes A, B and C.

 ISBN: 978-84-616-2723-3

problem size. Here we can see how, as the problem size increases, the $GFlops/sec$ count stabilizes, but not so the computational intensity, with all phases moving left. Anyway, this is clearly a computational bound program, and would have to use SIMD instructions to be optimised.

5 Conclusion

In this paper a set of extensions to the Berkeley Roofline Model, an showed their usefulness, have been presented. To create these models advantage of the PEBS counters of Intel processors was taken. A set of tools to automate the task were implemented. We have shown that, while the Berkeley Roofline Model remains a useful and simple model, it hides some characteristics of the applications that become important in many systems, specially manycore, NUMA or heterogeneous systems. We use these tools to show how parallel applications like the NPB-OMP benchmarks present complex behaviours and unbalances in NUMA systems. These problems can be easily modelled with our tools, without influencing the normal execution of the applications, and showing a realistic model of their performance. Thanks to our Dynamic Roofline Model a program behaviour, its phases or unbalances can be more easily detected, making it easier to correct performance issues.

Acknowledgements

This work has been partially supported by the Ministry of Education and Science of Spain, FEDER funds under contract TIN 2010-17541 and by the Xunta de Galicia (Spain) project 09TIC002CT. It has been developed in the framework of the European network HiPEAC-2 and the Spanish network CAPAP-H.

References

[1] O. G. Lorenzo, J. A. Lorenzo, J. C. Cabaleiro, D. B. Heras, M. Suarez, and J. C. Pichel, "A study of memory access patterns in irregular parallel codes using hardware counter-based tools," in *Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, Las Vegas (USA), 2011, pp. 920–923.

[2] S. Williams, A. Waterman, and D. Patterson, "Roofline: an insightful visual performance model for multicore architectures," *Commun. ACM*, vol. 52, no. 4, pp. 65–76, Apr. 2009. [Online]. Available: http://doi.acm.org/10.1145/1498765.1498785

[3] D. Mosberger and S. Eranian, *IA-64 Linux Kernel: Design and Implementation*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[4] *http://perfmon2.sourceforge.net/pfmon_intel_core.html#pebs*, Precise Event-Based Sampling (PEBS).

[5] H. Jin, M. Frumkin, and J. Yan, "The OpenMP implementation of NAS parallel benchmarks and its performance," Technical Report NAS-99-011, NASA Ames Research Center, Tech. Rep., 1999.

[6] D. R. Martínez, V. Blanco, J. C. Cabaleiro, T. F. Pena, and F. F. Rivera, "Modeling the performance of parallel applications using model selection techniques," *Concurrency and Computation: Practice and Experience*, 2013.

[7] X. Wu, *Performance, evaluation, prediction and visualization of parallel systems.* Kluwer Academic Publishers, 1999.

[8] V. Taylor, X. Wu, and R. Stevens, "Prophesy: An infrastructure for performance analysis and modeling of parallel and grid applications," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 4, pp. 13–18, 2003.

[9] *http://ark.intel.com/products/64592/*, Intel Ark.

[10] J. D. McCalpin, "Memory bandwidth and machine balance in current high performance computers," *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, Dec. 1995.

[11] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[12] *http://download.intel.com/products/processor/manual/253669.pdf*, Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Two new efficient methods for solving systems of nonlinear equations

Taher Lotfi [1], Katayoun Mahdiani[1], Parisa Bakhtiari[2], Alicia Cordero[3] and Juan R. Torregrosa[3]

[1] *Department of Mathematics, Islamic Azad University, Hamedan Branch, Hamedan, Iran*

[2] *Young Researchers Club, Islamic Azad University, Hamedan Branch, Hamedan, Iran*

[3] *Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain*

emails: Corresponding author: lotfitaher@yahoo.com, lotfi@iauh.ac.ir, mahdiani@iauh.com, bakhtiari@iauh.ac.ir, acordero@mat.upv.es, jrtorre@mat.upv.es

Abstract

In this paper, we present two new iterative methods to solve systems of nonlinear equations. These methods require the evaluation of first-order Frechet derivative and the main advantage of them is that we achieve high convergence orders using appropriate computations of Jacobians. The error analysis is presented to prove the convergence order.
Key words: Nonlinear systems; Matrix; LU factorization; Computational complexity

MSC 2000: 65H10

1 Introduction

Let the function $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ has at least, second-order Frechet derivatives with continuity on a convex set D. Suppose that the equation $F(x) = 0$ has a solution $\alpha \in D$. In this work, we introduce two methods of multi-step iterative methods free from second or higher-order Frechet derivatives for solving nonlinear systems of equations.

In this paper, we improve the convergence behavior of Weerakoon and Fernando method

[2] and also Parhi and Gupta [1] to solve systems of nonlinear equations.

It is widely known that the Newton's method in several variables could be written as

$$x^{(k+1)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \quad k = 0, 1, 2, \cdots . \tag{1.1}$$

Another famous scheme for solving systems of nonlinear equations is the Jarratt fourth-order method which is the generalization of the scheme in the scalar case given in [3] as follows

$$\begin{cases} y^{(k)} = x^{(k)} - \frac{2}{3}F'(x^{(k)})^{-1}F(x^{(k)}), \\ x^{(k+1)} = x^{(k)} - \frac{1}{2}(3F'(y^{(k)}) - F'(x^{(k)}))^{-1}(3F'(y^{(k)}) + F'(x^{(k)}))F'(x^{(k)})^{-1}F(x^{(k)}). \end{cases} \tag{1.2}$$

Note that although most of the works emphasizes on the numerical aspects of these iterations without some theoretics, there are two general ways for pursuing this aim analytically. One is based on the well-known n-dimensional Taylor expansion [4] and second is based on the matrix approach, which is so-called as Point of Attraction, introduced first in [6]. We here apply the first case by reminding the following:

Let $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be sufficiently Frechet differentiable in D. By using the notation introduced in [4], the qth derivative of F at $u \in \mathbb{R}^n$, $q \geq 1$, is the q-linear function $F^{(q)}(u) : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ such that $F^{(q)}(u)(v_1, \ldots, v_q) \in \mathbb{R}^n$. It is well known that, for $\mathbf{x}^* + h \in \mathbb{R}^n$ lying in a neighborhood of a solution $\mathbf{x}^*$ of the nonlinear system $F(\mathbf{x}) = 0$, Taylor's expansion can be applied and we have

$$F(\mathbf{x}^* + h) = F'(\mathbf{x}^*)\left[h + \sum_{q=2}^{p-1} C_q h^q\right] + O(h^p), \tag{1.3}$$

where $C_q = (1/q!)[F'(\mathbf{x}^*)]^{-1}F^{(q)}(\mathbf{x}^*)$, $q \geq 2$. We observe that $C_q h^q \in \mathbb{R}^n$ since $F^{(q)}(\mathbf{x}^*) \in \mathcal{L}(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$ and $[F'(\mathbf{x}^*)]^{-1} \in \mathcal{L}(\mathbb{R}^n)$. In addition, we can express F' as

$$F'(\mathbf{x}^* + h) = F'(\mathbf{x}^*)\left[I + \sum_{q=2}^{p-1} qC_q h^{q-1}\right] + O(h^p), \tag{1.4}$$

wherein I is the identity matrix, and $qC_q h^{q-1} \in \mathcal{L}(\mathbb{R}^n)$.

2 Development of the first iterative method

We here propose a contributed high-order method of this paper for finding solution of the nonlinear systems in what follows

$$\begin{cases} y^{(k)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} = x^{(k)} - 2(F'(x^{(k)}) + F'(y^{(k)}))^{-1}F(x^{(k)}), \\ x^{(k+1)} = z^{(k)} - (3F'(y^{(k)}) - F'(x^{(k)}))^{-1}(F'(x^{(k)}) + F'(y^{(k)}))F'(x^{(k)})^{-1}F(z^{(k)}), \end{cases} \tag{2.1}$$

 ISBN: 978-84-616-2723-3

Theorem 2.1. *Let $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be sufficiently Frechet differentiable at each point of an open convex neighborhood D of $x^* \in \mathbb{R}^n$, that is a solution of the system $F(x) = 0$. Let us suppose that $F'(x)$ is continuous and nonsingular in x^*. Then, the sequence $\{x^{(k)}\}_{k\geq 0}$ obtained using the iterative method (2.1) converges to x^* with convergence rate six.*

Proof. Note that in what follows, $e^{(k)} = x^{(k)} - x^*$ is the *error* in the kth iteration and $e^{(k+1)} = Le^{(k)^p} + O(e^{(k)^{p+1}})$ is the *error equation*, where L is a p-linear function, i.e. $L \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n, \ldots, \mathbb{R}^n)$ and p is the order of convergence. Observe that $e^{(k)^p} = (e^{(k)}, e^{(k)}, \ldots, e^{(k)})$.

From (1.3) and (1.4), we obtain $F(x^{(k)}) = F'(x^*)[e^{(k)} + C_2 e^{(k)^2} + C_3 e^{(k)^3} + C_4 e^{(k)^4} + C_5 e^{(k)^5} + C_6 e^{(k)^6}] + O(e^{(k)^7})$, and

$$F'(x^{(k)}) = F'(x^*)\big[I + 2C_2 e^{(k)} + 3C_3 e^{(k)^2} + 4C_4 e^{(k)^3} + 5C_5 e^{(k)^4} + 6C_6 e^{(k)^5}\big] + O(e^{(k)^6}), \quad (2.2)$$

where $C_k = (1/k!)[F'(x^*)]^{-1} F^{(k)}(x^*)$, $k = 2, 3, \ldots$. From (2.2), we have

$$[F'(x^{(k)})]^{-1} = \left[I + X_1 e^{(k)} + X_2 e^{(k)^2} + X_3 e^{(k)^3} + \cdots\right][F'(x^*)]^{-1} + O(e^{(k)^6}), \quad (2.3)$$

where $X_1 = -2C_2$, $X_2 = 4C_2^2 - 3C_3$, $X_3 = -8C_2^3 + 6C_2C_3 + 6C_3C_2 - 4C_4$, ... Note that $e^{(k)^p}$ is a singular matrix, not a vector. Then,

$$\begin{aligned}[F'(x^{(k)})]^{-1} F(x^{(k)}) &= e^{(k)} - C_2 e^{(k)^2} + 2(C_2^2 - C_3)e^{(k)^3} \\ &+(-4C_2^3 + 4C_2C_3 + 3C_3C_2 - 3C_4)e^{(k)^4} + \cdots + O(e^{(k)^7}), \end{aligned} \quad (2.4)$$

and subsequently $y^{(k)} = x^* + e^{(k)} + C_2 e^{(k)^2} + 2(C_2^2 - C_3)e^{(k)^3} + (-4C_2^3 + 4C_2C_3 + 3C_3C_2 - 3C_4)e^{(k)^4} + \cdots + O(e^{(k)^7})$. The Taylor expansion of the Jacobian matrix $F'(y^{(k)})$ is

$$\begin{aligned} F'(y^{(k)}) &= F'(x^*)[I + 2C_2(y^{(k)} - x^*) + 3C_3(y^{(k)} - x^*)^2 \\ &+ 4C_4(y^{(k)} - x^*)^3 + 5C_5(y^{(k)} - x^*)^4] + O(e^{(k)^5}) \\ &= F'(x^*)\left[I + N_2 e^{(k)^2} + N_3 e^{(k)^3}\right] + \cdots + O(e^{(k)^7}), \end{aligned} \quad (2.5)$$

where $N_2 = 2C_2^2$, and $N_3 = 2C_2(-2C_2^2 + 2C_3)$. Therefore, we obtain

$$z^{(k)} - x^* = \left(C_2^2 + \frac{C_3}{2}\right) e^{(k)^3} + \frac{1}{2}\left(-6C_2^3 + 2C_2C_3 + C_3C_2 + C_4\right) e^{(k)^4} + \cdots + O(e^{(k)^7}). \quad (2.6)$$

Hence, taking into account (2.6), it will be easy to write the Taylor series of $F(z^{(k)})$ as follows

$$F(z^{(k)}) = [F'(x^*)]\left(C_2^2 - \frac{C_3}{2}\right) e^{(k)^3} + \frac{1}{2}\left(-6C_2^3 + 2C_2C_3 + C_3C_2 + C_4\right) e^{(k)^4} + \cdots + O(e^{(k)^7}). \quad (2.7)$$

 ISBN: 978-84-616-2723-3

We now should find the Taylor series at the third step of (2.1), thus using (2.3) and (2.7), we have

$$F'(x^{(k)})^{-1}F(z^{(k)}) = \left(C_2^2 - \frac{C_3}{2}\right) e^{(k)^3} + \frac{1}{2}\left(-10C_2^3 + C_2C_3 + 2C_4\right) e^{(k)^4} + \cdots + O(e^{(k)^7}). \tag{2.8}$$

By using (2.8), and similar terminology, we have the final error equation

$$e^{(k+1)} = C_2\left(C_2^4 - 2C_2^2C_3 - \frac{5C_3^2}{4}\right) e^{(k)^6} + O(e^{(k)^7}), \tag{2.9}$$

which shows that the new method has sixth order of convergence for solving systems of nonlinear equations. □

3 Development of the second iterative method

We here propose a contributed high-order method of this paper for finding real solutions of the nonlinear systems in what follows

$$\begin{cases} y^{(k)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} = x^{(k)} - 2(F'(x^{(k)}) + F'(y^{(k)}))^{-1}F(x^{(k)}), \\ x^{(k+1)} = z^{(k)} - (F'(x^{(k)}) + F'(y^{(k)}))^{-1}(3F'(x^{(k)}) - F'(y^{(k)}))F'(x^{(k)})^{-1}F(z^{(k)}), \end{cases} \tag{3.1}$$

Theorem 3.1. *Let $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be sufficiently Frechet differentiable at each point of an open convex neighborhood D of $x^* \in \mathbb{R}^n$, that is a solution of the system $F(x) = 0$. Let us suppose that $F'(x)$ is continuous and nonsingular in x^*. Then, the sequence $\{x^{(k)}\}_{k\geq 0}$ obtained using the iterative method (3.1) converges to x^* with convergence rate five.*

Proof. Similar to the proof of Theorem 2.1, we have then the final error equation

$$e^{(k+1)} = C_2^2\left(4C_2^2 + 2C_3\right) e^{(k)^5} + O(e^{(k)^6}), \tag{3.2}$$

which shows that the new method has fifth order of convergence for solving systems of nonlinear equations. □

4 Computational complexities

In first iterative method (2.1), we solve four linear systems of equations, while three LU factorizations must be done due to multiple right hand sides. Also, for using the second

 ISBN: 978-84-616-2723-3

iterative method, we solve four linear systems of equations, while just two LU factorizations must be done due to multiple right hand sides. In these situations, one could compute a factorization of the (Jacobian) matrix and use it repeatedly.

The iterative method (2.1) has the following cost: n evaluations of scalar functions for $F(x)$, n evaluations of scalar functions for $F(z)$, n^2 evaluations of scalar functions for $F'(x)$, again n^2 evaluations of scalar functions for $F'(y)$ and three LU decompositions for solving the linear systems involved.

The computation of LU decomposition by any of the existing algorithms in the literature normally requires $\frac{2n^3}{3}$ flops in floating point arithmetic, while the floating point operations for solving the two triangular systems will be $2n^2$ when the right hand side of the systems is a vector, and $2n^3$, or roughly n^3 (as considered in this paper), when the right hand side is a matrix.

The computational efficiency of the new algorithms has now been computed by a practical efficiency index, defined by

$$FEI = p^{\frac{1}{C}}, \tag{4.1}$$

also known as flops-like efficiency index [5], wherein C stands for the total computational cost per iteration in terms of the number of functional evaluations along with cost of LU decompositions and solving two triangular systems (based on the flops), to observe the competence of distinctive methods. Note that for the flops-like efficiency indices, we have $FEI_{(1.1)} = 2^{\frac{1}{n+3n^2+\frac{2n^3}{3}}}$, $FEI_{(1.2)} = 4^{\frac{1}{n+4n^2+\frac{7n^3}{3}}}$,for frist method $FEI_{(2.1)} = 6^{\frac{1}{2n+8n^2+3n^3}}$ and for second method $FEI_{(3.1)} = 5^{\frac{1}{2n+8n^2+\frac{7n^3}{3}}}$.

5 Numerical implementation

In this section, we want to apply our methods to solve three examples, taken from [4]. Also we compare the computed results and justify the accuracy and applicability of the mentioned algorithm and theorems. In fact, we want to estimate the zeros of the following nonlinear systems. Also in tables A_i means $\|x^{(i)} - x^*\| + \|F(x^{(i)})\|$ and $a \times 10^{-b}$ was shown with $a(-b)$.

$$(i) \quad F(x_1, x_2) = (\sin(x_1) + x_2 \cos(x_1), x_1 - x_2), \quad x^* = (0,0)^T, \quad x^{(0)} = (0.8, 0.8)^T.$$

$$(ii) \quad F(x_1, x_2) = \left(\exp(x_1^2) - \exp(\sqrt{2}x_1), x_1 - x_2\right), \quad x^* = \left(\sqrt{2}, \sqrt{2}\right)^T, \quad x^{(0)} = (0.1, 1.1)^T.$$

$$(iii) \quad F(x_1, x_2) = \left(x_1^2 + x_2^2 - 1, x_1^2 - x_2^2 + \frac{1}{2}\right), \quad x^* = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^T, \quad x^{(0)} = (2, 2)^T.$$

As it can be observed from Tables 1-3, our developed methods work in action very efficiently.

 ISBN: 978-84-616-2723-3

Methods	A_1	A_2	A_3	A_4	A_5
Newton Method	1.081347	1.96980	0.24646	0.01065	2.44544(−8)
Jarratt Method	0.36510	0.00070	8.46419(−17)	4.10363(−97)	5.55178(−486)
Frist Method	1.95571	0.01098	6.31628(−19)	1.31219(−132)	2.19151(−928)
Second Method	1.88613	0.00190	1.41255(−21)	1.52068(−151)	0

Table 1: solutions for frist system

Methods	A_1	A_2	A_3	A_4	A_5
Newton	13.81680	4.61691	1.28240	0.21494	0.009634
Jarratt	3.10308	0.05366	4.98801(−8)	3.96362(−32)	1.58035(−128)
Frist method	0.16012	3.68313(−8)	3.17899(−48)	1.31439(−288)	0
Second method	5.059485	0.26508	1.26862(−5)	5.24590(−27)	6.34254(−134)

Table 2: solutions for Second system

Methods	A_1	A_2	A_3	A_4	A_5
Newton	3.16798	0.78191	0.11717	0.00431	6.55581(−6)
Jarratt	0.78191	0.00431	1.51951(−11)	2.35604(−45)	1.361744(−180)
Frist method	0.28530	1.29497(−6)	2.60515(−38)	1.72692(−228)	0
Second method	0.35841	2.38741(−4)	1.57255(−26)	1.42781(−128)	1.58712(−648)

Table 3: solutions for third system

 ISBN: 978-84-616-2723-3

6 Conclusion

Two new and efficient multi-point iterations have been developed for solving systems of nonlinear equations.

Acknowledgements

This work has been partially supported by Islamic Azad University- Hamedan Branch.

References

[1] S. K. Parhi, D. K. Gupta, *A sixth order method for nonlinear equations*, Appl. Math. Comput., **203** (2008), 50–55.

[2] S. Weerakoon, T. C. I. Fernando, *A variant of Newton's method with accelerated third-order convergence*, Appl. Math. Left. **13** (8), (2000), 87–93.

[3] P. Jarratt, *Some fourth order multipoint iterative methods for solving equations*, Math. Comput., **20** (1966), 434–437.

[4] A. Cordero, J. L. Hueso, E. Martinez, J. R. Torregrosa,*A modified Newton-Jarratt's composition*,Numer. Algor. **55** (2010), 87-99.

[5] F. Soleymani, T. Lotfi, P. Bakhtiari, *A multi-step class of iterative methods for nonlinear systems,* Optim. Lett., **7** 2013, DOI 10.1007/s11590-013-0617-6

[6] J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, New York, 1970.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Some three-step iterative methods with memory with highest efficiency index

Taher Lotfi [1], Elaheh Tavakoli[1], Katayoun Mahdiani[1], Alicia Cordero[2] and Juan R. Torregrosa[2]

[1] *Department of Mathematics, Islamic Azad University, Hamedan Branch, Hamedan, Iran*

[2] *Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain*

emails: Corresponding author: lotfitaher@yahoo.com, lotfi@iauh.ac.ir, tavakoli-18367@yahoo.com, mahdiani@iauh.com, acordero@mat.upv.es, jrtorre@mat.upv.es

Abstract

It is attempted to introduce some new methods with memory for solving nonlinear equations. This family uses four function evaluations per cycle and is derivative free with order of convergence 12.
Key words: nonlinear equations, iterative methods with memory

MSC 2000: 65H05

1 Introduction

Traub introduce for the first time an Steffensen-like method with memory in his book [4]. Indeed he could increase the order of Steffensen method [3] by applying a new parameter and polynomial interpolation to approximate the mentioned parameter in a genuine way. Traub's idea has not been considered until very recently [2, 5].

In this study our primary aim is to introduce a new class of three-point method with memory. To do this, we first develop an optimal new free-derivative class without memory. This class uses only four function evaluations per iterate. Then, we try to modify it using the idea on accelerator in such a way that without any extra function evaluation the order of convergence increase from 8 to 12. Finally, to justify our method numerically, we implement it using some example and compare it to some fresh methods in the same class.

2 Development a new derivative free three-point class without memory

Let α be a simple real zero of a real function $f : D \subset R \to R$ and let x_0 be an initial approximate to α. To construct derivative free three-point methods of optimal order eight, let us start from the tripled Newton's method (iteration indices are dropped for simplicity)

$$\begin{cases} y = x - \frac{f(x)}{f'(x)}, \\ z = y - \frac{f(y)}{f'(y)}, \\ \widehat{x} = z - \frac{f(z)}{f'(z)}. \end{cases} \tag{2.1}$$

It is well known that the order of convergence of scheme (2.1) is eight, but its computational efficiency is low. To improve this disadvantage, we substitute in all three steps by suitable approximations that use available data, not including calculation of derivatives. To provide these requirements, in the first step we approximate

$$f'(x) \approx f[x, w], \quad \text{where } w = x + \gamma f(x),\ 0 \neq \gamma \in R,$$

and $f[x, y] = \frac{f(x)-f(y)}{x-y}$ denotes a divided difference. Similarly, the other two derivative can be approximated by weight functions with two variables. We introduce approximations

$$f'(y) \approx \frac{f[y, w]}{H(t, u)},\ t = \frac{f(y)}{f(x)},\ u = \frac{f(w)}{f(x)},\ f'(z) \approx \frac{f[z, w]}{G(t, s)\, W(v, s)},\ s = \frac{f(z)}{f(y)},\ v = \frac{f(z)}{f(x)},$$

and apply them in the second and third step of (2.1), where H, G and W are weight functions. So, the following iterative family of three-point family is obtained

$$\begin{cases} y = x - \frac{f(x)}{f[x,w]}, \\ z = y - H(t, u)\, \frac{f(y)}{f[y,w]}, \\ \widehat{x} = z - G(t, s)\, W(v, s)\, \frac{f(z)}{f[z,w]}, \end{cases} \tag{2.2}$$

and the function H, G and W should be determined in such a way that the order of convergence of the three-point method (2.2) is eight.
Now we state the following convergent theorem for the family (2.2).

Theorem 2.1. *Let H(t,u), G(t,s) and W(v,s) be differentiable two-variable functions that satisfy the conditions* $H(0,0) = W(0,0) = G(0,0) = H_{1,0}(0,0) = G_{1,0}(0,0) = G_{0,1}(0,0) = 1,\ H_{0,1}(0,0) = H_{0,2}(0,0) = H_{0,3}(0,0) = H_{1,1}(0,0) = H_{1,2}(0,0) = H_{2,0}(0,0) = H_{2,1}(0,0) = G_{2,0}(0,0) = W_{1,0}(0,0) = W_{0,1}(0,0) = 0,\ G_{1,1}(0,0) = 2$ *and* $G_{3,0}(0,0) = H_{3,0}(0,0) - 6 - \frac{6}{1+\gamma f[x,w]}$. *If an initial approximation* x_0 *is sufficiently close to the root* α *of a function*

 ISBN: 978-84-616-2723-3

f, then the convergence order of the family of three-point method without memory (2.2) is equal to eight. Moreover, we have

$$\begin{aligned}\widehat{e} = &-\frac{1}{6}(c_2(1+\gamma f'(\alpha))^4(-c_3+c_2^2(3+\gamma f'(\alpha)))(-6c_2c_4+3c_3^2(-2+G_{0,2}+W_{0,2}) \\ &-3c_2^2c_3(-22+6G_{0,2}+G_{2,1}+6W_{0,2}+\gamma f'(\alpha)(-6+2G_{0,2}+G_{2,1}+2W_{0,2})) \\ &+c_2^4(-H_{3,0}(1+\gamma f'(\alpha))^2+3G_{2,1}(1+\gamma f'(\alpha))(3+\gamma f'(\alpha))+3G_{0,2}(3+\gamma f'(\alpha))^2 \\ &+3(W_{0,2}(3+\gamma f'(\alpha))^2-2(13+\gamma f'(\alpha)(7+\gamma f'(\alpha))))))e^8+O[e^9],\end{aligned} \tag{2.3}$$

We denote by $H_{i,j}$, $G_{i,j}\,W_{i,j}$, the $(i,j)th$ partial derivatives with respect to their variables.

Some explicit forms of weight functions

Now we present some simple weight functions satisfying the conditions of Theorem (2.1)

$$\begin{aligned}H(t,u) &= 1+t, \\ G(t,s) &= 1+t+s+2ts+(-1-\phi)t^3, \\ W(s,v) &= 1+s^2+v^2,\end{aligned} \tag{2.4}$$

and

$$\begin{aligned}H(t,u) &= 1+t, \\ G(t,s) &= \frac{\frac{1}{1+\phi}(1+t+s+2ts)+t^2}{\frac{1}{1+\phi}+t^2}, \\ W(s,v) &= 1+\frac{s^2}{v^2+1},\end{aligned} \tag{2.5}$$

where $\phi = 1/(1+\gamma f[x,w])$.

We apply these weight functions in the proposed method (2.2) as concretes in the numerical section and compare them with some existing methods in the same class.

3 New family of three-point methods with memory

We deduce from (2.3) that the order of convergence of the method (2.2) is eight when $\gamma \neq -1/f'(\alpha)$. If $\gamma = -1/f'(\alpha)$, then the order of convergence is greater than 8, but $f'(\alpha)$ is unknown in practice since α is unknown. To this end, one can approximate $f'(\alpha)$ using available data. Here we deal with this problem. In fact, we calculate the forth degree of Newton's interpolation for approximate $f'(\alpha)$. In other words

$$\gamma = \frac{-1}{f'(\alpha)} \approx \frac{-1}{N_4'(x_k)} = \gamma_k, \tag{3.1}$$

 ISBN: 978-84-616-2723-3

where

$$N_4'(x_k) = f[x_k, z_{k-1}] + f[x_k, z_{k-1}, y_{k-1}](x_k - z_{k-1}) + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](x_k - z_{k-1})(x_k - y_{k-1}) \tag{3.2}$$
$$+ f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}](x_k - z_{k-1})(x_k - y_{k-1})(x_k - x_{k-1}).$$

Considering Theorem (2.1) and (3.1) we construct the following new derivative free method with memory

$$\begin{cases} \gamma_0 \text{ and } x_0 \text{ are given,} \\ w_k = x_k + \gamma_k f(x_k), \\ y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\ z_k = y_k - H(t_k, u_k)\frac{f(y_k)}{f[y_k, w_k]}, \\ x_{k+1} = z_k - G(t_k, s_k)W(v_k, s_k)\frac{f(z_k)}{f[z_k, w_k]}, \quad k = 0, 1, \cdots, \\ \gamma_k = \frac{-1}{N_4'(x_k)}, \quad k = 1, 2, \cdots. \end{cases} \tag{3.3}$$

To determine the order of convergence of the method (3.3) with memory, where $\gamma \approx \gamma_k$ is estimated by (3.1), we state the following convergence theorem.

