ECCOMAS Thematic Conference

2nd International Conference on

Numerical and Symbolic Computation

Developments and Applications

PROCEEDINGS

26-27 March, UNIVERSIDADE DO ALGARVE

FARO, Portugal

idmec IST engenharia mecânica

APMTAC

ISBN 978-989-96264-7-8

ISBN 978-989-96264-7-8

SYMCOMP 2015 – 2nd International Conference on Numerical and Symbolic Computation: Developments and Applications

Proceedings in digital support (pen drive)

Edited by APMTAC – Associação Portuguesa de Mecânica Teórica, Aplicada e Computacional

Editors: Amélia Loja (IDMEC/IST, ISEL), Joaquim Infante Barbosa (IDMEC/IST, ISEL), José Alberto Rodrigues (CMAT/UM, ISEL)

March, 2015

1 – Introduction

The Organizing Committee of SYMCOMP2015 – 2nd International Conference on Numerical and Symbolic Computation: Developments and Applications welcomes all the participants and acknowledge the contribution of the authors to the success of this event.

This Second International Conference on Numerical and Symbolic Computation, is promoted by APMTAC - Associação Portuguesa de Mecânica Teórica, Aplicada e Computacional and it was organized in the context of IDMEC/IST - Instituto de Engenharia Mecânica. With this ECCOMAS Thematic Conference it is intended to bring together academic and scientific communities that are involved with Numerical and Symbolic Computation in the most various scientific areas

SYMCOMP 2015 elects as main goals:

To establish the state of the art and point out innovative applications and guidelines on the use of Numerical and Symbolic Computation in the numerous fields of Knowledge, such as Engineering, Physics, Mathematics, Economy and Management, Architecture, ...

To promote the exchange of experiences and ideas and the dissemination of works developed within the wide scope of Numerical and Symbolic Computation.

To encourage the participation of young researchers in scientific conferences.

To facilitate the meeting of APMTAC members (Portuguese Society for Theoretical, Applied and Computational Mechanics) and other scientific organizations members dedicated to computation, and to encourage new memberships.

We invite all participants to keep a proactive attitude and dialoguing, exchanging and promoting ideas, discussing research topics presented and looking for new ways and possible partnerships to work to develop in the future.

The Executive Committee of SYMCOMP2015 wishes to express his gratitude for the cooperation of all colleagues involved in various committees, from the Scientific Committee, Organizing Committee and the Secretariat. We hope everyone has enjoyed helping to birth this project, which we are sure will continue in the future. Our thanks to you all.

- Amélia Loja, Chairperson (IDMEC/LAETA, ADEM/ISEL)
- Ana Conceição, Co-Chairperson (CEAF/IST; DM/FCT/UAlg)
- António J. M. Ferreira (FEUP/INEGI)
- Joaquim Infante Barbosa (IDMEC/LAETA, ADEM/ISEL)
- José Alberto Rodrigues (CMAT/UM; ADM/ISEL)
- Marcín Kaminski (Technical University of Lodz, Poland)
- Stéphane Louis Clain (CMAT/UM,UM)

2 – CONFERENCE BOARD

Chairperson

Maria Amélia Ramos Loja, ADEM/ISEL ; IDMEC/LAETA

Área Departamental de Engenharia Mecânica

Instituto Superior de Engenharia de Lisboa

Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa

Email : amelialoja@dem.isel.ipl.pt, amelialoja@ist.utl.pt

Co-Chairperson

Ana Conceição, Co-Chairperson (CEAF/IST; DM/FCT/UAlg)

Center for Functional Analysis, Linear Structures and Applications (CEAFEL)

Departamento de Matemática

Faculdade de Ciências e Tecnologia da Universidade do Algarve

Campus de Gambelas 8005-139, Faro, Portugal

Email : aicdoisg@gmail.com

EXECUTIVE COMMITTEE

- Amélia Loja (IDMEC/IST, ISEL)
- Ana Conceição (CEAFEL/IST; DM/FCT/UAlg)
- Joaquim Infante Barbosa (IDMEC/IST, ISEL)
- José Alberto Rodrigues (CMAT/UM, ISEL)
- Tiago Silva (PhD student IDMEC/IST, ISEL)
- Gonçalo Bernardo (PhD student IDMEC/IST)
- Mosab Bazargani (IDMEC/IST)
- Diogo Costa (GI-MOSM/ISEL)

ORGANIZING COMMITTEE

Amélia Loja (Chair, IDMEC/IST, ISEL)

Ana Conceição (Co-Chair, UAlg)

António J. M. Ferreira (IDMEC/FEUP)

Joaquim Infante Barbosa (IDMEC/IST, ISEL)

José Alberto Rodrigues (CMAT/UM, ISEL)

Marcín Kaminski (Technical University of Lodz, Poland)
Stéphane Louis Clain (CMAT/UM,UM)

LOCAL ORGANIZING COMMITTEE

Ana Conceição (CEAFEL/IST, DM/FCT/ UAlg)

Celestino Coelho (DM/FCT/UAlg)

Rui Marreiros (CEAFEL/IST, DM/FCT/UAlg)

Susana Fernandes (DM/FCT/UAlg)

António Guerreiro (ESEC/UAlg)

José Luís Pereira (PhD student DEEI/FCT/UAlg, CEAFEL/IST, CENSE)

SCIENTIFIC COMMITTEE

Amélia Loja (IDMEC/IST, ISEL—Lisboa, Portugal)

Ana Conceição (UAlg—Algarve, Portugal)

Ana Madureira (ISEP/IPP, Porto, Portugal)

António J M Ferreira (IDMEC/FEUP— Porto, Portugal)

Carla A. F. Costa Ferreira (FCTUC—Coimbra, Portugal)

Carlos Alberto Mota Soares (IDMEC/IST—Lisboa, Portugal)

Cristóvão Manuel Mota Soares (IDMEC/IST—Lisboa, Portugal)
Elena Vásquez-Céndon (Universidad de Santiago de Compostela, Spain)
José Eugénio S. Garção (Universidade de Évora, Portugal)

Gaetano Giunta (Centre de Recherches George Tudor, Luxembourg)
Hélder C Rodrigues (IDMEC/IST— Lisboa, Portugal)

Helena Melão de Barros (FCTUC— Coimbra, Portugal)

Hoon Hong (North Carolina State University, United States of America)
J. N. Reddy (Texas A&M University, United States of America)

Joaquim Infante Barbosa (IDMEC/IST, ISEL— Lisboa, Portugal)

José Alberto Rodrigues (CMAT/UM, ISEL— Lisboa, Portugal)

José Miranda Guedes (IDMEC/IST-Lisboa, Portugal)

Józef Korbicz (University of Zielona-Góra, Poland)

Lina Vieira (ESTeSL/IPL, Lisboa, Portugal)

Luís Mateus (FA/UTL—Lisbon, Portugal)

Lorenzo Dozio (Politecnico Milano, Italy)

Marcin Kamiński (Technical University of Lodz, Poland)

Michal Bartys (Warsaw University of Technology, Poland)

Miguel Matos Neves (IDMEC/IST— Lisboa, Portugal)

Pedro Areias (Universidade de Évora, Portugal)

Paulo B. Vasconcelos (Universidade do Porto, Portugal)

Silvio Simani (Università Ferrara, Italy)
Stéphane Louis Clain (UM— Minho, Portugal)

Teresa Restivo (FEUP— Porto, Portugal)

Victor Mota Ferreira (FA/UTL, Lisbon, Portugal)

SPONSORS

ECCOMAS – European Community on Computational Methods in Applied Sciences

APMTAC – Associação Portuguesa de Mecânica Teórica, Aplicada e Computacional, ECCOMAS Member Association;

IDMEC/LAETA – Instituto de Engenharia Mecânica/Laboratório Associado de Energia, Transportes e Aeronáutica;

UAlg – Universidade do Algarve

ISEL/IPL – Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa

FCT – Fundação para a Ciência e Tecnologia (Project PTDC/ATQ/5355/2012)

TIMBERLAKE Consultores

CMF - Câmara Municipal de Faro

Wolfram Research

MapleSoft

ORGANIZING INSTITUTION

IDMEC/LAETA – Instituto de Engenharia Mecânica/Laboratório Associado de Energia, Transportes e Aeronáutica.

PLACE OF THE EVENT

Anfiteatro Paulo Freire

Complexo Pedagógico –ESEC - Universidade do Algarve

Campus da Penha

Contents

SYMCOMP 2015
Faro, March 26-27, 2015

ALGORITHMS FOR SYMBOLIC POLYNOMIALS, MATRICES AND DOMAINS

Stephen M. Watt

Department of Computer Science
University of Western Ontario
London, Canada

Abstract *This talk is about computing with mathematical quantities where the sizes or shapes are not known in advance. We will consider*

• *polynomials where the exponents can be given by symbolic expressions,*

• *matricies with blocks or other internal structure of symbolic size, and*

• *piecewise functions where the shapes of the domains are given by symbolic expressions.*

For symbolic polynomials, there are various operations we want to be able to do, such as squaring $x^{2n}-1$ *to get* $x^{4n}-2x^{2n}+1$ *, or differentiating it to get* $2nx^{2n-1}$ *.We present algorithms to compute their GCD, factorization and functional decomposition. For symbolic matrices, we show how to do arithmetic with blocks,bands and other structures where the dimensions are given by symbolic expressions. Finally, we consider the case of piecewise functions, where the regions of definition are symbolic. We show how hybrid sets, a generalization of multisets allowing negative multipiicities, can be used to reduce the computational complexity of working with these objects.*

SYMCOMP 2015
Faro, March 26-27, 2015

HIERARCHICAL OPTIMIZATION OF THE STRUCTURE AND ITS MATERIAL: APPLICATIONS IN COMPOSITE LAMINATE DESIGN

Helder C. Rodrigues

LAETA - IDMEC
IST
University of Lisbon

Abstract *This presentation deals with the problem of optimal design of mechanical structures; particularly describes computational models developed to address the problem of optimally design the mechanical structure and the material used for its production.*

Here we describe the hierarchical structural optimization model applied to the optimal design of composite laminates. The model assumes a mixed set of micro (material) and macro(structural) independent design variables, to characterize the distribution of two (or porous) materials to obtain the optimal composite microstructures at the micro design level as well as the optimal fiber orientation at the macro level.

Several examples will be presented to demonstrate and analyse the developed methodology.

The influence of the designer choices for "design subdomains" characterizing the macrostructure, and "material unit cell" representing the microstructure, will also be studied in the solutions obtained.

The examples show the effectiveness of the methodology developed, fully benefiting from an enlarged design space incorporating structural and material designs, and thus efficiently design structural components.

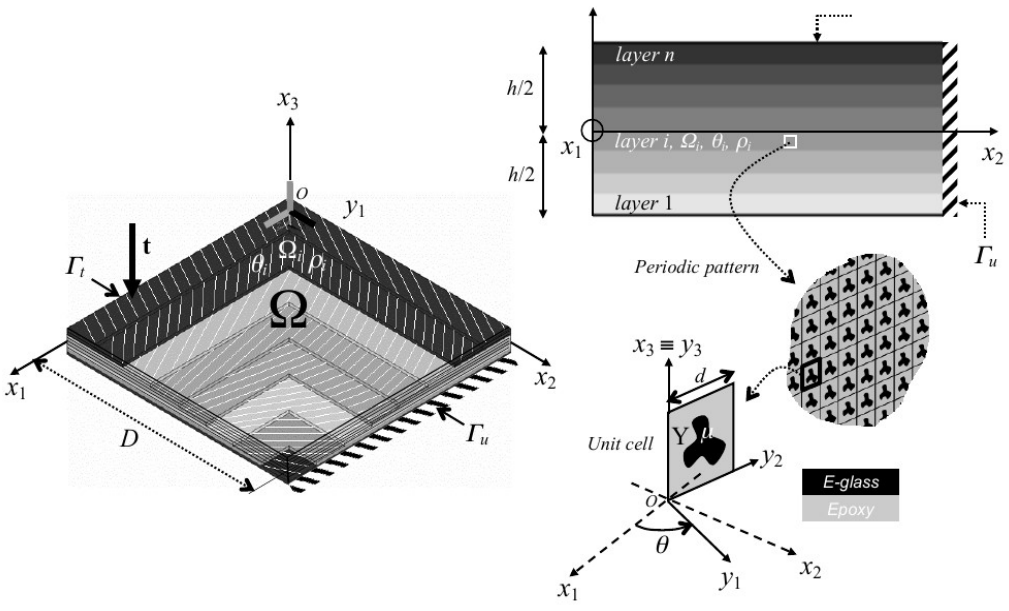
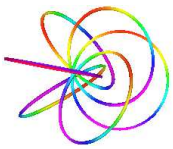


Figure 1. Hierarchical design model for laminated composite structures. Global $Ox_1x_2x_3$ and local $Oy_1y_2y_3$ coordinate systems and material directionality θ are shown.



SYMCOMP 2015
Faro, March 26-27, 2015

COMPUTATIONAL MECHANICS: SYMBOLIC COMPUTATION, NONLINEAR CONTINUA, CONTACT AND FRACTURE

P. Areias [1*]

1: Department of Physics
School of Sciences and Engineering
University of Évora
Rua Romão Ramalho, 59,
7002-554 vora, Portugal
e-mail: pmaa@uevora.pt,
web:**http://www.simplas-software.com**

Keywords: Symbolic computations, sparse matrices, semi-implicit algorithms, complementarity smoothing, remeshing, constraints

Abstract. *Our aim is to develop software for solving solid mechanics problems covering all observed finite strain behavior: linear elastic, elasto-plastic, contact with friction, shear bands and fracture. Three basic levels of discretization are covered: plane problems, shell problems and full 3D problems. We created a framework to deal with sparse linear algebra structures, then a Frontal Solver to deal with equality constraints and both plane and shell discretizations. An elaborate constitutive algorithm adapts to the specific discretization dimensions. Mathematica/Acegen combination allows extremely sophisticated* specific *constitutive laws and kinematic stencils to be coupled. In this work we describe in detail all the crucial points and derivations, focusing on the symbolic computations. Specifically: shell formulations for finite strains, multisurface elasto-plastic laws, fracture algorithms and the solution procedures.*

1 A semi-implicit integration algorithm in finite strains

Cauchy equations of equilibrium for a *rotated* reference configuration are obtained from the corresponding *spatial* equilibrium (derivations for the latter are shown in Ogden [26]). Using standard notation (cf. [26, 37]) we write the spatial equilibrium equations as

$$\frac{\partial \sigma_{ij}}{\partial x_{a_j}} + b_i = 0 \tag{1}$$

with the Cauchy tensor components σ_{ij} $(i,j=1,2,3)$. In (1) i is the direction index and j is the facet index. The components of the body force vector are b_i. In (1), coordinates x_{a_j} are the spatial, or deformed, coordinates of a given point under consideration. In addition, the following natural and essential boundary conditions hold on each part of the boundary $\Gamma_a = \Gamma_a^t \cup \Gamma_a^u$ where Γ_a^t is the natural boundary and Γ_a^u is the essential boundary:

$$\overline{\boldsymbol{t}} = \boldsymbol{\sigma}\cdot\boldsymbol{v} \quad \text{on} \quad \Gamma_a^t \tag{2}$$

$$\overline{\boldsymbol{u}} = \boldsymbol{u} \quad \text{on} \quad \Gamma_a^u \tag{3}$$

where $\overline{\boldsymbol{t}}$ is the known stress vector on Γ_a^t where $\boldsymbol{v}$ is the outer normal and $\overline{\boldsymbol{u}}$ is the known displacement field on Γ_a^u. It is assumed that (1) and (2-3) are satisfied for a time parameter $t_a \in [0,T]$ with T being the total time of observation and for a point with position $\boldsymbol{x}_a \in \Omega_a$ belonging to the deformed position domain at the time of analysis. Equilibrium configuration corresponds to the domain Ω_a and is identified by the subscript a. In tensor notation, equation (1) can be presented as:

$$\nabla\cdot\boldsymbol{\sigma}^T + \boldsymbol{b} = \boldsymbol{0} \tag{4}$$

with $\boldsymbol{\nabla} = {}^{\partial}/_{\partial \boldsymbol{x}_a}$ being the spatial gradient *operator*. After multiplication by the velocity field $\dot{\boldsymbol{u}}$, integration in the deformed configuration Ω_a and application of integration by parts component-wise, we obtain the following power form ($\dot{W}_{\text{int}}$ is the internal and $\dot{W}_{\text{ext}}$ is the external power):

$$\underbrace{\int_{\Omega_a} \boldsymbol{\sigma} : \boldsymbol{L}\mathrm{d}\Omega_a}_{\dot{W}_{\text{int}}} = \underbrace{\int_{\Omega_a} \boldsymbol{b}\cdot\dot{\boldsymbol{u}}\mathrm{d}\Omega_a + \int_{\Gamma_a^t} \overline{\boldsymbol{t}}\cdot\dot{\boldsymbol{u}}\mathrm{d}\Gamma_a}_{\dot{W}_{\text{ext}}} \tag{5}$$

where $\boldsymbol{L}$ is the velocity gradient: $\boldsymbol{L} = \frac{\partial \dot{\boldsymbol{u}}}{\partial \boldsymbol{x}_p}$. $\boldsymbol{L}$ can be decomposed in a symmetric part ($\boldsymbol{D}$, called strain rate) and a skew-symmetric part ($\boldsymbol{W}$, called vorticity), $\boldsymbol{L} = \boldsymbol{D} + \boldsymbol{W}$. Since $\boldsymbol{\sigma}$ has a symmetric component matrix, integrating in the initial configuration (subscript 0, which corresponds to the domain Ω_0), we obtain:

$$\dot{W}_{\text{int}} = \int_{\Omega_0} \boldsymbol{\tau} : \boldsymbol{D}\mathrm{d}\Omega_0 \tag{6}$$

where $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ is the Kirchhoff stress tensor with $J = \det \boldsymbol{F}$ and $\boldsymbol{F}$ is the deformation gradient. The use of equivalence between internal $\dot{W}_{\text{int}}$ and external $\dot{W}_{\text{ext}}$ power for

a continuum has been used to obtain objective rates of the Cauchy stress $\boldsymbol{\sigma}$. In the corotational case, we can rotate the Kirchhoff stress and the strain rate to obtain:

$$\dot{W}_{\text{int}} = \int_{\Omega_0} \left(\boldsymbol{R}_{0a}^T \boldsymbol{\tau} \boldsymbol{R}_{0a}\right) : \left(\boldsymbol{R}_{0a}^T \boldsymbol{D} \boldsymbol{R}_{0a}\right) \mathrm{d}\Omega_0 \tag{7}$$