Theorem 3.1. *If an initial approximation x_0 is sufficiently close to a simple zero α of f, then the R-order of convergence of the three-point method with memory (3.3) is at least 12.*

Proof. We will use Herzbergers matrix method [1] to determine the R-order of convergence for (3.3). In other words, the lower bound of order of a single step s-point method $x_k = G(x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4})$ is the spectral radius of a matrix $M^{(s)} = (m_{i,j})$, associated to this method, with elements

$$\begin{cases} m_{1,j} = \text{amount of information required at point} x_{k-j}, \ j = 1, 2, 3, 4. \\ m_{i,i-1} = 1, \ \ i = 2, 3, 4 \\ m_{i,j} = 0, \quad \text{otherwise.} \end{cases} \tag{3.4}$$

The lower bound of order of an s-point method $G = G_1 o G_2 o G_3 o G_4$ is the spectral radius of the product of matrices $M = M_1.M_2.M_3.M_4$. More precisely, we have

$$M = \begin{bmatrix} 8 & 4 & 4 & 4 & 4 \\ 4 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3.5}$$

with the eigenvalues $12, 0, 0, 0, 0$. Hence, the R-order of the method with memory (3.3) is at leat 12. □

Remark 3.2. *If we take $\gamma_k = 1/N_i'(x_k), \ i = 1, 2, 3$, then we achieve lower R-orders which has no practical interest.*

 ISBN: 978-84-616-2723-3

4 Numerical results and comparisons

In this section, the family of three-point methods (3.3) is tested on two nonlinear equations along with several three-point iterative methods of optimal order eight. Some of them are displayed bellow:

The first concrete of the new method with memory (3.3)

$$\begin{cases} \gamma_0 \text{ and } x_0 \text{ are given,} \\ w_k = x_k + \gamma_k f(x_k), \\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]}, \\ z_k = y_k - (1+t_k)\frac{f(y_k)}{f[y_k,w_k]}, \\ x_{k+1} = z_k - \big(1+t_k+s_k+2t_ks_k+(-1-\phi_k)t_k^3\big)(1+s_k^2+v_k^2)\frac{f(z_k)}{f[z_k,w_k]}, \quad k=0,1,\cdots, \\ \gamma_k = \frac{-1}{N_4'(x_k)}, \quad k=1,2,\cdots. \end{cases} \tag{4.1}$$

where $\phi_k = 1/(1+\gamma_k f[x_k,w_k])$.

The second concrete of the new method with memory (3.3)

$$\begin{cases} \gamma_0 \text{ and } x_0 \text{ are given,} \\ w_k = x_k + \gamma_k f(x_k), \\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]}, \\ z_k = y_k - (1+t_k)\frac{f(y_k)}{f[y_k,w_k]}, \\ x_{k+1} = z_k - \Big(\frac{\frac{1}{1+\phi_k}(1+t+s+2ts)+t^2}{\frac{1}{1+\phi_k}+t^2}\Big)\Big(1+\frac{s^2}{v^2+1}\Big)\frac{f(z_k)}{f[z_k,w_k]}, \quad k=0,1,\cdots, \\ \gamma_k = \frac{-1}{N_4'(x_k)}, \quad k=1,2,\cdots. \end{cases} \tag{4.2}$$

where $\phi_k = 1/(1+\gamma_k f[x_k,w_k])$.

For demonstration, among many numerical experiments, we have selected a test problem. The errors $|x_n-\alpha|$ of approximations to the zeros are given in Table 1 where $A(-h)$ denotes $A\times 10^{-h}$. These tables include the values of the computational order of convergence r_c calculated by the formula [5]

$$r_c = \frac{\log|f(x_n)/f(x_{n-1})|}{\log|f(x_{n-1})/f(x_{n-2})|}. \tag{4.3}$$

 ISBN: 978-84-616-2723-3

methods	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$	r_c (4.3)
NEW: (4.1)	0.71066(-4)	0.20396(-49)	0.49715(-596)	12.0
NEW: (4.2)	0.80715(-4)	0.15495(-49)	0.65738(-595)	12.0
[5]	0.90460(-5)	0.81817(-56)	0.30880(-670)	12.0

Table 1: $f(x) = \exp(x^2 + x\cos x - 1)\sin \pi x + x\log(x\sin x + 1),\ x_0 = 0.6,\ \alpha = 0,\ \gamma = -0.1$

5 Conclusion

A new class with memory has been introduced using four function evaluations per iterate. This family does not require any derivative and has 12 order of convergence.

Acknowledgements

This work has been partially supported by Islamic Azad University- Hamedan Branch.

References

[1] J. Herzberger, *Uber Matrixdarstellungen fur Iterationverfahren bei nichtlinearen Gleichungen*, Computing **12** (1974) 215222.

[2] F. Soleymani, *Some optimal iterative methods and their with memory variants*, Journal of the Egyptian Mathematical Society, (2013), http://dx.doi.org/10.1016/j.joems.2013.01.002

[3] J. F. Steffensen,Remarks on iteration, Skand. Aktuarietidskr **16** (1933) 6472.

[4] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, 1964.

[5] X. Wang, J. Dzunic, T. Zhang, *On an efficient family of derivative free three-point methods for solving nonlinear equations*, Applied Mathematics and Computation **219** (2012) 1749-1760.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

On generalization based on Bi et al iterative methods with eight-order convergence for solving nonlinear equations

Taher Lotfi [1], Maryam Mohammadi Zadeh[1] and Morteza Amir Abadi[1]

[1] *Department of Mathematics, Islamic Azad University, Hamedan Branch, Hamedan, Iran*

emails: lotfitaher@yahoo.com.com, lotfi@iauh.ac.ir, ,

Abstract

The primarily goal of this paper is to provide an optimal three-step class. In other words, using four function evaluation per iterate with eight-order convergence. Moreover, the proposed class includes both Bi et al families as particular cases.
Key words: nonlinear equations, optimal, iterative methods, without memory

MSC 2000: 65H05

1 Introduction

Multipoint methods for solving nonlinear equations $f(x) = 0$, where $f : D \subset R \to R$, possess an important advantage since they overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency [1, 2, 4]

In this paper we present a new optimal family of three-point methods without memory which employs the idea of weight functions in the second and third steps. The order of this family is eight requiring four function evaluations so supports the Kung and Traub conjecture [3]. The proposed class includes the most cited papers by Bi et al. [1, 2] in three last years. It is supposed they are the first optimal three point methods in this field and have made great progresses in the studies as well.

In order to construct new methods, we need the knowledge of divided differences. Let $f(x)$ be a function defined on an interval I, where I is the smallest interval containing $k+1$ distinct nodes $x_1, x_2, \ldots, x_k$. The divided difference $f[x_0, x_1, \ldots, x_k]$ with kth-order is defined as follows: $f[x_0] = f(x_0)$

$$f[x_0] = \frac{f[x_1] - f[x_0]}{x_1 - x_0},\ \cdots,\ f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, x_2, \ldots, x_k] - f[x_0, x_1, \ldots x_{k-1}]}{x_k - x_0}.$$

 ISBN: 978-84-616-2723-3

It is clear that the divided difference $f[x_0, x_1, \dots x_k]$ is a symmetric function of its arguments $x_0, x_1, \dots x_k$. Moreover if we assume that $f \in C^{(k+1)}(I_x)$ where I_x is the smallest interval containing the nodes $x_0, x_1, \dots x_k$ and x, then $f[x_0, x_1, \dots x_k] = \dfrac{f^{(k+1)}(\xi)}{(k+1)!}$, for a suitable $\xi \in I_x$. Specially, if $x_0 = x_1 = \ldots = x_k = x$, then

$$f[x, x, \dots, x, x] = \frac{f^{(k+1)}(x)}{(k+1)!}.$$

2 The method and analysis of convergence

Newton's method converges quadratically. To obtain a higher convergence order and a higher efficiency index than that of Newton's method, we consider the following three-step Newton's method (omitting iteration index for simplicity)

$$\begin{cases} y = x - \dfrac{f(x)}{f'(x)}, \\ z = y - \dfrac{f(y)}{f'(y)}, \\ \hat{x} = z - \dfrac{f(z)}{f'(z)}. \end{cases} \tag{2.1}$$

$f'(z)$ in the third step can be approximated given in [1, 2] as follows

$$f'(z) \approx f[z, y] + f[z, x, x](z - y).$$

We can easily prove that Scheme (2.1) has convergent order eighth and it requires six function evaluations . it has an efficiency index of [1] $8^{\frac{1}{6}} = 1.414$, which is the same as Newton's method. In other words, it does not increase the computational efficiency. To derive a scheme with a higher efficiency index, the following three step methods is proposed

$$\begin{cases} y = x - \dfrac{f(x)}{f'(x)}, \\ z = y - g(s)\dfrac{f(y)}{f'(x)}, \quad s = \dfrac{f(y)}{f(x)} \\ \hat{x} = z - h(t)\dfrac{f(z)}{f[z, y] + f[z, x, x](z - y)}, \quad t = \dfrac{f(z)}{f(x)}. \end{cases} \tag{2.2}$$

It is clear that the proposed method by (2.2) requires only four function evaluations per iteration while it is not eight order method. To recover the optimal eight order, we find some suitable conditions on the introduced weight functions $g(s)$, and $h(t)$.

To find the weight functions g and h in (2.2) providing order eight, we will use the method of undetermined coefficients and Taylor's series about 0 since $t \to 0$, $s \to 0$, when $x \to 0$.

 ISBN: 978-84-616-2723-3

Let

$$g(s) = g(0) + g'(0)s + g''(0)\frac{s^2}{2} + g'''(0)\frac{s^3}{6} + \ldots,$$

and

$$h(t) = h(0) + h'(0)t + h''(0)\frac{t^2}{2} + h'''(0)\frac{t^3}{6} + \ldots.$$

The simplest method for finding the coefficient of the above Taylor expansions is the use of symbolic computation by a computer algebra system. In the following we disclose our Mathematica code which provides required conditions. Because of simplicity of the given code, we prevent ourselves to add comments for them. Program (written in Mathematica)

$\mathtt{f[e\_]} = \mathtt{f1a}(\mathtt{e} + \sum_{\mathtt{k}=2}^{8} \mathtt{c_k e^k})$;
$\mathtt{e_y} = \mathtt{e} - \mathtt{Series}[\frac{\mathtt{f[e]}}{\mathtt{f'[e]}}, \mathtt{e}, 0, 8]$;
$\mathtt{s} = \frac{\mathtt{f[e_y]}}{\mathtt{f[e]}}$;
$\mathtt{g[s\_]} = \mathtt{g_0} + \mathtt{g_1 s} + \frac{\mathtt{g_2}}{2}\mathtt{s}^2 + \frac{\mathtt{g_3}}{6}\mathtt{s}^3 + \frac{\mathtt{g_4}}{24}\mathtt{s}^4$;
$\mathtt{e_z} = \mathtt{e_y} - \mathtt{g[s]} * \frac{\mathtt{f[e_y]}}{\mathtt{f'[e]}}$;
$\mathtt{t} = \frac{\mathtt{f[e_z]}}{\mathtt{f[e]}}$;
$\mathtt{h[t\_]} = \mathtt{h_0} + \mathtt{h_1 t} + \frac{\mathtt{h_2}}{2}\mathtt{t}^2 + \frac{\mathtt{h_3}}{6}\mathtt{t}^3 + \frac{\mathtt{h_4}}{24}\mathtt{t}^4$;
$\mathtt{f[a\_, b\_]} := \frac{\mathtt{f[a]-f[b]}}{\mathtt{a-b}}$;
$\mathtt{f[a\_, b\_, b\_]} := \frac{\mathtt{f[a,b]-f'[b]}}{\mathtt{a-b}}$; $\mathtt{e_T} = \mathtt{e_z} - \mathtt{h[t]} * \frac{\mathtt{f[e_z]}}{\mathtt{f[e_z,e_y]+f[e_z,e,e](e_z-e_y)}}$;
$\mathtt{a2} = \mathtt{Coefficient}[\mathtt{e_T}, \mathtt{e}^2]//\mathtt{Fullsimplify}$
$\mathtt{Out[a2]} = \mathtt{c_2}(-1 + \mathtt{g_0})(-1 + \mathtt{h_0})$
$\mathtt{g_0} = 1; \mathtt{h_0} = 1$ (* Vanish coefficient of e^2 *)
$\mathtt{a3} = \mathtt{Coefficient}[\mathtt{e_T}, \mathtt{e}^3]//\mathtt{Fullsimplify}$
0
$\mathtt{a4} = \mathtt{Coefficient}[\mathtt{e_T}, \mathtt{e}^4]//\mathtt{Fullsimplify}$
0
$\mathtt{a5} = \mathtt{Coefficient}[\mathtt{e_T}, \mathtt{e}^5]//\mathtt{Fullsimplify}$
$\mathtt{Out[a5]} = -\mathtt{c_2^4}(-2 + \mathtt{g_1})^2\mathtt{h_1}$
$\mathtt{g_1} = 2$; (* Vanishes coefficient of e^5 *)
$\mathtt{a6} = \mathtt{Coefficient}[\mathtt{e_T}, \mathtt{e}^6]//\mathtt{Fullsimplify}$
0
$\mathtt{Out[a7]} = -\frac{1}{4}\mathtt{c_2^2}(2\mathtt{c_3} + \mathtt{c_2^2}(-10 + \mathtt{g_2}))(2\mathtt{c_3}(-2 + \mathtt{h_1}) + \mathtt{c_2^2}(-10 + \mathtt{g_2})\mathtt{h_1})$
$\mathtt{g_2} = 10; \mathtt{h_1} = 2$; (*Vanish coefficient of e^7 *)
$\mathtt{e_T}//\mathtt{FullSimplify}$
$\mathtt{Out[e_T]} = -\frac{1}{3}(\mathtt{c_2^2 c_3}(-6\mathtt{c_2 c_3} + 3\mathtt{c_4} + \mathtt{c_2^3}(-84 + \mathtt{g_3})))\mathtt{e}^8 + \mathtt{O[e]}^9$.

To sum up, $g(0) = h(0) = 1$, $g'(0) = h'(0) = 2$, $g''(0) = 10$, $|g'''(0)| < \infty$.

According to the above analysis, we have proved the following theorem.

 ISBN: 978-84-616-2723-3

Theorem 2.1. *Assume that $f \in C^5(D)$. Suppose $x^* \in D$, $f(x^*) = 0$ and $f'(x^*) \neq 0$. If the initial point x_0 is sufficiently close to x^*, then the sequence x_n generated by any method of the family (2.2) has eight-order of convergence to x^* if g and h are any functions with $g(0) = h(0) = 1$, $g'(0) = h'(0) = 2$, $g''(0) = 10$ and $|h''(0)| < \infty$ and $|g''(0)| < \infty$.*

Corollary 2.1. *If we set $g(s) = \dfrac{1+as}{1+(a-2)s}$, $a \in R$, our proposed method results Bi et al's Method [1].*

Corollary 2.2. *If we set $h(t) = \dfrac{1+bt}{1+(b-2)t}$ $b \in R$, the proposed method results Bi et al's Method [2].*

3 Conclusion

In conclusion, a new optimal class of three-step methods without memory has been generalized based on Bi et al families [1, 2].

Acknowledgements

This work has been partially supported by Islamic Azad University- Hamedan Branch.

References

[1] W. Bi, Q. Wu, H. Ren, *A new family of eight-order iterative method for solving nonlinear equations*, Appl. Math. Comput. **214** (2009) 236-45.

[2] W. Bi, H. Ren, Q. Wu, *Three-step iterative methods with eight- order convergence for solving nonlinear equations*, J. Comput. Appl. Math. **225** (2009) 105-12.

[3] H. T. Kung, J. F. Traub, *Optimal order of one-point and multipoint iteration*, J. ACM, **21** (1974) 643-651.

[4] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, 1964.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Geodesic regression on spheres: a numerical optimization approach

L. Machado[1] and T. Monteiro[2]

[1] *Department of Mathematics & ISR, University of Trás-os-Montes and Alto Douro & University of Coimbra*

[2] *R&D Centre Algoritmi, Department of Production and Systems, University of Minho, Portugal*

emails: `lmiguel@utad.pt`, `tm@dps.uminho.pt`

Abstract

In this paper we address the problem of finding a geodesic curve that best fits a given set of time-labeled points on a sphere. Since the corresponding normal equations are highly non-linear, we formulate the problem as a constrained nonlinear optimization problem and solve it using the routine `fmincon` from MATLAB with the SQP (Sequential Quadratic Programming) algorithm.

Key words: Manifolds, geodesics, geodesic distance, normal equations, constrained nonlinear optimization, SQP.

1 Introduction

The astounding development of mechanical and robotics industry in the past few years has required the generalization of classical methods to more general curved spaces. This is mainly due to the fact that the configuration systems of the most part of mechanical systems are particular manifolds, like Lie groups or symmetric spaces. Although obtaining such generalizations is not as straightforward as we might expect.

In this paper we show how to generalize the linear regression problem on Euclidean spaces to the $n-$dimensional unit sphere. This technique of approximating data is a common procedure in several applications from a wide range of fields including statistics, computer vision, signal processing, fuzzi control, air traffic and aeronautics control.

In the classical problem, [6], we are given a collection of points and the same number of instants of time and the objective is to find a parameterized straight line that best fits

the data. However, finding the best approximant curve is not a trivial task especially when the points belong to some curved space. We refer to [5], where it has been developed a variational approach to generate fitting curves on the sphere and to [8] and [9], where this approach has been generalized to the more general context of Riemannian manifolds. The major difficulty that we face is the fact that, in general, explicit formulas to the analogues of straight lines, the so-called geodesics, are not available. Such is not the case of the unit $n-$sphere, when equipped with the metric induced by the metric in the embedding space, where geodesics are the great arc circles, [4].

To get some insight, we start, in Section 2, by recalling the classical linear regression problem on Euclidean spaces. In Section 3, we formulate the corresponding problem on the $n-$dimensional unit sphere and then present the first order necessary optimality conditions. Unlike the Euclidean case, this system of equations is extremely non-linear and numerical optimization methods are put to use in Section 4. In this section, the problem is formulated as a constrained nonlinear optimization problem and numerical experiments using the optimization toolbox from MATLAB will be provided. Some conclusions and future work ideas are carried out in Section 5.

2 Linear regression problem

Although in the most part of the literature, [6, 11, 3], the linear least squares problem is addressed for data in $\mathbb{R}$, the approach to more general Euclidean spaces is straightforward, [7].

Let us consider the Euclidean space $\mathbb{R}^n$ endowed with the usual inner product and let us denote by d the metric induced by the l^2-norm $\|a\| = \langle a, a\rangle^{\frac{1}{2}}$.

In the linear regression problem, we are given a finite set of points in $\mathbb{R}^n$, $p_0, \ldots, p_N$, and a set of instants of time, $t_0, \ldots, t_N$, and wish to find a parameterized straight line $t \longmapsto \gamma(t) = a_0 + a_1 t \in \mathbb{R}^n$, that best fits the given data, in the sense that the functional E, defined by

$$E(\gamma) = \sum_{i=0}^{N} d^2(p_i, \gamma(t_i)),$$

should be as small as possible.

Theorem 2.1. *For each $N \geq 1$, the parameterized straight line $t \mapsto \gamma(t) = a_0 + a_1 t$ that best fits the given data is unique and is the solution of the following system of equations*

$$\begin{cases} \sum_{i=0}^{N} \gamma(t_i) = \sum_{i=0}^{N} p_i \\ \sum_{i=0}^{N} t_i \gamma(t_i) = \sum_{i=0}^{N} t_i p_i \end{cases} . \tag{1}$$

 ISBN: 978-84-616-2723-3

The linear system of equations (1), known in the literature as the normal equations, can be solved explicitly and its unique solution is given by

$$\gamma(t) = \frac{\sum_{i=0}^{N} t_i^2 \sum_{i=0}^{N} p_i - \sum_{i=0}^{N} t_i \sum_{i=0}^{N} t_i p_i}{(N+1)\sum_{i=0}^{N} t_i^2 - \left(\sum_{i=0}^{N} t_i\right)^2} + \frac{(N+1)\sum_{i=0}^{N} t_i p_i - \sum_{i=0}^{N} t_i \sum_{i=0}^{N} p_i}{(N+1)\sum_{i=0}^{N} t_i^2 - \left(\sum_{i=0}^{N} t_i\right)^2} t.$$

This classical problem can be naturally generalized to more general curved spaces as long as explicit formulas for geodesics are available, [8]. Nevertheless, even in those cases obtaining exact solutions is not an easy task mainly because the counterpart of the normal equations give rise to nonlinear systems of equations.

In the next section we will present the generalization of this classical problem to the $n-$dimensional unit sphere S^n.

3 Regression problem on spheres

Let us consider the unit $n-$sphere S^n as an embedded submanifold of the Euclidean space $\mathbb{R}^{n+1}$. Since the tangent space of S^n at a point $p \in S^n$ is

$$T_p S^n = \left\{ v \in \mathbb{R}^{n+1} \ : \ \langle v, p \rangle = 0 \right\},$$

let us define an inner product in $T_p S^n$ by

$$\langle u, v \rangle = u^\top v, \quad u, v \in T_p S^n.$$

With this inner product, S^n can be considered a Riemannian manifold whose metric is the one induced by the Euclidean inner product in $\mathbb{R}^{n+1}$.

Geodesics with respect to this metric are the solutions of the second order differential equation

$$\ddot{\gamma} - \langle \ddot{\gamma}, \gamma \rangle \gamma = 0.$$

The unique geodesic $t \mapsto \gamma(t)$ with initial conditions $\gamma(0) = p \in S^n$ and $\dot{\gamma}(0) = v \in T_p S^n$ is given by

$$\gamma(t) = p \cos(t\|v\|) + \frac{v}{\|v\|} \sin(t\|v\|). \tag{2}$$

If we settled on the $2-$dimensional sphere, S^2, geodesics are simply the great arc circles.

In order to find out the geodesic distance with respect to the above metric, one just needs to compute the velocity vector of the geodesic that joins two points on the sphere. Let us assume that we are given two points p and q on S^n such that the angle between

 ISBN: 978-84-616-2723-3

them lies on the interval $]0,\pi[$. Then, the geodesic that joins p, at $t=0$, to q, at $t=1$, can be parameterized explicitly as

$$\gamma(t) = p\cos(\alpha t) + \frac{q - p\cos\alpha}{\sin\alpha}\sin(\alpha t),$$

where $t \in [0,1]$ and $\alpha = \arccos\langle p, q\rangle$, [2]. Therefore, the geodesic distance between p and q is given by the length of the velocity vector of γ at $t=0$, that is,

$$d(p,q) = \arccos\langle p, q\rangle. \tag{3}$$

3.1 Problem's formulation

Since we have already defined the distance function, we are now in conditions to formulate the analogous to the linear regression problem on S^n. Let us consider a collection of $N+1$ points, $p_0,\ldots,p_N$, on S^n and a monotone increasing sequence of instants of time $t_0 < t_1 \cdots < t_N$, that we assume, for simplicity, that they form a partition of the unit time interval $[0,1]$. Our main goal is to find a geodesic on S^n parameterized explicitly by

$$\gamma(t) = p\cos(t\|v\|) + \frac{v}{\|v\|}\sin(t\|v\|), \tag{4}$$

where $p \in S^n$ and $v \in T_pS^n$, that best fits the given data in the sense that it yields the minimum value for the functional

$$E(\gamma) = \sum_{i=0}^{N} d^2(p_i, \gamma(t_i)),$$

where d is the geodesic distance on S^n defined by (3).

Notice that finding γ is equivalent to find $p \in S^n$ and $v \in T_pS^n$ that minimize the function

$$F(p,v) = \sum_{i=0}^{N} \arccos^2\big\langle p_i, \gamma(t_i)\big\rangle. \tag{5}$$

Theorem 3.1. ([8]) *A necessary condition for $t \mapsto \gamma(t) = p\cos(\|v\|t) + \frac{v}{\|v\|}\sin(\|v\|t)$ to be the geodesic that best fits the given data (points and instants of time) is that the pair (p,v) satisfies the following system of equations:*

$$\begin{cases} \sum_{i=0}^{N} \frac{\alpha_i}{\sin\alpha_i}\cos\big(\|v\|t_i\big)\big(p_i - \langle p_i, p\rangle p\big) = 0 \\ \sum_{i=0}^{N} \frac{\alpha_i \sin(\|v\|t_i)}{\sin\alpha_i}\big(p_i - \langle p_i, p\rangle(t_i v + p) - \frac{\langle p_i, v\rangle}{\|v\|^2}v\big) = \sum_{i=0}^{N} \frac{-\alpha_i\cos(\|v\|t_i)}{\sin\alpha_i}\frac{\langle p_i, v\rangle}{\|v\|}t_i v \end{cases}, \tag{6}$$

where $\alpha_i = \arccos\big\langle p_i, p\cos\big(\|v\|t_i\big) + \frac{v}{\|v\|}\sin\big(\|v\|t_i\big)\big\rangle$, for $i = 0,\ldots,N$.

 ISBN: 978-84-616-2723-3

Analogously to what happens in the Euclidean case, we call to the system of equations (6) the normal equations for the geodesic regression problem on S^n.

Since the above system of equations is highly nonlinear, a numerical optimization approach to find out the approximate solutions of the proposed optimization problem will be considered in the next section.

4 Numerical tests using an optimization approach

Let us consider the constrained optimization problem:

$$\begin{aligned} \min_{p,v} \quad & F(p,v) = \sum_{i=0}^{N} \arccos^2\langle p_i, \gamma(t_i)\rangle \\ \text{s.t.} \quad & \|p\| = 1 \\ & \langle p, v\rangle = 0 \end{aligned} \tag{7}$$

whose objective function is defined in (5). The equality constraints mean that the point p must belong to the unit n-sphere S^n and v must be orthogonal to p.

4.1 Computational experiments

The computational experiments were made on a *2.0 GHz Intel Core i7* with 8GB of RAM, Windows 7 64-bit operating system. The MATLAB version used was 7.13.0.564 (R2011b).

The problem was codified in MATLAB and solved with the `fmincon` routine from the optimization toolbox. This routine, `fmincon`, attempts to find a constrained minimum of a scalar function of several variables starting at an initial estimate. This is generally referred to as constrained nonlinear optimization or nonlinear programming. `fmincon` has four algorithm options: interior-point, SQP, active-set and trust-region-reflective (default). In these experiments the SQP algorithm is used, [10]. SQP is an iterative method for nonlinear optimization used on problems for which the objective function and the constraints are twice continuously differentiable. SQP methods solve a sequence of optimization subproblems, each of which optimizes a quadratic model of the objective objective function subject to a linearization of the constraints.

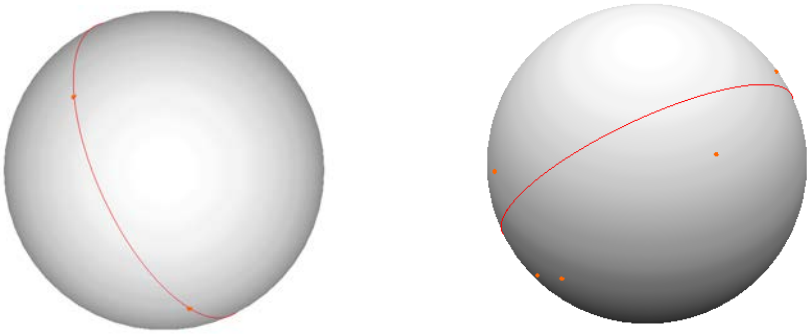
Several numerical tests were performed on the two dimensional unit sphere S^2. Table 1 reports information from two of them. In the first column it is indicated the number of points $(N+1)$ and in the second column the set of instants of time (t). The third column reports the $N+1$ points randomly generated on S^2. p^*, v^* and F^* are the optimal values of p, v and F, respectively, and *iter* denotes the number of iterations carried out by the `fmincon` routine.

 ISBN: 978-84-616-2723-3

Figure 1 shows the geodesic that best fits two (1a) and five given data (1b), respectively, corresponding to the tests mentioned on Table 1. Notice that, as expected, the geodesic that best fits two points is exactly the one that joins them (interpolation case).

Table 1: Numerical tests

$N+1$	t	Data					p^*	v^*	F^*	$iter$
2	{0,1}		0.1585 -0.8624 0.4807		-0.5605 0.4575 0.6903		0.1585 -0.8624 0.4807	-0.9351 0.5696 1.3302	$\approx 10^{-14}$	14
5	{0, 0.25, 0.5, 0.75, 1}	-0.89 -0.3941 -0.2292	0.8243 -0.2085 0.5264	-0.9164 -0.158 -0.3678	-0.9383 0.3022 0.1678	0.0192 -0.7635 0.6455	-0.8006 -0.2176 0.5583	4.6366 -32,542 5.3801	3.4203	47



(a) Two data points (b) Five data points

Figure 1: Geodesics on S^2

5 Conclusions and future work

In this paper we formulated the analogous linear regression problem to the unit $n-$sphere. In contrast to what happens in the Euclidean case this optimization problem cannot be solved analytically due to nonlinearity of the counterpart of the normal equations (6). To overcome this difficulty we have successfully used the MATLAB optimization toolbox routine `fmincon`.

Since in the Euclidean case higher order degree polynomials can be used to fit a given data set of points, as future work we aim to find more general fitting curves on spheres.

 ISBN: 978-84-616-2723-3

Acknowledgements

This work is funded by FEDER funds through Operational Programme for Competitiveness Factors - COMPETE and National Funds through FCT - Foundation for Science and Technology in Projects scope: FCOMP-01-0124-FEDER-022674 and PTDC/EEA-CRO/122812/2010.

References

[1] S. R. Buss and J. P. Fillmore, *Spherical Averages and Applications to Spherical Splines and Interpolation*, ACM Transactions on Graphics, **2**, no. 20, (2001), 95–126.

[2] P. Crouch, G. Kun and F. Silva Leite, *The De Casteljau Algorithm on Lie Groups and Spheres*, J. of Dyn. Control Sys., **5**, no. 3, (1999) 397–429.

[3] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1990.

[4] J. Jost *Riemannian geometry and geometric analysis* (6 Ed.), Universitext, Springer, 2011.

[5] P. E. Jupp and J. T. Kent, *Fitting Smooth Paths to Spherical Data*, Appl. Statist., **36**, no. 1, (1987), 34–46.

[6] P. Lancaster and K. Salkauskas, *Curve and Surface Fitting*, Academic Press, 1990.

[7] L. Machado, *Least squares Problem on Riemannian manifolds*, PhD Thesis, University of Coimbra, 2006.

[8] L. Machado and F. Silva Leite, *Fitting Smooth Paths on Riemannian Manifolds*, Int. J. Appl. Math. Stat., **4**, no. J06, (2006), 25–53.

[9] L. Machado, F. Silva Leite and K. Krakowski, *Higher-order smoothing splines versus least squares problems on Riemannian manifolds*, J. of Dyn. Control Sys., **16**, no. 1, (2010), 121–148.

[10] J. Nocedal and S. J. Wright, *Numerical Optimization, Second Edition*, Springer Series in Operations Research, Springer Verlag, 2006.

[11] M. J. D. Powell, *Approximation Theory and Methods*, Cambridge University Press, 1981.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2013 24–27 June, 2013.

Stochastic amplification and childhood diseases in large geographical areas

Ramona Marguta[1] **and Andrea Parisi**[1]

[1] *Centro de Fsica da Materia Condensada, Universidade de Lisboa, Av. Prof Gama Pinto 2, 1649-003 Lisboa (Portugal)*

emails: margutaramona@hotmail.com, parisia@ptmat.fc.ul.pt

Abstract

We study the spread of childhood infectious diseases in geographically detailed population focussing on stochastic amplification. We use an individual based SIR model with demography where individuals reside in small geographical areas representing a portion of land, and long-distance transmission among different geographical areas occurs due to mobility of individuals. Mobility is implemented using the recently introduced radiation model [Simini et al., Nature 484, 96 (2012)]. Parameterizing the model for measles, we observe that some features of the data available for this disease can be understood within the framework of stochastic amplification, but also that the interplay between mobility and disease dynamics influences the resulting time-series.

Key words: stochastic amplification, human mobility, measles, SIR

1 Introduction

Stochastic fluctuations around the equilibrium for epidemiological models have been extensively studied in recent years [1, 2]. Substantial research has shown that the time series observed for various childhood diseases can be understood in terms of stochastic amplification, that is the increase in amplitude and regularization of the fluctuations around the equilibrium value observed for finite populations in epidemiological models driven by stochasticity. Such studies have shown that the frequency and amplitude observed for recurrent epidemics in available datasets for various infectious diseases could be understood within this framework [2, 3]. The work we present here investigates this idea in a more realistic setup by using realistic geographically detailed populations, where individuals move

around different geographical areas following a mobility model recently introduced that was shown to reproduce observed human mobility patterns [4]. We show that some of the features observed in real datasets can be understood in terms of stochastic amplification; however we also show that further fine tuning is needed to properly reproduce the observed data.

2 Methods

We perform individual based simulation of a SIR model with demography, in which we assume equal death and birth rates, so that the population size is kept constant. Individual based simulations for this kind of compartmental models show fluctuations of the number of infective individuals around the endemic equilibrium [2]; these fluctuations have a preferred frequency which can be easily determined calculating their power spectrum. Spatial and temporal correlations enhance the coherence and amplitude of such fluctuations [5, 6]. This phenomenon, known as stochastic amplification, has been suggested as an explanation for the incidence patterns observed for some childhood diseases.

Our aim is to study this phenomenon using the detailed geographic distribution of human population provided by the Gridded Population of the World database [7], which provides estimates for the population of a given geographical area on a regular grid of cells of angular size 2.5 arc-minutes. This corresponds roughly to 5 km at the equator. Each of these cells is considered by us as a well mixed population, and disease evolution in described by a simple individual based SIR model. Since the geographical areas we consider include population sizes of the order of tens of millions of individuals, we use parallel computation which grants unlimited complexity. A simulated annealing technique is used to partition the map under consideration into regions that are similar in population size and compact in shape. Each of these regions is then assigned a different node of a computer cluster.

Interactions among cells is introduced by moving individuals among the cells: when an individual leaves a cell, it ceases to participate to the dynamics of that cell, while it starts to participate to the dynamics of the new cell it moves to. Thus, by moving around, individuals can spread the disease from their cell of origin to new cells, leading to long-distance transmission. The way individuals move among different cells reflects the recently introduced radiation model for human mobility described in ref. [4]. This model, which gives the mobility fluxes among a set of locations, is influenced by a single parameter N_c/N which corresponds to the fraction of individuals in the population that participates to such long-distance displacements (N is the total population, N_c is the total number of individuals moving, or *commuters*). This parameter controls the level of long-distance transmission and hence can be estimated by data on human mobility. Following previous studies [8], individuals are assigned a set of preferred locations that they visit with a frequency compatible with the fluxes provided by the radiation model, allowing us to limit

 ISBN: 978-84-616-2723-3

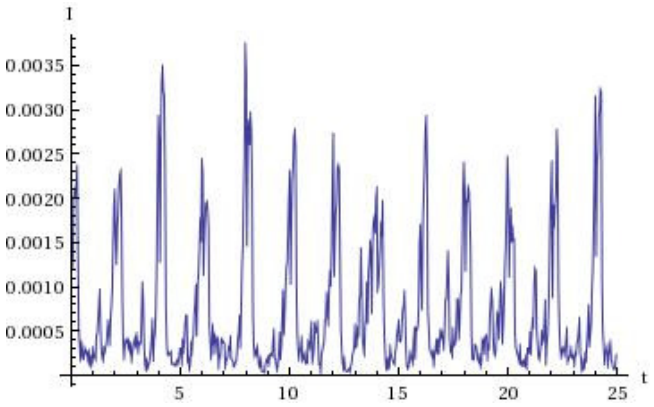


Figure 1: Infective incidence as a function of time for a grid cell in the city of London; time is in years. Simulations parameters: $\beta_0 = 1.175$ days^{-1}, $\beta_1 = 0.35$, $\gamma = 1/13$ days^{-1}, $\mu = 5.5 \times 10^{-5}$ days^{-1}, $N_c/N = 0.1$

the computational requirements of our simulations.

3 Results

We considered parameters corresponding to measles, a disease that received considerable attention [9] and was shown to be compatible with the stochastic amplification mechanism [2]. Since there are available datasets for England, we performed simulations for the geographical area of the British Isles that we assumed to be isolated: no imports from outer regions were considered. We introduced seasonality by modulating the contact rate through term-time forcing:

$$\beta(t) = \beta_0 \left[1 + \beta_1 \ \mathrm{Term}(t)\right]$$

where $\mathrm{Term}(t)$ is a function taking values $+1$ during school terms and -1 otherwise [9].

From the time series of the number of infective individuals for different locations in the area, we calculated the corresponding power spectra which we found to be in agreement with those calculated in previous studies [2]. The time series also show regular biennial cycles (see figure 1) for some location as well as propagation of waves from big centres towards less populated areas. We found however that the regularity of the sequence of peaks observed in the time series for some locations depends on the intensity of human mobility, controlled by the parameter N_c/N. Low populated cities are characterized by extinctions of the disease that must be imported by another location, however if N_c/N is low, such import events might be considerably delayed, leading to a separation among peaks larger than biennial

 ISBN: 978-84-616-2723-3

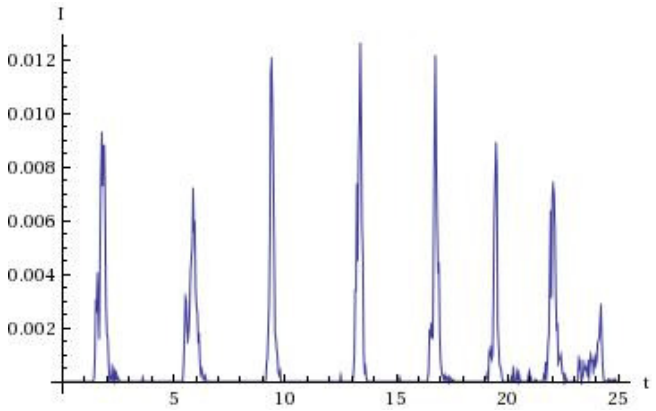


Figure 2: Infective incidence as a function of time for a grid cell in the city of York; time is in years. Due to the low N_c/N, peaks appear temporally more separated than the biennial sequence observed for London. Simulations parameters: same as in figure 1, except $N_c/N = 0.02$

(see figure 2). When N_c/N is high, long-distance transmission becomes efficient, but on the other hand this leads to higher infectiveness and broader peaks.

Although some of the features observed can also be interpreted in terms of broadness of the power spectra, we think that additional features should be considered in order to get a better agreement. An improved description for measles is that obtained using an SEIR model: our first attempts show that this change influences the time series but does not solve the discrepancies observed with data. Instead, we believe that more realistic recovery profiles using multiple infective classes would lead to a higher regularization of the epidemic sequence while reducing inter-epidemic infectivity. Also, contact restriction on individuals a few days after being infectives would also act to reduce the overall infectiveness observed in simulations.

Acknowledgements

This work was funded by the Fundação para a Ciência e a Tecnologia (FCT) within the framework of project PTDC/SAU-EPI/112179/2009.

References

[1] A.J. McKane, T.J. Newman, Phys. Rev. Lett. 94, 218102 (2005)

[2] D. Alonso, A.J. McKane and M. Pascual, J. R. Soc. Interface 4, 575-582 (2007)

[3] G. Rozhnova, A. Nunes, J. R. Soc. Interface 9 (2012)

[4] Simini et al., Nature 484, 96 (2012).

[5] Simoes et al., J. R. Soc. Interface 5, 555-566 (2008)

[6] A.J. Black, A.J. McKane, A. Nunes and A. Parisi, Phys. Rev. E 80, 021922 (2009)

[7] Center for International Earth Science Information Network (CIESIN)/Columbia University, United Nations Food and Agriculture Programme (FAO), and Centro Internacional de Agricultura Tropical (CIAT). 2005. Gridded Population of the World, Version 3 (GPWv3): Population Count Grid. Palisades, NY: NASA Socioeconomic Data and Applications Center (SEDAC).

[8] Marta C. Gonzalez, Cesar A. Hidalgo and Albert L. Barabasi, Nature 453, 779 (2008).

[9] Matt J. Keeling, Pejman Rohani, Bryan T. Grenfell, Physica D, 148, 317-335, (2001).

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

A model to generate process logs with equi-probable runs

Pasquale Marinaro[1], Irene Diaz[2] and Luigi Troiano[3]

[1] *Intelligentia srl, Italy. Department of Computer Science, University of Oviedo, Spain*

[2] *Department of Computer Science, University of Oviedo, Spain*

[3] *Department of Engineering, University of Sannio, Italy*

emails: pasquale.marinaro@intelligentia.it, sirene@uniovi.es, troiano@unisannio.it

Abstract

The evaluation of process models require the availability of execution logs, which hardly are available. Therefore, some techniques have been proposed to generate logs from a process model. The problem of generating a complete event log is NP-hard, therefore for complex process model looking for a complete event log might be infeasible. In this paper we propose an approach,based on an order preserving heuristic, for sampling the log space which produces equiprobable execution runs.

Key words: process mining, business process, log generation

1 Introduction

Modern enterprises rely on well-defined business processes in order to pursue their mission. A process can be defined as a set of actions or activities that happen over the time, but which are related to each other by a common goal. It is possible to identify processes in companies, hospitals, government institutions, universities, and so on. Process Mining is a knowledge discovery approach which represents an emerging area of interest able to cover problems such as the analysis or design of processes, their evolution over the time, the adaptation of the original process definition to their practice and experience. [4,5]

Data Mining and Knowledge Discovery approaches have been deeply developed with respect to tabular data, spatial information, text and other specific sources. However, when applied to business processes it unveils new and specific issues to be faced. Event logs provide the starting point for process mining techniques, [2,3] as they collect most of relevant

 ISBN: 978-84-616-2723-3

information to understand how a process is instanced, operated and evolved. They describe a flow of activities along a graph of functional dependencies between tasks, even when this structure is implicit in practice. Basically, a process log is made or runs according to a given abstract process model where each run represents a specific process instance. A great amount of work in process mining tries to discover from a log the process model that best describes the set of process instances. To do that, process instances can be represented as a directed graph, a finite state machine or a Petri net.[4,5]

As logs represent all the information related to a given process, their analysis and generation represent an important issue because Process Mining approaches normally rely on the assumption that the log to be mined is complete. However, the problem of generating all possible runs given a model is NP-hard. This might lead to make infeasible the production of a complete event log. An alternative way is to sample the log space by picking a given number of execution runs. In order to avoid any statistical bias, log sampling should rely on a pseudo random generation of runs which would approximate uniform distribution. i.e. each run should be equiprobably generated.

This work proposes a heuristic method to sample a process log in a uniform way, which produces runs considering the correct interleaving of activities according to the given model. This heuristic is inspired to a general method to obtain linear extensions from a partial order.[1] Log generation is complicated by the different gateways which control the process workflow. In particular, we cope with parallel and exclusive gateways, as these are the most common in practice. In addition, we did not take into account cycles, as they generally reflects anomalous executions which require to redo some activities.

In the following section we provide a detailed description of the method as well as some insights about its application.

2 Algorithm

In this section we present an algorithm based on heuristics able to generate runs quasi-uniformly. Generating equiprobable runs avoid to bias the log sampling when more extended and complex process model are considered. Indeed, in this case it is not feasible to determine in advance which and how many different runs are possible. Algorithm for generating process runs is described by pseudo code in Algorithm 1.

The algorithm takes as input A, the list of activities entailed by the process; G, the adjacency matrix of activities in A, so that $G(a_j, a_k) = 1$ if a_k follows immediately after a_j; s, the activity starting the process; E, the set of activities ending the process.

As first step, we compute R the transitive closure of the matrix G, and we assign s as starting activity; R_0 is the closure matrix without considering a_0; variable i provides the current run step (lines 1-4). For each step, we collect in P activities that has no predecessor in R_i: they are the candidate to be selected (line 5). If there is only one activity in P, we

 ISBN: 978-84-616-2723-3

Algorithm 1 RunGenerator

Input: A, activities
Input: G, adjacency matrix
Input: s, stating activity
Input: E, ending activities
Output: $(a_0, a_1, \ldots, a_m)$, the run

1: $R \leftarrow closure(G)$
2: $a_0 \leftarrow s$
3: $R_0 \leftarrow R$ cleared of a_0
4: $i \leftarrow 0$
5: **repeat**
6: $P \leftarrow \{a \in A \mid a$ has no predecessor in $R_i\}$
7: **if** $|P| = 1$ **then**
8: $a_{i+1} \leftarrow a \in P$
9: **else**
10: **if** $context(a_i) = MAIN, PAR$ **then**
11: $N_i \leftarrow \{(u, v) \in A_i^2 | R_i(u, v) = 0\}$
12: **for all** $a \in P$ **do**
13: $Q(a) \leftarrow \#\ (a, \cdot) \in N$
14: **end for**
15: $a_{i+1} \leftarrow roulette(P, Q)$
16: $R_{i+1} \leftarrow R_i$ cleared of a_{i+1}
17: **else**
18: $a_{i+1} \leftarrow roulette(P)$
19: $R_{i+1} \leftarrow R_i$ cleared of a_{i+1} and unreachable nodes in R_i
20: **end if**
21: **end if**
22: $i \leftarrow i + 1$
23: **until** $a_i \in E$

 ISBN: 978-84-616-2723-3

select it as the next in the run (lines 7-8); otherwise we have to choose. Choice depends on the context of a_i, that is determined by the last gateway entered. Context values can be $MAIN$, PAR and XOR. The different context will affect how the next activity will be selected. In the case of PAR and $MAIN$ (line 9), we assign to activities a chance of being that inversely proportional to the number of possible paths that can be originated. To estimate this number we collect in N_i all pairs activities that are not connected in R_i. For each activity a in P we compute the number of times it appears as first element in a pair $(a, \cdot)$ considered by N_i and assigned to Q (line 13). We select an activity from P with a probability that inversely proportional to Q (line 15). We reduce the closure matrix R_{i+1} by removing a_{i+1} from R_i and we move to the next step (line 16). Instead, if the context is XOR, we just make a random selection in P (line 18), and we prepare R_{i+1} by removing a_{i+1} and unreachable nodes from R_i (line 19). As output we get the run made of activities $a_0, a_1, \ldots, a_m$.

The crucial point here is related to the selection of each activity at step i because it is necessary to distinguish between parallel or exclusive activities. In a flow, if control originates a double path which leads to two activities a_k and a_j (parallel gateway), we can obtain runs in which both activities are interleaved. By contrast, if a decision is made in a_i which leads to a_k or alternatively to a_j (exclusive gateway), we will get runs in which only one of the two activities will appear. These two possibilities have to be taken into account in generating a path.

If the activities belong to an exclusive gateway, the same probability of being selected is assigned to each activity. Instead, if the activities belong to parallel gateway, it's not fair to assign to each activity the same probability to be selected, because the final execution flows will have different probabilities of being generated.

3 An example

Let's check the algorithm with an example that contains both parallel and exclusive gateways. Only activities belonging to the same gateway will be referred. Figure 1 shows a process with 8 activities. Activities $a2$ and $a3$ are exclusive. That is, a certain log instance contains either $a2$ or $a3$, but not both. All possible execution flows are showed in Figure 2.

The run tree showed in Figure 3 is explored for the determination of the execution flow assigning to each activity the same probability of being selected. As it can be seen, the activity starting the process is $a1$ and $a8$ the ending one. Therefore each log instance starts with activity $a1$. The activities following next the current activity are $a2$ and $a3$. As they are exclusive, the probability of being selected is $1/2$. If $a2$ is selected just after $a1$, the first log instance of Figure 2 is obtained.

On the other side, if activity $a2$ is selected after $a1$, there is more than one possibility for the third activity. The candidates to be placed in third position are $a4$ and $a5$. As

 ISBN: 978-84-616-2723-3

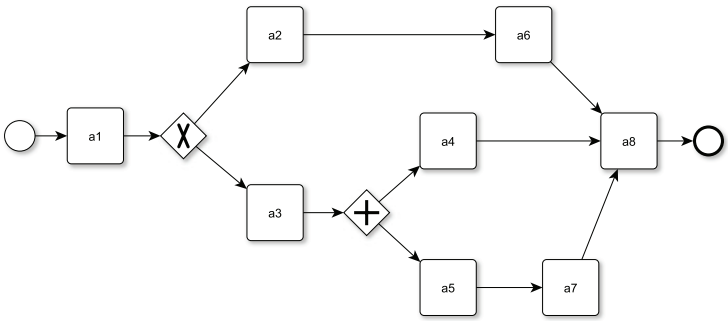


Figure 1: Example of a process model by BPMN notation.

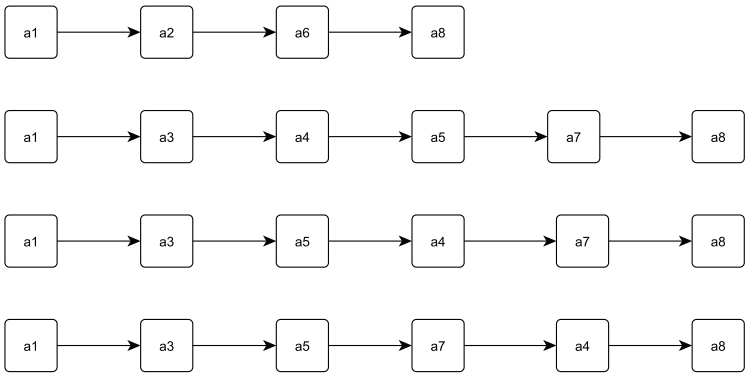


Figure 2: Log instances

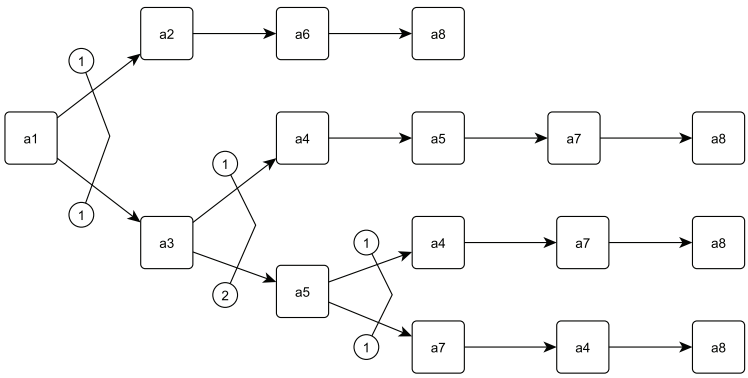


Figure 3: The run tree (generator). Number in arcs represent the relative weight for any node

 ISBN: 978-84-616-2723-3

these activities can be executed in parallel, then the probability of selecting one of them is inversely proportional to the number of possible paths that can be originated. If $a4$ is executed at third step, it will affect one log instance (the second in Figure 2), If $a5$ is executed at third step, it will affect two log instances (the third and fourth in Figure 2). Therefore, the probability of selecting $a4$ ($a5$) at third step is $1/3$ ($2/3$). For the logs starting by $a1, a3, a5$ we repeat the proces for selecting the fourth activity. We then continue the process until some ending activity is reached.

4 Conclusion

The algorithm described in previous section is able to generate different execution runs according to a given process model. Runs are generated with the nearly equal probability. This allows a uniform sampling of the log space when it is not feasible to generate a complete log. The run generation depends on the nature of the different gateways met along the path. In this paper we considered two main gateways: parallel and exclusive. Other gateways are possible. In the future we aim to take into account them along the run generation. Also, we did not consider cycles, as they are generally related to anomalous executions, while we were interested to outline the nominal behaviour.

Having a uniform log sampling might help to answer questions concerning log completeness, as non trivial criteria for testing it are not known. There are some works focused on studying the completeness of logs[6] but all of them make assumptions hardly to verify in practice. Other works take a probabilistic approach in order to guarantee, with some probability, that the event log is complete and the discovered process is similar to the generating process. We aim to investigate relationship between uniform log sampling and log completeness more deeply in the future.

Acknowledgements

Author 2 acknowledges financial support Grant TEC2012-38142-C04-04 from Ministry of Education and Science, Government of Spain.

References

[1] E.F. Combarro, I. Díaz, and P. Miranda. On random generation of fuzzy measures. *Fuzzy Sets and Systems*, (0):–, 2012.

[2] Diogo R. Ferreira and Daniel Gillblad. Discovering process models from unlabelled event logs. In *Proceedings of the 7th International Conference on Business Process Management*, BPM '09, pages 143–158, Berlin, Heidelberg, 2009. Springer-Verlag.

[3] Laura Maruster, A. J. M. M. Weijters, Wil M. P. van der Aalst, and Antal van den Bosch. Process mining: Discovering direct successors in process logs. In *Proceedings of the 5th International Conference on Discovery Science*, DS '02, pages 364–373, London, UK, UK, 2002. Springer-Verlag.

[4] A. Tiwari, C. J. Turner, and B. Majeed. A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal*, 14(1):5–22, 2008.

[5] W. M. P. van der Aalst and A. J. M. M. Weijters. Process mining: a research agenda. *Comput. Ind.*, 53(3):231–244, April 2004.

[6] Kees M. van Hee, Zheng Liu, and Natalia Sidorova. Is my event log complete? - a probabilistic approach to process mining. In *RCIS*, pages 1–7, 2011.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Modelling the effect of *MDR-TB* on Tuberculosis epidemic

Silvia Martorano Raimundo[1], Hyun Mo Yang[1] and Ezio Venturino[2]

[1] *Faculdade de Medicina da Universidade de São Paulo*
LIM01, HCFMUSP
Rua Teodoro Sampaio, 115
CEP: 05405-000 - São Paulo, SP, Brazil
, Departamento de Matemática Apliacada, IMECC
Universidade de Campinas
Campinas, SP, Brasil

[2] *Dipartimento di Matematica "Giuseppe Peano",*
Via Carlo Alberto 10,
10123 Torino, Italy, Università di Torino

emails: silviamr@dim.fm.usp.br, hyunynag@ime.unicamp.br,
ezio.venturino@unito.it

Abstract

Multidrug-resistant tuberculosis (*MDR-TB*) is a specific and particularly a dangerous form of drug-resistant *TB*, which is defined as the form of disease caused by resistance of a strain of *Mycobacterium tuberculosis (MTB)* to two or more of the antituberculosis drugs. *Key words: Mycobacterium tuberculosis - TB - MDR-TB - Basic reproductive number - treatment - drug-sensitive strains - drug-resistant strains*

1 Main results.

MDR-TB is generally treatable, however, the efficacy of treatment of drug-resistant cases is reduced compared with that of drug-sensitive cases. Studies have found that drug resistance develops because of inadequate or erratic therapy, although it has been shown that persons previously treated for drug-sensitive tuberculosis can be reinfected with drug-resistant strains.

 ISBN: 978-84-616-2723-3

A number of theoretical studies have been performed on the mathematical modelling of coexistence of different pathogens (strains) in the same host [1], [3], [9], [11], [12], [13]. One of the first mathematical model that included the dynamics of both drug-sensitive and drug-resistant tuberculosis was published by Blower et.al. [2]. More recently others have modeled the emergence of drug-resistant tuberculosis and predict the future burden of *MDR-TB* [4], [5], [8], [10], [14].

This work is concerned with a mathematical model to evaluate the effect of *MDR-TB* on *TB* epidemic and its control by assessing the transmission dynamics of both drug-susceptible and drug-resistant *TB*.

Our mathematical model monitors the temporal dynamics of susceptible individuals (not infected but susceptible to infection), latent individuals (infected but unable to infect others) and the active-TB infections, given by the infectious individuals (i.e., infected individuals that are able to infect others). Since the model assesses the drug-resistant and drug-susceptible tuberculosis transmission, two subclasses of latent and infectious *TB* individuals are required to build it. Hence, the total population (N), is divided into five classes, namely, S, the susceptible individuals; L_S, the drug-sensitive latent individuals; L_R, the drug-resistant latent individuals; TB_S, the drug-sensitive with active-*TB* individuals, and TB_R, the drug-resistant with active-*TB* individuals.

We assume that *MTB* infection is transmitted by infectious individuals with active-*TB* (TB_S and TB_R), and the infection propagates following the pseudo mass-action incidence [6], [11]. The susceptible individuals (S) can be infected with either a drug-sensitive strain or a drug-resistant strain. The rate of newly recruited drug-sensitive and drug-resistant cases are $\beta_S TB_S S$ and $\beta_R TB_R S$, respectively. The transmission coefficients, β_S and β_R specify the transmissibility of drug-sensitive *TB*, and the transmissibility of drug-resistant *TB* individuals, respectively. The transmission of drug-resistant *TB* occurs via two independent but interacting processes: (i) transmission of drug-resistant to susceptible individuals (transmitted resistance) and (ii) conversion of sensitive cases to drug-resistant cases during the treatment (acquired resistance).

The dynamics of epidemic models can be understood in terms of the basic reproductive number of infection, R_0, which is the average number of secondary cases caused by one infectious case in a completely susceptible population. It is well-known that the condition $R_0 < 1$ is necessary for disease eradication [7]. Here, the *relative reproductive fitness* function will be approximated by the basic reproductive number of infection (R_0) in the absence of treatment or the effective reproductive number (R) in the presence of treatment.

We identify the steady states of the model to analyse their stability. We find that the basic reproductive number is composed of two critical values, relative reproductive number for sensitive (strains sensitive to all drugs) and *MDR* strains (strains resistant to all drugs).

Paradoxically, we have found that even *MDR* strains that are considerably less fit (and thus less transmissible) than the drug-sensitive strains can lead to a high *MDR* incidence.

 ISBN: 978-84-616-2723-3

Drug-resistant pathogens gain an advantage over drug-sensitive pathogens because treatment is less effective against drug-resistant strains.

Acknowledgements

This work was supported by grants from FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), Brazil, Fondazione CRT/Progetto Lagrange and INdAM (Istituto Nazionale di Alta Matematica), Italy.

S.M.Raimundo gratefully acknowledges, with thanks, the support of FAPESP; Fondazione CRT/Progetto Lagrange and INdAM.

References

[1] Baggaley, R.F., Garnet, G.P, Ferguson, N.M.. Modelling the Impact of Antiretroviral Use in Resource-Poor Settings. PLoS Med (2006) 3(4):e124

[2] Blower, S.M., Small, P.M., Hopewell, P. Control strategies for tuberculosis epidemics: new models for old problems. Science (1996) 273:497-500.

[3] Bowong S., Kurths, J. Modeling and analysis of the transmission dynamics of tuberculosis without and with seasonality. Nonlinear Dynamics (2012) 67:2027-2051.