In a shell, it is convenient to use local frames. We use three directions, where $\overline{\boldsymbol{e}}_{1a}$ and $\overline{\boldsymbol{e}}_{2a}$ are tangent to the mid-surface and $\overline{\boldsymbol{e}}_{3a}$ is normal. In that case, we define $\boldsymbol{R}_{0a}$ as:

$$\boldsymbol{R}_{0a} = [\overline{\boldsymbol{e}}_{1a}, \overline{\boldsymbol{e}}_{2a}, \overline{\boldsymbol{e}}_{3a}] \tag{8}$$

where each column is a unit vector and $\overline{\boldsymbol{e}}_{ia} \cdot \overline{\boldsymbol{e}}_{ja} = \delta_{ij}$. If required, one of the directions (here $\overline{\boldsymbol{e}}_{1a}$) can be taken to be parallel to a fiber in an anisotropic material. We then use the following convention for writing tensors: the bold symbol indicates tensor *components*. Therefore, the frame where the tensor components are defined is identified as a superscript. For example, the second-order tensor $\boldsymbol{S}_{ab}$ with the equilibrium configuration Ω_a and reference configuration Ω_b has the following components in frame c:$\boldsymbol{S}_{ab}^c$. If global frame 0 is adopted, component transformation between frame 0 and frame c follows:

$$\boldsymbol{S}_{ab}^c = \boldsymbol{R}_{c0} \boldsymbol{S}_{ab}^0 \boldsymbol{R}_{c0}^T \tag{9}$$

Generalizing (9), *change of basis* from d to c results in the following transformation:

$$\boxed{\boldsymbol{S}_{ab}^c = \boldsymbol{R}_{cd} \boldsymbol{S}_{ab}^d \boldsymbol{R}_{cd}^T} \tag{10}$$

In addition, change in *reference* configuration reads:

$$\boxed{\boldsymbol{S}_{ac}^0 = \boldsymbol{R}_{bc} \boldsymbol{S}_{ab}^0 \boldsymbol{R}_{bc}^T} \tag{11}$$

Using (11) and transformation (10) for coinciding reference configuration and frame,

$$\boxed{\boldsymbol{S}_{ac}^c = \boldsymbol{S}_{ab}^b} \tag{12}$$

From the relation between the time derivative right Cauchy-Green tensor $\boldsymbol{C}$ and the strain rate $\boldsymbol{D}$, we obtain:

$$\dot{\boldsymbol{C}} = 2\boldsymbol{F}^T \boldsymbol{D} \boldsymbol{F} \tag{13}$$

J. Simo [34] derived the following one-step scheme for integrating the strain rate in frame 0, producing the following "strain" from integration of $\boldsymbol{D}$:

$$\boldsymbol{e}_{ab,\beta}^0 \cong \frac{1}{2} \boldsymbol{F}_{b\beta}^T \left[\boldsymbol{F}_{a\beta}^T \boldsymbol{F}_{a\beta} - \boldsymbol{I}\right] \boldsymbol{F}_{b\beta} \tag{14}$$

Use the previous power conjugacy (7) ($\boldsymbol{\tau} = J\boldsymbol{\sigma}$), internal power can be written as:

$$\dot{W}_{\rm int} = \int_{\Omega_0} \boldsymbol{S}^0_{a0} : \boldsymbol{D}^0_{a0} {\rm d}\Omega_0$$

where $\boldsymbol{S}^0_{a0} = \boldsymbol{R}^T_{0a}\boldsymbol{\tau}^0_a\boldsymbol{R}_{0a}$ and $\boldsymbol{D}^0_{a0} = \boldsymbol{R}^T_{0a}\boldsymbol{d}^0_a\boldsymbol{R}_{0a}$. Using an hypoelastic relation, stress updating is performed as:

$$\boldsymbol{S}^0_{a0} = \boldsymbol{S}^0_{b0} + \Delta\check{\boldsymbol{S}}_a(\boldsymbol{R}_{\beta 0}\boldsymbol{e}^0_{ab,\beta}\boldsymbol{R}^T_{\beta 0})$$

where $\Delta\check{\boldsymbol{S}}_a$ is a function of the strain $\boldsymbol{e}^0_{ab,\beta}$. This term $\Delta\check{\boldsymbol{S}}_a$ is called the constitutive part of the stress. However, since $\boldsymbol{S}^0_{a0} = \boldsymbol{S}^b_{ab}$ and $\boldsymbol{S}^0_{b0} = \boldsymbol{S}^b_{bb}$, we have:

$$\boldsymbol{S}^b_{ab} = \boldsymbol{S}^b_{bb} + \Delta\check{\boldsymbol{S}}_a(\boldsymbol{R}_{\beta 0}\boldsymbol{e}^0_{ab,\beta}\boldsymbol{R}^T_{\beta 0})$$

In the case $\beta = b$, the result is:

$$\boldsymbol{S}^b_{ab} = \boldsymbol{S}^b_{bb} + \Delta\check{\boldsymbol{S}}_a(\boldsymbol{e}^b_{ab}) \tag{15}$$

with

$$\boxed{\boldsymbol{e}^b_{ab} \cong \frac{1}{2}\boldsymbol{R}_{b0}\left[\boldsymbol{F}^T_{ab}\boldsymbol{F}_{ab} - \boldsymbol{I}\right]\boldsymbol{R}^T_{b0}} \tag{16}$$

We now insert the Jacobian for configuration Ω_b in (15) as:

$$\underbrace{\frac{1}{J_{b0}}\boldsymbol{S}^b_{ab}}_{\boldsymbol{S}^{\star b}_{ab}} = \underbrace{\frac{1}{J_{b0}}\boldsymbol{S}^b_{bb}}_{\boldsymbol{\sigma}^b_b} + \frac{1}{J_{b0}}\Delta\check{\boldsymbol{S}}_a(\boldsymbol{e}^b_{ab}) \tag{17}$$

Defining

$$\boldsymbol{S}^{\star b}_{ab} = \frac{1}{J_{b0}}\boldsymbol{S}^b_{ab} \tag{18}$$

then we obtain the Cauchy stress in configuration Ω_a and frame a as:

$$\boldsymbol{\sigma}^a_a = \boldsymbol{S}^{\star a}_{aa} = \frac{1}{J_{a0}}\boldsymbol{S}^a_{aa} = \frac{1}{J_{a0}}\boldsymbol{S}^b_{ab} = \frac{J_{b0}}{J_{a0}}\boldsymbol{S}^{\star b}_{ab} \tag{19}$$

If $\Delta\check{\boldsymbol{S}}_a(\boldsymbol{e}^b_{ab})$ is homogeneous of degree one, then it follows that:

$$\boxed{\boldsymbol{S}^{\star b}_{ab} = \boldsymbol{\sigma}^b_b + \Delta\check{\boldsymbol{S}}_a\left(\widetilde{\boldsymbol{e}}^b_{ab}\right)}$$

where

$$\boxed{\widetilde{\boldsymbol{e}}^b_{ab} = \frac{1}{J_{b0}}\boldsymbol{e}^b_{ab}} \tag{20}$$

For hyperelastic models, we require the total strain, which is the strain from configuration Ω_a with respect to configuration Ω_0. Using frame b, we have the following update formula for the Green-Lagrange strain:

$$\boldsymbol{e}_{a0}^{b} = \boldsymbol{e}_{b0}^{b} + \boldsymbol{F}_{b0}^{bT}\boldsymbol{e}_{ab}^{b}\boldsymbol{F}_{b0}^{b} \tag{21}$$

We can observe that $\boldsymbol{S}_{ab}^{\star b}$ is work-conjugate to $\boldsymbol{e}_{a0}^{b}$ in configuration Ω_b. Global Cauchy stress can be obtained as: $\boldsymbol{\sigma}_b^0 = \boldsymbol{R}_{0b}\boldsymbol{\sigma}_b^b\boldsymbol{R}_{0b}^T$. In frame a, $\boldsymbol{e}_{a0}^{a}$ reads:

$$\boldsymbol{e}_{a0}^{a} = \boldsymbol{R}_{ba}^{T}\boldsymbol{e}_{a0}^{b}\boldsymbol{R}_{ba} \tag{22}$$

For symmetric tensors, such as $\boldsymbol{S}_{ab}^{\star b}$ and $\boldsymbol{e}_{ab}^{b}$, it is preferable to use the Voigt notation. Upright bold symbols denote Voigt form of symmetric tensors:

$$\mathbf{S}_{ab}^{\star b} = \text{Voigt}[\boldsymbol{S}_{ab}^{\star b}] = \frac{1}{J_{b0}}\mathscr{V}_S\left(\boldsymbol{F}_{b0}^{b}\right)\mathbf{S}_{a0}^{b} \tag{23}$$

where $\mathbf{S}_{a0}^{b} = \text{Voigt}[\boldsymbol{S}_{a0}^{b}]$. We note that, omitting indices a, b, and 0 $\mathscr{V}_S(\boldsymbol{F})$ can be written as:

$$\mathscr{V}_S(\boldsymbol{F}) = \begin{bmatrix} F_{11}^2 & F_{21}^2 & F_{31}^2 & 2F_{21}F_{11} & 2F_{31}F_{11} & 2F_{31}F_{21} \\ F_{12}^2 & F_{22}^2 & F_{32}^2 & 2F_{22}F_{12} & 2F_{32}F_{12} & 2F_{32}F_{22} \\ F_{13}^2 & F_{23}^2 & F_{33}^2 & 2F_{23}F_{13} & 2F_{33}F_{13} & 2F_{33}F_{23} \\ F_{11}F_{12} & F_{21}F_{22} & F_{31}F_{32} & F_{21}F_{12}+F_{11}F_{22} & F_{31}F_{12}+F_{11}F_{32} & F_{31}F_{22}+F_{21}F_{32} \\ F_{11}F_{13} & F_{21}F_{23} & F_{31}F_{33} & F_{21}F_{13}+F_{11}F_{23} & F_{31}F_{13}+F_{11}F_{33} & F_{31}F_{23}+F_{21}F_{33} \\ F_{12}F_{13} & F_{22}F_{23} & F_{32}F_{33} & F_{22}F_{13}+F_{12}F_{23} & F_{32}F_{13}+F_{12}F_{33} & F_{32}F_{23}+F_{22}F_{33} \end{bmatrix}$$

and, for the strain quantities, $\mathscr{V}_E(\boldsymbol{F}) = \mathscr{V}_S^T(\boldsymbol{F})$. The full algorithm is given in Table 1. We establish the constitutive quantities by a general nonlinear *constitutive* equation system. Omitting the frame superscript, it reduces to the root finding for the following nonlinear system:

$$\boxed{\boldsymbol{\varphi}\left(\overbrace{\underbrace{\mathbf{S}_{ab}, Q_{ab}}_{\text{Unknown}}, \mathbf{e}_{ab}^{U}}^{\text{Governing}}; \underbrace{\mathbf{S}_{ab}^{K}, \mathbf{e}_{ab}^{K}, \mathbf{e}_{ab}, T_a}_{\text{Known}}; \underbrace{\boldsymbol{\chi}_a}_{\text{Internal var.}}\right) = \mathbf{0}} \tag{24}$$

where $\mathbf{S}_{ab}^{K}$ is a set of known stresses and $\mathbf{e}_{ab}^{U}$ is the set of unknown strains. Newton iteration for (24) provides the solution for the constitutive unknowns $\mathbf{S}_{ab}$, Q_{ab}, $\mathbf{e}_{ab}^{U}$ and $\boldsymbol{\chi}_a$. A smoothed version of (24) is introduced by using the Chen-Mangasarian replacement functions, which depend on a parameter `Error` such that (cf. [8]):

$$\lim_{\texttt{Error}\to 0}\boldsymbol{\varphi}_{\texttt{Error}} = \boldsymbol{\varphi} \tag{25}$$

Algorithm 1 Relative Lagrangian formulation (Voigt notation adopted).

Given $\boldsymbol{F}_{ab}^b$, $\mathbf{e}_{ab}^b$ (*both in frame b*), $\boldsymbol{R}_{0b}$ and $\boldsymbol{R}_{0a}$ (*both in frame* 0) and T_b and T_a Recovered from storage $\boldsymbol{F}_{b0}^b$, $\boldsymbol{\sigma}_b^0$, $\mathbf{e}_{b0}^b$ ($\boldsymbol{\sigma}_b$ is stored in frame 0 for purpose of representation)	
Represent Cauchy stress in frame b	$\boldsymbol{\sigma}_b^b = \mathscr{V}_S\left(\boldsymbol{R}_{0b}^T\right)\boldsymbol{\sigma}_b^0$
Relevant Jacobian determinants	$J_{b0} = \det \boldsymbol{F}_{b0}^b$ $J_{ab} = \det \boldsymbol{F}_{ab}^b$
Relative rotation and total deformation gradient update	$\boldsymbol{R}_{ab} = \boldsymbol{R}_{0a}^T\boldsymbol{R}_{0b}$ $\boldsymbol{F}_{a0}^b = \boldsymbol{F}_{ab}^b\boldsymbol{F}_{b0}^b$
Thermal effect in the relative strain	$\mathbf{e}_{ab}^{b\circ} = \mathbf{e}_{ab}^b + \alpha\left(T_a - T_b\right)\mathbf{I}$
Total strain update	$\mathbf{e}_{a0}^b = \mathbf{e}_{b0}^b + \mathscr{V}_E\left(\boldsymbol{F}_{b0}^{bT}\right)\mathbf{e}_{ab}^{b\circ}$
Corrected relative strain for **UL**	$\widetilde{\boldsymbol{e}}_{ab}^{b\circ} = \frac{1}{J_{b0}}\boldsymbol{e}_{ab}^{b\circ}$
(UL) *Determine* $\mathbf{S}_{ab}^{\star b} = \boldsymbol{\sigma}_b^b + \Delta\check{\mathbf{S}}_a\left(\widetilde{\mathbf{e}}_{ab}^{b\circ}\right)$ and Q_{ab} along with sensitivity quantities $\mathscr{C}_{ab} = \frac{1}{J_{b0}}\frac{\partial\Delta\check{\mathbf{S}}_a}{\partial\widetilde{\mathbf{e}}_{ab}^{b\circ}}$, $\frac{\partial\Delta\check{\mathbf{S}}_a}{\partial T_a}$, $\frac{1}{J_{b0}}\frac{\partial Q_{ab}}{\partial\widetilde{\mathbf{e}}_{ab}^{b\circ}}$ and $\frac{\partial Q_{ab}}{\partial T_a}$	
(TL) *Determine* $\mathbf{S}_{a0}^b\left(\mathbf{e}_{a0}^b\right)$ and Q_{ab}, along with sensitivity quantities $\mathscr{C}_{a0} = \frac{\partial\mathbf{S}_{a0}^b}{\partial\mathbf{e}_{a0}^b}$, $\frac{\partial\mathbf{S}_{a0}^b}{\partial T_a}$, $\frac{\partial Q_{ab}}{\partial\mathbf{e}_{a0}^b}$ and $\frac{\partial Q_{ab}}{\partial T_a}$	
(TL) Determine relative stresses	$\mathbf{S}_{ab}^{\star b} = \frac{1}{J_{b0}}\mathscr{V}_S\left(\boldsymbol{F}_{b0}^b\right)\mathbf{S}_{a0}^b$
(TL) Determine relative sensitivities	$\mathscr{C}_{ab} = \frac{1}{J_{b0}}\mathscr{V}_S\left(\boldsymbol{F}_{b0}^b\right)\mathscr{C}_{a0}\mathscr{V}_E(\boldsymbol{F}_{b0}^{bT})$ $\frac{\partial\mathbf{S}_{ab}^{\star b}}{\partial T_a} = \frac{1}{J_{b0}}\mathscr{V}_S\left(\boldsymbol{F}_{b0}^b\right)\frac{\partial\mathbf{S}_{a0}^b}{\partial T_a}$ $\frac{\partial Q_{ab}}{\partial\mathbf{e}_{ab}^b} = \frac{\partial Q_{ab}}{\partial\mathbf{e}_{a0}^b}\mathscr{V}_E\left(\boldsymbol{F}_{b0}^{bT}\right)$
Update temperature sensitivity	$\frac{\partial\mathbf{S}_{ab}}{\partial T_a} \leftarrow \frac{\partial\mathbf{S}_{ab}}{\partial T_a} + \alpha\mathscr{C}_{ab}\mathbf{I}$
Determine strain in frame a	$\mathbf{e}_{a0}^a = \mathscr{V}_E\left(\boldsymbol{R}_{ab}\right)\mathbf{e}_{a0}^b$
Determine deformation gradient in frame a	$\boldsymbol{F}_{a0}^a = \boldsymbol{R}_{ab}\boldsymbol{F}_{a0}^b\boldsymbol{R}_{ab}^T$
Determine Cauchy stress in frame 0	$\boldsymbol{\sigma}_a^0 = \frac{1}{J_{ab}}\mathscr{V}_S\left(\boldsymbol{R}_{0a}\right)\mathbf{S}_{ab}^{\star b}$
Store $\boldsymbol{F}_{a0}^a$, $\boldsymbol{\sigma}_a^0$, $\mathbf{e}_{a0}^a$	
Return Q_{ab}, $\mathbf{S}_{ab}^{\star b}$, $\mathscr{C}_{ab}$, $\frac{\partial\mathbf{S}_{ab}^{\star b}}{\partial T_a}$, $\frac{\partial Q_{ab}}{\partial\mathbf{e}_{ab}^b}$ and $\frac{\partial Q_{ab}}{\partial T_a}$	