[4] Colijin, C. Cohen, T., Ganesh, A., Murray, Spontaneous emergence of multiple drug resistance in Tuberculosis before and during therapy. PLos ONE (2011) 6(3), e18327.

[5] Feng, Z., Ianelli, M., Milner, F.A. A two-strain Tuberculosis model with age of infection. (2002) SIAM J. Appl. Math. 62(5),1634-1656.

[6] Jong M. C. M., Diekmann O., Heesterbbeek J. A. P. , 1994. How does transmission of infection depend on population size? in: D. Mollison (Ed.), Epidemic Models: Their Structure and Relation to Data, Cambridge University, Cambridge, vol. 5, p. 84.

[7] Hethcote, W. H. The mathematcs of infectious diseases, SIAM Review (42)4 (2000), 599-653.

[8] Liu, Y., Sun, Z., Sun, G., Zhong, Q., Jinag, L., Zhou, L., Qiao, Y., Jia, Z. Modeling transmission of Tuberculosis with MDR and undetected cases (2011). Discrete Dynamics in Nature and Society doi:10.1155/2011/296905

[9] Moghadas, S.M., Bowman, C.S., Rost, G., Wu, J. Population-wide emergence of antiviral resistance during pandemic influenza. PLos ONE, (2008) 3(3),e1839.

 ISBN: 978-84-616-2723-3

[10] Okuonghae, D., Omosigho, S.E. Analysis of a mathematical model for tuberculosis: What could be done to increase case detection. J. Theoretical Biology (2011) 269:31-45.

[11] Raimundo, S.M., Yang, H.M., Venturino, E., Massad, E. Modeling the emergence of HIV-1 drug-resistance resulting from antiretroviral therapy: Insights from theoretical and numerical studies. BioSystems (2012) 10:1-13.

[12] Raimundo, S.M., Yang, H.M., Bassanezi, R.C., Ferreira, M;A.C. The attracting basins and the assessment of the transmission coefficients for HIV and M. Tuberculosis infections among women inmates. J. Biol. Syst. (2002) 10,61-83 .

[13] Raimundo, S.M., Engel, A.B., Yang, H.M., Bassanezi, R.C. An approach to estimating the transmission coefficients for AIDS and for tuberculosis using mathematical models. Systm. Anal. Model. Simul. (2003) 43,423-442 .

[14] Rodrigues, P., Gomes, MGM, Rebelo, C. (2007) Drug resistance in tuberculosis - a reinfection model. Theor. Popul. Biol. 71(2):196-212.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Modelling pack hunting and prey herd behavior

Diego Melchionda[1], Elisa Pastacaldi[1], Chiara Perri[1] and Ezio Venturino[1]

[1] *Dipartimento di Matematica "Giuseppe Peano", Università di Torino,*
via Carlo Alberto 10, 10123 Torino, Italy

emails: melchionda.diego@gmail.com, elisa.pastacaldi@libero.it,
chiarapperri@gmail.com, email: ezio.venturino@unito.it

Abstract

We consider here coordinated behavior of predators, in correspondence of prey individual and group gathering. Two models for analysing these situations are introduced. In both cases the system can thrive at coexistence, but in case of prey herd defense, quite unexpectedly, we observe that the system may very well collapse, if environmental conditions are suitable. In addition, in this very same case limit cycles are discovered, when a certain bifurcation parameter crosses a critical value. Finally, the population values at coexistence for these two models are compared among themselves as well as against those of the classical Lotka-Volterra model with prey logistic correction, and with a previously proposed prey herd behavior system.

Key words: predator-prey, group gathering, stability, ecosystems
MSC 2000: AMS codes 92D25, 92D40

1 Introduction

In this paper we consider two predator-prey models. Common to both, and contrary to classical models in which predators search for prey on an individual basis, see for instance the first chapters in [7], is the assumption that they hunt the prey in a coordinate fashion. In other words, the pack hunts the prey, but the strongest individuals in general have the upper hand, in the sense that either they occupy the forefront positions of the pack in order to get the best benefits from the action, or simply they obtain them afterwards. We assume these positions to be on the outskirts of the pack. In this situation the individuals in these places will be the first to fall upon the prey. As for the latter, we use the standard assumption of individual behavior, for which they are kind of isolated, and can be attacked by the

ISBN: 978-84-616-2723-3

predators' pack individually, but we also assume that they can gather together in herds, as recently introduced in the literature, [2]. In such situation therefore the consequences of the attacks are felt mostly by the prey on the perimeter of the herd, as they are more exposed to the predators' hunting.

In the literature the idea of group defense, first expressed along these lines in [1], had been explored long time ago, with rather different assumptions, [6]. More recently, it has been further investigated, [3] and explored also in the context of ecoepidemiology, [4].

Two models are then presented along these lines. Mathematically, the pack and the herd assumptions correspond to replacing the classical mass action law terms by a nonlinear function of the populations. Specifically, as discussed in [2], these terms involve the square root of the population densities, rather than the populations themselves. In other words, here we suitably combine Gompertz-like interaction terms, where the exponent is not generic, but fixed to the value $\frac{1}{2}$, as this value expresses the relationship between area and perimeter of the territory occupied by the herd or the pack.

2 Modelling pack predation

Here we consider at first the predators gathering together to hunt for food, while the prey have an individualistic behavior. Let $P(\tau)$ represent the density of the predators, namely the number of individuals per surface unit. As described in [2], if the pack occupies an area A, the individuals who stay at the outskirts of the pack are in number proportional to the perimeter of the patch on which the pack sits, so that its length is proportional to $\sqrt{A}$. This dependence is thus reflected by the density square root, i.e. $\sqrt{P}$. We assume that the interactions with the prey population occur mainly via these peripheral individuals, hence the interaction term must be proportional to $Q\sqrt{P}$, where Q denotes the prey population. The model is thus

$$\frac{dQ}{d\tau} = r\left(1 - \frac{Q}{K}\right)Q - q\sqrt{P}Q, \quad \frac{dP}{d\tau} = -mP + p\sqrt{P}Q. \tag{1}$$

The first equation describes the logistic dynamics of the prey population, while the second one expresses the corresponding evolution of the predators. The interactions of the predators with the prey occur essentially through the individuals lying at the outskirts of the herd. The parameter r is the growth rate of the prey, K is the environment's carrying capacity, m represents the predators natural death rate. The predation rate is represented by the parameter q, which affects negatively the prey, while p is the hunting profit, i.e. the benefit for the predator. Since not the whole prey is converted in food for the predators we need to impose that $p < q$. All the parameters are nonnegative.

 ISBN: 978-84-616-2723-3

2.1 Model simplification

The model (1) has a singularity in the Jacobian, due to the presence of the square root term. It is therefore advisable to remove it, before proceeding to the analysis. We rescale the variables as follows

$$X = \frac{Q}{K}, \qquad Y = \frac{q\sqrt{P}}{m}, \qquad t = m\tau,$$

and define the new parameters

$$b = \frac{r}{m}, \qquad c = \frac{pq}{2Km^2}.$$

The adimensionalized system for $Y \neq 0$ can be written as

$$\frac{dX}{dt} = b\,(1-X)\,X - XY, \quad \frac{dY}{dt} = -\frac{1}{2}Y + cX, \tag{2}$$

while in absence of predators, the system reduces just to the first equation. In this case, easily, the prey follow a logistic growth, toward the adimensionalized carrying capacity $X_1 = 1$. We have defined the new dependent variable Y by taking P's square root, thus Y is nonnegative. Similarly, but straightforwardly from its definition, we have also that X must be nonnegative. Also the new parameters b and c are combinations of the old parameters r, m, p, q, K which are nonnegative. As a consequence, they must be nonnegative as well.

2.2 Boundedness

Let us introduce the total population in the environment, $Z(t) = X(t) + Y(t)$. Summing the equations in (2), we find

$$\frac{dZ}{dt} = -\frac{1}{2}Y + cX + bX - bX^2 - XY = -\frac{1}{2}Z + \left(c + b + \frac{1}{2} - bX - Y\right)X.$$

Thus the following estimates follow on taking the maximum of the parabola in X

$$\frac{dZ}{dt} + \frac{1}{2}Z \leq \left(c + b + \frac{1}{2} - bX\right)X \leq \frac{\left(c + b + \frac{1}{2}\right)^2}{4b} \equiv \bar{M}.$$

From the theory of differential inequalities we obtain

$$Z(t) \leq e^{-\frac{1}{2}t} + 2\bar{M}\left(1 - e^{-\frac{1}{2}t}\right) \leq 1 + 2\bar{M} = M$$

and in view of the fact that the total population is bounded, each subpopulation X and Y must be bounded as well.

 ISBN: 978-84-616-2723-3

2.3 Equilibria

For the system (2) there are only two equilibria $E_i = (X_i, Y_i)$, namely the origin $E_0 = (0, 0)$, at which point both populations become extinct, and the coexistence equilibrium

$$E_2 = \left(\frac{b}{b+2c}, \frac{2bc}{b+2c} \right),$$

which is always feasible.

As mentioned earlier, there is also the equilibrium point $E_1 = (1, 0)$ for the subsystem without predators. But this does not satisfy the second equation in (2) and furthermore it is unstable. In fact, since at any point (X^*, ϵ), with $\epsilon > 0$ but arbitrarily small and X^* close to X_1 we have $\frac{dY}{dt} = cX^* + o(\epsilon) > 0$, it follows that solution trajectories of (2) move away from E_1.

2.4 Stability

The Jacobian of (2) is

$$J \equiv \begin{pmatrix} b - 2bX - Y & -X \\ c & -\frac{1}{2} \end{pmatrix}.$$

At the origin E_0, the eigenvalues are $\lambda_1 = b > 0$ and $\lambda_2 = -\frac{1}{2}$, so that it is unstable.

Let us denote by J_2 the Jacobian matrix evaluated at E_2. Applying the Routh-Hurwitz criterion to the characteristic equation, we find

$$\det(J_2) = -\frac{1}{2}b + \frac{b^2 + 2bc}{b+2c} = \frac{1}{2}b > 0, \quad \operatorname{tr}(J_2) = -\frac{1}{2} + b - \frac{2b^2 + 2bc}{b+2c} = -\frac{2b^2 + 2c + b}{2(b+2c)} < 0.$$

Thus both conditions are always satisfied. This means that J_2 has eigenvalues with negative real part and therefore E_2 is always a stable equilibrium. The phase plane picture also sustains this conclusion as we can see in Figure 1. Summarizing, we found the following result.

Theorem 1. The equilibria E_0 and E_1 are always unstable equilibria. The coexistence equilibrium E_2 is always locally asymptotically stable.

Furthermore, taking the point $N^* \equiv (N, 2cN)$ on the isocline through the origin in Figure 1, with $N > 1$, we can identify a compact set Ω, the rectangle having N^* and the origin as opposite vertices, which is positively invariant. On its right vertical side indeed

$$\frac{dX}{dt}|_{X=N} = b(1-N)N - 2cN^2 < 0$$

 ISBN: 978-84-616-2723-3

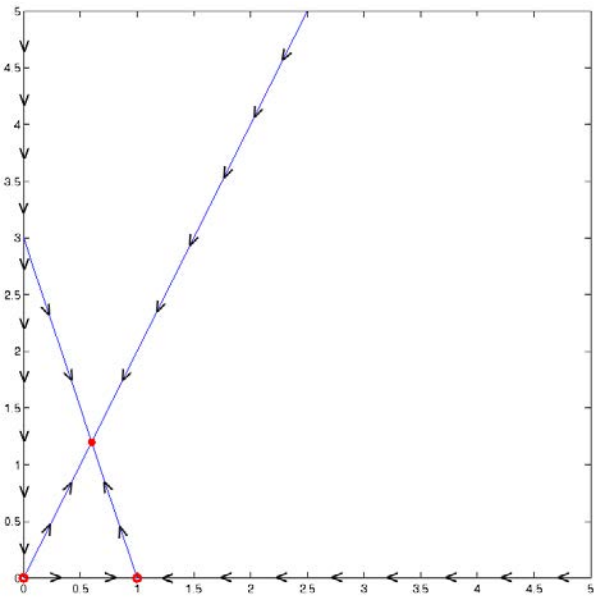


Figure 1: Phase plane sketch of the model (2) with parameters value $r = 0.9$, $m = 0.3$, $p = 0.6$, $q = 1.5$ and $K = 5$.

so that trajectories enter into Ω from the right. Similarly on the upper horizontal side, for $X < N$, we have

$$\frac{dY}{dt}|_{Y=2cN} = -\frac{1}{2}2cN + cX < 0$$

and again trajectories of (2) enter into Ω from above. Applying the Poincaré-Bendixson theorem, global stability follows. Hence

Theorem 2. The coexistence equilibrium E_2 is also globally asymptotically stable.

No Hopf bifurcations can arise at this point, since $\mathrm{tr}(J_2) < 0$ strictly.

3 Pack hunting with herd response

In this second model we still assume the predator population to hunt in packs, but in addition the prey population here exhibits an herd behaviour. Using the same notations, the interactions between the two populations occur via peripheral individuals, those who take the outermost position in both herds. Thus the interactions between predators and prey are proportional to the product of the square roots of their densities, $\sqrt{P}\sqrt{Q}$. The

 ISBN: 978-84-616-2723-3

model is therefore given by

$$\frac{dQ}{d\tau} = r\left(1 - \frac{Q}{K}\right)Q - q\sqrt{P}\sqrt{Q}, \quad \frac{dP}{d\tau} = -mP + p\sqrt{P}\sqrt{Q}. \tag{3}$$

Again the first equation describes the evolution of the prey population, while the second one describes the same for the predators. The parameters retain their meaning as for (1).

3.1 Model simplification

We proceed in the same way as we did for (1). We rescale the variable as follows

$$X = \sqrt{\frac{Q}{K}}, \qquad Y = \frac{q}{2m}\sqrt{\frac{P}{K}}, \qquad t = m\tau,$$

and define the new parameters, which are nonnegative as a result of the nonnegativity of the original ones,

$$e = \frac{r}{2m}, \qquad f = \frac{pq}{4m^2}.$$

For $Y \neq 0$, the adimensionalized system takes the form

$$\frac{dX}{dt} = e(1 - X^2)X - Y, \quad \frac{dY}{dt} = -\frac{1}{2}Y + fX. \tag{4}$$

Again, for $Y = 0$, the prey evolve toward the equilibrium $\widehat{E}_1 \equiv (1, 0)$.

3.2 Boundedness

Once again we define the total population $Z(t) = X(t) + Y(t)$. Summing the equations in (4), we have

$$\frac{dZ}{dt} = eX - eX^3 - Y - \frac{1}{2}Y + fX = X\left(e + f + \frac{3}{2} - eX^2\right) - \frac{3}{2}Z.$$

Thus we obtain the following estimates

$$\frac{dZ}{dt} + \frac{3}{2}Z \leq \left(\frac{2}{3}e + \frac{2}{3}f + 1\right)\sqrt{\frac{2e + 2f + 3}{6e}} \equiv \bar{M}.$$

The last estimate follows on taking the maximum of the cubic in X.

Finally, from the theory of differential inequalities we have

$$Z(t) \leq e^{-\frac{3}{2}t} + \frac{2}{3}\bar{M}\left(1 - e^{-\frac{3}{2}t}\right) \leq 1 + \frac{2}{3}\bar{M} = M.$$

Each subpopulation X and Y is therefore bounded since their sum is.

 ISBN: 978-84-616-2723-3

3.3 Equilibria

The two equilibria $\widehat{E}_i = (\widehat{X}_i, \widehat{Y}_i)$ of (4) are once again the origin $\widehat{E}_0 = (0,0)$ and the coexistence equilibrium

$$\widehat{E}_2 = \left(\sqrt{\frac{e-2f}{e}}, 2f\sqrt{\frac{e-2f}{e}} \right).$$

The latter is feasible if

$$e \geq 2f. \tag{5}$$

In Figures 2 and 3 we compare the two situations in which $\widehat{E}_2$ is feasible and when it is unfeasible.

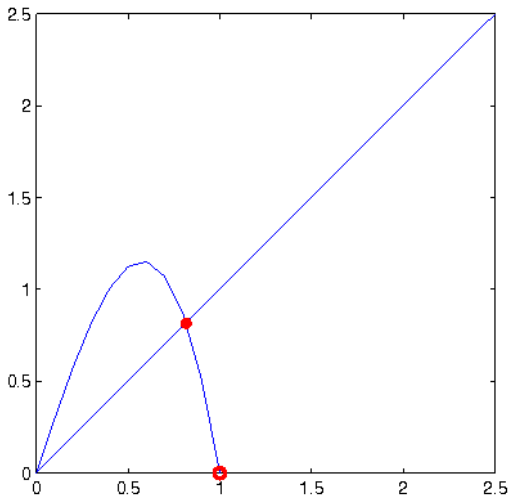


Figure 2: Phase plane of model (4) with $e \geq 2f$, both $\widehat{E}_0$ and $\widehat{E}_2$ exist.

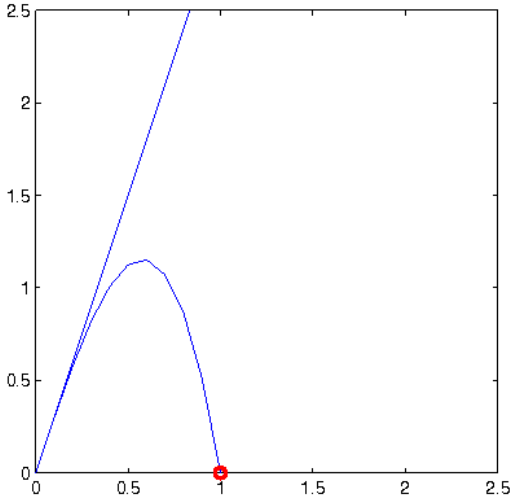


Figure 3: Phase plane of model (4) with $e < 2f$, $\widehat{E}_2$ is unfeasible.

As we can see in Figure 2 also the point $\widehat{E}_1 = (1,0)$ seems to be an equilibrium, but it does not satisfy (4). Since at any point $(\widehat{X}^*, \epsilon)$, with an arbitrarily small $\epsilon > 0$ and $\widehat{X}^*$ near $\widehat{X}_1$ we have $\frac{dY}{dt} = f\widehat{X}^* - o(\epsilon) > 0$, it follows that $\widehat{E}_1$ cannot be a stable equilibrium.

3.4 Stability

The Jacobian of (4) is

$$\widehat{J} \equiv \begin{pmatrix} e(1-3X^2) & -1 \\ f & -\frac{1}{2} \end{pmatrix}.$$

 ISBN: 978-84-616-2723-3

At the origin $\widehat{E}_0$, the characteristic polynomial is

$$\lambda^2 + \left(\frac{1}{2} - e\right)\lambda + f - \frac{1}{2}e = 0.$$

Applying the Routh-Hurwitz criterion we obtain the stability conditions for $\widehat{E}_0$ as

$$2f > e, \qquad e < \frac{1}{2}. \tag{6}$$

Thus in this situation the ecosystem may well disappear. Also comparing the first condition in (6) with (5) we have the following result.

Theorem 3. The origin of system (4) is locally asymptotically stable if (6) holds. There is a transcritical bifurcation for which $\widehat{E}_2$ emanates from the origin when the parameter e raises up to attain the critical value $e^* = 2f$.

Letting $\widehat{J}_2$ denote the Jacobian evaluated at $\widehat{E}_2$, the Routh-Hurwitz conditions are $\det(\widehat{J}_2) = e - 2f > 0$ and

$$\mathrm{tr}(\widehat{J}_2) = -2e + 6f - \frac{1}{2} < 0. \tag{7}$$

The first one is always true in view of the feasibility condition (5) of $\widehat{E}_2$. As for the second condition we have several cases, which are represented in Figure 4. In summary

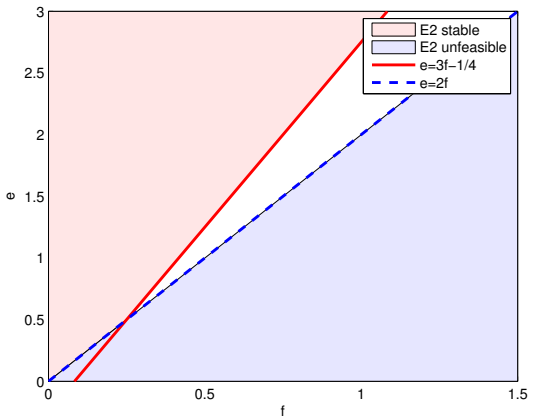


Figure 4: Region of the $e-f$ parameter space in which the coexistence equilibrium is stable.

Theorem 4. The coexistence equilibrium of system (4) is locally asymptotically stable in the following cases:

 ISBN: 978-84-616-2723-3

if $2f < e < \frac{1}{2}$, (7) holds and then E_2 is stable;

if $e > \max\left\{\frac{1}{2}, 3f - \frac{1}{4}\right\}$, (7) holds as well and thus E_2 is stable;

if $\frac{1}{2} < e < 3f - \frac{1}{4}$, (7) is not true, so that E_2 is unstable.

Theorem 5. The equilibria $\widehat{E}_0$ and $\widehat{E}_2$ are both globally asymptotically stable, whenever they are locally asymptotically stable.

Proof. We proceed as for the former model (2). Take in this case a point $\widehat{L}_* \equiv (L, 2fL)$ with

$$L > \max\left\{1, \frac{e}{3\sqrt{3}f}\right\}.$$

The rectangle $\widehat{\Omega}$ with the origin and $\widehat{L}_*$ as opposite vertices is a positively invariant set since on its right vertical and upper horizontal sides, for the latter recalling that here $X < L$, we have

$$\frac{dX}{dt}|_{X=L} = e(1-L^2)L - Y < 0, \quad \frac{dY}{dt}|_{Y=2fL} = -f(L-X) < 0.$$

Hence all trajectories enter into $\widehat{\Omega}$ and therefore the only locally asymptotically stable equilibrium in its interior, be it $\widehat{E}_0$ or $\widehat{E}_2$, which are mutually exclusive in view of the transcritical bifurcation of Theorem 3, must also be globally asymptotically stable.

We can sum up the situation of the equilibria of system (4) in the following table.

Parameters		$\widehat{E}_0$	$\widehat{E}_2$	Bifurcation
$e < \frac{1}{2}$	$f > \frac{e}{2}$	STABLE	UNFEASIBLE	
$e < \frac{1}{2}$	$e^* = ef$			Transcritical
$e < \frac{1}{2}$	$f < \frac{e}{2}$	UNSTABLE	STABLE	
$e > \frac{1}{2}$	$e > 3f - \frac{1}{4}$	UNSTABLE	STABLE	
$e > \frac{1}{2}$	$e = e^\dagger = 3f - \frac{1}{4}$			Hopf
$e > \frac{1}{2}$	$2f < e < 3f - \frac{1}{4}$	UNSTABLE	UNSTABLE	
$e > \frac{1}{2}$	$f > \frac{e}{2}$	UNSTABLE	UNFEASIBLE	

3.5 Bifurcations

In addition to the transcritical of Theorem 3, we now try to establish whether for special parameters combinations Hopf bifurcations originate near $\widehat{E}_2$. We need purely imaginary

 ISBN: 978-84-616-2723-3

eigenvalues for the characteristic equation, and this occurs when the trace of the Jacobian vanishes, i.e. (7) becomes an equality and the constant term is positive, $\det(\widehat{J}_2) = e - 2f > 0$, which is true in view of (5). Thus, in summary

Theorem 6. The system (4) admits a Hopf bifurcation at the coexistence equilibrium when the bifurcation parameter e crosses the critical value

$$e^{\dagger} = 3f - \frac{1}{4}. \tag{8}$$

For the dimensionalized model (3), Figure 5 contains the simulations of the system trajectories and the corresponding limit cycles in the phase plane. The experiments are run over long times to show that the oscillations obtained are really persistent.

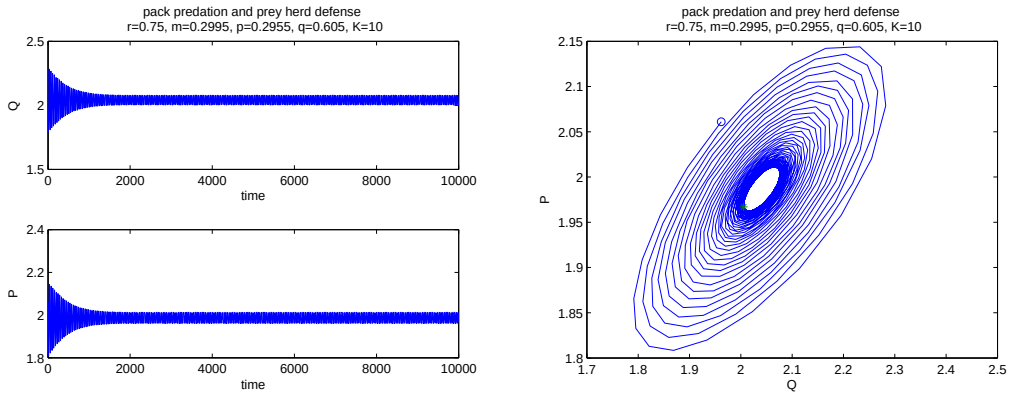


Figure 5: Left: time series of the system trajectories (3); Right: corresponding limit cycle in the phase plane. The original parameter values are $r = 0.75$, $m = 0.2995$, $p = 0.2955$, $q = 0.605$, $K = 10$, with coexistence equilibrium $E_2 = (2.041, 1.9869)$; they correspond to $e = 1.2521$, $f = 0.4983$ in the rescaled model (4).

4 Conclusions

We have examined predators' pack hunting, in correspondence both of individual prey behavior, as well as their gathering in herds. The major difference between the two models is that when the prey move loose in their habitat, the ecosystem always attains a coexistence equilibrium. Instead, if they gather together, presumably for defense purposes, the ecosystem may be prone to extinction. This is a rather counterintuitive result, since one would expect the defensive strategy to work. However, it can be interpreted observing that

 ISBN: 978-84-616-2723-3

perhaps the group defense is less effective when predators hunt in pack, since the prey perhaps are more easily discovered. Clearly, if the herd is wiped out, then the predators will disappear as well, as the model assumes that no other food source is available for them. Note that ecosystem extinction does occur, but rarely, in the model without pack predation, [8]. Furthermore, in case of herd defense, we have seen that the populations can exhibit persistent oscillations. Other than this, in both situations we can find only that populations coexist, as the prey-only equilibrium is unstable. However, the population values at coexistence in the two cases differ. Therefore the two proposed models differ almost only in the equilibria location in the phase plane, but their qualitative behavior is about similar, apart from the limit cycles arising in the latter system.

In order to better compare these results from the quantitative point of view, we consider also the classical model, with logistic correction. However, in such case, since the square root is absent, if we rescale it, we end up with different adimensionalized variables and coefficients than those used here. It is therefore imperative to return to the original dimensionalized models, for this comparison, both for the classical model and for those introduced here.

Now, if we consider the classical Lotka-Volterra model with logistic correction for the prey, i.e. (1) with mass action terms and in place of the square root interaction term, it is easy to find its equilibria. In particular the coexistence point C_* is feasible for $pK > m$ and in such case always stable. When unfeasible, the equilibrium $(K, 0)$ takes its place, via a transcritical bifurcation. The coexistence equilibria of the two models presented here, respectively with loose prey and herd defense, in dimensional form are given by

$$E_2 \equiv \left(K\frac{mr}{mr+pqK}, \frac{p^2}{m^2}K^2\left(\frac{mr}{mr+pqK}\right)^2\right), \quad \widehat{E}_2 \equiv \left(K\left(1-\frac{pq}{mr}\right), \frac{p^2}{m^2}K\left(1-\frac{pq}{mr}\right)\right).$$

For the classical model and the model with loose hunting and prey group defense instead the coexistence equilibrium reads, [1],

$$C_* \equiv \left(\frac{m}{p}, \frac{r}{q}\left(1-\frac{m}{pK}\right)\right), \quad \widetilde{E}_2 = \left(\frac{m^2}{p^2}, \frac{mr}{pq}\left(1-\frac{m^2}{p^2K}\right)\right).$$

To assess the population levels we need essentially to compare m and p, i.e. the predators' mortality and predation efficiency. The ultimate prey population values of the latter two models depend on model parameters that belong to predators and not on their reproductive capabilities nor to the environment carrying capacity. When the predators instead pack together, the prey equilibrium values at coexistence involve also their own intrinsic characteristics. Still in the last two models if the predators' hunting efficiency exceeds their own mortality, $m < p$, the prey settle at a much lower value when gathering together for defense purposes; the predators instead will reach a higher population value. On the contrary for $m > p$ the prey defensive strategy keeps them at higher numbers than when they behave individualistically; the predators instead will have the opposite result, reaching lower values when the prey use a defensive behavior, and higher ones with individualistic prey.

 ISBN: 978-84-616-2723-3

The reverse situation occurs for pack hunting and herd defense. The ratio of the predators' hunting efficiency p and their mortality m tells whether their population will ultimately be larger than that of the prey. A similar result could hold for the model of pack hunting coupled with loose prey, but the predators population at equilibrium depends on the prey population squared. If the latter is smaller than 1 the conclusion is not immediate. At E_2 and at $\widehat{E}_2$ the prey populations are given by a term in the bracket, smaller than 1, multiplied by the carrying capacity, so that its value depends on K and may be large, most likely, but also small. A population smaller than 1 is in fact not counterintuitive, since we can not just count individuals, but also can measure the population by its weight. Depending on the units chosen, the claim therefore makes sense.

References

[1] V. Ajraldi, E. Venturino, *Mimicking spatial effects in predator-prey models with group defense*, in J. Vigo Aguiar, P. Alonso, S. Oharu, E. Venturino, B. Wade (Eds), Proceedings of the International Conference CMMSE 2009, 1 (2009) 57-66.

[2] V. Ajraldi, M. Pittavino, E. Venturino, *Modeling herd behavior in population systems*, Nonlinear Analysis: Real World Applications **12** (2011) 2319-2338.

[3] P.A. Braza, *Predator prey dynamics with square root functional responses*, Nonlinear Analysis: Real World Applications **13** (2012) 1837-1843.

[4] *E. Cagliero, E. Venturino Ecoepidemics with group defense and infected prey protected by herd*, Proceedings of the 12th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 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 (Editors) **1** (2012) 247-266.

[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) 65-75.

[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] H. Malchow, S. Petrovskii, E. Venturino, *Spatiotemporal patterns in Ecology and Epidemiology*, CRC, Boca Raton, 2008.

[8] E. Venturino, S. Petrovskii, *Spatiotemporal Behavior of a Prey-Predator System with a Group Defense for Prey*, Ecological Complexity **14** (2013) 37-47.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Numerical Optimization Experiments Using the Hyperbolic Smoothing Strategy to Solve MPCC

Teófilo M. M. Melo[1], João L. H. Matias[2] and M. Teresa T. Monteiro[3]

[1] *CIICESI, School of Technology and Management of Felgueiras, Polytechnic of Porto*

[2] *Centre of Mathematics, CM-UTAD, University of Trás-os-Montes e Alto Douro*

[3] *R&D Centre Algoritmi, Department of Production and Systems, University of Minho*

emails: tmm@estgf.ipp.pt, j_matias@utad.pt, tm@dps.uminho.pt

Abstract

In this work we solve Mathematical Programs with Complementarity Constraints using the hyperbolic smoothing strategy. Under this approach, the complementarity condition is relaxed through the use of the hyperbolic smoothing function, involving a positive parameter that can be decreased to zero. An iterative algorithm is implemented in MATLAB language and a set of AMPL problems from MacMPEC database were tested.

Key words: complementarity constraints, hyperbolic smoothing, SQP

1 Introduction

Mathematical Programs with Complementarity Constraints (MPCC) is a subclass of more general Mathematical Programs with Equilibrium Constraints (MPEC). These kind of constraints may come in the form of a game, a variational inequality or as stationary conditions of an optimization problem. The main applications areas are Engineering and Economics [1], [2], [3]. They are so widespread in this areas because the concept of complementarity is synonymous with the notion of system equilibrium. They are very difficult to solve as the usual constraint qualifications necessary to guarantee the algorithms convergence fail in all feasible points [4]. This complexity is caused by the disjunctive nature of the complementarity constraints. They have been proposed some nonlinear approaches to solve MPCC, starting with the smoothing scheme [5], [6], the regularization scheme [7], [8] the interior

point methods [10], the penalty approaches [11], [12], [13] and the "elastic mode" for nonlinear programming in conjunction with a sequential quadratic programming (SQP) algorithm [14]. On this paper we present the hyperbolic smoothing strategy [15] and we apply it for solving MPCC. The proposed method adopts a C^∞ differential class function, in order to overcome the difficulties on solving the complementarity constraints.
This paper is organized as follows. Next section defines the MPCC problem. Some optimal issues are presented in Section 3. The hyperbolic smoothing technique and the MATLAB algorithm are described in Section 4. Numerical experiments using the hyperbolic smoothing algorithm are reported in Section 5. Some conclusions and future work are exposed in Section 6.

2 Problem definition

We consider Mathematical Program with Complementarity Constraints (MPCC):

$$\begin{array}{ll} \min & f(x) \\ s.t. & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \\ & 0 \leq x_1 \perp x_2 \geq 0, \end{array} \tag{1}$$

where f and c are the nonlinear objective function and the constraint functions, respectively, assumed to be twice continuously differentiable. E and I are two disjoined finite index sets with cardinality p and m, respectively. A decomposition $x = (x_0, x_1, x_2)$ of the variables is used where $x_0 \in \mathbb{R}^n$ (control variables) and $(x_1, x_2) \in \mathbb{R}^{2q}$ (state variables). The expressions $0 \leq x_1 \perp x_2 \geq 0 : \mathbb{R}^{2q} \to \mathbb{R}^q$ are the q complementarity constraints. One attractive way of solving (1) is to consider its equivalent nonlinear programming formulation:

$$\begin{array}{ll} min & f(x) \\ s.t. & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \\ & x_1 \geq 0,\ x_2 \geq 0, \\ & X_1 x_2 \leq 0, \end{array} \tag{2}$$

where X_1 is a diagonal matrix with x_1 as diagonal. On this formulation the complementarity constraints are replaced by a set of nonlinear inequalities, such as $x_{1j}\, x_{2j} \leq 0$, $j = 1, \ldots, q$, enabling the use of standard NLP solvers to solve the complementary constraints.

3 Optimal issues

This section introduces some concepts related to stationarity and first order conditions. The optimality concepts follow the development of [16] and the corresponding proves can

 ISBN: 978-84-616-2723-3

be consulted in this work. Consider two index sets: $X_1, X_2 \subset \{1, \ldots, q\}$ with $X_1 \cup X_2 = \{1, \ldots, q\}$, denoting the corresponding complements in $\{1, \ldots, q\}$ by $X_1^{\perp}$ e $X_2^{\perp}$. For each pair of index one define the relaxed NLP corresponding to (1):

$$\begin{array}{ll} min & f(x) \\ s.t. & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \\ & x_{1j} = 0,\ \forall j \in X_2^{\perp}, \\ & x_{2j} = 0,\ \forall j \in X_1^{\perp}, \\ & x_{1j} \geq 0,\ \forall j \in X_2, \\ & x_{2j} \geq 0,\ \forall j \in X_1. \end{array} \tag{3}$$

Concepts like constraints qualification, stationarity and second order conditions of the MPCC problem will be defined in terms of (3). The linear independence constraint qualification, LICQ, is extended to MPCC that is MPCC-LICQ:

Definition 1. MPCC-LICQ Consider $x_1, x_2 \geq 0$ and define: $X_1 = \{j : x_{1j} = 0\}$, $X_2 = \{j : x_{2j} = 0\}$. The MPCC problem verifies the MPCC-LICQ at x if the corresponding (3) verifies the LICQ.

If x^* is a local solution of (3) and satisfies ${x_1^*}^T x_2^* = 0$, then x^* is also a local solution of original MPCC.

There are several kinds of stationarity defined for MPCC problem. Among them, the strong stationarity is the following one:

Definition 2. Strong stationarity x^* is a strong stationary point if exist Lagrange multipliers λ, $\widehat{\nu}_1$ and $\widehat{\nu}_2$ so that:

$$\begin{array}{l} \nabla f^* - [\nabla(c_i^*), i \in E \ : \ \nabla(c_i^*), i \in I]\, \lambda - \begin{pmatrix} 0 \\ \widehat{\nu}_1 \\ \widehat{\nu}_2 \end{pmatrix} = 0, \\ c_i^* = 0, i \in E, \\ c_i^* \geq 0, i \in I, \\ x_1^* \geq 0, \\ x_2^* \geq 0, \\ x_{1j}^* = 0 \text{ or } x_{2j}^* = 0, \\ \lambda_i \geq 0, i \in I, \\ c_i \lambda_i = 0, \\ x_{1j}^* \widehat{\nu}_{1j} = 0, \\ x_{2j}^* \widehat{\nu}_{2j} = 0, \\ \text{if } x_{1j}^* = x_{2j}^* = 0 \text{ then } \widehat{\nu}_{1j} \geq 0 \text{ and } \widehat{\nu}_{2j} \geq 0. \end{array} \tag{4}$$

 ISBN: 978-84-616-2723-3

Note that (4) are the first order optimality conditions of the problem (3) at x^*. As theoretical support, we summarized some known results concerning constraint qualifications and first order optimality conditions of MPCC. Based on these ideas, a computational implementation of a hyperbolic smoothing strategy was developed. Details of the corresponding hyperbolic smoothing algorithm are in next section.

4 Hyperbolic smoothing

They have been proposed several smoothing approaches, the most obvious smoothing analysed by [9] is to replace $X_1 x_2 \leq 0$ by $X_1 x_2 \leq \epsilon_k$, and solve a sequence of NLPs, decreasing ϵ_k to zero. Another similar approach studied by [7] is to gather the complementarity constraints into a single constraint by $x_1^T x_2 \leq \epsilon_k$. Other alternative is to penalize the complementarity constraints [12], solving a sequence of NLPs where the objective is modified as

$$\min f(x) + \rho_k x_1^T x_2$$

for a sequence of increasing penalty parameters $\rho_k > 0$.
Another smoothing idea [6] is to replace the complementarity constraints by the smoothed function,

$$\psi_\mu(x_{1j}, x_{2j}) = \sqrt{(x_{1j} - x_{2j})^2 + 4\mu} - x_{1j} - x_{2j} = 0,$$

for $j = 1, \ldots, q$, where $\mu > 0$ is a parameter that decreases to zero.
On this work we consider the following NLP:

$$\begin{array}{ll} min & f(x) \\ s.t. & c_i(x) = 0,\ i \in E, \\ & c_i(x) \geq 0,\ i \in I, \\ & x_1 \geq 0,\ x_2 \geq 0, \\ & \phi(x_1, x_2) \leq 0, \end{array} \tag{5}$$

where $\phi(x_1, x_2) = (\varphi_\tau(x_{11}, x_{21}), \ldots, \varphi_\tau(x_{1q}, x_{2q}))$ is a vector and φ_τ is the hyperbolic smoothing function defined as follows:

$$\varphi_\tau(x_{1j}, x_{2j}) = \frac{1}{2}\left(x_{1j}x_{2j} + \sqrt{(x_{1j}x_{2j})^2 + \tau^2}\right),$$

for $j = 1, \ldots, q$ and $\tau \to 0$. An algorithm was implemented (Algorithm 1) to iteratively solve problem (5) with $\tau \to 0$. This algorithm has two iterative procedures, the inner one is performed by `fmincon` routine from MATLAB Optimization toolbox, that uses the SQP method.

 ISBN: 978-84-616-2723-3

Algorithm 1 Hyperbolic smoothing

1: Take initial values x_0, $\tau_0 > 0$ and tolerances ϵ_1, ϵ_2.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Solve the minimization problem (5) with x_k, τ_k obtaining x_{k+1}.
4: **if** $\|\nabla L(x_{k+1}, \ldots)\| \leq \epsilon_1$ and $\|x_1^T x_2\| \leq \epsilon_2$ **then**
5: STOP.
6: **else**
7: $\tau_{k+1} = r\tau_k,\ 0 < r < 1$.
8: **end if**
9: **end for**

To evaluate the stop criterium in the algorithm, we consider the following equality in the solution x^*:

$$\nabla L(x^*, \delta, \gamma, \xi) = \nabla f(x^*) - \sum_{i=1}^{m} \delta_i \nabla c_i(x^*) - \sum_{i=1}^{p} \gamma_i \nabla c_i(x^*) + \sum_{j=1}^{q} \xi_i \nabla \varphi_{\tau,j}(x^*)$$

where for $j = 1, \ldots, q$ and $x \in \mathbb{R}^n$ we have

$$\nabla \varphi_{\tau,j}(x^*) = \frac{1}{2}\left(\nabla x_{1j}x_{2j} + \frac{x_{1j}x_{2j}\nabla x_{1j}x_{2j}}{\sqrt{(x_{1j}x_{2j})^2 + \tau^2}}\right).$$

The Lagrange multipliers δ, γ and ξ are an output of the `fmincon` routine from MATLAB. The tolerances used in the stop criterium are $\epsilon_1 = \epsilon_2 = 10^{-4}$. The initial choices, $\tau_0 = 0.25$ and $r = 0.25$ were considered. Next section reports the numerical results using 45 test problems.

5 Numerical results

This section describes the numerical experiments with an implementation of the hyperbolic smoothing scheme for problem (1). The computational experiments were made on a *2.26 GHz Intel Core 2 Duo* with 8GB of RAM, MAC OS 10.6.8 operating system. The MATLAB version used was 7.11.0 (R2010b). The `fmincon` routine is connected to the modeling language AMPL [17] by a MATLAB mex interface and the test problems are from MacMPEC database [18].
Table 1 reports the numerical results achieved by Algorithm 1, the first column indicates the name of the test problem, from column 2 to 5, the problem dimensions are presented. Column f^* shows the final objective function value and column $\|\nabla L\|$ presents the norm of the Lagrangian function of problem (5). The last three columns give information about the performance of the algorithm. Column `int` presents the number of internal iterations performed by the `fmincon` routine from MATLAB, column `ext` shows the number of external

 ISBN: 978-84-616-2723-3

Problem	n	m	p	q	f^*	$\|\nabla L\|$	int	ext	nfe
bard1	5	3	1	3	17.000	1.781910e-15	63	10	661
bard3	6	2	3	1	-12.679	8.149116e-09	19	18	161
bard1m	6	3	1	3	17.000	9.949933e-16	40	10	480
bard2m	12	4	5	3	-6598.000	7.831440e-06	104	10	2076
bilevel2	16	9	4	8	-6600.000	5.560937e-05	229	11	5275
bilevel2m	16	9	4	8	-6600.000	5.560937e-05	229	11	5275
dempe	3	1	1	1	28.258	5.577955e-06	175	26	3952
desilva	6	2	2	2	-1.000	7.450581e-09	35	18	677
df1	2	3	0	1	0.000	9.425165e-09	13	10	54
ex9.1.1	13	5	7	5	-13.000	1.159107e-15	27	20	419
ex9.1.2	8	2	5	2	-3.000	1.364484e-07	35	22	521
ex9.1.4	8	2	5	2	-37.000	4.322747e-15	22	17	229
ex9.1.5	13	5	7	5	-1.000	6.182457e-15	27	21	421
ex9.1.8	11	4	5	3	-3.250	4.965068e-16	23	19	311
ex9.1.10	11	4	5	3	-3.250	4.965068e-16	23	19	311
ex9.2.9	9	3	5	3	2.000	5.787607e-16	15	10	281
flp2	4	2	0	2	0.000	1.568440e-07	58	12	706
flp4-1	80	60	0	30	0.000	5.132135e-12	11	10	902
flp4-2	110	110	0	60	0.000	9.102046e-12	11	10	1232
flp4-3	140	170	0	70	0.000	7.670026e-12	11	10	1562
flp4-4	200	250	0	100	0.000	9.583691e-12	11	10	2222
incid-set1-8	118	54	50	49	0.000	1.147902e-14	15	11	1949
incid-set1c-8	117	61	49	49	0.000	1.288618e-07	14	10	1844
jr1	2	1	0	1	0.500	2.349058e-08	63	3	457
kth1	2	1	0	1	0.000	0	3	2	12
kth2	2	1	0	1	0.000	2.014697e-08	13	10	53
kth3	2	1	0	1	0.500	6.852287e-06	23	10	236
nash1a	6	2	2	2	0.000	1.701382e-07	13	10	104
nash1b	6	2	2	2	0.000	2.359488e-07	16	10	187
nash1c	6	2	2	2	0.000	2.359719e-07	14	10	130
nash1d	6	2	2	2	0.000	2.359995e-07	16	10	153
outrata31	5	4	0	4	3.208	2.704044e-08	85	10	903
outrata32	5	4	0	4	3.449	2.823092e-08	139	15	3419
outrata33	5	4	0	4	4.604	6.415710e-07	144	12	2674
outrata34	5	4	0	4	6.593	4.768317e-07	172	17	2662
qpec1	30	20	0	20	80.000	1.055822e-14	13	10	416
scholtes1	3	1	0	1	2.000	5.176421e-08	107	10	2137
scholtes2	3	1	0	1	15.000	1.336380e-07	107	10	2137
scholtes3	2	1	0	1	0.500	2.279238e-07	36	11	380
scholtes5	3	2	0	2	1.000	7.377520e-06	223	10	1987
scale1	2	1	0	1	1.000	9.643328e-05	107	21	824
scale2	2	1	0	1	1.000	1.491230e-06	74	17	729
scale3	2	1	0	1	1.000	2.210586e-05	28	12	177
scale5	2	1	0	1	100.000	7.805362e-05	43	11	422
sl1	8	3	2	3	0.0001	3.266939e-08	13	10	157
stackelberg1	3	1	1	1	-3266.667	1.101177e-06	65	9	946

Table 1: Numerical results.

iterations and the last column reports the number of function evaluations. The solutions obtained by Algorithm 1 are similar to the ones reported in MacMPEC database with good accuracy.

6 Conclusions and future work

An iterative algorithm in MATLAB language to solve MPCC was implemented. The algorithm aims to compute a local optimal solution joining the hyperbolic smoothing the SQP strategy. The algorithm is still in an improvement phase but some conclusions can already be taken: the promising numerical results present good accuracy of the solutions when compared with the ones provided from the MacMPEC test problem database. As future work, it is intended to test the method on large scale test problems and compare the hyperbolic

 ISBN: 978-84-616-2723-3

smoothing strategy with others smoothing methods suggested in literature.

Acknowledgements

This work is funded by FEDER funds through Operational Programme for Competitiveness Factors - COMPETE and National Funds through FCT - Foundation for Science and Technology in Project scope: FCOMP-01-0124-FEDER-022674. The work is also supported by PEst-OE/MAT/UI4080/2011. The authors are very grateful to all the financial support.

References

[1] Z. Luo, J. Pang and D. Ralph, *Mathematical programs with equilibrium constraints*, Cambridge University Press, 1996.

[2] M. Ferris and J. Pang, *Engineering and economic applications of complementarity constraints*, SIAM Review **39** (1997) 669–713.

[3] J. Outrata, M. Kocvara and J. Zowe, *Nonsmooth approach to optimization problems with equilibrium constraints*, Kluwer Academic Publishers, Dordrecht, 1998.

[4] X. Chen, M. Florian, *The nonlinear bilevel programming problem: formulations, regularity and optimality conditions*, Optimization **32** (1995) 193–209.

[5] M. Fukushima, and J. Pang, *Convergence of a smoothing continuation method for mathematical programs with complementarity constraints*, Lectures Notes in Economics and Mathematical Systems, Springer-Verlag **447** (1999) 99–110.

[6] F. Facchinei, H. Jiang and L. Qi, *A smoothing method for mathematical programs with equilibrium constraints*, Mathematical Programming **85** (1999) 107–134.

[7] S. Scholtes, *Convergence proprieties of a regularization schemes for mathematical programs with complementarity constraints*, SIAM Journal on Optimization **11** (2001) 918–936.

[8] M. Monteiro and H. Rodrigues, *Combining the regularization strategy and the SQP to solve MPCC – A MATLAB implementation*, Journal of Computational and Applied Mathematics **235** (2011) 5348–5356.

[9] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints: Stationarity, optimality and sensitivity*, Mathematics of Operations Research **25** (2001) 1–22.

 ISBN: 978-84-616-2723-3

[10] Raghunathan, A. and Biegler, L., *Interior point methods for mathematical programs with complementarity constraints*, CAPD technical report, Department of Chemical Engineering, Carnegie Mellon University, June 2003

[11] X. Hu and D. Ralph, *Convergence of a penalty method for mathematical programs with complementarity constraints*, Journal of Optimization Theory and Application **123** (2004) 365–390.

[12] D. Ralph and S. Wright, *Some proprieties of regularization and penalization schemes for MPECs*, Optimization Methods and Software **19** (2004) 527–556.

[13] M. Monteiro and J. Meira, *A penalty method and a regularization strategy to solve MPCC*, International Journal of Computers Mathematics **88** (2011) 145–149.

[14] M. Anitescu, *On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints*, SIAM Journal on Optimization **15** (2005) 1203–1236.

[15] A. Xavier and A. Oliveira, *Optimal Covering of Plane Domains by Circles Via Hyperbolic Smoothing*, Journal of Global Optimization **31** (2005) 493–504

[16] R. Fletcher, S. Leyffer, D. Ralph and S. Scholtes, *Local convergence of SQP methods for mathematical programs with equilibrium constraints*, SIAM Journal on Optimization **17** (2006) 259–286.

[17] R. Fourer and B. Kernighan, *AMPL: A modeling language for mathematical programming*, Duxburg Press, Massachusetts, 1993.

[18] S. Leyffer, *MacMPEC AMPL collection of mathematical programs with equilibrium constraints*, http://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

A Batched Cholesky Solver for Local RX Anomaly Detection on GPUs

J. M. Molero[1], E. M. Garzón[1], I. García[2], E.S. Quintana-Ortí[3] and A. Plaza[4]

[1] *Department of Informatics and Agrifood Campus of International Excellence (CEIA3), University of Almería, SPAIN*

[2] *Department of Computer Architecture, University of Málaga, SPAIN*

[3] *Department of Engineering and Computer Science, Universidad Jaime I, SPAIN*

[4] *Hyperspectral Computing Laboratory, University of Extremadura, SPAIN*

emails: jmp384@ual.es, gmartin@ual.es, igarciaf@uma.es, quintana@icc.uji.es, aplaza@unex.es

Abstract

We investigate the problem of simultaneously solving thousands of dense linear systems of moderate size via the Cholesky factorization on Graphics Processing Units (GPUs). We propose the concept of a batched solver to calculate hundreds of different and independent Cholesky, thus addresing the corresponding linear systems concurrently. This problem arises, among others, in the context of anomaly detection in hyperspectral images, an important task for Earth observation data exploitation. The approach studied for this purpose is the local version of the well-known RX (Reed-Xiaoli) algorithm (LRX), which applies the same concept as the original RX to a local sliding window centered around each image pixel. LRX has a very high computational cost and can be expressed in terms of batched solvers which are evaluated in this application context.

Key words: Batched Cholesky solver, GPU computing, anomaly detection, remote sensing, RX algorithm.

ISBN: 978-84-616-2723-3

1 Introduction

The Cholesky factorization is a well-known method for the numerical solution of linear systems of equations when the coefficient matrix is symmetric and positive definite (SPD) [1]. Blocked implementations of this method are available from the LAPACK library[1], or the MKL library[2], while MAGMA[3] includes a hybrid CPU-GPU implementation of this factorization. All these implementations are designed to accelerate the calculation of a single Cholesky factorization, being specially tuned to deal with very large dense matrices.

On the other hand, there are applications which require a different computation scheme, where it is necessary to solve many independent linear systems of small-to-moderate dimension. The linear algebra operations related to this particular context are often referred to as *batched* [3]. Two parallel levels can be exploited in this kind of solvers: (1) the parallelism of the standard solver, which is constrained by the data dependencies and the size of the local problem; and (2) the concurrency intrinsic to the solution of multiple linear systems. Very few references address the parallel implementation of batched solvers on GPUs [4]. In this work, we develop a CUDA version of the Batched Cholesky Solver (hereafter, `CuBCholS`) for NVIDIA GPUs [5,6], describing the major implementation details and performing an experimental evaluation on an NVIDIA GeForce GTX 680 GPU, for the matrix cases encountered in the field of remotely sensed hyperspectral image processing. This field is characterized by the availability of remotely sensed images with hundreds of different channels (corresponding to different wavelengths, typically in the visible and near infra-red part of the spectrum) for the same area on the surface of the Earth. These data open tremendous possibilities from the viewpoint of remote Earth observation, but these come at the expense of very high computational cost that can be addressed using specialized architectures.

2 Batched Cholesky Solver on GPU

Given a SPD matrix $A \in \mathbb{R}^{n \times n}$, the Cholesky factorization computes the decomposition $A = LL^T$, where $L \in \mathbb{R}^{n \times n}$ is lower triangular and directly yields the solution to the linear system $Ax = b$ via two simple triangular system solves [1, 7].

In this work, our interest focuses on scenarios where it is necessary to solve thousands of dense SPD linear systems of moderate dimension (concretely, $50 \leq n \leq 500$) using the Cholesky factorization. Therefore, the global problem has a high computational load, clearly asking for the adoption of high performance platforms. In this paper, we propose a batched implementation to solve these systems on GPUs (`CuBCholS`). For this purpose, we develop a tiled algorithm that leverages the computational resources of the GPU, comput-

[1] http://www.netlib.org/lapack/
[2] http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation
[3] http://icl.cs.utk.edu/magma/

 ISBN: 978-84-616-2723-3

ing hundreds of independent linear systems simultaneously. Two levels of parallelism are exploited by `CuBCholS` on GPU: (1) internally, each system can be decomposed into a series computations (tasks) with dependences among these tasks; and (2) externally, the computation of the linear systems can be performed independently. According to this scheme, in our GPU implementation of the Cholesky method, each CUDA block computes a Cholesky factorization and solves the corresponding system. In this context, we optimize these linear algebra operations taking into account the use of the resources of a single CUDA block. Moreover, the set of blocks solves the systems associated to `CuBCholS`, efficiently exploiting the resources of the GPU platform.

3 Local RX based on `CuBPOTRF`

In this work, we consider the local RX algorithm as a case study for `CuBCholS`. This approach [9–11] is specially suited for local anomaly detection in hyperspectral images as, for each pixel $\mathbf{x}$, the LRX filter is computed independently using a square window of size $\kappa \times \kappa$, centered at pixel $\mathbf{x}$. Consequently, the filter is defined by:

$$\delta_{\kappa}^{LRX}(\mathbf{x}) = \mathbf{x}^T \mathbf{R}_{\kappa\times\kappa}(\mathbf{x})^{-1}\mathbf{x}, \tag{1}$$

where $\mathbf{R}$ is the correlation matrix and $\mathbf{x}$ is the hyperspectral pixel (a vector of dimension equal to the number bands, B, of the hyperespectral image, typically in the order of a couple hundreds).

Bearing in mind the previous descriptions, three stages can be identified in the LRX algorithm for every pixel of the image ($\mathbf{x}$): (1) evaluation of the correlation matrices $\mathbf{R}_{\kappa\times\kappa}(\mathbf{x})$; (2) computation of the intermediate vector $y(\mathbf{x}) = \mathbf{R}^{-1}\mathbf{x}$; (this stage can be expressed in terms of the batched Cholesky solver as, for every pixel, a system of dimension B is involved;) and (3) computation of the output filter $\delta(\mathbf{x}) = \mathbf{x}^T\mathbf{y}$. Stages 1 and 2 exhibit a high computational cost when LRX is applied to a real hyperspectral scene. The acceleration of Stage 1 in GPUs has been described in [12], while our focus here is on Stage 2.

4 Experimental Evaluation

Our evaluation of `CuBCholS` is carried out on a computing platform equipped with an Intel Xeon E5640 (2.67 GHz) multicore processor with 12 GB of RAM memory. The GPU connected to the system is an NVIDIA GeForce GTX 680 GPU (GK104 "Kepler" architecture) [8], with 8 multiprocessors and 192 cores per multiprocessor (for a total of 1536 cores), clock rate of 1.06 GHz, and 2 GB of global memory.

In order to evaluate the performance of `CuBCholS`, we have considered a real hyperspectral data set, which was collected in the framework of the HYperspectral Digital Image Collection Experiment (HYDICE). The original scene [13] consists of 64×64 pixels and

 ISBN: 978-84-616-2723-3

Table 1: Computing time (in milliseconds) for the most important stages of LRX, processing a line of the real hyperpectral scene (HYDICE) using different number of spectral bands.

Bands	Cholesky + Solver (CPU)	`CuBCholS` (GPU)	SpeedUp
64	23.4	6.9	3.3 ×
128	104.9	13.6	7.7 ×
160	147.2	21.7	6.7 ×
169	153.6	25.2	6.1 ×

B=169 bands, but we defined several tests with different number of bands, from 64 to 169, for our experiments. As a preliminary study, Table 1 shows the cost of processing a single line (64 pixels) of the image. Although this time is small, the speed-up obtained with the GPU illustrates the benefits of routine `CuBCholS`.

5 Conclusions

This work provides a new batched GPU implementation that is especially appropriate to handle hundreds of small to moderate dense SPD linear systems. In our approach, we map an independent Cholesky factorization and the corresponding linear system solver to an entire GPU CUDA block. This routine is applied and evaluated in the context of anomaly detection in hyperspectral images via the LRX algorithm, and preliminary results show the advantage of this routine in terms of performance, even with small test images of reduced dimensions. While the batched solver is tailored to suit the special needs of hyperspectral image processing, our approach carries beyond this particular application, and we believe that it is also valid for other batched applications.