Newton iteration on (24), using $\boldsymbol{\varphi}_{\texttt{Error}}$ as a replacement provides the following scheme:

$$\underbrace{\left[\begin{array}{cccc} \frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial \mathbf{S}_{ab}} & \frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial Q_{ab}} & \frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial \mathbf{e}^U_{ab}} & \frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial \boldsymbol{\chi}_a} \end{array}\right]}_{\boldsymbol{J}} \left\{\begin{array}{c} \Delta\mathbf{S}_{ab} \\ \Delta Q_{ab} \\ \Delta\mathbf{e}^U_{ab} \\ \Delta\boldsymbol{\chi}_a \end{array}\right\} = -\boldsymbol{\varphi}_{\texttt{Error}} \tag{26}$$

After achieving solution using the Newton scheme (26), we calculate the sensitivities from the following equation:

$$\left\{\begin{array}{c} \mathrm{d}\mathbf{S}_{ab} \\ \mathrm{d}Q_{ab} \\ \mathrm{d}\mathbf{e}^U_{ab} \\ \mathrm{d}\boldsymbol{\chi}_a \end{array}\right\} = -\boldsymbol{J}^{-1} \left[\frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial \mathbf{e}^K_{ab}} \frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial \mathbf{e}_{ab}} \frac{\partial \boldsymbol{\varphi}_{\texttt{Error}}}{\partial T_{ab}}\right] \left\{\begin{array}{c} \mathrm{d}\mathbf{e}^K_{ab} \\ \mathrm{d}\mathbf{e}_{ab} \\ \mathrm{d}T_{ab} \end{array}\right\} \tag{27}$$

We provide two examples: hyperelastic and elasto-plasticity with shells. For hyperelasticity, the system is independent of `Error` , since classical hyperelasticity is smooth:

$$\boldsymbol{\varphi}(\mathbf{S}_{ab}, e^U_{33}; S^K_{33}, \mathbf{e}_{ab}) = \left\{\begin{array}{c} S_{11} \\ S_{22} \\ S^K_{33} = 0 \\ S_{12} \\ S_{13} \\ S_{23} \end{array}\right\} - \frac{\partial \psi(\mathbf{e}_{ab}, e^U_{33})}{\partial \left\{\begin{array}{c} e_{11} \\ e_{22} \\ e^U_{33} \\ e_{12} \\ e_{13} \\ e_{23} \end{array}\right\}} \tag{28}$$

For elasto-plasticity in shells we provide a general approach, with details being given in [8]:

$$\boldsymbol{\varphi}(\mathbf{S}_{ab}, e^U_{33}; S_{33}, \mathbf{e}_{ab}) = \left\{\begin{array}{c} \mathbf{e}_{ab} - \mathscr{C}^{-1}_{\text{linear}} \Delta\check{\mathbf{S}}_{ab} - \mathbf{n}\Delta\gamma \\ S_{33} \\ \mu^\star \Delta\gamma - \langle \mu^\star \Delta\gamma + \phi \rangle_{\texttt{Error}} \\ \Delta\boldsymbol{\chi}_a - \Delta\gamma\omega(\boldsymbol{\chi}_a) \end{array}\right\} \tag{29}$$

where $\Delta\gamma$ is the plastic multiplier increment, $\mu^\star$ is a dimensional parameter described in [8] and the smooth ramp function of Chen and Mangasarian [16] is used for the third equation, which depends on `Error`. Function ϕ in (29) is the yield function and ω are the internal variable evolution functions. In addition, the flow vector $\mathbf{n}$ is also required for the flow rule, and it is defined as:

$$\mathbf{n} = \frac{\partial \phi}{\partial \mathbf{S}_{ab}} \tag{30}$$

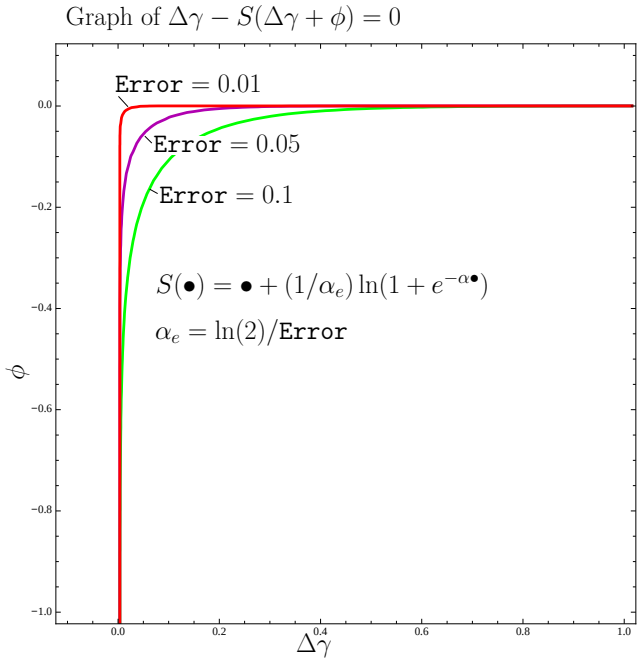


Figure 1: Replacement of $\mu^\star\Delta\gamma - \langle\mu^\star\Delta\gamma + \phi\rangle$ by $\mu^\star\Delta\gamma - S(\mu^\star\Delta\gamma + \phi)$ as a function of a Error parameter ($\mu^\star = 1$ is depicted).

Figure 1 shows the effect of `Error` in the satisfaction of the complementarity condition. The prototype yield functions employed are von-Mises, Hill and Barlat 91. All are treated in the same format, without requirement of particular implementations (which are only possible with von-Mises and Hill criteria). The reason for a unique treatment for *all* yield functions is that any particular yield function is inserted by means of ϕ, $\boldsymbol{n}$ and derivatives of $\boldsymbol{n}$. We summarize the yield functions in Table 1.

2 Shell element technology

Shells further constrain the metric, as described by the classical work of Antman and Marlow [2], resulting in a specific form of the position vector. For a shell, it is known that a possible $\boldsymbol{x}_a \in \Omega_a$ is the following:

$$\boldsymbol{x}_a = \boldsymbol{r}_a + \frac{H_a\xi_3}{2}\boldsymbol{d}_a \tag{31}$$

where $\boldsymbol{r}_a$ is the mid-surface position vector in Ω_a, ξ_3 is a non-dimensional, thickness-like coordinate, H_a is the thickness at position $\boldsymbol{r}_a$ and, finally, $\boldsymbol{d}_a$ is a unit director. We use H for the undeformed thickness. This nomenclature is standard and is adopted in [12]. For $\boldsymbol{d}_a$ we use 3 rotation parameters, $\beta_1, \beta_2, \beta_3$ and determine the director $\boldsymbol{d}_a$ as:

Table 1: Prototype yield functions employed

	Yield function
von-Mises	$\phi = \frac{\sqrt{\frac{1}{2}(\sigma_{11}-\sigma_{22})^2+\frac{1}{2}(\sigma_{11}-\sigma_{22})^2+\frac{1}{2}(\sigma_{11}-\sigma_{22})^2+3\sigma_{12}^2+3\sigma_{13}^2+3\sigma_{23}^2}}{\sigma_y} - 1$
Hill	$\phi = \frac{\sqrt{F(\sigma_{22}-\sigma_{33})^2+G(\sigma_{33}-\sigma_{11})^2+H(\sigma_{11}-\sigma_{22})^2+2s_{12}\sigma_{12}^2+2s_{13}\sigma_{13}^2+2s_{23}\sigma_{23}^2}}{\sigma_y} - 1$ $F = \frac{1}{2}\left[\frac{1}{R_{22}^2}+\frac{1}{R_{33}^2}-\frac{1}{R_{11}^2}\right]$ $G = \frac{1}{2}\left[\frac{1}{R_{11}^2}+\frac{1}{R_{33}^2}-\frac{1}{R_{22}^2}\right]$ $H = \frac{1}{2}\left[\frac{1}{R_{11}^2}+\frac{1}{R_{22}^2}-\frac{1}{R_{33}^2}\right]$ $s_{12} = \frac{3}{2R_{12}^2}$ $s_{13} = \frac{3}{2R_{13}^2}$ $s_{23} = \frac{3}{2R_{23}^2}$
Barlat 91 (BCC: $m=6$)	$\phi = \frac{\left[\frac{1}{2}(T_1-T_2)^m+\frac{1}{2}(T_2-T_3)^m+\frac{1}{2}(T_3-T_1)^m\right]^{1/m}}{\sigma_y} - 1$ where T_i are the eigenvalues of $\boldsymbol{T}$ $\boldsymbol{T} = \begin{bmatrix} \frac{1}{3}\left[P_3(\sigma_{11}-\sigma_{22})-P_2(\sigma_{33}-\sigma_{11})\right] & P_6\sigma_{12} & P_5\sigma_{13} \\ P_6\sigma_{12} & \frac{1}{3}\left[P_1(\sigma_{22}-\sigma_{33})-P_3(\sigma_{11}-\sigma_{22})\right] & P_4\sigma_{23} \\ P_5\sigma_{13} & P_4\sigma_{23} & \frac{1}{3}\left[P_2(\sigma_{33}-\sigma_{11})-P_1(\sigma_{22}-\sigma_{33})\right] \end{bmatrix}$

$$\boldsymbol{d}_a = \boldsymbol{T}(\beta_1,\beta_2,\beta_3)\boldsymbol{d}_b \tag{32}$$

The rotation matrix $\boldsymbol{T}_r(\beta_1,\beta_2,\beta_3)$ is given by a reworked version of Rodrigues formula (cf. [28]):

$$\boldsymbol{T}_r(\beta_1,\beta_2,\beta_3) = \boldsymbol{I} + \frac{\sin\beta}{\beta}\boldsymbol{W}(\boldsymbol{\beta}) + \frac{2\sin^2\left(\frac{\beta}{2}\right)}{\beta^2}\boldsymbol{W}^2(\boldsymbol{\beta}) \tag{33}$$

where $\beta = \|\boldsymbol{\beta}\|$ is the norm of $\{\beta_1,\beta_2,\beta_3\}$ and $\boldsymbol{W}(\boldsymbol{\beta})$ is the following skew-symmetric matrix:

$$\boldsymbol{W}(\boldsymbol{\beta}) = \begin{bmatrix} 0 & -\beta_3 & \beta_2 \\ \beta_3 & 0 & -\beta_1 \\ -\beta_2 & \beta_1 & 0 \end{bmatrix} \tag{34}$$

First and second derivatives of $\boldsymbol{T}_r(\beta_1,\beta_2,\beta_3)$ are determined using the software Wolfram Mathematica [31] with the add-on AceGen [23]. Since $\boldsymbol{d}_a$ is a unit vector, the property $\boldsymbol{d}_a \cdot \frac{\partial \boldsymbol{d}_a}{\partial \xi_i} = 0$ for $i = 1,2$ results in the following metric components:

$$[\boldsymbol{m}_{aa}]_{ij} = \boldsymbol{r}_{a_i} \cdot \boldsymbol{r}_{a_j} + \frac{H_a \xi_3}{2}\left[\left(\boldsymbol{r}_{a_i} \cdot \boldsymbol{d}_{a_j} + \boldsymbol{r}_{a_j} \cdot \boldsymbol{d}_{a_i}\right) + \frac{H_a \xi_3}{2} \boldsymbol{d}_{a_i} \cdot \boldsymbol{d}_{a_j}\right] \tag{35}$$

$$[\boldsymbol{m}_{aa}]_{i3} = \kappa_s \frac{H_a}{2} \boldsymbol{r}_{a_i} \cdot \boldsymbol{d}_a \tag{36}$$

$$[\boldsymbol{m}_{aa}]_{33} = \frac{H_a^2}{4} \tag{37}$$

with $i = 1, 2$ and $j = 1, 2$. In (35), we consider $\boldsymbol{r}_{a_i}$ as the derivative of $\boldsymbol{r}_a$ with respect to ξ_i (the parent-domain coordinate). Finally, κ_s is the shear correction parameter, taken here as 5/6, which scales the shear terms so that the correct energy is obtained. After $\boldsymbol{m}_{aa}$ is calculated from relations (35-37), compatible $\boldsymbol{C}_{ab}$ is calculated as:

$$\boldsymbol{C}_{ab} = \boldsymbol{y}_b^T \boldsymbol{m}_{aa} \boldsymbol{y}_b \tag{38}$$

where $\boldsymbol{y}_b = \left[\frac{\partial \boldsymbol{x}_b}{\partial \boldsymbol{\xi}}\right]^{-1}$. Of course, if *fixed* non-orthogonal (but normed) coordinates are used for $\boldsymbol{C}_{ab}$, four versions can be determined (covariant/covariant, contravariant/contravariant, contravariant/covariant and covariant/contravariant), the latter being used here and identified as $\boldsymbol{C}_{ab}^\star$:

$$\boldsymbol{C}_{ab}^\star = \overline{\boldsymbol{y}}_b^{-T} \boldsymbol{C}_{ab} \overline{\boldsymbol{y}}_b^T \tag{39}$$

where the line over a quantity indicates an evaluation at the element's center coordinates (i.e. $\overline{\boldsymbol{y}}_b \equiv \boldsymbol{y}_b|_{\boldsymbol{\xi} \to \boldsymbol{0}}$). As in Simo's [33] and Pian's [27] works, fixed non-orthogonal coordinates are used to:

- Extend the bending modes of a rectangle to a general quadrilateral.
- Ensure the satisfaction of the Patch-test for distorted geometries.

Fixed frames are also required for mixed transverse shear components. The use of covariant/contravariant coordinates is a consequence of mesh distortion sensitivity studies (see [6]). Note that the corresponding strain is obtained from the following formula:

$$\boldsymbol{e}_{ab}^\star = \frac{1}{2}\left(\boldsymbol{C}_{ab}^\star - \boldsymbol{I}\right) \tag{40}$$

The rationale for using this transformation is the extension of assumed-strain formulations to non-regular quadrilaterals. For convenience, Voigt form is used. We can write the Voigt form of strain (using upright notation) as:

$$\mathbf{e}_{ab}^\star = \boldsymbol{T}_E(\overline{\boldsymbol{x}}_b, \overline{\boldsymbol{y}}_b)\mathbf{e}_{ab} \tag{41}$$

where $\boldsymbol{T}_E(\overline{\boldsymbol{x}}_b, \overline{\boldsymbol{y}}_b)$ is the 6×6 matrix resulting from the use of Voigt form of the transformation (39).

2.1 Assumed strain formulation

For conciseness of notation, we temporarily omit the subscripts a and b. A mixed formulation for conservative problems is used to derive the weak form of equilibrium from a total potential with Lagrange multiplier terms to impose the constraints (cf. [35, 33]). To further simplify the notation, we use Voigt convention for symmetric tensors and assume that all strain components follow the same scheme. The consistent approach to the mixed interpolation for the conservative case is given by the following total potential:

$$\Pi(\{\boldsymbol{u}, \boldsymbol{d}\}, \widetilde{\mathbf{e}}, \boldsymbol{S}) = \int_{\Omega_b} \psi\left[\widetilde{\mathbf{e}}\right] \mathrm{d}\Omega_b + \int_{\Omega_b} \mathbf{S}^T \left(\mathbf{e} - \widetilde{\mathbf{e}}\right) \mathrm{d}\Omega_b - W_{\text{ext}}\left(\boldsymbol{u}, \boldsymbol{d}\right) \tag{42}$$

where ψ is the strain energy density function (cf. [26]), in this case a function of the independent strain increment. W_{ext} is the external work for conservative forces. In (42), $\boldsymbol{u}$ and $\boldsymbol{d}$ are the displacement and director fields, respectively, grouped as a pair $\{\boldsymbol{u}, \boldsymbol{d}\}$. In (42), the components $\mathbf{S}$ are the Lagrange multipliers corresponding to the constraints $\mathbf{e} - \widetilde{\mathbf{e}} = 0$ and also represent mixed second Piola-Kirchhoff stresses. For simplicity reasons, we temporarily assume that all components of the relative strain $\widetilde{\mathbf{e}}$ are independent, not detracting from the aimed generality. The stationarity condition of (42) results in

$$\int_{\Omega_b} \left(\mathbf{e} - \widetilde{\mathbf{e}}\right)^T \overset{\triangle}{\mathbf{S}} \mathrm{d}\Omega_b = 0 \tag{43}$$

$$\int_{\Omega_b} \left(\frac{\partial \psi\left[\widetilde{\mathbf{e}}\right]}{\partial \widetilde{\mathbf{e}}} - \mathbf{S}\right)^T \overset{\triangle}{\widetilde{\mathbf{e}}} \mathrm{d}\Omega_b = 0 \tag{44}$$

$$\int_{\Omega_b} \mathbf{S}^T \overset{\triangle}{\mathbf{e}} \mathrm{d}\Omega_b = \overset{\triangle}{W}_{\text{ext}} \tag{45}$$

Equations (43-45) are used as a basis for assumed-strain methods. Using $\widetilde{\mathbf{S}} = \frac{\partial \psi[\widetilde{\mathbf{e}}]}{\partial \widetilde{\mathbf{e}}}$, we now introduce a linear relation between $\boldsymbol{S}$ and $\widetilde{\boldsymbol{S}}$ as:

$$\mathbf{S} = \boldsymbol{Q}\boldsymbol{M}^{-1} \int_{\Omega_b} \boldsymbol{Q}^T \widetilde{\mathbf{S}} \mathrm{d}\Omega_b \tag{46}$$

where $\boldsymbol{Q}$ is a function of the spatial coordinates and $\boldsymbol{M}$ is given by:

$$\boldsymbol{M} = \int_{\Omega_b} \boldsymbol{Q}^T \boldsymbol{Q} \mathrm{d}\Omega_b \tag{47}$$

Energy-conjugacy $\int_{\Omega_b} \mathbf{S}^T \overset{\triangle}{\mathbf{e}} \mathrm{d}\Omega_b = \int_{\Omega_b} \widetilde{\mathbf{S}}^T \overset{\triangle}{\widetilde{\mathbf{e}}} \mathrm{d}\Omega_b$ then allows the determination of $\overset{\triangle}{\widetilde{\mathbf{e}}}$ as:

$$\overset{\triangle}{\widetilde{\mathbf{e}}} = \boldsymbol{Q}\boldsymbol{M}^{-1} \int_{\Omega_b} \boldsymbol{Q}^T \overset{\triangle}{\mathbf{e}} \mathrm{d}\Omega_b \tag{48}$$

Note that (48) could be obtained by using a weighed least-squares approach with the interpolation matrix $\boldsymbol{Q}$. By inserting (46) into (45), we can write the energy conjugacy condition (previously alluded) as:

$$\begin{aligned} \int_{\Omega_b} \mathrm{S}_i \overset{\triangle}{\mathrm{e}}_i \mathrm{d}\Omega_b &= \int_{\Omega_b} \left[\overset{\triangle}{\mathrm{e}}_i Q_{ij} M^{-1}_{jk} \left(\int_{\Omega_b} Q_{lk} \widetilde{\mathrm{S}}_l \mathrm{d}\Omega_b \right) \right] \mathrm{d}\Omega_b & (49) \\ &= \int_{\Omega_b} \widetilde{\mathrm{S}}_i \overset{\triangle}{\widetilde{\mathrm{e}}}_i \mathrm{d}\Omega_b & (50) \end{aligned}$$

Equation (45) then becomes:

$$\int_{\Omega_b} \widetilde{\mathbf{S}}^T \overset{\triangle}{\widetilde{\mathbf{e}}} \mathrm{d}\Omega_b = \overset{\triangle}{W}_{\mathrm{ext}} \tag{51}$$

As for equation (43), it is trivially satisfied from (46) and (48). The same occurs with (44) as a consequence of using the assumed form of $\mathbf{S}$ and $\widetilde{\mathbf{e}}$. Of course, if a non-conservative problem is solved, the form (51) can still be applied originating from the principle of virtual power (e.g. [1]). Note that we can write equation (48) as the result of a least square calculation.