Acknowledgements

This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117, TIN2011-23283, TIN2012-37483-C03 and AYA2011-29334-C02-02), Junta de Andalucia (P10-TIC-6002) and Junta de Extremadura (PRI09A110 and GR10035) in part financed by the European Regional Development Fund (ERDF). Moreover, it has been developed in the framework of the network High Performance Computing on Heterogeneous Parallel Architectures (CAPAP-H4), supported by the Spanish Ministry of Science and Innovation (TIN2011-15734-E).

References

[1] G. Golub and C. V. Loan, *Matrix Computations Third Edition*, The Johns Hopkins University Press, 1996.

[2] *Matrix Algebra on GPU and Multicore Architectures*, http://icl.cs.utk.edu/magma

[3] NVIDIA Cublas manual, *http://docs.nvidia.com/cuda/cublas/index.html*

[4] M. J. Anderson, D. Sheffield and K. Keutzer, *A Predictive Model for Solving Small Linear Algebra Problems in GPU Registers*, IEEE 26th International Parallel and Distributed Processing Symposium (IPDPS). (2012) p 2–13.

[5] NVIDIA CUDA C Programming Guide, *http://developer.nvidia.com/category/cuda-zone*, October, 2012.

[6] R. Farber, *CUDA, Application Design and Development*, Morgan Kaufman, 2010.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C (2nd ed.): the art of scientific computing*, Cambridge University Press, 1992.

[8] NVIDIA GeForce GTX 680 Whitepaper, *http://www.geforce.com/Active/en_US/en_US/pdf/GeForce-GTX-680-Whitepaper-FINAL.pdf* (2012).

[9] J. M. Molero, E. M. Garzón, I. García and A. Plaza, *Analysis and Optimizations of Global and Local Versions of the RX Algorithm for Anomaly Detection in Hyperspectral Data*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. vol. 6, no. 2, pp. 801-8014, April 2013.

[10] I. S. Reed and X. Yu, *Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution.*, IEEE Trans. Acoustics Speech and Signal Processing. 138 (1990) 1760–1770.

[11] S. Matteoli, M. Diani, and G. Corsini, *A tutorial overview of anomaly detection in hyperspectral images*, IEEE Aerospace and Electronic Systems Magazine. 25 (2010) 5–28.

[12] J. M. Molero, E. M. Garzón, I. García, E. S. Quintana and A. Plaza, *Accelerating the KRX Algorithm for Anomaly Detection in Hyperspectral Data on GPUs*, Proceedings of the 12th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2012.

[13] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Norwell, MA: Kluwer. (2003).

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Designs and binary codes constructed from the simple group Ru of Rudvalis

J Moori[1] and B Rodrigues[2]

[1] *School of Mathematical Sciences, North-West University (Mafikeng), Mmabatho 2735, South Africa*

[2] *School of Mathematical Sciences , University of KwaZulu-Natal, Durban 4041, South Africa*

emails: jamshid.moori@nwu.ac.za, rodrigues@ukzn.ac.za

Abstract

This talk is based on an article written by the authors that has appeared in the Journal of Algebra [14]. In [10] we developed two methods for constructing codes and designs from finite groups. In this paper we apply the Method 1 to the sporadic simple group Ru of Rudvalis and study the binary codes C_n (where $n \in \{1755, 1756, 2304, 2305, 4059\}$), defined by the rank-3 primitive permutation representation of degree 4060 on the cosets of the Ree group $2_{F_4(2)}$. These codes are obtained from the row span of the incidence matrices of the designs defined by the union of the orbits of $2_{F_4(2)}$. We prove that $\dim(C_{1756}) = 29$, $\dim(C_{2304}) = 28$, $C_{1756} \supset C_{2304}$ and Ru acts irreducibly on C_{2304}. Furthermore we have $C_{1755} = C_{2305} = C_{4059} = V_{4060}(\mathbb{F}_2)$, $\mathrm{Aut}(\mathcal{D}_{1755}) = \mathrm{Aut}(\mathcal{D}_{1756}) = \mathrm{Aut}(\mathcal{D}_{2304}) = \mathrm{Aut}(\mathcal{D}_{2305}) = \mathrm{Aut}(C_{1756}) = \mathrm{Aut}(C_{2304}) = \mathrm{Ru}$ while $\mathrm{Aut}(\mathcal{D}_{4059}) = \mathrm{Aut}(C_{1755}) = \mathrm{Aut}(C_{2305}) = \mathrm{Aut}(C_{4059}) = S_{4060}$. We also determine the weight distribution of C_{1756} and C_{2304} and that of their duals. For each word w_l of weight l, in the codes $C_{\Gamma_{1756}}$ or C_{2304} we determine the stabilizer $(\mathrm{Ru})_{w_l}$, and some primitive designs held by particular codewords.

Key words: Design, Code, Sporadic Simple Group, Rudvalis Group, Primitive Permutation Representation

MSC 2000: AMS codes (05E20, 94B25, 20D08)

1 Introduction

The simple group Ru of Rudvalis is one the 26 sporadic simple groups. It has a rank-3 primitive permutation representation of degree 4060 which can be used to construct a

strongly regular graph Γ with parameters $v = 4060$, $k = 1755$, $\lambda = 730$ and $\mu = 780$ or its complement a strongly regular $\widetilde{\Gamma} = (4060, 2304, 1328, 1280)$ graph. The stabilizer of a vertex u in this representation is a maximal subgroup isomorphic to the Ree group $2_{F_4(2)}$ producing orbits $\{u\}$, Δ_1, Δ_2 of lengths 1, 1755, and 2304 respectively. The regular graphs Γ, $\widetilde{\Gamma}, \Gamma^R, \widetilde{\Gamma}^R$, Γ^S are constructed from the sets Δ_1, Δ_2, $\{u\} \cup \Delta_1$, $\{u\} \cup \Delta_2$, and $\Delta_1 \cup \Delta_2$, respectively. If A denotes an adjacency matrix for Γ then $B = J - I - A$, where J is the all-one and I the identity 4060×4060 matrix, will be an adjacency matrix for the graph $\widetilde{\Gamma}$ on the same vertices. We examine the neighbourhood designs $\mathcal{D}_{1755}$, $\mathcal{D}_{1756}$, $\mathcal{D}_{2304}$, $\mathcal{D}_{2305}$ and $\mathcal{D}_{4059}$ and corresponding binary codes C_{1755}, C_{1756}, C_{2304}, C_{2305}, and C_{4059} defined by the binary row span of A, $A + I$, B, $B + I$ and $A + B$ respectively. Note that $A + I$ and $B + I$ are adjacency matrices for the graphs $\Gamma^R, \widetilde{\Gamma}^R$ obtained from Γ and $\widetilde{\Gamma}$, respectively, by including all loops, and thus referred to as reflexive graphs. Since Γ^S is the complete graph on 4060 vertices, we do not need to use Ru to construct it. Also clearly $\mathrm{Aut}(\Gamma^S) = \mathrm{Aut}(\mathcal{D}_{4059}) = \mathrm{Aut}(C_{4059}) = S_{4060}$. So throughout the paper we omit discussions on Γ^S, $\mathcal{D}_{4059}$ and C_{4059}.

In the theorem given below, we summarize our results; the specific results relating to the codes are given as propositions and lemmas in the following sections.

Theorem 1 *Let G be the simple Rudvalis group* Ru *and Γ, $\widetilde{\Gamma}, \Gamma^R, \widetilde{\Gamma}^R$ be the regular graphs defined by the union of the orbits of $2_{F_4(2)}$, and C_i where $i \in \{1755, 1756, 2304, 2305\}$ be the codes defined by the binary row span of their adjacency matrices. Then*

(i) $\mathrm{Aut}(\mathcal{D}_{1755}) = \mathrm{Aut}(\mathcal{D}_{1756}) = \mathrm{Aut}(\mathcal{D}_{2304}) = \mathrm{Aut}(\mathcal{D}_{2305}) = \mathrm{Aut}(C_{1756}) = \mathrm{Aut}(C_{2304}) =$ Ru.

(ii) $\dim(C_{1756}) = 29$, $\dim(C_{2304}) = 28$, $C_{1756} \supset C_{2304}$ *and* Ru *acts irreducibly on* C_{2304}.

(iii) $C_{1755} = C_{2305} = V_{4060}(\mathbb{F}_2)$.

(iv) $\mathrm{Aut}(C_{1755}) = \mathrm{Aut}(C_{2305}) = S_{4060}$.

Note that Theorem 1 (i) implies that $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(\Gamma^R) = \mathrm{Aut}(\widetilde{\Gamma}) = \mathrm{Aut}(\widetilde{\Gamma}^R) =$ Ru. The proof of the theorem follows from a series of lemmas in Sections 7, 6, 8, and 9 respectively. We will show that the codes C_{1756} and C_{2304} are of types $[4060, 29, 1756]_2$ and $[4060, 28, 1792]_2$, respectively. Moreover,

$$C_{1756} = \langle C_{2304}, \boldsymbol{\jmath} \rangle = C_{2304} \cup \{w + \boldsymbol{\jmath} : w \in C_{2304}\} = C_{2304} \oplus \langle \boldsymbol{\jmath} \rangle,$$

where $\boldsymbol{\jmath}$ denotes the all-ones vector. Let W_l denote the set of all codewords of C_{1756} of weight l and let A_l be the size of W_l, that is $|W_l| = A_l$. Then clearly $W_l + \{\boldsymbol{\jmath}\} = W_{4060-l} \subset C_{1756}$ and $|W_l| = A_l = |W_{4060-l}| = A_{4060-l}$. We find the weight distributions of C_{1756} and that of C_{2304}. The structure of the stabilizers $(\mathrm{Ru})_{w_l}$, for all nonzero weight l, where $w_l \in C_{1756}$ is

 ISBN: 978-84-616-2723-3

a codeword of weight l (see Table 4) is determined in Section 9. The weight distributions and the structures of the stabilizers $(\mathrm{Ru})_{w_l}$ for C_{2304} follows clearly from those of C_{1756}.

The paper is organized as follows: after a brief description of our terminology and some background, Section 5 outlines the construction of graphs, designs and codes, and in Sections 6, 7, 8, and 9 we present our results.

2 Terminology and notation

Our notation will be standard, and it is as in [1] and $\mathbb{ATLAS}$ [4]. For the structure of groups and their maximal subgroups we follow the $\mathbb{ATLAS}$ notation. The groups $G.H$, $G:H$, and $G^{\cdot}H$ denote a general extension, a split extension and a non-split extension respectively. For a prime p, the symbol p^n denotes an elementary abelian group of that order. If p is an odd prime, p_+^{1+2n} and p_-^{1+2n} denote the extraspecial groups of order p^{1+2n} and exponent p or p^2 respectively.

Terminology for **graphs** is standard: the graphs, $\mathcal{G} = (V, E)$ with vertex set V and edge set E, are undirected and the **degree** of a vertex is the number of edges containing the vertex. A graph is **regular** if all the vertices have the same degree; a regular graph is **strongly regular** of type (n, k, λ, μ) if it has n vertices, degree k, and if any two adjacent vertices are together adjacent to λ vertices, while any two non-adjacent vertices are together adjacent to μ vertices. A rank 3 graph is a graph that admits an automorphism group which is transitive on the vertices, edges, and nonedges. Note that any rank 3 graph is a strongly regular graph. The converse is not always true. The complementary graph of a strongly regular graph with parameters (n, k, λ, μ) is a strongly regular graph with parameters $(n, n-k-1, n-2k+\mu-2, n-2k+\lambda)$. The **neighbourhood design** of a regular graph is the 1-design formed by taking the points to be the vertices of the graph and the blocks to be the sets of neighbours of a vertex, for each vertex. The **code** of a graph Γ over a finite field F is the row span of an adjacency matrix A over the field F, denoted by $C_F(\Gamma)$ or $C_F(A)$. The dimension of the code is the rank of the matrix over F, also written $\mathrm{rank}_p(A)$ if $F = \mathbb{F}_p$, in which case we will speak of the **p-rank** of A or Γ, and write $C_p(\Gamma)$, $C_p(A)$ or simply C_Γ for the code. A connected strongly regular graph has diameter 2. If v and w are vertices of a connected strongly regular graph Γ such that $d(v, w) = i$, $i = 0, 1, 2$, then the number p_{ij} of neighbors of w whose distance from v is j, $j = 0, 1, 2$, are the intersection numbers of Γ. The 3×3-matrix with entries p_{ij}, $i, j = 0, 1, 2$, is called the *intersection matrix* of Γ. The weight enumerator of C_Γ is defined as $W_{C_\Gamma}(x) = \sum_{i=0}^{n} A_i x^i$, where A_i denotes the number of codewords of weight i in C_Γ. The *dual* code ${C_\Gamma}^\perp$ is the orthogonal complement under the standard inner product $(,)$, i.e. ${C_\Gamma}^\perp = \{v \in F^n \,|\, (v, c) = 0 \text{ for all } c \in C_\Gamma\}$. A code C_Γ is *self-orthogonal* if $C_\Gamma \subseteq {C_\Gamma}^\perp$ and it is *self-complementary* if it contains the all-one vector. The all-one vector will be denoted by $\mathbf{1}$, and is the constant vector of weight the length of the code. A binary code

 ISBN: 978-84-616-2723-3

C_Γ is *doubly-even* if all codewords of C_Γ have weight divisible by four. Two linear codes of the same length and over the same field are **isomorphic** if they can be obtained from one another by permuting the coordinate positions. An **automorphism** of a code C_Γ is an isomorphism from C_Γ to C_Γ. The automorphism group will be denoted by $\mathrm{Aut}(C_\Gamma)$.

3 Preliminary results

The designs and codes in this paper come from the following construction, described in [7, Proposition 1] and corrected in [9]. See also [8]:

Result 2 *Let G be a finite primitive permutation group acting on the set Ω of size n. Let $\alpha \in \Omega$, and let $\Delta \neq \{\alpha\}$ be an orbit of the stabilizer G_α of α. If $\mathcal{B} = \{\Delta^g \mid g \in G\}$ and, given $\delta \in \Delta$, $\mathcal{E} = \{\{\alpha, \delta\}^g \mid g \in G\}$, then $\mathcal{D} = (\Omega, \mathcal{B})$ forms a symmetric 1-$(n, |\Delta|, |\Delta|)$ design. Further, if Δ is a self-paired orbit of G_α then $\Gamma = (\Omega, \mathcal{E})$ is a regular connected graph of valency $|\Delta|$, $\mathcal{D}$ is self-dual, and G acts as an automorphism group on each of these structures, primitive on vertices of the graph, and on points and blocks of the design.*

In fact one can use any union of orbits of a point-stabilizer in this construction, and this is the approach that we will adopt in the paper. The above construction has been applied by the authors to various sporadic groups, for example see [11], [12] and [13].

4 The Rudvalis group Ru

In this section we give a brief but complete overview of the Rudvalis group Ru. For more information on the Rudavlis group we refer the reader to [4, p.126] or [17, Section 5.9.3]. The simple group Ru of Rudvalis is the automorphism group of a certain 28-dimensional lattice over $\mathbb{Z}[i]$. There is a monomial group of type $(2^6{:}U_3(3)){:}2 \cong 2^6 \cdot G_2(2)$ which may be constructed by taking the 28 non-zero isotropic vectors in a unitary 3-space over $\mathbb{F}_9$. The square of a non-zero element $a \in \mathbb{F}_9$ is a fourth root of unit which can be lifted to the complex number $a^{[2]} = \pm 1$ or $\pm \mathrm{i}$. The Rudvalis group can be described as the automorphism group of a regular graph Γ of degree 1755 or $\tilde{\Gamma}$ of degree 2304 on 4060 vertices. The vertex stabilizer is $2_{F_4(2)}$, and in the degree 1755 case the stabilizer of a pair of non-adjacent vertices is $L_2(25) \cdot 2^2$. The stabilizer of an edge is a non-maximal subgroup isomorphic to $2^{1+4+6} \cdot 5 \cdot 4$ contained in the centralizer of an involution of shape $2^{1+4+6}S_5$. The orbit of length 1755 corresponds to the points of generalized octagon for $2_{F_4(2)}$. Two points are joined in the graph when they are perpendicular in the 26-dimensional natural representation of $2_{F_4(2)}$. In the degree 2304 case, the vertices of the graph are the minimal vectors considered modulo the unit scalar factors ± 1, $\pm \mathrm{i}$. These have norm 4; a vertex is joined (inner product ± 1, or $\pm \mathrm{i}$) to 2304 others or is disjoined (orthogonal) to the remaining 1755. Four mutually orthogonal vertices form a quartet if any further vertex orthogonal to three of them is orthogonal to

 ISBN: 978-84-616-2723-3

the fourth. Modulo $1+\mathsf{i}$ the lattice yields a 28-dimensional orthogonal representation over $\mathbb{F}_2$ in which many of the vector stabilizers are maximal subgroups.

Result 3 *(Wilson [16]) The Rudvalis simple group* Ru *has just fifteen conjugacy classes of maximal subgroups, as follows:*
(A) *Four classes of* 2*-local subgroups:*
$2 \cdot 2^{4+6}{:}S_5$, $2^{3+8}{:}L_3(2)$, $(2^6{:}U_3(3)){:}2$, $(2^2 \times Sz(8)){:}3$.
(B) *Four classes of odd-local subgroups:*
$3A_6 \cdot 2^2$, ${5_+}^{1+2}{:}[2^5]$, $5{:}4 \times A_5$, $5^2{:}4S_5$.
(C) *Seven classes of non-local subgroups:*
$A_6 \cdot 2^2,\ A_8,\ L_2(25) \cdot 2^2,\ L_2(29),\ L_2(13){:}2,\ U_3(5){:}2,\ 2_{F_4(2)}$.

The primitive representations referred to in Result 3 are listed in Table 1. The first column gives the ordering of the primitive representations as given by Magma [2], [3] (or the ATLAS [4]) and as used in our computations; the second gives the maximal subgroups; the third gives the degree (the number of cosets of the point stabilizer);

No.	Max. sub.	Deg.	No.	Max. sub.	Deg.
1	$2_{F_4(2)}$	4060	9	$L_2(29)$	11980800
2	$(2^6{:}U_3 3){:}2$	188500	10	$5^2{:}4S_5$	12160512
3	$(2^2 \times S_z(8)){:}3$	417600	11	$3 \cdot A_6 \cdot 2^2$	33779200
4	$2^{3+8}{:}L_3(2)$	424125	12	${5_+}^{1+2}{:}[2^5]$	36481536
5	$U_3(5){:}2$	579072	13	$L_2(13){:}2$	66816000
6	$2 \cdot 2^{4+6}{:}S_5$	593775	14	$A_6 \cdot 2^2$	101337600
7	$L_2(25) \cdot 2^2$	4677120	15	$5{:}4 \times A_5$	121605120
8	A_8	7238400			

Table 1: Maximal subgroups of Ru

5 The graphs, designs and codes

Observe from Result 3 and Table 1 that there is just one class of maximal subgroups of Ru of index 4060. The stabilizer of a vertex u in this representation is a maximal subgroup isomorphic to $2_{F_4(2)}$, producing orbits $\{u\}$, Δ_1, and Δ_2 of lengths 1, 1755 and 2304 respectively. The regular graphs $\Gamma, \Gamma^R, \widetilde{\Gamma}, \widetilde{\Gamma}^R$ are constructed from the sets Δ_1, $\{u\} \cup \Delta_1$, Δ_2 and $\{u\} \cup \Delta_2$, respectively. Notice that the orbit containing a single element, has been omitted, as it would produce a trivial design. The binary codes C_{1755}, C_{1756}, C_{2304}, C_{2305} whose properties we will be examining are obtained as described below.

 ISBN: 978-84-616-2723-3

- The rows of an adjacency matrix A for Γ give the blocks of the neighbourhood design of Γ which we will denote $\mathcal{D}_{1755}$. Notice that $\mathcal{D}_{1755}$ is a self-dual symmetric 1-$(4060, 1755, 1755)$ design. We write C_{1755} to denote the binary code spanned by the rows of $\mathcal{D}_{1755}$.
- From the rows of an adjacency matrix $A + I$ of the reflexive graph Γ^R we obtain the self-dual symmetric 1-$(4060, 1756, 1756)$ design $\mathcal{D}_{1756}$, and the binary code C_{1756}.
- The rows of an adjacency matrix B for $\widetilde{\Gamma}$ yield the neighbourhood 1-$(4060, 2304, 2304)$ design $\mathcal{D}_{2304}$. This is a self-dual symmetric design, and the binary row span of gives the code C_{2304}.
- From the rows of an adjacency matrix $B + I$ of the reflexive graph $\widetilde{\Gamma}^R$ we get the self-dual symmetric 1-$(4060, 2305, 2305)$ design $\mathcal{D}_{2305}$. We write C_{2305} to denote the binary code of $\mathcal{D}_{2305}$.

The Rudvalis group Ru acts on each of these graphs, designs and codes and it is always the full automorphism group. In the sequel, when necessary we will use the graphs and their corresponding designs interchangeably. This will be noticed for example in the proofs of Lemma 6 and Lemma 8. In Sections 6, 7, 8 we deal with these designs and respective binary codes.

6 Designs $\mathcal{D}_{1755}$ and $\mathcal{D}_{2305}$

In Lemma 4 below we examine the properties of the designs $\mathcal{D}_i$, and of their binary codes C_i spanned by the rows of the incidence matrices of each $\mathcal{D}_i$.

Lemma 4 *Let G be the Rudvalis group* Ru *and $\mathcal{D}_i$ and C_i where $i \in \{1755, 2305, 4059\}$ be the designs and binary codes constructed from the primitive rank-3 permutation action of G on the cosets of $2_{F_4(2)}$. Then*

(i) $\mathrm{Aut}(\mathcal{D}_{1755}) = \mathrm{Aut}(\mathcal{D}_{2305}) =$ Ru *and $\mathcal{D}_{1755}$ is the unique point-primitive and flag-transitive symmetric design on* 4060 *points.*

(ii) $C_{1755} = C_{2305} = V_{4060}(\mathbb{F}_2)$.

(iii) $\mathrm{Aut}(C_{1755}) = \mathrm{Aut}(C_{2305}) = S_{4060}$.

Proof: (i) The definition of Ω and $\mathcal{B}$ emerges from Result 2, and from this it is clear that $G \subseteq \mathrm{Aut}(\mathcal{D}_{1755})$. It follows from Result 2, and also from the Atlas [4, p.126] that G acts primitively on both Ω and $\mathcal{B}$ of degree $|\Omega| = |\mathcal{B}| = 4060$, and the stabilizer of a vertex u (point) has exactly three orbits in Ω. Hence G_u fixes setwise each of $\{u\}$, Δ_1

 ISBN: 978-84-616-2723-3

and $\Omega \setminus (\Delta_1 \cup \{u\}) = \Delta_2$ and these are all possible G_u-orbits. This shows that $\mathcal{D}_{1755}$ is a point primitive, symmetric 1-design. It remains to show that $G = \mathrm{Aut}(\mathcal{D}_{1755})$. Now $G \subseteq \mathrm{Aut}(\mathcal{D}_{1755}) \subseteq S_{4060}$, so $\mathrm{Aut}(\mathcal{D}_{1755})$ is a primitive permutation group on Ω of degree 4060. Moreover, $\mathrm{Aut}(\mathcal{D}_{1755})_u$ must fix Δ_1 setwise, and hence $\mathrm{Aut}(\mathcal{D}_{1755})_u$ also has orbits of lengths 1, 1755, and 2304 in Ω. The only primitive group of degree 4060, such that $\mathrm{Aut}(\mathcal{D}_{1755})_u$ can have orbit lengths 1, 1755, and 2304 is Ru, see [6, Theorem 18]. Hence $G = \mathrm{Aut}(\mathcal{D}_{1755})$. Since $\mathcal{D}_{2305} = \tilde{\mathcal{D}}_{1755}$, we deduce that $\mathrm{Aut}(\mathcal{D}_{2305}) = \mathrm{Aut}(\mathcal{D}_{1755}) = \mathrm{Ru}$. Recall that there is a unique class of maximal subgroups of Ru of type $2_{F_4(2)}$. Now, given a subgroup K in that class, its normalizer is twice bigger in Ru, meaning that there are exactly two subgroups $2_{F_4(2)}$ that contain K, and so we derive a contradiction. Thus, we conclude that there is a unique 1-$(4060, 1755, 1755)$ symmetric design invariant under Ru, and since the block stabilizer acts transitively on the points of the block the claim on flag-transitivity holds.

(ii) For $p = 2$, the row span of $\mathcal{D}_{1755}$ and $\mathcal{D}_{2305}$, respectively, yield the full space $V_{4060}(\mathbb{F}_2)$. That is

$$C_{1755} = C_{2305} = V_{4060}(\mathbb{F}_2).$$

(iii) Since $\mathrm{Aut}(V_{4060}(\mathbb{F}_2)) = S_{4060}$, we have

$$\mathrm{Aut}(C_{1755}) = \mathrm{Aut}(C_{2305}) = S_{4060}.$$

□

7 The code of the graph Γ^R

The rows of an adjacency matrix $A + I$ for the reflexive graph Γ^R give the blocks of the design $\mathcal{D}_{1756}$. In Lemma 5 we determine the automorphism group of this design, and in Lemma 6 we examine some of the properties of its binary code C_{1756}.

Lemma 5 *For* Ru *of degree 4060, the automorphism group of the graph Γ^R or design $\mathcal{D}_{1756}$ is a non-abelian finite simple group of order 145926144000. Moreover this group is isomorphic to the simple sporadic group* Ru.

Proof: This follows readily by computations with Magma. □

Lemma 6 *The group* Ru *is the automorphism group of the $[4060, 29, 1756]_2$ code C_{1756} obtained from $\mathcal{D}_{1756}$. The code C_{1756} is self-orthogonal doubly-even. Its dual is a $[4060, 4031, 4]_2$ code. Moreover, $\jmath \in C_{1756}$.*

 ISBN: 978-84-616-2723-3

Proof: It is clear that the rows of the adjacency matrix of Γ^R generate C_{1756} (this is also the case if we regard the adjacency matrix for Γ^R as the incidence matrix for the 1-$(4060, 1756, 1756)$ design $\mathcal{D}_{1756}$ and by Lemma 5 we have $\mathrm{Aut}(\mathcal{D}_{1756}) = \mathrm{Ru}$. Furthermore, direct calculations reveal that $|\mathrm{Aut}(\mathcal{D}_{1756})| = |\mathrm{Aut}(C_{1756})|$, thus we have $\mathrm{Aut}(C_{1756}) \cong \mathrm{Ru}$. Now, using the fact that the block size of $\mathcal{D}_{1756}$ is divisible by four, i.e. $1756 \equiv 0 \pmod 4$ and that all block intersection numbers are even (in fact two distinct blocks intersect either in 780 or 732 points), we have that the design $\mathcal{D}_{1756}$ is self-orthogonal. Thus, the rows of the block-point incidence matrix of $\mathcal{D}_{1756}$ span a self-orthogonal binary code of length 4060 namely C_{1756}. Since the incidence vectors of the blocks of the design span the code, and the vectors have weight 1756, we deduce that C_{1756} is doubly-even. In fact Magma gives the weight enumerator which is listed below.

$$\begin{aligned} W_{C_{1756}} = 1 \;&+\; 4060\, x^{1756} + 188500\, x^{1792} + 417600\, x^{1820} + 4677120\, x^{1952} \\ &+\; 33779200\, x^{1980} + 38001600\, x^{1984} + 95597775\, x^{2012} \\ &+\; 95769600\, x^{2016} + 95769600\, x^{2044} + 95597775\, x^{2048} \\ &+\; 38001600\, x^{2076} + 33779200\, x^{2080} + 4677120\, x^{2108} \\ &+\; 417600\, x^{2240} + 188500\, x^{2268} + 4060\, x^{2304} + x^{4060}. \end{aligned}$$

Notice that there are 4060 codewords of minimum weight 1756 in C_{1756}. Thus, the incidence matrix of $\mathcal{D}_{1756}$ is determined by the set of all minimum weight codewords up to a column permutation. In addition note that the blocks of $\mathcal{D}_{1756}$ are of even size, so $\boldsymbol{\jmath}$ meets evenly every vector of C_{1756}, and thus $\boldsymbol{\jmath} \in {C_{1756}}^{\perp}$. From computations with Magma we deduce that the sum (modulo 2) of all rows of the generator matrix for the code is the all-one vector, hence $\boldsymbol{\jmath} \in C_{1756}$. That ${C_{1756}}^{\perp}$ has minimum weight 4 was found using Magma. The full weight distribution can be obtained. Notice that $w_{4060} = \boldsymbol{\jmath} \in C_{1756}$ and that $\langle \boldsymbol{\jmath} \rangle$ is an 1-dimensional Ru$-$invariant subspace of C_{1756}. Also $A_{4060-l} = |\{w_l + \boldsymbol{\jmath} : w_l \in C_{1756}\}| = |\{w_l : w_l \in C_{1756}\}| = A_l$. Direct calculations show that $\dim(C_{1756}) = 29$ (it will be shown in Lemma 8 that $C_{1756} \supseteq C_{2304}$, and that C_{1756} is in fact C_{2304} adjoined by the all-one vector), and hence C_{1756} is a $[4060, 29, 1756]_2$ code. □

Remark: Notice that the minimum weight 1756 of C_{1756} is the valency of the graph Γ^R, thus the minimum weight codewords are precisely the rows of the adjacency matrix $A + I$.

8 The code of the graph $\widetilde{\Gamma}$

The rows of an adjacency matrix B for the graph $\widetilde{\Gamma}$ gives the blocks of the neighbourhood design $\mathcal{D}_{2304}$. In Lemma 7 below we determine the automorphism group of this design. Further, in Lemma 8 we examine the properties of its binary code C_{2304}.

 ISBN: 978-84-616-2723-3

Lemma 7 *For* Ru *of degree 4060, the automorphism group of the design* $\mathcal{D}_{2304}$ *is isomorphic to the group* Ru.

Proof: Since $\mathcal{D}_{2304} = \tilde{\mathcal{D}}_{1756}$, we have $\mathrm{Aut}(\mathcal{D}_{2304}) = \mathrm{Aut}(\tilde{\mathcal{D}}_{1756}) = \mathrm{Aut}(\mathcal{D}_{1756})$. Now the proof follows from Lemma 5. □

It should now be clear that an adjacency matrix for the graph $\widetilde{\Gamma}$ is also an incidence matrix for the neighbourhood design $\mathcal{D}_{2304}$. The reader will notice that in the sequel we use either structures interchangeably. We will see an instance of this interplay in the proof of Lemma 8 where we examine the binary code $C_{2304} = [4060, 28, 1792]_2$ spanned by the rows of the adjacency matrix of the graph $\widetilde{\Gamma}$. In that lemma we establish some of the properties of C_{2304}, and show that this code is in fact the unique irreducible 28-dimensional module invariant under the Rudvalis group.

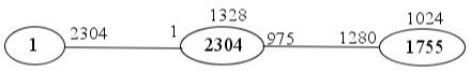
Lemma 8 *The group* Ru *is the automorphism group of* C_{2304}*. The code* C_{2304} *is self-orthogonal doubly-even, with minimum weight 1792. Its dual is a* $[4060, 4032, 4]_2$*. Moreover,* Ru *acts irreducibly on* C_{2304} *as an* $\mathbb{F}_2$*-module,* $C_{2304} \subset C_{1756}$*, and* $\mathrm{Aut}(C_{2304}) = \mathrm{Ru}$.

Proof: We will use the strong regularity of $\widetilde{\Gamma}$ to show that the code C_{2304} is self-orthogonal. Notice first that C_{2304} is obtained from the strongly regular graph $\widetilde{\Gamma}$ with parameters $(4060, 2304, 1328, 1280)$ and intersection matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 2304 & 1328 & 1280 \\ 0 & 975 & 1024 \end{bmatrix}.$$

It can be seen from Figure 1 below that if we fix a vertex v in $\widetilde{\Gamma}$ we can divide the remaining vertices into two sets, namely $\widetilde{\Gamma}'$ of size 2304 and $\widetilde{\Gamma}''$ of size 1755, with $\widetilde{\Gamma}'$ being the set of vertices adjacent to v, and $\widetilde{\Gamma}''$ the set of vertices non-adjacent to v. Now, from the second column of the above matrix we deduce that each vertex in $\widetilde{\Gamma}'$ is adjacent to v and to 1328 other vertices in $\widetilde{\Gamma}'$, thus to 975 vertices in $\widetilde{\Gamma}''$ while from the third column shows that a vertex in $\widetilde{\Gamma}''$ is adjacent to 1280 vertices in $\widetilde{\Gamma}'$, and so to 1024 vertices in $\widetilde{\Gamma}''$. The structure of the graph and the orbit joins are summarized in the following diagram.

Figure 1: Number of joins between orbits of a stabilizer



The valency 2304 ensures that generating codewords have length zero (mod 2) and the 1328 and the 1280 ensure that (i) any two generating codewords have an even number of

 ISBN: 978-84-616-2723-3

non-zero entries in common, and (ii) that any two generating codewords are orthogonal to one another. Hence C_{2304} is self-orthogonal, and since all non-zero codewords have weights divisible by 4, it follows that C_{2304} is doubly-even. Moreover, the blocks of $\mathcal{D}_{2304}$ are of even size, so $\boldsymbol{\jmath}$ meets evenly every vector of C_{2304}, so $\boldsymbol{\jmath} \in {C_{2304}}^{\perp}$. It can be deduced from [5, Section 3] that the 2-rank of $\widetilde{\Gamma}$ is 28, and so the dimension of C_{2304} follows.

We used Magma to calculate the weight enumerator of C_{2304} which is as follows:

$$\begin{aligned} W_{C_{2304}} = 1 \quad &+ \quad 188500\ x^{1792} + 4677120\ x^{1952} + 38001600\ x^{1984} \\ &+ \quad 95769600\ x^{2016} + 95597775\ x^{2048} + 33779200\ x^{2080} \\ &+ \quad 417600\ x^{2240} + 4060\ x^{2304}. \end{aligned}$$

Finally, notice that the 2-modular character table of the group Ru is completely known (see [15]) and it follows from it that the irreducible 28-dimensional $\mathbb{F}_2$-representation is unique. Moreover, 28 is the smallest dimension for any non-trivial irreducible $\mathbb{F}_2$-module invariant under Ru. Using the above weight distribution, we can easily see that C_{2304} does not contain a invariant subspace of dimension 1. If C_{2304} were reducible, it would contain an invariant irreducible subspace E of dimension d where $1 \leq d \leq 27$, which is not possible. Hence C_{2304} is irreducible and must be isomorphic to the 28-dimensional $\mathbb{F}_2$-module on which Ru acts irreducibly. Further, notice that $\mathcal{D}_{1756}$ is the complement of $\mathcal{D}_{2304}$, so the inclusion follows as C_{1756} is C_{2304} adjoined by the $\boldsymbol{\jmath}$ vector. Finally we can also use this fact to show that $\mathrm{Aut}(C_{2304}) = \mathrm{Ru}$. Clearly Ru sits in $\mathrm{Aut}(C_{2304})$. If $\alpha \in \mathrm{Aut}(C_{2304})$, then since $\alpha(\boldsymbol{\jmath}) = \boldsymbol{\jmath}$ and $C_{1756} = \langle C_{2304}, \boldsymbol{\jmath} \rangle$, we have $\alpha \in \mathrm{Aut}(C_{1756})$. So that $\mathrm{Aut}(C_{2304}) \subseteq \mathrm{Aut}(C_{1756})$. Now, order arguments and Lemma 6 show that $\mathrm{Aut}(C_{2304}) = \mathrm{Ru}$. □

9 Stabilizer in Ru of a word w_i of weight i

Due to symmetry (since $\boldsymbol{\jmath} \in C_{1756}$) we will only consider the action of Ru on the code C_{2304}. In fact, the structures of the stabilizers of the codewords of C_{1756} can be described by means of the stabilizers of the codewords of C_{2304}. Thus, let w_i be a word of weight i in C_{2304}, in this section we determine the structures of $(\mathrm{Ru})_{w_i}$ i.e, the stabilizers of w_i in Ru. These are listed in Table 3. Also, for each w_i, we take the support of w_i and orbit its image under the action of $G = \mathrm{Ru}$ to form the blocks of the $1-(4060, i, k_i)$ designs $\mathcal{D}_{w_i}$, where $k_i = |(w_i)^G| \times \frac{i}{4060}$. Information on these designs is listed in TABLE 4.

Proposition 9, which follows, deals with the action of Ru on the words of C_{2304}. Since Ru acts as an automorphism group of C_{2304} we consider this action and determine the structure of $(\mathrm{Ru})_{w_i}$ where i is in L, with L as defined below.

Let $L = \{1792, 1952, 2080, 2240, 2304\}$ and $\overline{L} = \{1984,\ 2016, 2048\}$. For $i \in L \cup \overline{L}$ we define $W_i = \{w_i \in C_{2304} \mid \mathrm{wt}(w_i) = i\}$, with $\mathrm{wt}(w_i)$ denoting the weight of the word w_i,

 ISBN: 978-84-616-2723-3

and let $|W_i| = |A_i|$. We show in Proposition 9 that $(\mathrm{Ru})_{w_i}$ is a maximal subgroup of Ru, for all $i \in L$. Also, for $w_i \in W_i$ we take the support of w_i and orbit that under Ru to form the blocks of the 1-designs D_{w_i}. We show that Ru acts primitively on D_{w_i}, for all $i \in L$.

Proposition 9 *Let $i \in L$ and $w_i \in W_i$. Then $(\mathrm{Ru})_{w_i}$ is a maximal subgroup of* Ru. *Furthermore* Ru *is primitive on $\mathcal{D}_{w_i}$ for each i.*

Proof: Let $i \in L$ with L is a described above. Our computations show that ${w_i}^{\mathrm{Ru}} = W_i$. Therefore each W_i forms an orbit under the action of Ru and thus Ru is transitive on each W_i. For $w_i \in W_i$ we construct $(\mathrm{Ru})_{w_i}$ as a permutation group inside Ru. Furthermore, we determine its structure by computing its composition factors. We deduce that for $i \in L$ we have $(\mathrm{Ru})_{w_i} \in \{2_{F_4(2)},\ (2^6{:}U_3(3){:}2),\ (2^2 \times S_z(8)){:}3,\ L_2(25){:}2^2,\ 3{\cdot}A_6{\cdot}2^2\}$. By the transitivity of Ru on the code coordinates, the codewords of W_i form a 1-design $\mathcal{D}_{w_i}$ with A_i blocks. This implies that Ru is transitive on the blocks of D_{w_i} for each w_i and since $(\mathrm{Ru})_{w_i}$ is a maximal subgroup of Ru, we deduce that Ru acts primitively on $\mathcal{D}_{w_i}$. □

Remark: Note that if $i \in \overline{L}$ and $w_i \in W_i$. Then $(\mathrm{Ru})_{w_i}$ is not necessarily a maximal subgroup of Ru. In Table 3, for $i = 1984$ we have $(\mathrm{Ru})_{w_{1984}} \cong 2^6{:}A_5$ which is not a maximal subgroup Ru.

10 Observations

(i) In Table 3 the first column represents the words of weight i and the second column represents the stabilizer in Ru of a codeword w_i of W_i. In the final column we test the maximality of $(\mathrm{Ru})_{w_i}$ in Ru.

Table 3
Stabilizer in Ru of a word w_i

i	$(\mathrm{Ru})_{w_i}$	Maximality
1792	$(2^6 : U_3(3){:}2)$	Yes
1952	$L_2(25){\cdot}2^2$	Yes
1984	$2^6{:}A_5$	No
2080	$3{\cdot}A_6{\cdot}2^2$	Yes
2240	$(2^2 \times Sz(8)){:}3$	Yes
2304	$2_{F_4(2)}$	Yes

(ii) In Table 4 the first column represents the words of weight i and the second column gives the structure of the designs $\mathcal{D}_{w_i}$ which were defined in Section 9. In the third column we list the number of blocks of $\mathcal{D}_{w_i}$. We test the primitivity for the action of Ru on $\mathcal{D}_{w_i}$ in the final column.

 ISBN: 978-84-616-2723-3

Table 4
1-designs $\mathcal{D}_{w_i}$ from Ru

i	$\mathcal{D}_{w_i}$	No. of blocks	Primitivity
1792	1-(4060, 1792, 83200)	188500	Yes
1952	1-(4060, 1952, 2248704)	4677120	Yes
1984	1-(4060, 1984, 18570240)	38001600	No
2080	1-(4060, 2080, 17305600)	33779200	Yes
2240	1-(4060, 2240, 230400)	417600	Yes
2304	1-(4060, 2304, 2304)	4060	Yes

11 Concluding remarks

In the paper we uncover many interesting links between combinatorial designs, graphs, groups, codes and modular representation theory. We show that C_{2304} is irreducible when regarded as a Ru-invariant $\mathbb{F}_2$-module. Moreover, we show that the stabilizers of several sets of codewords are maximal subgroups of the automorphism group. Thus, some designs are primitive.

Acknowledgements

The authors would like to thank Markus Grassl from the Centre for Quantum Technologies at the National University of Singapore, for the valuable help provided in the computation of the orbits of the automorphism groups of the codes. The authors also thank the anonymous referee for helpful and constructive remarks and suggestions.

Acknowledgements

This work was supported by the NRF (South Africa), the Universities of KwaZulu-Natal and North-West (Mafikeng).

References

[1] E. F. Assmus, Jr and J. D. Key. *Designs and their Codes*. Cambridge: Cambridge University Press, 1992. Cambridge Tracts in Mathematics, Vol. 103 (Second printing with corrections, 1993).

[2] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *J. Symb. Comp.*, 24, 3/4:235–265, 1997.

 ISBN: 978-84-616-2723-3

[3] J. Cannon, A. Steel, and G. White. Linear codes over finite fields. In J. Cannon and W. Bosma, editors, *Handbook of Magma Functions*, pages 3951–4023. Computational Algebra Group, Department of Mathematics, University of Sydney, 2006. V2.13, http://magma.maths.usyd.edu.au/magma.

[4] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson. *An Atlas of Finite Groups*. Oxford: Oxford University Press, 1985.

[5] K. Coolsaet. A construction of the simple group of Rudvalis from the group $U_3(5)$:2. *J. Group Theory*, **1** (1998), no. 2, 146–163.

[6] Hannah J. Coutts, Martyn Quick and Colva M. Roney-Dougal. The primitive permutation groups of degree less than 4096. *Comm. Algebra.*, **39** (2011), 3526–3546.

[7] J. D. Key and J. Moori. Designs, codes and graphs from the Janko groups J_1 and J_2. *J. Combin. Math. and Combin. Comput.* **40** (2002), 143–159.

[8] J. D. Key, J. Moori, and B. G. Rodrigues. On some designs and codes from primitive representations of some finite simple groups. *J. Combin. Math. and Combin. Comput.* **45** (2003), 3–19.

[9] J. D. Key and J. Moori. Correction to: "Codes, designs and graphs from the Janko groups J_1 and J_2" [J. Combin. Math. Combin. Comput. **40** (2002), 143–159], *J. Combin. Math. Combin. Comput.* **64** (2008), 153.

[10] J. Moori, Finite groups, designs and codes, *Information Security, Coding Theory and Related Combinatorics*, Nato Science for Peace and Security Series D: Information and Communication Security, **29** (2011), 202–230, IOS Press (ISSN 1874-6268).

[11] J. Moori and B. G. Rodrigues. A self-orthogonal doubly even code invariant under M^{c}L:2. *J. Combin. Theory Ser. A*, **110** (2005), no. 1, 53–69.

[12] J. Moori and B. G. Rodrigues. Some designs and codes invariant under the simple group Co_2. *J. of Algebra* **316** (2007), 649–661.

[13] J. Moori and B. G. Rodrigues. A self-orthogonal doubly-even code invariant under M^{c}L. *Ars Combin.* **91** (2009), no. 1, 321–332.

[14] J. Moori and B. G. Rodrigues. Some designs and binary codes preserved by the simple group Ru of Rudvalis, *J. Algebra* **372** (2012), 702–710.

[15] R. A. Parker and R. A. Wilson. The 2-modular character table of the Rudvalis group, unpublished, 1998.

 ISBN: 978-84-616-2723-3

[16] R. A. Wilson. The geometry and maximal subgroups of the simple groups of A. Rudvalis and J. Tits. *Proc. London Math. Soc.* **3** (1984), 533–563.

[17] R. A. Wilson. *The finite simple groups*. London: Springer-Verlag London Ltd., 2009. Graduate Texts in Mathematics, Vol. 251.

 ISBN: 978-84-616-2723-3

Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.

Relaxing the Role of Adjoint Pairs in Multi-adjoint Logic Programming

G. Moreno, J. Penabad and C. Vázquez[1]

[1] *University of Castilla-La Mancha, Faculty of Computer Science Engineering*
02071, Albacete (Spain),

emails: {Gines.Moreno,Jaime.Penabad,Carlos.Vazquez}@uclm.es

Abstract

Multi-adjoint logic programming, MALP in brief, represents a modern and powerful approach for fuzzifying pure logic programming, by dealing with truth degrees and connectives collected from lattices more complex than the trivial one based on two simple values $true$ and $false$. Each MALP program is associated with its own multi-adjoint lattice, equipped with a complete or partial ordering among the (possibly infinite set of) elements and a wide set of fuzzy connectives where it is crucial to connect each implication symbol with a proper conjunction thus conforming constructs of the form $(\leftarrow_i, \&_i)$. Initially used for modeling a fuzzy variant of the classical *modus ponens* inference rule, the use of the so-called *adjoint pairs* strongly affects the definitions of both declarative and operational semantics (as well as their corresponding soundness and completeness properties) of the MALP framework. In this work we relax this impact in two stages. Firstly, we show a sub-class of MALP programs for which it is possible to define simpler versions of models and computational steps non depending on adjoint pairs. Next, and what is the best, we show a very simple technique able to map every MALP program with other one in this sub-class, thus moving the need for using adjoint pairs from the semantic core of MALP to a purely syntactic pre-process.

Key words: Multi-adjoint Logic Programming, Adjoint Pairs
MSC 2000: Multi-adjoint Lattice

1 Introduction

During the last years, our research group has provided both theoretical and practical advances in the design of declarative languages, applications and tools with a fuzzy taste (visit

$\&_{\mathtt{P}}(x,y)$	$\triangleq$	$x * y$	$\leftarrow_{\mathtt{P}}(x,y)$	$\triangleq$	$\min(1, x/y)$	*Product*
$\&_{\mathtt{G}}(x,y)$	$\triangleq$	$\min(x,y)$	$\leftarrow_{\mathtt{G}}(x,y)$	$\triangleq$	$\begin{cases} 1 & \text{if } y \leq x \\ x & \text{otherwise} \end{cases}$	*Gödel*
$\&_{\mathtt{L}}(x,y)$	$\triangleq$	$\max(0, x+y-1)$	$\leftarrow_{\mathtt{L}}(x,y)$	$\triangleq$	$\min\{x-y+1, 1\}$	*Łukasiewicz*

Figure 1: Adjoint pairs in $[0,1]$ for *Łukasiewicz*, *Gödel* and *product* fuzzy logics

`http://dectau.uclm.es/floper/` and `http://dectau.uclm.es/fuzzyXPath/`) as shown in [3, 9, 10, 8, 1], where we focus on the multi-adjoint framework.

In essence, the notion of multi-adjoint lattice considers a *carrier* set L (whose elements verify a concrete ordering $\leq$) equipped with a set of connectives like implications, conjunctions, disjunctions and other *hybrid operators* (not always belonging to an standard taxonomy) with the particularity that for each implication symbol there exists its adjoint conjunction used for modeling the *modus ponens* inference rule in a fuzzy logic setting. For instance, some adjoint pairs -i.e. conjunctors and implications- in the lattice $([0,1], \leq)$ are presented in Figure 1, where labels `L`, `G` and `P` mean respectively *Łukasiewicz logic*, *Gödel logic* and *product logic* (with different capabilities for modeling *pessimist*, *optimist* and *realistic scenarios*, respectively).

Definition 1.1. *Let $(L, \leq)$ be a lattice. A* multi-adjoint lattice *is a tuple $(L, \leq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ such that:*

i) $(L, \leq)$ *is a complete lattice, namely,* $\forall S \subset L, S \neq \emptyset, \exists\ inf(S), sup(S)$[1].

ii) $(\&_i, \leftarrow_i)$ *is an* adjoint pair *in* $(L, \leq)$*, i.e.:*

1) $\&_i$ *is non-decreasing in both arguments, for all* i, $i = 1, \ldots, n$.

2) $\leftarrow_i$ *is non-decreasing in the first argument and non-increasing in the second one.*

3) $x \leq (y \leftarrow_i z)$ *if and only if* $(x \&_i z) \leq y$*, for any* $x, y, z \in L$ *(adjoint property)*[2].

iii) $\top \&_i v = v \&_i \top = v$ *for all* $v \in L, i = 1, \ldots, n$*, where* $\top = sup(L)$.

In what follows, we present a short summary of the main features of the MALP language (we refer the reader to [7, 5, 6] for a complete formulation). We work with a first order language, $\mathcal{L}$, containing variables, function symbols, predicate symbols, constants, quantifiers ($\forall$ and $\exists$), and several (arbitrary) connectives to increase language expressiveness. In our fuzzy setting, we use implication connectives $(\leftarrow_1, \leftarrow_2, \ldots, \leftarrow_m)$ and also other connectives

[1] Then, it is a bounded lattice, i.e. it has bottom and top elements, denoted by $\bot$ and $\top$, respectively.

[2] This condition is the most important feature of the framework.

 ISBN: 978-84-616-2723-3

which are grouped under the name of *aggregators* or *aggregation operators*. They are used to combine/propagate truth values through the rules. The general definition of aggregation operators subsumes conjunctive operators (denoted by $\&_1, \&_2, \ldots, \&_k$), disjunctive operators ($\vee_1, \vee_2, \ldots, \vee_l$), and average and hybrid operators (usually denoted by $@_1, @_2, \ldots, @_n$). Although the connectives $\&_i$, $\vee_i$ and $@_i$ are binary operators, we usually generalize them as functions with an arbitrary number of arguments. By definition, the truth function for an n-ary aggregation operator $[\![@]\!] : L^n \to L$ is required to be monotone and fulfills $[\![@]\!](\top, \ldots, \top) = \top$, $[\![@]\!](\bot, \ldots, \bot) = \bot$. Additionally, our language $\mathcal{L}$ contains the elements of a multi-adjoint lattice, $(L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$, equipped with a collection of adjoint pairs $(\leftarrow_i, \&_i)$, where each $\&_i$ is a conjunctor intended to the evaluation of *modus ponens*. In general, the set of truth values L may be the carrier of any complete lattice.

A *rule* is a formula $H \leftarrow_i \mathcal{B}$, where H is an atomic formula (usually called the *head*) and $\mathcal{B}$ (which is called the *body*) is a formula built from atomic formulas $B_1, \ldots, B_n$ ($n \geq 0$), truth values of L and conjunctions, disjunctions and aggregations. Rules with an empty body are called *facts*. A *goal* is a body submitted as a query to the system. Variables in a rule are assumed to be governed by universal quantifiers. Roughly speaking, a MALP program is a set of pairs $\langle \mathcal{R}; v \rangle$, where $\mathcal{R}$ is a rule and v is a *truth degree* (a value of L) expressing the confidence which the user of the system has in the truth of the rule $\mathcal{R}$.

Both operational and declarative semantics of this fuzzy framework are presented in Section 2, whose strong dependence of adjoint pairs is largely weakened for an interesting sub-class of MALP programs in Section 3. Before concluding in Section 5, we provide in Section 4 a mapping which links any MALP program with its corresponding one into this subset, thus deviating the need for using adjoint pairs to just a simple, purely syntactic and semantics-preserving pre-process.

2 Operational and Declarative Semantics of MALP

In order to describe the procedural semantics of the MALP language, in the following we denote by $\mathcal{C}[A]$ a formula where A is a sub-expression (usually an atom) which occurs in the –possibly empty– context $\mathcal{C}[]$ whereas $\mathcal{C}[A/A']$ means the replacement of A by A' in context $\mathcal{C}[]$. Moreover, $\mathcal{V}ar(s)$ denotes the set of distinct variables occurring in the syntactic object s, $\theta[\mathcal{V}ar(s)]$ refers to the substitution obtained from θ by restricting its domain to $\mathcal{V}ar(s)$ and $mgu(E)$ denotes the *most general unifier* of a set of expressions E. In the next definition, we always consider that A is the selected atom in goal $\mathcal{Q}$ and L is the multi-adjoint lattice associated to $\mathcal{P}$.

Definition 2.1 (Admissible Step)**.** *Let $\mathcal{Q}$ be a goal and let σ be a substitution. The pair $\langle \mathcal{Q}; \sigma \rangle$ is a* state*. Given a program $\mathcal{P}$, an* admissible computation *is formalized as a state transition system, whose transition relation $\overset{AS}{\rightsquigarrow}$ is the smallest relation satisfying the following* admissible rules*:*

 ISBN: 978-84-616-2723-3

1) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS}{\rightsquigarrow} \langle (\mathcal{Q}[A/v \&_i \mathcal{B}])\theta; \sigma\theta \rangle$ *if* $\theta = mgu(\{H = A\})$, $\langle H \leftarrow_i \mathcal{B}; v \rangle$ *in* $\mathcal{P}$ *and* $\mathcal{B}$ *is not empty.*

2) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS}{\rightsquigarrow} \langle (\mathcal{Q}[A/v])\theta; \sigma\theta \rangle$ *if* $\theta = mgu(\{H = A\})$, *and* $\langle H \leftarrow_i; v \rangle$ *in* $\mathcal{P}$.

3) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS}{\rightsquigarrow} \langle (\mathcal{Q}[A/\bot]); \sigma \rangle$ *if there is no rule in* $\mathcal{P}$ *whose head unifies with* A *(this case copes with possible unsuccessful branches).*

An *admissible derivation* is a sequence $\langle \mathcal{Q}; id \rangle \overset{\text{AS}}{\rightsquigarrow}{}^* \langle \mathcal{Q}'; \theta \rangle$. As usual, rules are taken renamed apart. We shall use the symbols $\overset{\text{AS1}}{\rightsquigarrow}$, $\overset{\text{AS2}}{\rightsquigarrow}$ and $\overset{\text{AS3}}{\rightsquigarrow}$ to distinguish between computation steps performed by applying one of the specific admissible rules. The application of a rule on a step will be annotated as a superscript of the $\overset{\text{AS}}{\rightsquigarrow}$ symbol.

If we exploit all atoms of a given goal, by applying admissible steps as much as needed during the operational phase, then it becomes a formula with no atoms (a L-expression) which can be then interpreted w.r.t. lattice L as follows.

Definition 2.2 (Interpretive Step and Fuzzy Computed Answer)**.** *Let* $\mathcal{P}$ *be a program,* $\mathcal{Q}$ *a goal and* σ *a substitution. Assume that* $[\![@]\!]$ *is the truth function of connective* @ *in the lattice* $(L, \leq)$ *associated to* $\mathcal{P}$*, such that, for values* $r_1, \ldots, r_n, r_{n+1} \in L$*, we have that* $[\![@]\!](r_1, \ldots, r_n) = r_{n+1}$*. Then, we formalize the notion of* interpretive computation *as a state transition system, whose transition relation* $\overset{IS}{\rightsquigarrow}$ *is defined as the least one satisfying:*

$$\langle \mathcal{Q}[@(r_1, \ldots, r_n)]; \sigma \rangle \quad \overset{IS}{\rightsquigarrow} \quad \langle \mathcal{Q}[@(r_1, \ldots, r_n)/r_{n+1}]; \sigma \rangle$$

An interpretive derivation *is a sequence* $\langle \mathcal{Q}; \sigma \rangle \overset{IS}{\rightsquigarrow} \cdots \overset{IS}{\rightsquigarrow} \langle \mathcal{Q}'; \sigma \rangle$*. When* $\mathcal{Q}' = r \in L$*, the state* $\langle r; \sigma \rangle$ *is called a* fuzzy computed answer *(f.c.a.) for that derivation.*

Moreover, in the MALP framework [7, 5, 6], each program has its own associated multi-adjoint lattice and each program rule is "weighted" with an element of L, whereas the components in its body (i.e., atoms and elements of L) are *linked* with connectives of the lattice.

We formally introduce now the semantic notions of Herbrand interpretation and Herbrand model, or directly, interpretation and model for short, for a MALP program $\mathcal{P}$, in a similar way to [7] and [3].

Definition 2.3 (Herbrand Interpretation)**.** *A Herbrand interpretation is a map* $\mathcal{I} : B_\mathcal{P} \to L$*, where* $B_\mathcal{P}$ *is the Herbrand base of the* MALP *program* $\mathcal{P}$ *and* $(L, \leq)$ *is the multi-adjoint lattice associated to program* $\mathcal{P}$.

$\mathcal{I}$ is extended in a natural way to the set of ground formulae of the language. In order to interpret a non ground formula A (closed, and universally quantified in the case of the

 ISBN: 978-84-616-2723-3

MALP language), it suffices to take $\mathcal{I}(A) = inf\{\mathcal{I}(A\xi) : A\xi \text{ is a ground instance of A}\}$. Let $\mathcal{H}$ be the set of interpretations whose order is induced from the order of L:

$$\mathcal{I}_j \leq \mathcal{I}_k \Longleftrightarrow \mathcal{I}_j(F) \leq \mathcal{I}_k(F), \forall F \in B_{\mathcal{P}}$$

It is trivial to check that $(\mathcal{H}, \leq)$ inherits the structure of complete lattice from the multi-adjoint lattice $(L, \leq)$.