2.2 Specific discretization for a single shell element

A discretization of the Euler-Lagrange equations for one element (superscript e) makes use of the following interpolations:

$$\boldsymbol{u}^e = \boldsymbol{N}_u \boldsymbol{u}_N \tag{52}$$

$$\boldsymbol{d}^e = \frac{\boldsymbol{N}_d \boldsymbol{d}_N}{\|\boldsymbol{N}_d \boldsymbol{d}_N\|_2} \tag{53}$$

The specific forms for $\boldsymbol{N}_u$, $\boldsymbol{N}_d$ are standard:

$$\boldsymbol{N}_u(\xi_1, \xi_2) = \begin{bmatrix} \cdots & N_K(\xi_1,\xi_2) & 0 & 0 & \cdots \\ \cdots & 0 & N_K(\xi_1,\xi_2) & 0 & \cdots \\ \cdots & 0 & 0 & N_K(\xi_1,\xi_2) & \cdots \end{bmatrix}_{3\times 12} \tag{54}$$

$$\boldsymbol{N}_d(\xi_1,\xi_2)=\left[\begin{array}{ccccc}\cdots & N_K(\xi_1,\xi_2) & 0 & 0 & \cdots\\ \cdots & 0 & N_K(\xi_1,\xi_2) & 0 & \cdots\\ \cdots & 0 & 0 & N_K(\xi_1,\xi_2) & \cdots\end{array}\right]_{3\times 12} \tag{55}$$

with scalar shape functions given by:

$$N_K(\xi_1,\xi_2)=\frac{1}{4}\left(1+\xi_{1K}\xi_1\right)\left(1+\xi_{2K}\xi_2\right) \tag{56}$$

Note that, in (53), normalization is required to ensure correctness in the metric coefficients. To clarify our approach we decompose the assumed strains in two terms: one for the out-of-plane strain obtained from ANS (here with notation $\widetilde{\mathbf{e}}_I^e$) and another in-plane least-squares interpolation (here with notation $\widetilde{\mathbf{e}}_{II}^e$):

$$\widetilde{\mathbf{e}}^e=\widetilde{\mathbf{e}}_I^e+\widetilde{\mathbf{e}}_{II}^e \tag{57}$$

In terms of transverse shear metric components, we have the modified MITC4 interpolation (cf. [9]), noting that tensor notation is used again:

$$\widetilde{\boldsymbol{e}}_I^e=\boldsymbol{y}_b^T\left[\begin{array}{ccc}0 & 0 & \left(\frac{2-\xi_2}{4}\right)m_{13A}+\left(\frac{2+\xi_2}{4}\right)m_{13B}\\ 0 & 0 & \left(\frac{2-\xi_1}{4}\right)m_{23C}+\left(\frac{2+\xi_1}{4}\right)m_{23D}\\ \left(\frac{2-\xi_2}{4}\right)m_{13A}+\left(\frac{2+\xi_2}{4}\right)m_{13B} & \left(\frac{2-\xi_1}{4}\right)m_{23C}+\left(\frac{2+\xi_1}{4}\right)m_{23D} & 0\end{array}\right]\boldsymbol{y}_b \tag{58}$$

where:

$$m_{13A} = m_{13}\big|_{\xi_1\to 0,\xi_2\to -1,\xi_3\to 0} \tag{59}$$
$$m_{13B} = m_{13}\big|_{\xi_1\to 0,\xi_2\to 1,\xi_3\to 0} \tag{60}$$
$$m_{23C} = m_{23}\big|_{\xi_1\to -1,\xi_2\to 0,\xi_3\to 0} \tag{61}$$
$$m_{23D} = m_{23}\big|_{\xi_1\to 1,\xi_2\to 0,\xi_3\to 0} \tag{62}$$

In addition, at the element level, the assumed strain is given by its components in covariant/contravariant coordinates, properly transformed by the 6×6 matrix $\boldsymbol{T}_E(\overline{\boldsymbol{x}}_b,\overline{\boldsymbol{y}}_b)$:

$$\widetilde{\mathbf{e}}_{II}^e=\mathscr{L}^{-1}\boldsymbol{T}_E^{-1}(\overline{\boldsymbol{x}}_b,\overline{\boldsymbol{y}}_b)\boldsymbol{Q}\underbrace{\boldsymbol{M}^{-1}\int_{\Omega_b}\boldsymbol{Q}^T\boldsymbol{T}_E(\overline{\boldsymbol{x}}_b,\overline{\boldsymbol{y}}_b)\mathscr{L}\mathbf{e}^e\mathrm{d}\Omega_b}_{\boldsymbol{\alpha}} \tag{63}$$

Where $\boldsymbol{\alpha}$ is a constant for each element, i.e. the dependence on $\boldsymbol{\xi}$ is completely contained in $\boldsymbol{Q}$. The interpolation matrix is given as:

$$\boldsymbol{Q}(\xi_1,\xi_2)=\left[\begin{array}{ccccc}1&\xi_2&0&0&0\\0&0&1&\xi_1&0\\0&0&0&0&0\\0&0&0&0&1\\0&0&0&0&0\\0&0&0&0&0\end{array}\right] \tag{64}$$

and the constitutive matrix $\mathscr{L}$ is given by:

$$\mathscr{L}=\mathscr{C}^{-1} \tag{65}$$

Matrix (64) is the Pian-Sumihara stress mode matrix [27] with:

- Affine direct in-plane strains.
- Constant shear in-plane strain.

As an illustration, in-plane and out-of-plane bending results for one element are shown in Figure 2. In-plane solution recovers 81.281% of the exact result and out-of-plane solution recovers the full 100% exact result. Two elements are sufficient to obtain the full in-plane solution.

3 Hybrid element technology

3.1 Tetrahedron

Quasi-incompressible constitutive laws severely reduce the performance of low-order displacement-based elements. This is due to the number of constraints present in low-order elements and the inability of displacement-based elements to separate incompressibility constraints from quadrature points [20, 6]. Usually, mixed elements are a solution for avoiding locking in quasi-incompressible problems. The low-order MINI element by Douglas Arnold ([13]), see also Bathe [15] for the two-field method, is based on a two-field formulation where:

- Pressure is linearly interpolated using the corner nodes.
- An internal shape function, called a bubble, enriches the velocity or displacement fields.

This passes the inf-sup condition and is easily extended to finite strains. Since Cauchy stress is calculated from the constitutive stress $\mathbf{S}_{ab}^{\star b}$ as:

$$\boldsymbol{\sigma}_a^a=\frac{1}{J_{ab}}\mathbf{S}_{ab}^{\star b} \tag{66}$$

, Cauchy pressure is obtained as:

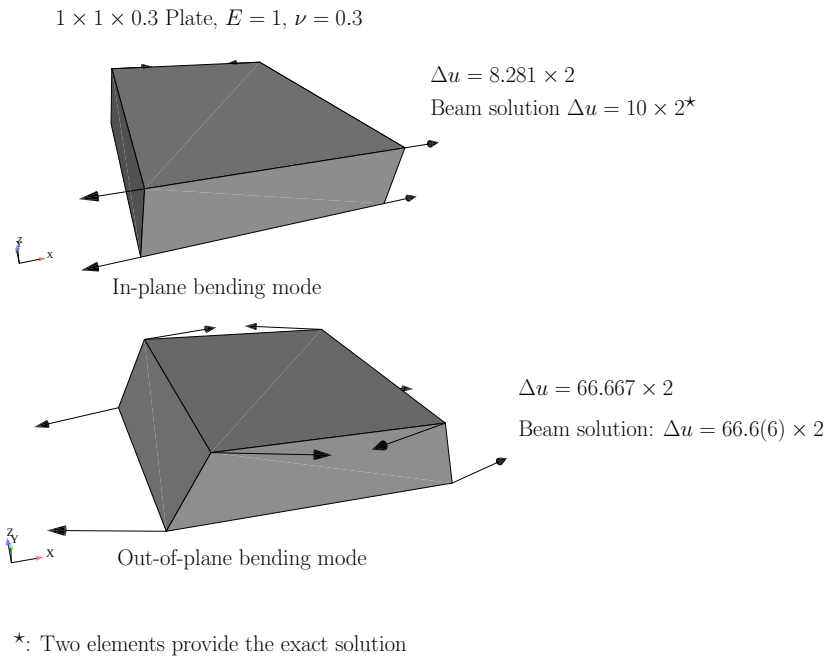


Figure 2: In-plane and out-of-plane bending results for one shell element. Poisson effect is obtained from the use of matrix $\mathscr{L}$.

$$p_a = -\frac{\left(\mathbf{S}_{ab}^{\star b}\right)^T \mathbf{I}_3}{3J_{ab}} \tag{67}$$

We can therefore write $\mathbf{S}_{ab}^{\star b}$ in Voigt form as a sum of deviatoric and pressure terms:

$$\mathbf{S}_{ab}^{\star b} = -J_{ab} p_a \mathbf{I}_3 + \mathbf{T}_{\rm dev} \mathbf{S}_{ab}^{\star b} \tag{68}$$

where $\mathbf{T}_{\rm dev}$ is the following sparse matrix:

$$\mathbf{T}_{\rm dev} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{69}$$

In terms of power balance, we use the following relation, where $\widetilde{\mathbf{S}}_{ab}^{\star b}$ depends on the independent pressure $\widetilde{p}$. It corresponds to a classical two-field variational principle:

$$\underbrace{\int_{\Omega_b} \left(\widetilde{\mathbf{S}}_{ab}^{\star b}\right)^T \dot{\mathbf{e}}_{ab}^b \mathrm{d}\Omega_b + \int_{\Omega_b} \left(J_{ab}\widetilde{p} + \frac{\left(\mathbf{S}_{ab}^{\star b}\right)^T \mathbf{I}_3}{3} \right) \dot{\widetilde{p}} \mathrm{d}\Omega_b}_{\dot{W}_{\rm int}} = \dot{W}_{\rm ext} \tag{70}$$

where the relative Jacobian J_{ab} is used to ensure correct volume calculation. We note that the product $J_{ab}\widetilde{p}$ cannot be used as an unknown field. In (70), we have the following quantities:

$$\widetilde{\mathbf{S}}_{ab}^{\star b} = -J_{ab}\widetilde{p}\mathbf{I}_3 + \mathbf{T}_{\rm dev}\mathbf{S}_{ab}^{\star b} \tag{71}$$

Discretization follows the standard MINI formulation:

$$\boldsymbol{u}(\boldsymbol{\xi}) = \sum_{K=1}^{5} N_K(\boldsymbol{\xi})\, \boldsymbol{u}_K \tag{72}$$

Independent pressure $\widetilde{p}$ is interpolated using the corner nodes:

$$\widetilde{p}(\boldsymbol{\xi}) = \sum_{K=1}^{4} N_K(\boldsymbol{\xi})\, \widetilde{p}_K \tag{73}$$

$$N_1(\boldsymbol{\xi}) = 1-\xi_1-\xi_2-\xi_3 \tag{74a}$$
$$N_2(\boldsymbol{\xi}) = \xi_2 \tag{74b}$$
$$N_3(\boldsymbol{\xi}) = \xi_3 \tag{74c}$$
$$N_4(\boldsymbol{\xi}) = \xi_1 \tag{74d}$$

The bubble function, for tetrahedra, is given by:

$$N_5(\boldsymbol{\xi}) = \xi_1\xi_2\xi_3\left(1-\xi_1-\xi_2-\xi_3\right) \tag{74e}$$

For the calculation of the stiffness matrix, the variation of (70) is required. Not all quantities are determined by hand-derivation, and we use Mathematica [31] with the AceGen (cf. [23]) add-on to calculate some derivatives. Using (70), we obtain:

$$\mathrm{d}\dot{W}_{\mathrm{int}} = \int_{\Omega_b}\left(\left(\mathrm{d}\widetilde{\mathbf{S}}^{\star b}_{ab}\right)^T\dot{\mathbf{e}}^b_{ab}+\left(\widetilde{\mathbf{S}}^{\star b}_{ab}\right)^T\mathrm{d}\dot{\mathbf{e}}^b_{ab}\right)\mathrm{d}\Omega_b \tag{75}$$
$$+\int_{\Omega_b}\left(\mathrm{d}J_{ab}\widetilde{p}+J_{ab}\mathrm{d}\widetilde{p}+\frac{\left(\mathrm{d}\mathbf{S}^{\star b}_{ab}\right)^T\mathbf{I}_3}{3}\right)\dot{\widetilde{p}}\mathrm{d}\Omega_b \tag{76}$$

where the following notation was used:

$$\mathrm{d}\mathbf{S}^{\star b}_{ab} = \mathscr{C}_{ab}\mathrm{d}\mathbf{e}^b_{ab} \tag{77}$$
$$\mathrm{d}\widetilde{\mathbf{S}}^{\star b}_{ab} = \mathbf{T}_{\mathrm{dev}}\mathscr{C}_{ab}\mathrm{d}\mathbf{e}^b_{ab}-\mathrm{d}J_{ab}\widetilde{p}\mathbf{I}_3-J_{ab}\mathrm{d}\widetilde{p}\mathbf{I}_3 \tag{78}$$

Specifically, the terms $\mathrm{d}\mathbf{e}^b_{ab}$ and $\mathrm{d}J_{ab}$ are determined by AceGen. A specialization for triangles is straightforward and therefore omitted. A depiction representing the degree-of-freedom distribution of a MINI tetrahedron is shown in Figure 3.
In terms of rotation tensor, we use the following process to obtain $\overline{\boldsymbol{e}}_1$, $\overline{\boldsymbol{e}}_2$ and $\overline{\boldsymbol{e}}_3$. Since degrees-of-freedom associated to node each local node 5 are internal to the element, we condense them out.

A note on the axisymmetric and plane-strain triangles

Dimensional reduction for plane-strain requires careful consideration of out-of-plane stress, which is in general non-null. We therefore explicitly enforce the plane-strain condition by setting 13 and 23 components of both stress and strain as zero, but including the out-of-plane stress $\left[\widetilde{\boldsymbol{S}}^{\star b}_{ab}\right]_{33}$ as active. Therefore, for plane strain, we only omit out-of-plane shear components of stress and strain. In the axisymmetric case, we use reduced integration, which is sufficient to remove locking in the near-incompressible case.

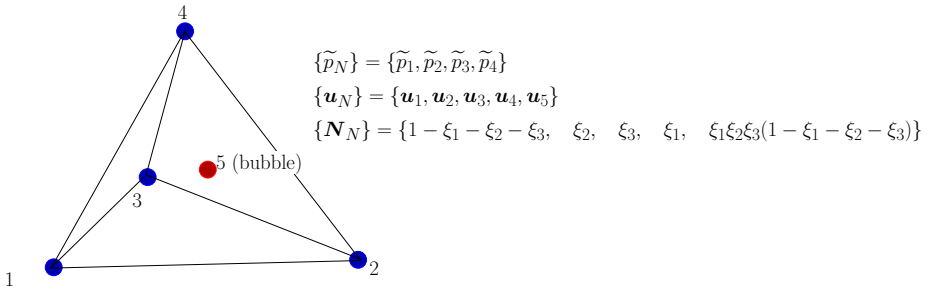


Figure 3: MINI tetrahedron.