Definition 2.4 (Herbrand Model)**.** *An interpretation $\mathcal{I}$ satisfies (or is* model *of) a rule $\langle H \leftarrow_i \mathcal{B}; \alpha_i \rangle$ if, and only if, $v \leq \mathcal{I}(H \leftarrow_i \mathcal{B})$. An interpretation $\mathcal{I}$ is a Herbrand model of $\mathcal{P}$ if, and only if, all rules in $\mathcal{P}$ are satisfied by $\mathcal{I}$.*

In [3] we have defined, for the first time in the literature using model theory, a declarative semantics for multi-adjoint logic programming in terms of the least fuzzy Herbrand model. This construction reproduces, in our fuzzy context, the classic construction of least Herbrand model of pure logic programming [4, 2], which has been traditionally accepted as the declarative semantics of logic programs.

In the last years, other adaptations of this concept have been provided, using model theory too ([13, 11, 12]), in alternative fuzzy logic programming frameworks different from MALP.

Furthermore, in [3] we have related our notion of least fuzzy model with the already existing procedural semantics and fix-point semantics, and we have given revealing examples in which our declarative semantics has still sense beyond the multi-adjoint case, while the previously mentioned ones remain undefined.

Definition 2.5 (Least Fuzzy Herbrand Model)**.** *Let $\mathcal{P}$ be a* MALP *program with associated lattice $(L, \leq)$. The interpretation $\mathcal{I}_{\mathcal{P}} = inf\{\mathcal{I}_j : \mathcal{I}_j \text{ is model of } \mathcal{P}\}$ is called least fuzzy Herbrand*[3] *model of $\mathcal{P}$.*

The following result justifies that the previous interpretation $\mathcal{I}_{\mathcal{P}}$ can be thought really as the least fuzzy Herbrand model.

Theorem 2.6 ([3])**.** *Let $\mathcal{P}$ be a* MALP *program with associated lattice L. The map $\mathcal{I}_{\mathcal{P}} = inf\{\mathcal{I}_j : \mathcal{I}_j \text{ is a model of } \mathcal{P}\}$ is the least model of $\mathcal{P}$.*

In the proof of this result provided in [3], it is essential that the lattice associated to the program be a multi-adjoint lattice. In this reference it is possible to contrast the necessity of this hypothesis for Theorem 2.6.

In Figure 2 we illustrate most definitions presented in this section, where we wish to remark that:

[3]Sometimes we will say only least fuzzy model or least model.

 ISBN: 978-84-616-2723-3

- In the first rule of $\mathcal{P}$, we mix connectives belonging to three different fuzzy logics, whose truth functions appear in Figure 1, having too that $|_{\mathtt{L}}(x,y) \triangleq min(x+y,1)$.
- Note that since there are no rules defining predicate s, the last step in the admissible derivation reduces $s(Y_2)$ to 0 by applying an $\overset{\text{AS3}}{\rightsquigarrow}$ step, which contrasts with crisp logic languages such as Prolog which would abort the whole derivation. Hence, in our fuzzy setting we can reach computed answers (at the end of the interpretive phase) even in the presence of non defined predicates.
- When describing the least fuzzy Herbrand model of $\mathcal{P}$, we omit those elements of the Herbrand Base interpreted as 0.

3 A MALP Sub-class non Depending on Adjoint Pairs

From now on, we call $\mathcal{M}_\top$ to the set of MALP programs whose rules are always labeled with the top element $\top$ of their associated lattices. We now speak about $\top$-programs, $\top$-rules and so on. For executing these programs, we can conceive the following operational semantics which is simpler than the one seen in the previous section (see Definition 2.1).

Definition 3.1 ($\top$-Admissible Step)**.** *Let $\mathcal{Q}$ be a goal and let σ be a substitution. The pair $\langle \mathcal{Q}; \sigma \rangle$ is a* state*. Given a $\top$-program $\mathcal{P} \in \mathcal{M}_\top$, a* $\top$-admissible computation *is formalized as a state transition system, whose transition relation $\overset{AS^\top}{\rightsquigarrow}$ is the smallest relation satisfying the following* $\top$-admissible rules*:*

1) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/\mathcal{B}])\theta; \sigma\theta \rangle$ *if $\theta = mgu(\{H = A\})$, $\langle H \leftarrow_i \mathcal{B}; \top \rangle$ in $\mathcal{P}$ and $\mathcal{B}$ is not empty.*

2) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/\top])\theta; \sigma\theta \rangle$ *if $\theta = mgu(\{H = A\})$, and $\langle H \leftarrow_i; \top \rangle$ in $\mathcal{P}$.*

3) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/\bot]); \sigma \rangle$ *if there is no $\top$-rule in $\mathcal{P}$ whose head unifies with A (this case copes with possible unsuccessful branches).*

A $\top$-*admissible derivation* is a sequence $\langle \mathcal{Q}; id \rangle \overset{\text{AS}^\top}{\rightsquigarrow} {}^*\langle \mathcal{Q}'; \theta \rangle$. We shall use the symbols $\overset{\text{AS1}^\top}{\rightsquigarrow}$, $\overset{\text{AS2}^\top}{\rightsquigarrow}$ and $\overset{\text{AS3}^\top}{\rightsquigarrow}$ to distinguish between computation steps performed by applying one of the specific admissible rules. Note that this definition (which is very close to the classical one of SLD-resolution used in Prolog) differs for Definition 2.1 just in the first case, since it does not make use on states of the $\&_i$ conjunction adjoint to the $\leftarrow_i$ implication symbol of $\top$-rules.

 ISBN: 978-84-616-2723-3

Multi-adjoint logic program:

$$\mathcal{P} = \begin{cases} \mathcal{R}_1: & \langle p(X) & \leftarrow_{\mathtt{P}} & q(X,Y)\ \&_{\mathtt{G}}\ (r(Y)\ |_{\mathtt{L}}\ s(Y)) & ; & 0.8\rangle \\ \mathcal{R}_2: & \langle q(a,Y) & \leftarrow & & ; & 0.9\rangle \\ \mathcal{R}_3: & \langle r(b) & \leftarrow & & ; & 1\rangle \end{cases}$$

Admissible derivation:

$$\begin{array}{ll} \langle \underline{p(X)}; id\rangle & \overset{\text{AS1}}{\rightsquigarrow}\ \mathcal{R}_1 \\ \langle 0.8\ \&_{\mathtt{P}}\ (\underline{q(X_1,Y_1)}\ \&_{\mathtt{G}}\ (r(Y1)\ |_{\mathtt{L}}\ s(Y1)));\{X/X_1\}\rangle & \overset{\text{AS2}}{\rightsquigarrow}\ \mathcal{R}_2 \\ \langle 0.8\ \&_{\mathtt{P}}\ (0.9\ \&_{\mathtt{G}}\ (\underline{r(Y2)}\ |_{\mathtt{L}}\ s(Y2)));\{X/a, X_1/a, Y_1/Y_2\}\rangle & \overset{\text{AS2}}{\rightsquigarrow}\ \mathcal{R}_3 \\ \langle 0.8\ \&_{\mathtt{P}}\ (0.9\ \&_{\mathtt{G}}\ (1\ |_{\mathtt{L}}\ \underline{s(Y2)}));\{X/a, X_1/a, Y_1/b, Y_2/b\}\rangle & \overset{\text{AS3}}{\rightsquigarrow} \\ \langle 0.8\ \&_{\mathtt{P}}\ (0.9\ \&_{\mathtt{G}}\ (1\ |_{\mathtt{L}}\ \underline{0}));\{X/a, X_1/a, Y_1/b, Y_2/b\}\rangle & \end{array}$$

Interpretive derivation:

$$\begin{array}{ll} \langle 0.8\ \&_{\mathtt{P}}\ (0.9\ \&_{\mathtt{G}}\ \underline{(1\ |_{\mathtt{L}}\ 0)});\{X/a\}\rangle & \overset{\text{IS}}{\rightsquigarrow} \\ \langle 0.8\ \&_{\mathtt{P}}\ \underline{(0.9\ \&_{\mathtt{G}}\ 1)};\{X/a\}\rangle & \overset{\text{IS}}{\rightsquigarrow} \\ \langle \underline{0.8\ \&_{\mathtt{P}}\ 0.9};\{X/a\}\rangle & \overset{\text{IS}}{\rightsquigarrow} \\ \langle 0.72;\{X/a\}\rangle & \end{array}$$

— f.c.a. means "*p(X)* is proved with truth degree 0.72 when $X = a$".

Least fuzzy Herbrand model:

$$\mathcal{I}_{\mathcal{P}} : \mathcal{B}_{\mathcal{P}} \to [0,1] \text{ s.t. } \mathcal{I}_{\mathcal{P}}(p(a)) = 0.72,\ \mathcal{I}_{\mathcal{P}}(q(a,a)) = \mathcal{I}_{\mathcal{P}}(q(a,b)) = 0.9,\ \text{and } \mathcal{I}_{\mathcal{P}}(r(b)) = 1.$$

Figure 2: Illustrative examples of MALP syntax and semantics

The following result establishes that, when dealing with $\top$-programs, the derivations built with admissible (together with subsequent interpretive) steps as well as those ones based on $\top$-admissible (and interpretive) steps, lead to the same set of fuzzy computed answers.

Theorem 3.2. *Let $\mathcal{P} \in \mathcal{M}_\top$ be a* MALP *$\top$-program with associated lattice L, $\mathcal{Q}$ a goal, σ a substitution and $v \in L$. Then,*

$$\langle \mathcal{Q}; id\rangle \overset{AS}{\rightsquigarrow}{}^* \cdots \overset{IS}{\rightsquigarrow}{}^* \langle v;\sigma\rangle \text{ w.r.t. } \mathcal{P} \quad \textit{if and only if} \quad \langle \mathcal{Q}; id\rangle \overset{AS^\top}{\rightsquigarrow}{}^* \cdots \overset{IS}{\rightsquigarrow}{}^* \langle v;\sigma\rangle \text{ w.r.t. } \mathcal{P}.$$

Proof. In our our proof we simply need to show that the effects produced by $\overset{\text{AS}}{\rightsquigarrow}$ steps on a generic state of the form $\langle \mathcal{Q}[A];\sigma\rangle$, are replicated by $\overset{\text{AS}^\top}{\rightsquigarrow}$ steps and viceversa. We consider three different cases:

 ISBN: 978-84-616-2723-3

1) If $\langle H \leftarrow_i \mathcal{B}; \top\rangle \in \mathcal{P}$, where $\mathcal{B}$ is not empty and $\theta = mgu(\{H = A\})$, then $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS1}}{\rightsquigarrow} \langle(\mathcal{Q}[A/\top \&_i \mathcal{B}])\theta; \sigma\theta\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS1}^\top}{\rightsquigarrow} \langle(\mathcal{Q}[A/\mathcal{B}])\theta; \sigma\theta\rangle$, since $\top \&_i \mathcal{B} \equiv \mathcal{B}$ and hence $(\mathcal{Q}[A/\top \&_i \mathcal{B}])\theta \equiv (\mathcal{Q}[A/\mathcal{B}])\theta$.

2) If $\langle H \leftarrow_i; \top\rangle \in \mathcal{P}$, where $\theta = mgu(\{H = A\})$, then $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS2}}{\rightsquigarrow} \langle(\mathcal{Q}[A/\top])\theta; \sigma\theta\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS2}^\top}{\rightsquigarrow} \langle(\mathcal{Q}[A/\top])\theta; \sigma\theta\rangle$.

3) If there is no $\top$-rule in $\mathcal{P}$ whose head unifies with A then, $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS3}}{\rightsquigarrow} \langle(\mathcal{Q}[A/\bot]); \sigma\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS3}^\top}{\rightsquigarrow} \langle(\mathcal{Q}[A/\bot]); \sigma\rangle$.

□

Let us continue now with declarative semantics aspects related with $\top$-programs.

Definition 3.3 ($\top$-model)**.** *An interpretation $\mathcal{I}$ $\top$-satisfies (or is* $\top$-model *of) a $\top$-rule $\langle H \leftarrow_i \mathcal{B}; \top\rangle$ if, and only if, $\mathcal{I}(\mathcal{B}) \leq \mathcal{I}(H)$. An interpretation $\mathcal{I}$ is $\top$-model of a $\top$-program $\mathcal{P}$ if, and only if, all $\top$-rules in $\mathcal{P}$ are $\top$-satisfied by $\mathcal{I}$.*

Definition 3.4 (Least Fuzzy Herbrand $\top$-Model)**.** *Let $\mathcal{P} \in \mathcal{M}_\top$ be a* MALP $\top$*-program with associated lattice $(L, \leq)$. The interpretation $\mathcal{I}_\mathcal{P}^\top = inf\{\mathcal{I}_j : \mathcal{I}_j$ is $\top$-model of $\mathcal{P}\}$ is the least fuzzy Herbrand $\top$-model of $\mathcal{P}$.*

In the following result we prove that the notion of least fuzzy Herbrand model of a given MALP $\top$-program is just the same construct than its least fuzzy Herbrand $\top$-model.

Theorem 3.5. *The least fuzzy Herbrand model of a* MALP $\top$*-program $\mathcal{P} \in \mathcal{M}_\top$ coincides with its least fuzzy Herbrand $\top$-model, that is, $\mathcal{I}_\mathcal{P} = \mathcal{I}_\mathcal{P}^\top$.*

Proof. Consider a generic $\top$-rule $\mathcal{R} : \langle H \leftarrow_i \mathcal{B}; \top\rangle \in \mathcal{P}$ and an interpretation $\mathcal{I}$. In order to prove that $\mathcal{I}$ satisfies $\mathcal{R}$ if and only if it $\top$-satisfies $\mathcal{R}$, it suffices by showing that $\top \leq \mathcal{I}(H \leftarrow_i \mathcal{B})$ becomes into $\mathcal{I}(\top \&_i \mathcal{B}) \leq \mathcal{I}(H)$ thanks to the adjoint property, and then this expression can be simplified to $\mathcal{I}(\mathcal{B}) \leq \mathcal{I}(H)$ (since obviousy $\top \&_i v = v$, for any $v \in L$), as desired. So, since the set of models for a given $\top$-program $\mathcal{P}$ is the same as its set of $\top$-models, the infimmun of such set is just $\mathcal{I}_\mathcal{P} = \mathcal{I}_\mathcal{P}^\top$, which concludes our proof. □

It is noteworthy that, in the previous definitions related with $\top$-models and $\mathcal{I}_\mathcal{P}^\top$, we don't require that the lattice associated to MALP $\top$-programs be a multi-adjoint lattice (in fact, we never use adjoint pairs), as occurred too when defining the new operational semantics for $\top$-programs. For this reason, from now on we can simplify the syntax of $\top$-rules, by removing the label of their implication symbols as well as their weights (or associated truth degrees), i.e., instead of $\langle H \leftarrow_i \mathcal{B}; \top\rangle$ we will simply write $H \leftarrow \mathcal{B}$.

 ISBN: 978-84-616-2723-3

Multi-adjoint logic $\top$-program:

$$\mathcal{P}' = \mathcal{P}^{\mathcal{M}_\top} = \begin{cases} \mathcal{R}'_1: & p(X) & \leftarrow & 0.8 \;\&_{\mathtt{P}}\; (q(X,Y) \;\&_{\mathtt{G}}\; (r(Y) \;|_{\mathtt{L}}\; s(Y)) \\ \mathcal{R}'_2: & q(a,Y) & \leftarrow & 0.9 \\ \mathcal{R}'_3: & r(b) & \leftarrow & \end{cases}$$

$\top$-Admissible derivation:

$$\begin{array}{ll} \langle \underline{p(X)}; id\rangle & \overset{\text{AS1}^\top}{\rightsquigarrow}\ \mathcal{R}'_1 \\ \langle 0.8 \;\&_{\mathtt{P}}\; (\underline{q(X_1,Y_1)} \;\&_{\mathtt{G}}\; (r(Y1) \;|_{\mathtt{L}}\; s(Y1)));\{X/X_1\}\rangle & \overset{\text{AS1}^\top}{\rightsquigarrow}\ \mathcal{R}'_2 \\ \langle 0.8 \;\&_{\mathtt{P}}\; (0.9 \;\&_{\mathtt{G}}\; (\underline{r(Y2)} \;|_{\mathtt{L}}\; s(Y2)));\{X/a, X_1/a, Y_1/Y_2\}\rangle & \overset{\text{AS2}^\top}{\rightsquigarrow}\ \mathcal{R}'_3 \\ \langle 0.8 \;\&_{\mathtt{P}}\; (0.9 \;\&_{\mathtt{G}}\; (1 \;|_{\mathtt{L}}\; \underline{s(Y2)}));\{X/a, X_1/a, Y_1/b, Y_2/b\}\rangle & \overset{\text{AS3}^\top}{\rightsquigarrow} \\ \langle 0.8 \;\&_{\mathtt{P}}\; (0.9 \;\&_{\mathtt{G}}\; (1 \;|_{\mathtt{L}}\; 0));\{X/a, X_1/a, Y_1/b, Y_2/b\}\rangle & \end{array}$$

Figure 3: Illustrative examples of concepts defined in Sections 3 and 4

4 A Mapping from MALP Programs to $\mathcal{M}_\top$

The following definition represents a very simple, purely syntactic pre-process which, by making use of adjoint pairs, is able to link MALP programs with $\top$-programs.

Definition 4.1. *We define a mapping that associates to each* MALP *program $\mathcal{P}$ a $\top$-program in $\mathcal{M}_\top$ with the following shape:*

$$\mathcal{P}^{\mathcal{M}_\top} = \{\mathcal{R}^{\mathcal{M}_\top} \;:\; \mathcal{R} \in \mathcal{P}\}$$

where for each $\top$-rule $\mathcal{R}: \langle H \leftarrow_i \mathcal{B}; v\rangle \in \mathcal{P}$, the mapping is defined too as:

$$\mathcal{R}^{\mathcal{M}_\top} \;=\; \begin{cases} H \leftarrow v \&_i \mathcal{B} & \text{if} \quad v \neq \top \\ H \leftarrow \mathcal{B} & \text{otherwise} \end{cases}$$

In Figure 3 we illustrate this definition as well as other concepts introduced in the previous section. Note that:

- The $\top$-program $\mathcal{P}'$ coincides with the transformation of the MALP program $\mathcal{P}$ seen in Figure 2, that is $\mathcal{P}' = \mathcal{P}^{\mathcal{M}_\top}$, since $\mathcal{R}'_1 = \mathcal{R}_1^{\mathcal{M}_\top}$, $\mathcal{R}'_2 = \mathcal{R}_2^{\mathcal{M}_\top}$ and $\mathcal{R}'_3 = \mathcal{R}_3^{\mathcal{M}_\top}$. In this last case, we have simply removed the weight of the rule (since it is just 1, i.e., the top element of lattice $[0,1]$) and both $\mathcal{R}_1$ and $\mathcal{R}'_1$ are facts in $\mathcal{P}$ and $\mathcal{P}'$, respectively.

 ISBN: 978-84-616-2723-3

- On the other hand, note that even when $\mathcal{R}_2 \in \mathcal{P}$ is a fact, $\mathcal{R}'_2 \in \mathcal{P}'$ is not a fact, since its body is not empty (it is composed by just a truth degree). For this reason, while the second admissible step of the admissible derivation in Figure 2 is of kind $\overset{\text{AS2}}{\rightsquigarrow}$, the corresponding second $\top$-admissible step of the $\top$-admissible derivation in Figure 3 is not a $\overset{\text{AS2}^\top}{\rightsquigarrow}$ but a $\overset{\text{AS1}^\top}{\rightsquigarrow}$ step.
- The sequence of states in the admissible derivation of Figure 2 coincides with the sequence of states in the $\top$-admissible of Figure 3, and after applying exactly the same sequence of interpretive steps drawn in Figure 2 (for this reason we have omitted it in Figure 3), the same fuzzy computed answer is reached.
- Note that even when the notion of $\top$-model[4] is less involved than the one of model, the least fuzzy Herbrand $\top$-model of $\mathcal{P}'$ coincides with $\mathcal{I}_\mathcal{P}$ in Figure 2, as wanted.

The following result establishes that derivations built with admissible (together with subsequent interpretive) steps lead to the same set of fuzzy computed answers than those ones based on $\top$-admissible (and interpretive) steps when dealing with $\top$-programs obtained from previous MALP programs after being transformed according Definition 4.1.

Theorem 4.2. *Let $\mathcal{P}$ be a* MALP *program with associated lattice L, $\mathcal{Q}$ a goal, σ a substitution and $v \in L$. Then,*

$\langle \mathcal{Q}; id\rangle \overset{AS}{\rightsquigarrow} * \cdots \overset{IS}{\rightsquigarrow} * \langle v; \sigma\rangle$ *w.r.t.* $\mathcal{P}$ *if and only if* $\langle \mathcal{Q}; id\rangle \overset{AS^\top}{\rightsquigarrow} * \cdots \overset{IS}{\rightsquigarrow} * \langle v; \sigma\rangle$ *w.r.t.* $\mathcal{P}^{\mathcal{M}_\top}$.

Proof. We distinguish four cases for showing that the effects produced by $\overset{\text{AS}}{\rightsquigarrow}$ steps on a generic state of the form $\langle \mathcal{Q}[A]; \sigma\rangle$, are replicated by $\overset{\text{AS}^\top}{\rightsquigarrow}$ steps and viceversa.

1) Note that $\langle H \leftarrow_i \mathcal{B}; v\rangle \in \mathcal{P}$, where $\mathcal{B}$ is not empty and $\theta = mgu(\{H = A\})$, if and only if $\langle H \leftarrow v \&_i \mathcal{B}\rangle \in \mathcal{P}^{\mathcal{M}_\top}$, and hence $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS1}}{\rightsquigarrow} \langle (\mathcal{Q}[A/v\&_i\mathcal{B}])\theta; \sigma\theta\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS1}^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/v\&_i\mathcal{B}])\theta; \sigma\theta\rangle$.

2) Now, $\langle H \leftarrow; v\rangle \in \mathcal{P}$, where $v \neq \top$ and $\theta = mgu(\{H = A\})$, if and only if $\langle H \leftarrow v\rangle \in \mathcal{P}^{\mathcal{M}_\top}$, and hence $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS2}}{\rightsquigarrow} \langle (\mathcal{Q}[A/v])\theta; \sigma\theta\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS1}^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/v])\theta; \sigma\theta\rangle$ (note this particular correspondence between $\overset{\text{AS2}}{\rightsquigarrow}$ and $\overset{\text{AS1}^\top}{\rightsquigarrow}$ steps).

3) Observe that $\langle H \leftarrow; \top\rangle \in \mathcal{P}$, where $\theta = mgu(\{H = A\})$, if and only if $\langle H \leftarrow\rangle \in \mathcal{P}^{\mathcal{M}_\top}$, and hence $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS2}}{\rightsquigarrow} \langle (\mathcal{Q}[A/\top])\theta; \sigma\theta\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS2}^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/\top])\theta; \sigma\theta\rangle$ (now we have shown the equivalence between $\overset{\text{AS2}}{\rightsquigarrow}$ and $\overset{\text{AS2}^\top}{\rightsquigarrow}$ steps).

[4]This concept does not make use of adjoint pairs and weights of program rules.

 ISBN: 978-84-616-2723-3

4) Finally, there is no rule in $\mathcal{P}$ whose head unifies with A if and only if there is no rule in $\mathcal{P}^{\mathcal{M}_\top}$ whose head unifies with A and so, $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS3}}{\rightsquigarrow} \langle (\mathcal{Q}[A/\bot]); \sigma\rangle$ if and only if $\langle \mathcal{Q}[A]; \sigma\rangle \overset{\text{AS3}^\top}{\rightsquigarrow} \langle (\mathcal{Q}[A/\bot]); \sigma\rangle$.

□

In the following result we prove that the notion of least fuzzy Herbrand model of MALP programs is just the same construct than the least fuzzy Herbrand $\top$-model of those $\top$-programs obtained by applying the transformation process described in Definition 4.1.

Theorem 4.3. *The least fuzzy Herbrand model of a* MALP *program $\mathcal{P}$ coincides with the least fuzzy Herbrand $\top$-model of its associated $\top$-program $\mathcal{P}^{\mathcal{M}_\top}$, that is, $\mathcal{I}_\mathcal{P} = \mathcal{I}^\top_{\mathcal{P}^{\mathcal{M}_\top}}$.*

Proof. Consider a generic rule $\mathcal{R} : \langle H \leftarrow_i \mathcal{B}; v\rangle \in \mathcal{P}$ and correspondingly $\mathcal{R}^{\mathcal{M}_\top} : H \leftarrow v \&_i \mathcal{B} \in \mathcal{P}^{\mathcal{M}_\top}$. Assume an interpretation $\mathcal{I}$ such that $\mathcal{I}$ satisfies $\mathcal{R}$ if and only if $\mathcal{I}$ $\top$-satisfies $\mathcal{R}^{\mathcal{M}_\top}$ since by the adjoint property $v \leq \mathcal{I}(H \leftarrow_i \mathcal{B})$ if and only if $\mathcal{I}(v \&_i \mathcal{B}) \leq \mathcal{I}(H)$ and hence, the set of models of $\mathcal{P}$ coincides with the set of $\top$-models of $\mathcal{P}^{\mathcal{M}_\top}$ and, in particular, the infimmum of such set is $\mathcal{I}_\mathcal{P}$ as well as $\mathcal{I}^\top_{\mathcal{P}^{\mathcal{M}_\top}}$, as wanted. □

5 Conclusions and Future Work

The high expressive power (and even the sense of its name) of the MALP language, very often relies on the possibility of using multiple adjoint pairs when coding programs. Although we have shown that the adjoint property plays an important role when defining and proving the properties of MALP, it somehow restricts (at least under a theoretical point of view) the class of lattices for being safely used in fuzzy programs.

In this paper we have presented a semantics-preserving transformation which makes use of adjoint pairs just once in order to produce new MALP programs with a very simple shape, which will no longer depend on *adjoint constraints*, thus opening the door for future developments intended to increase the range of fuzzy logic programs beyond MALP. We are nowadays implementing the technique described so far into our $\mathcal{FLOPER}$ platform, which is freely available from `http://dectau.uclm.es/floper/`.

References

[1] J. M. Almendros-Jiménez, A. Luna, and G. Moreno. A Flexible XPath-based Query Language Implemented with Fuzzy Logic Programming. In N. Bassiliades et al., editor, *Proc. of 5th International Symposium on Rules: Research Based, Industry Focused, RuleML'11*, pages 186–193. Lecture Notes in Computer Science 6826, 2011.

 ISBN: 978-84-616-2723-3

[2] K. R. Apt. *From Logic Programming to Prolog.* International Series in Computer Science, Prentice Hall, 1997.

[3] P. Julián, G. Moreno, and J. Penabad. On the declarative semantics of multi-adjoint logic programs. In *Proc. of the 10th International Work-Conference on Artificial Neural Networks, IWANN'09*, pages 253–260. Lecture Notes in Computer Science 5517, 2009.

[4] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, Berlin, 1987.

[5] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. *Proc. of Logic Programming and Non-Monotonic Reasoning, LPNMR'01, Lecture Notes in Artificial Intelligence*, 2173:351–364, 2001.

[6] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. A procedural semantics for multi-adjoint logic programing. *Progress in Artificial Intelligence, EPIA'01, Lecture Notes in Artificial Intelligence*, 2258(1):290–297, 2001.

[7] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Similarity-based Unification: a multi-adjoint approach. *Fuzzy Sets and Systems*, 146:43–62, 2004.

[8] P. J. Morcillo, G. Moreno, J. Penabad, and C. Vázquez. A Practical Management of Fuzzy Truth Degrees using FLOPER. In M. Dean et al., editor, *Proc. of 4th International Symposium on Rule Interchange and Applications, RuleML'10*, pages 119–126. Lecture Notes in Computer Science 6403, 2010.

[9] P. J. Morcillo, G. Moreno, J. Penabad, and C. Vázquez. Dedekind-MacNeille completion and cartesian product of multi-adjoint lattices. *Int. J. Computer Mathematics (extended version of a previous work in CMMSE'11)*, 89(13-14): 1742–1752, 2012.

[10] G. Moreno, J. Penabad, and C. Vázquez. On fuzzy correct answers and logical consequences in multi-adjoint logic programming. In J. Vigo-Aguiar, editor, *Proc. 12th International Conference on Mathematical Methods in Science and Engineering, CMMSE'12*, volume III, pages 864–875, 2012.

[11] M. I. Sessa. Approximate reasoning by similarity-based SLD resolution. *Fuzzy Sets and Systems*, 275:389–426, 2002.

[12] U. Straccia, M. Ojeda-Aciego, and C. V. Damásio. On fixed-points of multivalued functions on complete lattices and their application to generalized logic programs. *SIAM Journal on Computing*, 38(5):1881–1911, 2009.

[13] P. Vojtáš and L. Paulík. Soundness and completeness of non-classical extended SLD-resolution. In R. Dyckhoff et al, editor, *Proc. of ELP'96 Leipzig*, pages 289–301. Lecture Notes in Computer Science 1050, 1996.

 ISBN: 978-84-616-2723-3