4 Contact

Figure 4 summarizes the ingredients for *discrete* contact with Coulomb friction. Given the contact normal $\boldsymbol{n}$, the following rank-one tensor

$$\boldsymbol{Q} = \boldsymbol{n} \otimes \boldsymbol{n}, \tag{79}$$

the friction coefficient μ and the contact force $\boldsymbol{f}$, the Coulomb friction cone is defined as:

$$K_\mu = \left\{ \boldsymbol{f} \in \mathbb{R}^3 | \boldsymbol{f} \cdot \boldsymbol{f} - (1+\mu^2) \boldsymbol{f} \cdot (\boldsymbol{Q}\boldsymbol{f}) \leq 0 \right\} \tag{80}$$

Note that $\boldsymbol{Q}$ is symmetric and corresponds to a projection, i.e. $\boldsymbol{Q} \cdot \boldsymbol{Q} = \boldsymbol{Q}$. All surface forces (either contact or no-contact) are members of K_μ. The *no contact* case corresponds to a contact force belonging to the cone apex. The corresponding polar cone is given as:

$$\begin{aligned} K_{\mu^\star} &= \{\boldsymbol{v} \in \mathbb{R}^3 | \boldsymbol{v} \cdot \boldsymbol{w} \leq 0, \forall \boldsymbol{w} \in K_\mu\} && (81) \\ &\equiv \{\boldsymbol{f} \in \mathbb{R}^3 — \mu \| (\boldsymbol{I} - \boldsymbol{Q}) \boldsymbol{f} \| \leq -\boldsymbol{n} \cdot \boldsymbol{f}\} && (82) \end{aligned}$$

where $\| \bullet \|$ is the Euclidean norm. Using the relative velocity $\dot{\boldsymbol{g}}$ at the contact point, which is power-conjugate to the contact force $\boldsymbol{f}$, the Coulomb friction model is written as the following equality and inequality system:

$$\begin{aligned} \boldsymbol{f}^T \boldsymbol{Q} \dot{\boldsymbol{g}} &= 0 \quad \text{absence of normal dissipation} && (83) \\ \boldsymbol{n} \cdot \boldsymbol{f} &\geq 0 \quad \text{absence of adhesion} && (84) \\ \boldsymbol{n} \cdot \dot{\boldsymbol{g}} &\geq 0 \quad \text{impenetrability} && (85) \\ (\boldsymbol{I} - \boldsymbol{Q}) \dot{\boldsymbol{g}} &= \dot{\gamma} (\boldsymbol{I} - \boldsymbol{Q}) \boldsymbol{f} \quad \text{tangential flow law} && (86) \\ \dot{\gamma} \left[\boldsymbol{f} \cdot \boldsymbol{f} - \left(1+\mu^2\right) \boldsymbol{f} \cdot (\boldsymbol{Q}\boldsymbol{f}) \right] &= 0 \quad \text{tangential complementarity} && (87) \\ \dot{\gamma} &\geq 0 \quad \text{positive tangential dissipation} && (88) \\ \left[\boldsymbol{f} \cdot \boldsymbol{f} - \left(1+\mu^2\right) \boldsymbol{f} \cdot (\boldsymbol{Q}\boldsymbol{f}) \right] &\leq 0 \quad \text{tangential yield criterion} && (89) \end{aligned}$$

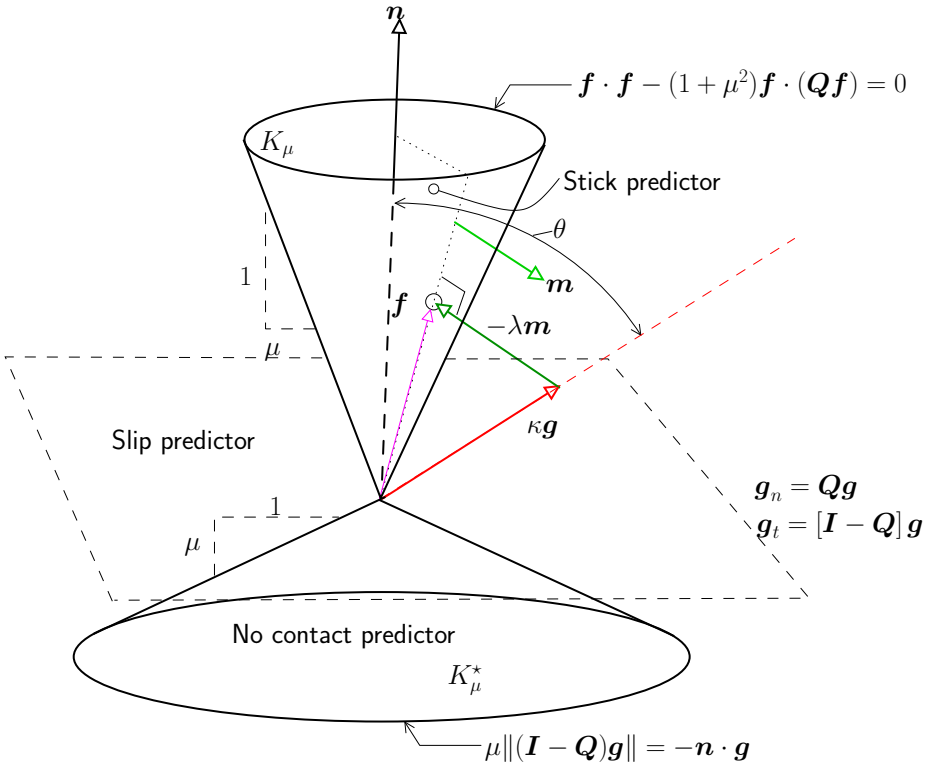


Figure 4: Coulomb friction cone, predictor step (κ is a single penalty).

Further details concerning this topic are given in the book by Wriggers, cf. [40].
Objective time integration of the relative velocity $\dot{\boldsymbol{g}}$ (see, e.g. [24, 40]) results in the so-called gap vector $\boldsymbol{g}$. Figure 4 presents the relevant data corresponding to this problem in a graphic form. Given the representation in Figure 4, it is advantageous to use an additive decomposition of the contact force $\boldsymbol{f}$,

$$\boldsymbol{f}=\kappa\boldsymbol{g}-\lambda\boldsymbol{m} \tag{90}$$

where κ is the single penalty parameter, λ is a scalar unknown and $\boldsymbol{m}$ is the cone's outer normal. We also introduce a unit surface normal $\boldsymbol{n}$. This results in the following convenient definitions for the yield function and the normal vector:

$$\varphi = \boldsymbol{f}\cdot\boldsymbol{f}-(1+\mu^2)\boldsymbol{f}\cdot(\boldsymbol{Q}\boldsymbol{f}) \tag{91}$$

$$\boldsymbol{m} = \tfrac{1}{2}\,\frac{\partial\varphi}{\partial\boldsymbol{f}}=\left[\boldsymbol{I}-(1+\mu^2)\boldsymbol{Q}\right]\boldsymbol{f} \tag{92}$$

Making use of (90) and (91), condition $\varphi=0$ reads:

$$\boldsymbol{g}^T\boldsymbol{H}(\lambda)^T\boldsymbol{H}(\lambda)\boldsymbol{g} - \left(1+\mu^2\right)\boldsymbol{g}^T\boldsymbol{H}(\lambda)^T\boldsymbol{Q}\boldsymbol{H}(\lambda)\boldsymbol{g} = 0 \tag{93}$$

where

$$\boldsymbol{H}(\lambda) = \left[(1+\lambda)\,\boldsymbol{I} - \lambda\left(1+\mu^2\right)\boldsymbol{Q}\right]^{-1} \tag{94}$$

Note that $\det\left[(1+\lambda)\,\boldsymbol{I} - \lambda\,(1+\mu^2)\,\boldsymbol{Q}\right] = (1+\lambda)^2\,(1-\lambda\mu^2)$. Since the solution requires $\lambda \geq 0$, matrix $\boldsymbol{H}(\lambda)$ does not exist if $\lambda = 1/\mu^{\mathbf{2}}$. Solving the equation $\varphi = 0$ for λ, we obtain two roots, the smallest one is the closest projection on the friction cone:

$$\lambda_{\text{sol}} = \frac{(\boldsymbol{g}\cdot\boldsymbol{g})\,\mu - (1+\mu^2)\sqrt{(\boldsymbol{g}\cdot\boldsymbol{n})^2\,(\boldsymbol{g}\times\boldsymbol{n})\cdot(\boldsymbol{g}\times\boldsymbol{n})}}{\mu\left[(1+\mu^2)\,(\boldsymbol{g}\times\boldsymbol{n})\cdot(\boldsymbol{g}\times\boldsymbol{n}) - (\boldsymbol{g}\cdot\boldsymbol{g})\right]} \tag{95}$$

we note that λ_{sol} is independent of κ. In addition, using the internal angle θ between $\boldsymbol{n}$ and $\boldsymbol{g}$, λ_{sol} is re-written as:

$$\lambda_{\text{sol}} = \frac{\mu - (1+\mu^2)\,|\cos\theta\sin\theta|}{\mu\left[(1+\mu^2)\sin^2\theta - 1\right]} \tag{96}$$

Since the absolute value of $\sin(2\theta)/2$ is apparent in equation (96), we have non-differentiability of λ_{sol} for $\theta = \frac{\pi}{2}i$, $i \in \mathbb{Z}$. Two common situation correspond to either perfect orthogonality or perfect alignment between $\boldsymbol{n}$ and $\boldsymbol{g}$. The derivative of λ_{sol} with respect to θ is given by:

$$\begin{aligned}\frac{\mathrm{d}\lambda_{\text{sol}}}{\mathrm{d}\theta} \;=\; & \frac{(\mu^2+1)\left\{\cos(2\theta)\left[(\mu^2+1)\cos(2\theta) - \mu^2 + 1\right]|\sin(\theta)\cos(\theta)|'\right\}}{2\mu\left((\mu^2+1)\sin^2(\theta) - 1\right)^2} + \\ & \frac{(\mu^2+1)\sin(2\theta)\left[(\mu^2+1)\,|\sin(2\theta)| - 2\mu\right]}{2\mu\left((\mu^2+1)\sin^2(\theta) - 1\right)^2}\end{aligned} \tag{97}$$

where

$$|\bullet|' = \begin{cases} 1 & \bullet > 0 \\ \text{undefined} & \bullet = 0 \\ -1 & \bullet < 0 \end{cases} \tag{98}$$

Given (97), we determine the derivatives of λ_{sol} with respect to $\boldsymbol{n}$ and $\boldsymbol{g}$ as:

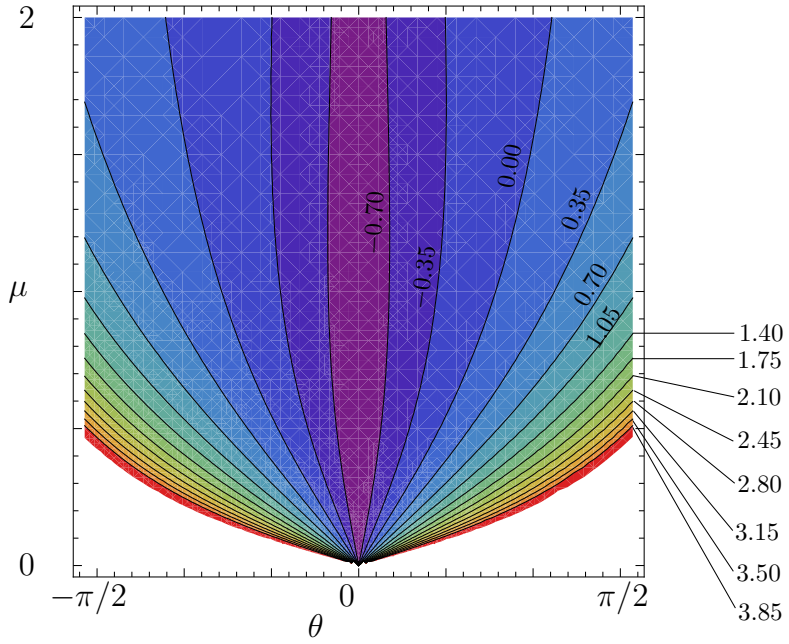


Figure 5: λ contour plot as a function of θ and μ. Negative values of λ correspond to the interior of the cone.

$$\frac{\partial \lambda_{\text{sol}}}{\partial \boldsymbol{g}} = \frac{\mathrm{d}\lambda_{\text{sol}}}{\mathrm{d}\theta} \frac{1}{(\boldsymbol{g}\cdot\boldsymbol{g})\sqrt{(\boldsymbol{g}\times\boldsymbol{n})\cdot(\boldsymbol{g}\times\boldsymbol{n})}} \left\{ \begin{array}{c} -g_2^2 n_1 + g_1 g_2 n_2 + g_3(-g_3 n_1 + g_1 n_3) \\ g_1 g_2 n_1 - g_1^2 n_2 + g_3(-g_3 n_2 + g_2 n_3) \\ g_3(g_1 n_1 + g_2 n_2) - (g_1^2 + g_2^2) n_3 \end{array} \right\}$$

$$\frac{\partial \lambda_{\text{sol}}}{\partial \boldsymbol{n}} = \frac{\mathrm{d}\lambda_{\text{sol}}}{\mathrm{d}\theta} \frac{1}{\sqrt{(\boldsymbol{g}\times\boldsymbol{n})\cdot(\boldsymbol{g}\times\boldsymbol{n})}} \left\{ \begin{array}{c} g_2 n_1 n_2 + g_3 n_1 n_3 - g_1(n_2^2 + n_3^2) \\ n_2(g_1 n_1 + g_3 n_3) - g_2(n_1^2 + n_3^2) \\ -g_3(n_1^2 + n_2^2) + (g_1 n_1 + g_2 n_2) n_3 \end{array} \right\}$$

We represent the contour plot of λ_{sol} as a function of θ and μ in Figure 5. The contact force $\boldsymbol{f}$ is therefore determined as:

$$\boldsymbol{f} = \kappa \left[(1 + \lambda_{\text{sol}})\, \boldsymbol{I} - \lambda_{\text{sol}} \left(1 + \mu^2\right) \boldsymbol{Q} \right]^{-1} \boldsymbol{g} \tag{99}$$

The three contact cases are summarized in Algorithm 2. Note that the tree-like decision process of the classical predictor/corrector algorithm (cf. [40]) is circumvented. Advantages of this approach are:

- No requirements for special frames and normal/tangential decompositions, the gap $\boldsymbol{g}$ is defined in global coordinates. As a consequence, the second variation of $\boldsymbol{g}$ is always zero;

Algorithm 2 Displacement contact algorithm for Coulomb friction

Case	Displacement condition	$\boldsymbol{f}$
Case I (no contact)	$\boldsymbol{n}\cdot\boldsymbol{g} < -\mu\|(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{g}\|$	$\boldsymbol{0}$
Case II (stick)	$\boldsymbol{g}\cdot\boldsymbol{g} < (1+\mu^2)\boldsymbol{g}\cdot(\boldsymbol{Q}\boldsymbol{g})$	$\kappa\boldsymbol{g}$
Case III (slip)	$\boldsymbol{g}\cdot\boldsymbol{g} \geq (1+\mu^2)\boldsymbol{g}\cdot(\boldsymbol{Q}\boldsymbol{g})$	$\kappa\left[(1+\lambda_{\rm sol})\,\boldsymbol{I}-\lambda_{\rm sol}\,(1+\mu^2)\,\boldsymbol{Q}\right]^{-1}\boldsymbol{g}$ with $\lambda_{\rm sol}=\frac{(\boldsymbol{g}\cdot\boldsymbol{g})\mu-\left(1+\mu^2\right)\sqrt{(\boldsymbol{g}\cdot\boldsymbol{n})^2(\boldsymbol{g}\times\boldsymbol{n})\cdot(\boldsymbol{g}\times\boldsymbol{n})}}{\mu[(1+\mu^2)(\boldsymbol{g}\times\boldsymbol{n})\cdot(\boldsymbol{g}\times\boldsymbol{n})-(\boldsymbol{g}\cdot\boldsymbol{g})]}$
Input: $\boldsymbol{n}$, $\boldsymbol{g}$ and μ,κ		
Output: $\boldsymbol{f}$, $\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{g}}$, $\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{n}}$		

- Decoupling of kinematic quantities $\boldsymbol{n}$ and $\boldsymbol{g}$ (calculated separately) from the contact algorithm;
- Unified approach for 2D and 3D (only the definition of $\boldsymbol{g}$ and $\boldsymbol{n}$ change according to the discretization);
- Quadratic form of the friction cone avoids the ill-conditioning for small values of the tangential displacement;
- Exact return to the friction cone in the slipping case.

Low order prototype contact elements

The gap vector is determined according to the discretization method. We use a variant of node-to-segment (and node-to-face in 3D) method (e.g. Zavarise and De Lorenzis [43] and Wriggers [41]) for low order continuum meshes, which is classical and addresses the well-known corresponding shortcomings. Figure 6 depicts the two geometrical constructions required to provide the Algorithm 2 with the values of $\boldsymbol{g}$ and $\boldsymbol{n}$. We use Mathematica [31] with the Acegen add-on [23] to generate the following quantities from the geometrical construction depicted in Figure 6:

- $\boldsymbol{g} = \boldsymbol{p}_{1a} - \boldsymbol{p}^{\star}_{1a}\left(\boldsymbol{\xi}_b\right)$,
- $\boldsymbol{n} = \boldsymbol{e}_3 \times {}^{(\boldsymbol{p}_{3a}-\boldsymbol{p}_{2a})}\!/_{\|\boldsymbol{p}_{3a}-\boldsymbol{p}_{2a}\|}$ for 2D and $\boldsymbol{n} = {}^{[(\boldsymbol{p}_{2a}-\boldsymbol{p}_{3a})\times(\boldsymbol{p}_{4a}-\boldsymbol{p}_{3a})]}\!/_{\|[(\boldsymbol{p}_{2a}-\boldsymbol{p}_{3a})\times(\boldsymbol{p}_{4a}-\boldsymbol{p}_{3a})]\|}$ for 3D,
- $\frac{\partial\boldsymbol{g}}{\partial\boldsymbol{p}_{Ia}}$ and $\frac{\partial\boldsymbol{n}}{\partial\boldsymbol{p}_{Ia}}$.

For detection, we use the incident node averaged normal to chose a main target edge or face, cf. [4]. This averaged normal is only used for detection of the first edge or face and not used for contact enforcement. For the contact elements, we allow more than one target edge or face to share an incident node.

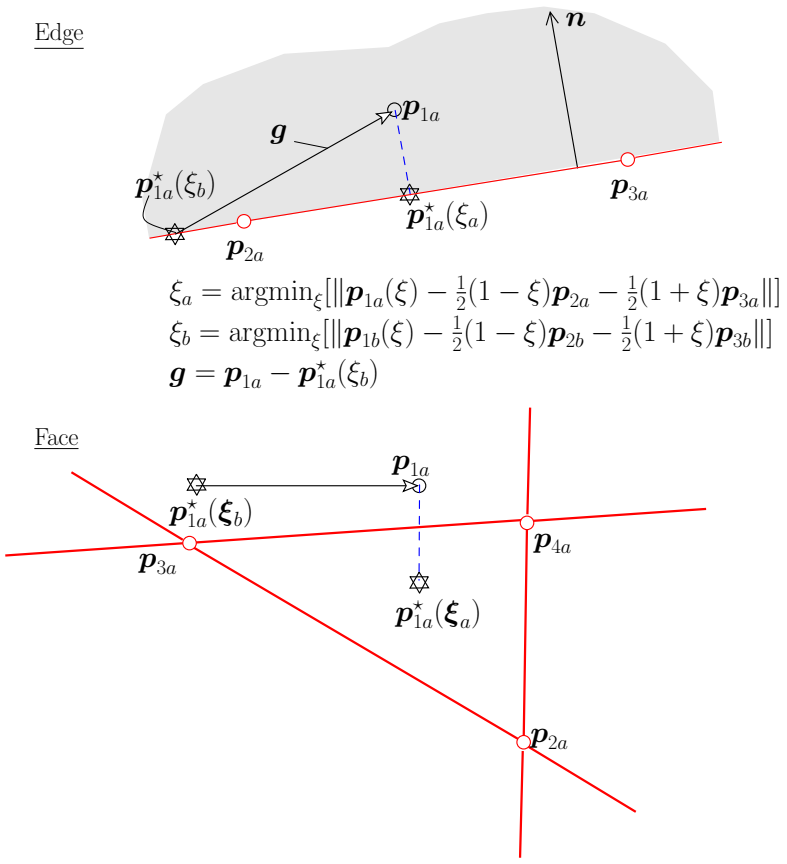


Figure 6: Node-to-segment and node-to-face definition of the gap vector.

4.1 Details on the contact area and the multiple targets of each incident node

We use the area estimation previously introduced by Areias *et al.* [5] and given by:

$$A_{est} = \frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\boldsymbol{n} \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{n})} \tag{100}$$

where $\boldsymbol{\sigma}$ is given by the nodal fitting corresponding to the least-squares using super-convergent patch recovery [44]. Two steps with fixed loading are employed:

- The first step uses $\kappa = \gamma \frac{f_c}{l_c}$ as a penalty.
- The second step uses $\kappa = \gamma \frac{f_c A_{est}}{l_c^2}$ as a penalty .

where γ is a user-defined parameter. This approach can be circumvented if Lagrange multipliers are introduced. Since this entails the discussion of further ingredients, it is omitted in this work. To attenuate the oscillations, each single incident node can have multiple targets simultaneously, as depicted in Figure 7. Note that this approach does not require additional housekeeping in terms of implementation: force and stiffness arrays are added to the global arrays according to the number of targets for each node (in graph theory, this is a clique summation. An interesting alternative is the one by Neto *et al.* [25] who use quadratic smoothing of the target faces (with Nagata patches) to obviate the need for multiple targets.

5 Some examples (shells and constitutive laws)

5.1 Right angle cantilever beam

To assess our rotational approach for frame-invariance, we test a one-dimensional problem consisting of a clamped beam with distributed end-moment (case I) and distributed end-load (case II). This was introduced by Chrścielewski, Makowski and Stumpf [18]. Figure 8 shows the relevant data for this problem and sequences of deformed meshes. Results with the rotational approach show remarkably similarity with the total Lagrangian formulation with the Kirchhoff/Saint-Venant and Neo Hookean hyperelastic models (see Figure 9).

5.2 Channel beam problem

The channel beam problem was proposed by Chrścielewski's group [17], with a longer beam version being introduced by Wagner and Gruttmann [38]. More recently, Chrścielewski and Witkowski [18] compared the results for both versions. We use these references for comparison with our results, in addition to the TUBA element implementation by Vladimirov (cf. [22]) and our corotational *triangle* [11]. Relevant data is shown in Figure 10 for the two cases. In case I, comparison with results of Chrścielewski and Witkowski (EAS14m1), cf. Figure 11, show that the present element is less stiff and we achieve a total displacement of 6 consistent units, when compared with 2.5 in other references. Our

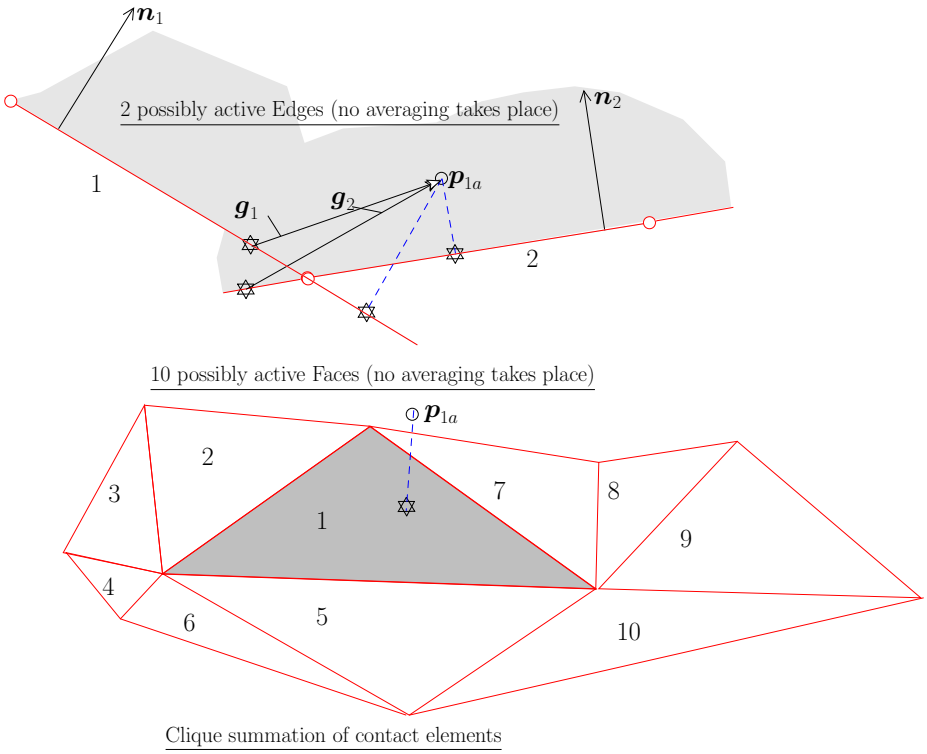


Figure 7: Examples of multiple targets for each incident node.

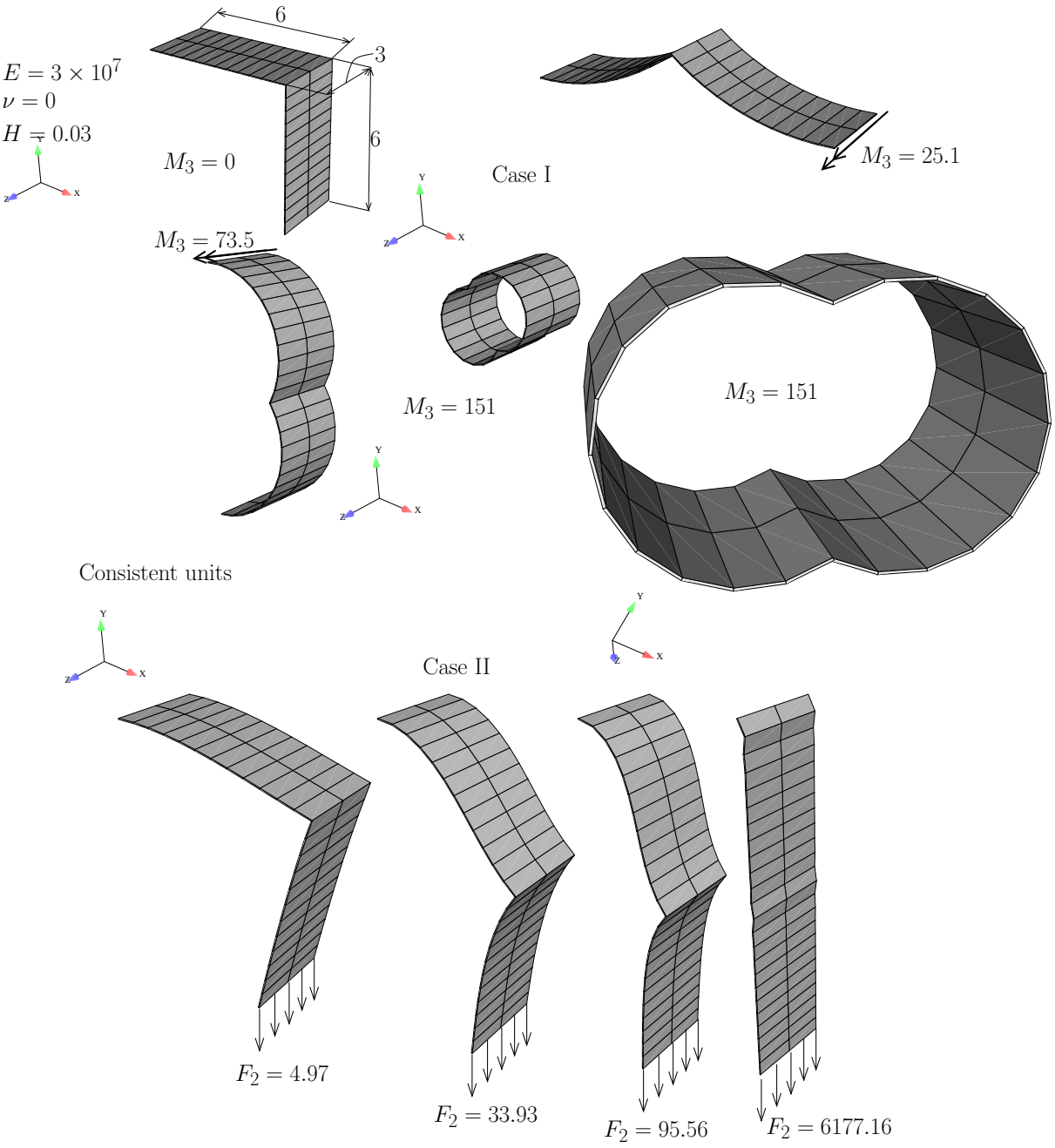
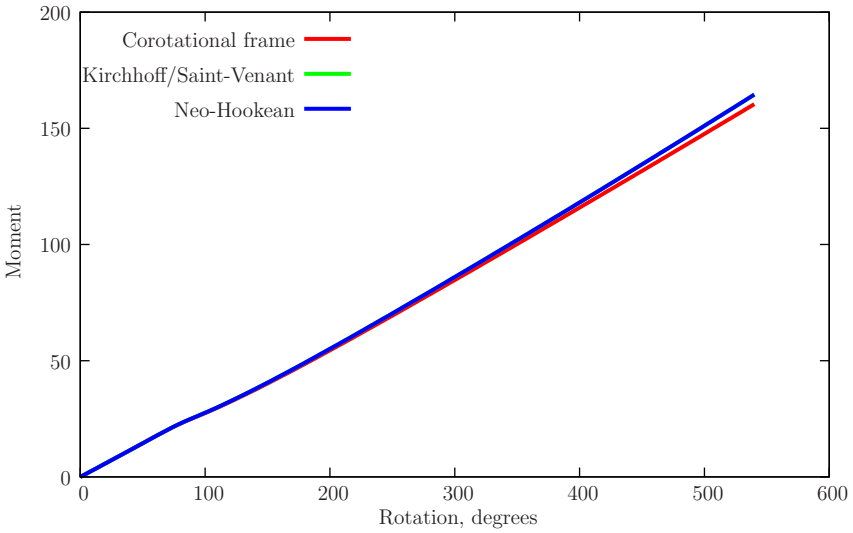
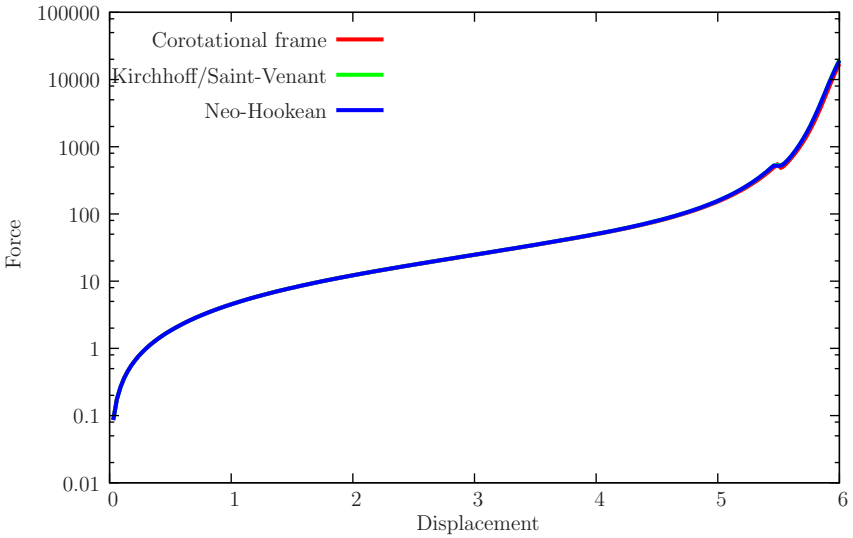


Figure 8: Right angle cantilever beam: sequences for cases I and II



(a) Right-angle frame: case I



(b) Right-angle frame: case II

Figure 9: Right angle cantilever beam: moment/rotation results and force/displacement.

corotational *triangle* (cf. [11]) produces slightly stiffer results with a comparable mesh. Only TUBA 13 triangle, using a sixth order polynomial, achieves more flexible results than the ones of the present element. We also test step-size sensitivity in Figure 12 which perfectly illustrates the robustness of our formulation. Results for case II are shown in Figure 13, where we can observe that Wagner and Gruttmann results are slightly stiffer, a fact also reported by Chrścielewski and Witkowski [18] .

5.3 Ring test

Another essential example for testing corotational frames is the pulled ring introduced by Basar and Ding [14], cf. Figure 14. We here use the well-known solution by Sansour and Kollmann [32] as comparison. The problem consists of a circular ring plate with a radial cut which is clamped on one side and loaded along its radial free edge by a uniform load q, see Figure 14, where the deformed configurations for several values of load q are shown. Load q maintains its direction perpendicular to the original ring plane. Linear control (described in [7]) is adopted to obtain the solution in this problem. A comparison with the results of Sansour and Kollmann is presented in Figure 15. We can observe that much higher values of loading and deformation are achieved by the present element and overall there is good agreement with published results (with a slightly less stiff results with our element).

5.4 Thickness variation test

A test is introduced for comparing three techniques for determination of the thickness:

1. Small-strain zero normal stress condition enforced by using the small-strain $\mathscr{C}_{\text{linear}}$. No thickness update is performed.

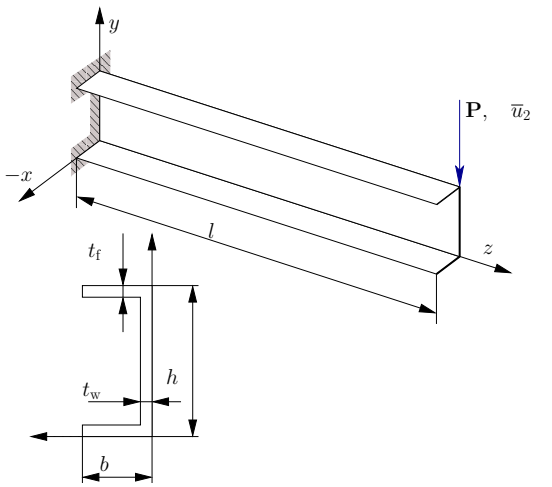
2. Single-parameter thickness update (similar to the approach of Hughes and Carnoy [21]). Thickness update is performed.

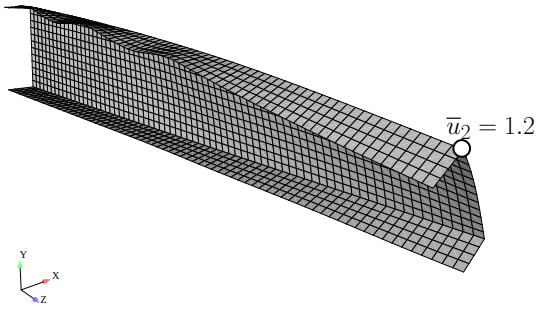
3. The two-parameter thickness update.

The test consists of a beam, depicted in Figure 16. Three constitutive laws are tested:

- Elastic, incremental Kirchhoff Saint-Venant law (here quasi-incompressible).
- Von-Mises yield criterion.
- Rousselier yield criterion (full detailing of the constitutive properties is performed in reference [10]).

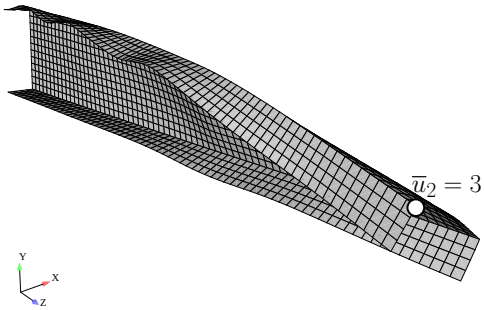
In terms of imposed displacement/load results, Figure 17 shows the relevant results for the three constitutive laws. For the thickness variation, Figure 18 shows the evolution with the imposed displacement. From the observation of these Figures, we conclude the following:





Data for case 1
Geometry:
$l = 36$, $h = 6$, $b = 2$
$t_f = t_w = 0.05$

Loading:
$P = 100$
Material:
$E = 10^7$
$\nu = 0.3$



Data for case 2
Geometry:
$l = 900$, $h = 30$, $b = 10$
$t_f = 1.6$
$t_w = 1.0$
Loading:
$P = 1$
Material:
$E = 21000$
$\nu = 0.3$

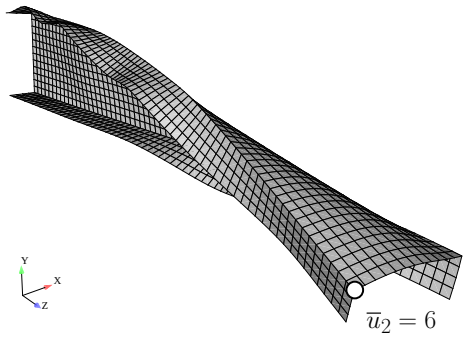


Figure 10: Channel beam: relevant data and 3 deformed configurations for 72 longitudinal elements in the beam web (for case I).

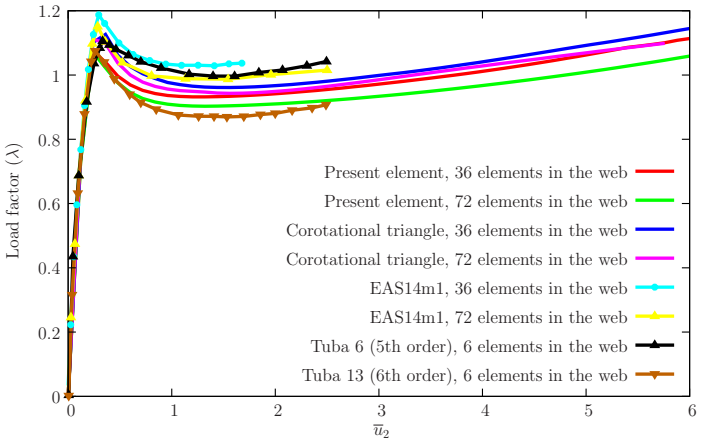


Figure 11: Channel beam, case I: comparison with corotational triangles (cf. [11]), EAS14m1 elements (cf. [18]) and TUBA elements (cf. [22]), 50 steps.

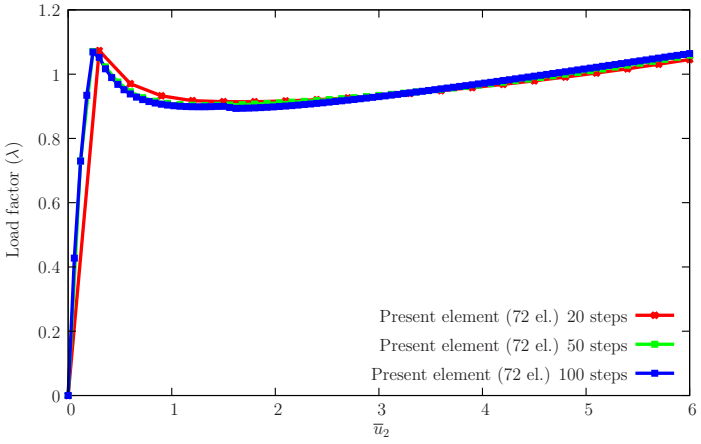


Figure 12: Channel, case I: effect of step size with incremental Kirchhoff/Saint-Venant elasticity.

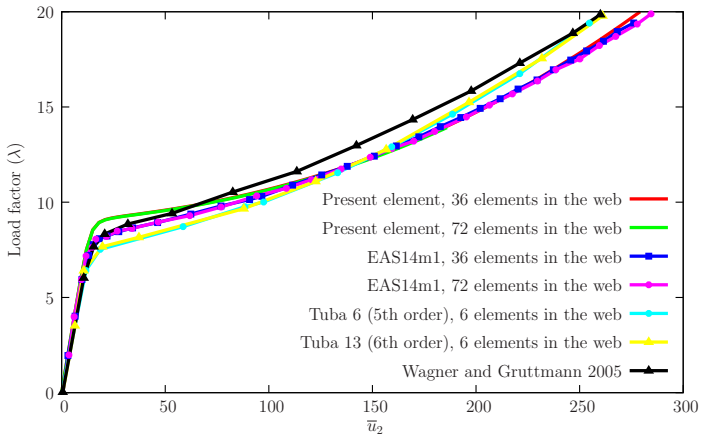


Figure 13: Channel, case II: comparison with EAS14m1 [18], TUBA [22] and Wagner and Gruttmann element (cf. [38]).

- Small-strain zero normal stress typically results in stiffer results, with a more linear reaction curve in the elastic case.
- When compared with the use of one thickness parameter, using two thickness parameters results in a more flexible behavior, as well as different total thickness evolution.
- Quasi-incompressibility in elasticity produces a marked difference between the three approaches.
- The Rousselier yield function is very sensitive to thickness variation. This is caused by the pressure term in the yield function (cf. [10]).

A slight dependence in the number of imposed displacement steps is also observed, since thickness parameters are not treated as degrees-of-freedom.

5.5 Finite strain plasticity shell problems

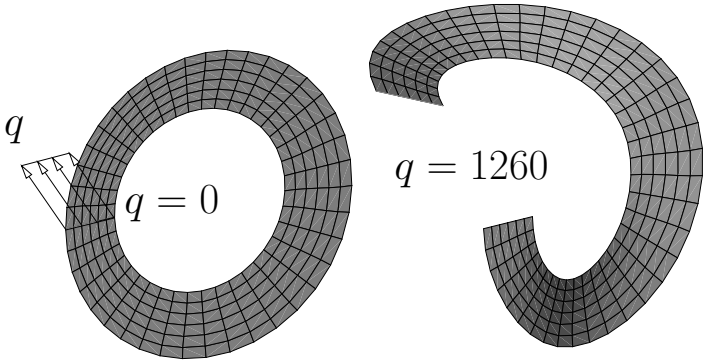
Two classical finite-strain shell problems with thickness extensibility are solved: the pinched cylinder (see Figures 19 and 21) and the plate under pressure (see reference [9] for relevant data and Figure 24). Besides the classical von-Mises plasticity here implemented with our algorithm (radial return *cannot be* used, since the zero normal stress condition invalidates its use), we also test the Hill criterion with local frame angles $\pi/4$ (pinched cylinder) and $\pi/6$ (plate under pressure). For the pinched cylinder, excellent results were obtained with a very small number of steps (5), and these agree with Wagner, Klinkel and Gruttmann [39]. Figure 20 shows the comparison. Robustness is very high and is related to the small elastic strains involved (see also Figures 21 and 22 for the results with the anisotropic Hill criterion). In both isotropic and anisotropic cases, a

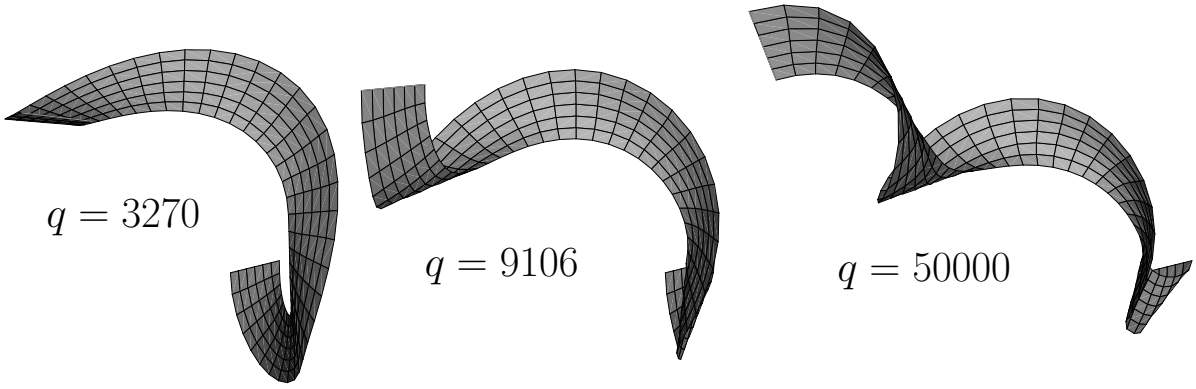
H=3.00000 $\times$ 10^{-02}

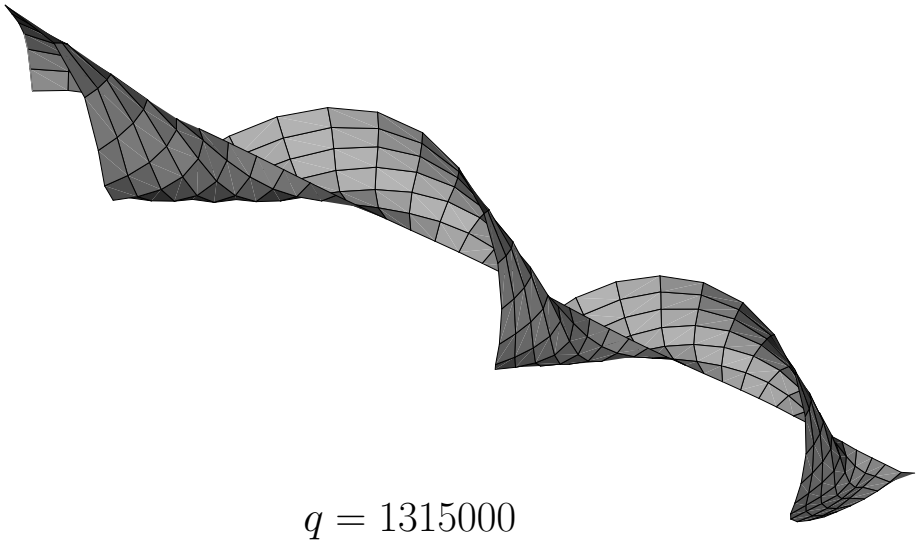
E=2.10000 $\times$ 10^{+10}

R_e=10

R_i=6







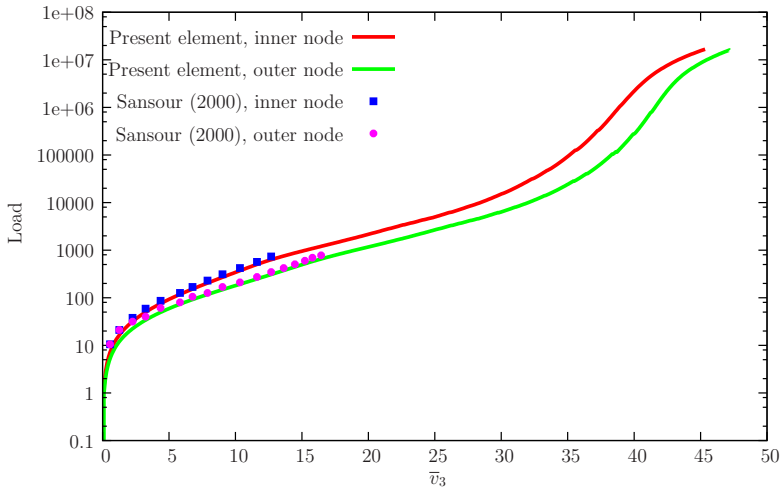


Figure 15: Ring test: comparison with the results of Sansour and Kollmann

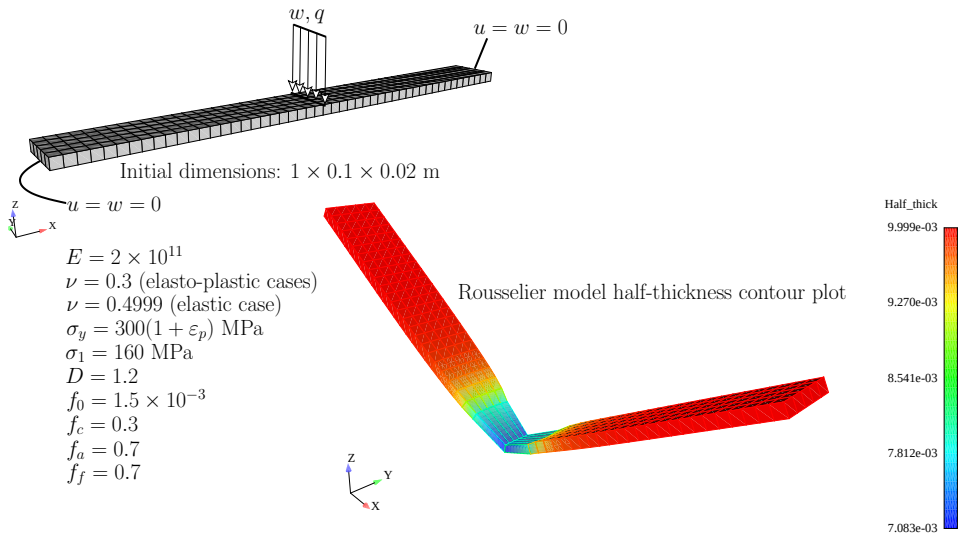
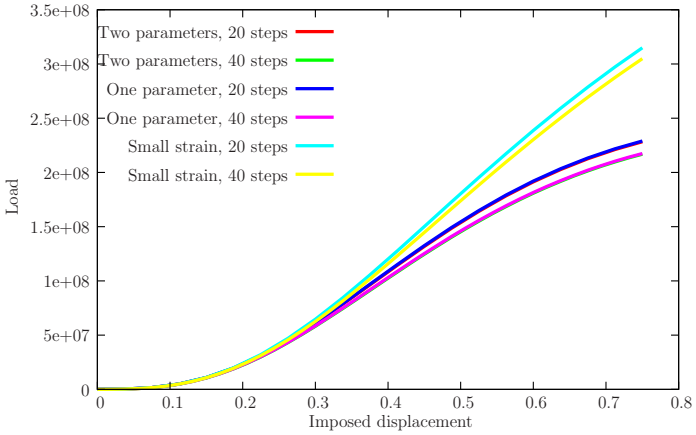
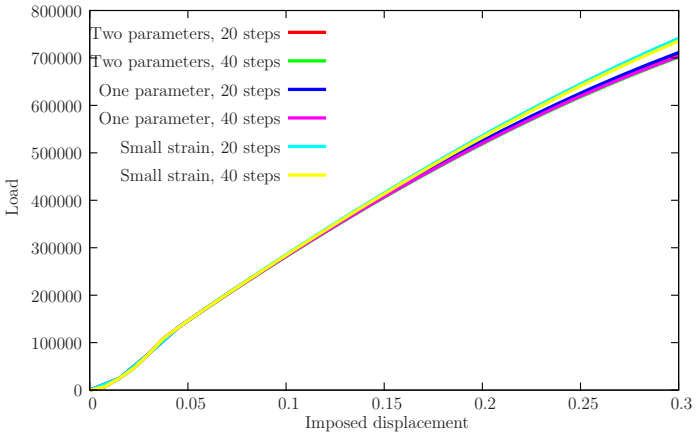


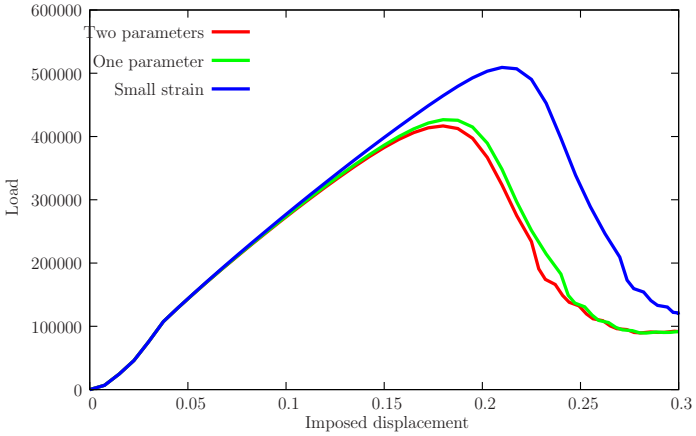
Figure 16: Simply supported beam: relevant data for the elastic case, von-Mises criterion and Rousselier yield criterion.



(a) Kirchhoff Saint-Venant

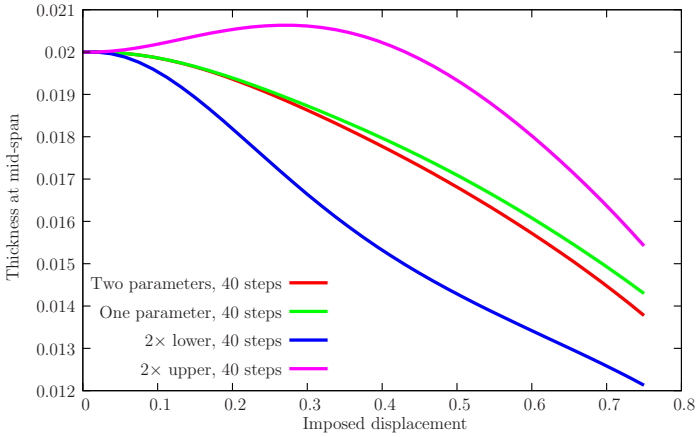


(b) Von-Mises yield function

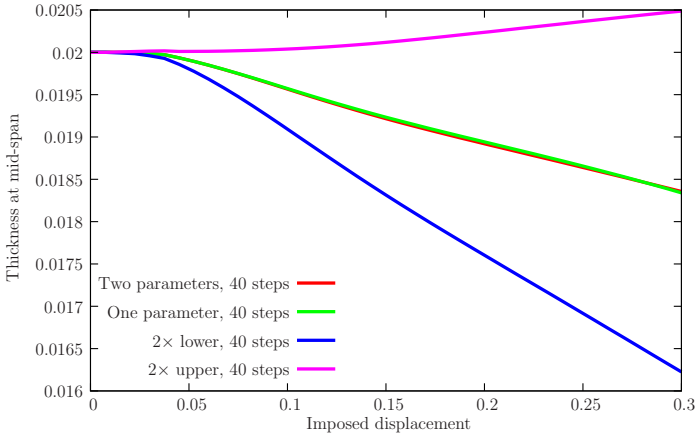


(c) Rousselier yield function

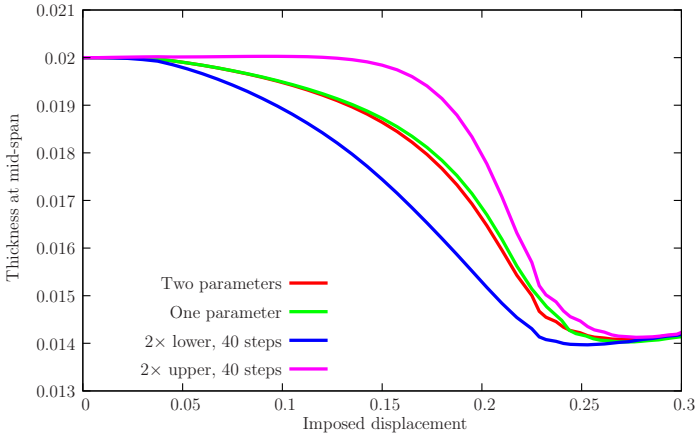
Figure 17: Simply supported beam: imposed displacement/load results for three constitutive models. The label "Small strain" indicates a small-strain approach for the zero normal stress condition.



(a) Kirchhoff Saint-Venant



(b) Von-Mises yield function



(c) Rousselier yield function

Figure 18: Simply supported beam: imposed displacement/mid-span thickness for three constitutive models. The label "Small strain" indicates a small-strain approach for the zero normal stress condition.

high level of robustness is observed. For the plate under pressure, Figures 25 and 26 show some drifting when 5 or 10 steps are used, but overall the results are very robust. Note that it is usual to use 50 or more steps in these problems.

6 Some examples (contact)

We implemented the present algorithm in a simulation code (cf. [3]) and use mixed finite elements to perform the discretization, except when using low order tetrahedra. Differences in element technology affect the results and explain the slightly lower reactions in some problems, when compared with published results.

6.1 2D ironing problem

This benchmark has been used to advocate the use of *mortar* discretizations in contact problems, cf. Yang, Laursen and Meng [42]. Figure 27 shows the relevant geometric and constitutive data, agreeing with the references [42] and [19]. We compare the present approach with the results of these References in Figure 28. Differences exist in the magnitude of forces, and we concluded that this is due to the continuum finite element technology. We use finite strain B-Bar elements (cf. [35]), known to be more compliant than the standard Q4 isoparametric elements. Only slight oscillations in the reactions are obtained and the deformation follows closely what is observed in the mortar case.

6.2 Post buckling of a cylinder

This classical elasto-plastic axisymmetric problem, relevant for crashworthiness studies was introduced by Laursen (cf. [24]). In his work, Laursen used the operator split method with the augmented Lagrangian method to solve the crushing of a cylinder. He used $\mu = 0$ and $\mu = 0.2$ to assess the implementation. We here reproduce this problem, and solve it with $\mu = 0$, $\mu = 0.2$ and $\mu = 1$. We use the consistent elasto-plastic integration algorithm discussed in [8]. Results are shown in Figures 29 and 30 allow the following conclusions:

- For $\mu = 0$, our algorithm predicts the post-buckling behavior earlier than Laursen. In addition, higher imposed displacements are possible with our algorithm. We note that a slightly lower penetration in the rigid fixture when compared with Laursen.
- For $\mu = 0.2$, our algorithm produces very similar results to the ones reported by Laursen.
- We also tested $\mu = 1$ with visible differences with respect to the case $\mu = 0.2$ in the fifth wrinkle.

The sequence of wrinkle formation is shown in Figure 31. Although the pattern is the same as in Laursen (cf. [24]), wrinkles start to form near the top instead of near the bottom of the tube. This is caused by differences in element technology. However, it does not affect the reaction behavior, as can be observed in Figure 30.

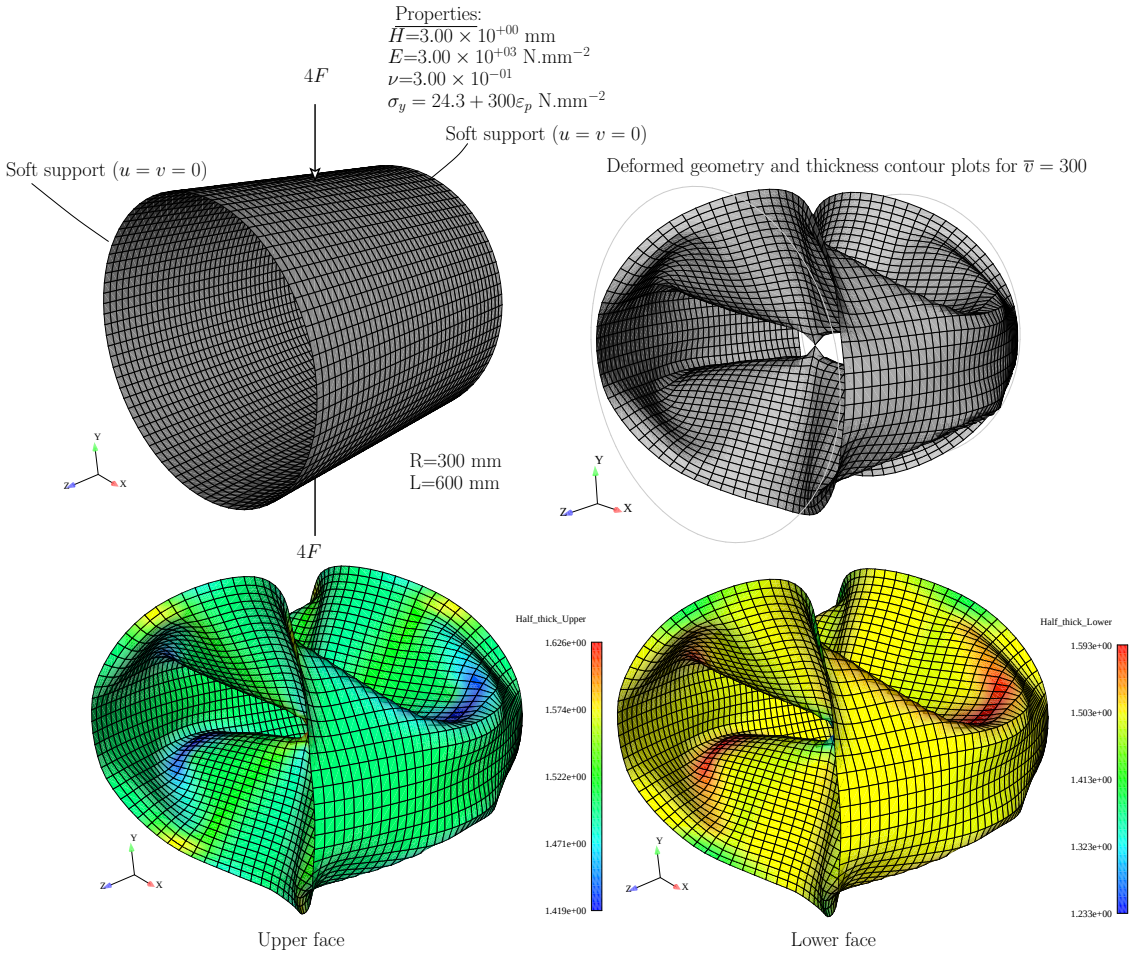


Figure 19: Pinched cylinder with von-Mises yield criterion (full constitutive system with stress condition).

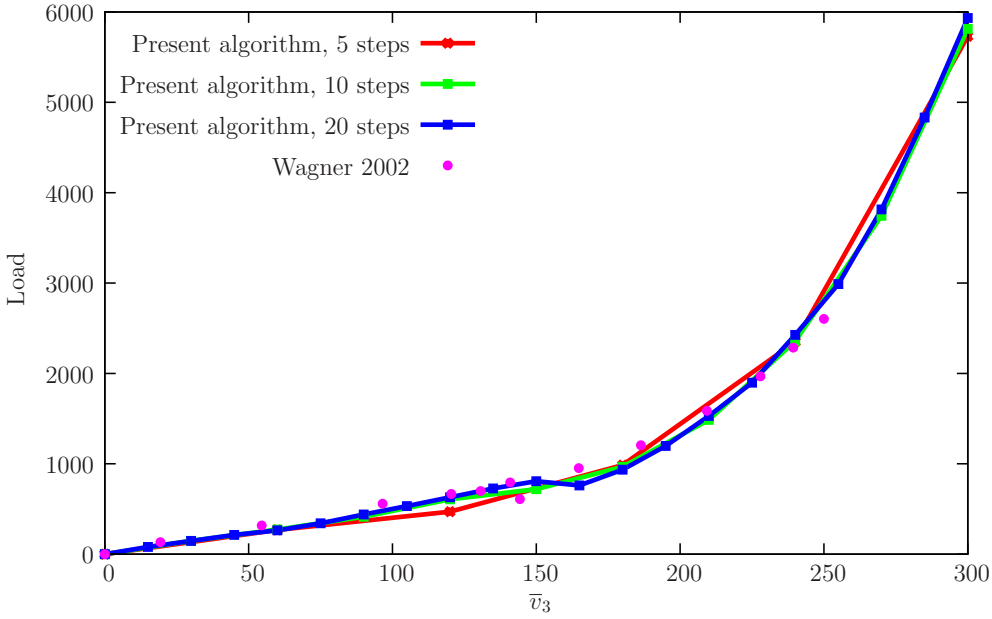


Figure 20: Pinched cylinder: effect of time-step and comparison with Wagner, Klinkel and Gruttmann [39].

6.3 Two classical frictional tests with rings

These are other examples typically adopted within the mortar formulation contexts (cf. Yang *et al.* [42]). We here use these examples to assess the (quadratic) cone projection method for problems typically solved by mortar discretization techniques. Figure 32 shows the relevant data for the two tests. In test I, a Neo-Hookean constitutive law is used and in test II J_2 plasticity is used with our recently proposed algorithm [8]. A sequence of results produced by the proposed algorithm is shown in Figure 33 for both tests. The effective plastic strain contour plot for test II is very similar to the one obtained by Yang *et al.* [42]. The evolution of reactions is also compared with published results in Figure 34. Good agreement with the mortar technique can be observed. We also show results for $\mu = 1$ and $\mu = 2$ and for these high values, slight oscillations appear.
The effect of choosing the upper or lower surface as master is shown in Figure 35 where it can be observed that only the end values differ slightly.

6.4 3D ironing with cylindrical die

We reproduce the problem of Puso and Laursen [30] who detected locking with node-to-segment approach. Relevant problem data is shown in Figure 36. Two stages of displacement of the deformable die (whose center is located at 2.5 units from the slab end) are used: in the first stage, $t \in [0, 0.2]$ the die lowers into the slab a total of 1.4 units.

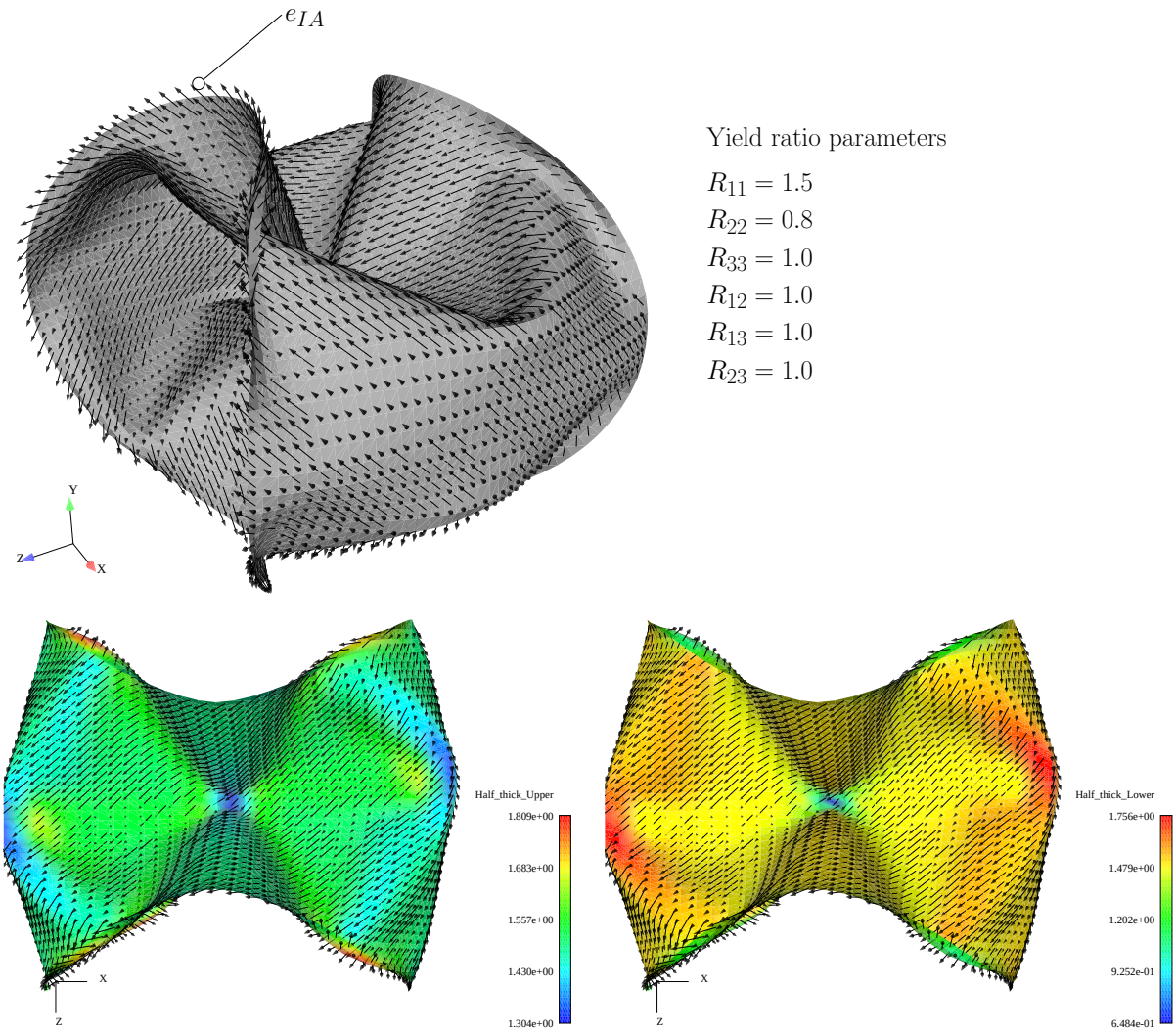


Figure 21: Pinched cylinder with Hill yield criterion (full constitutive system with stress condition). The anisotropic direction I is shown, as well as a top perspective

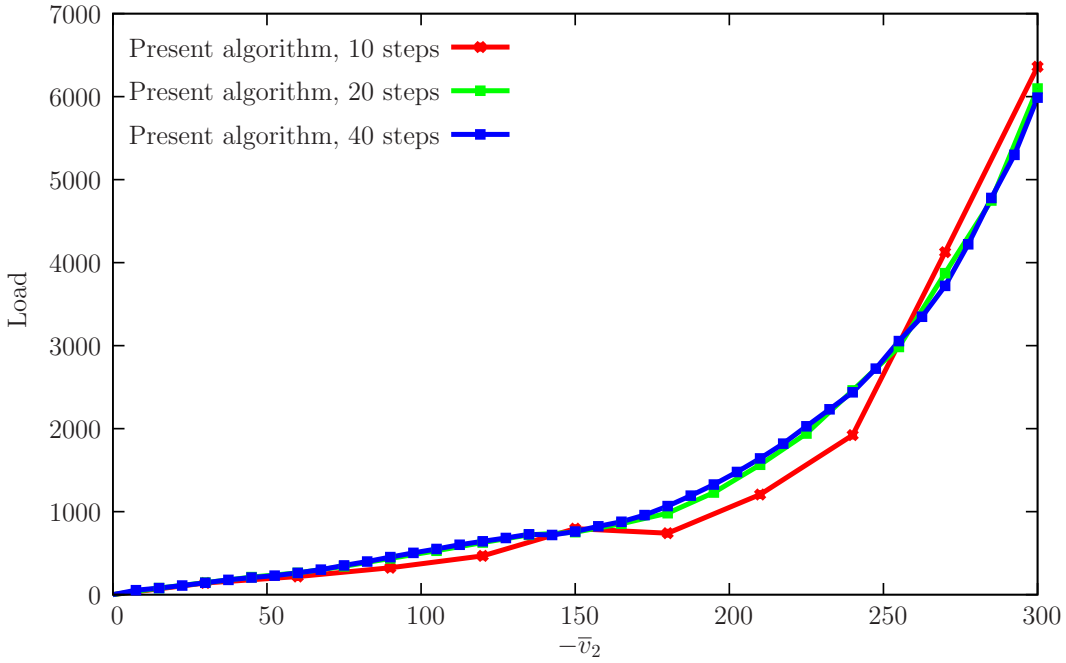


Figure 22: Pinched cylinder with Hill yield criterion. Step size sensitivity.

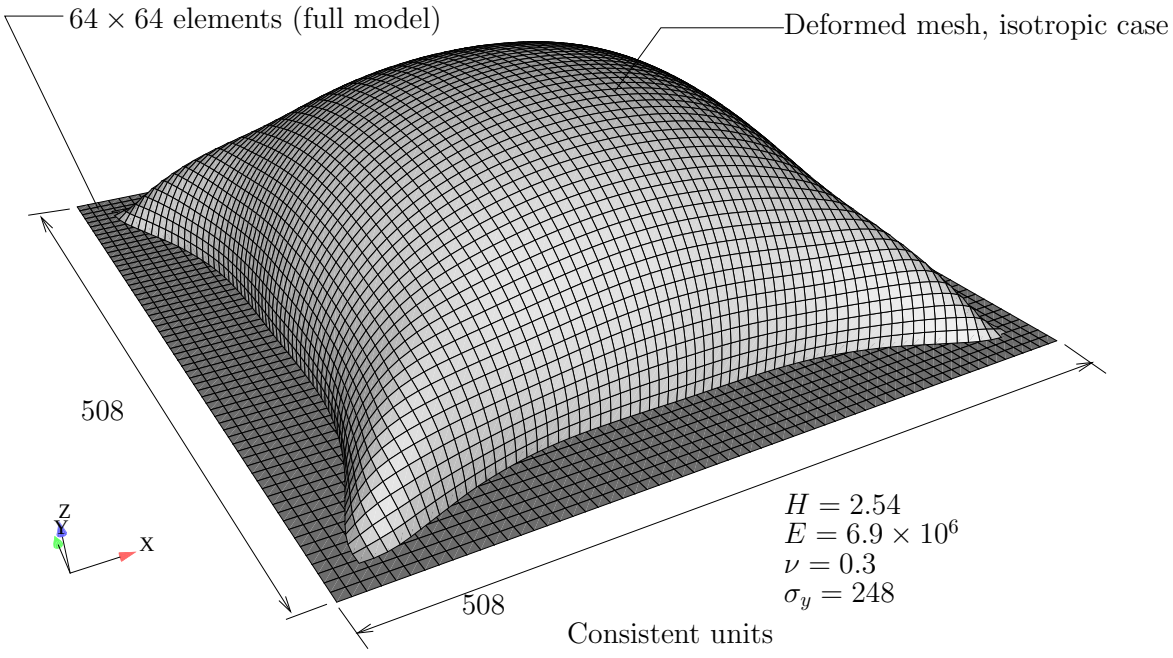


Figure 23: Square plate under pressure: relevant data.

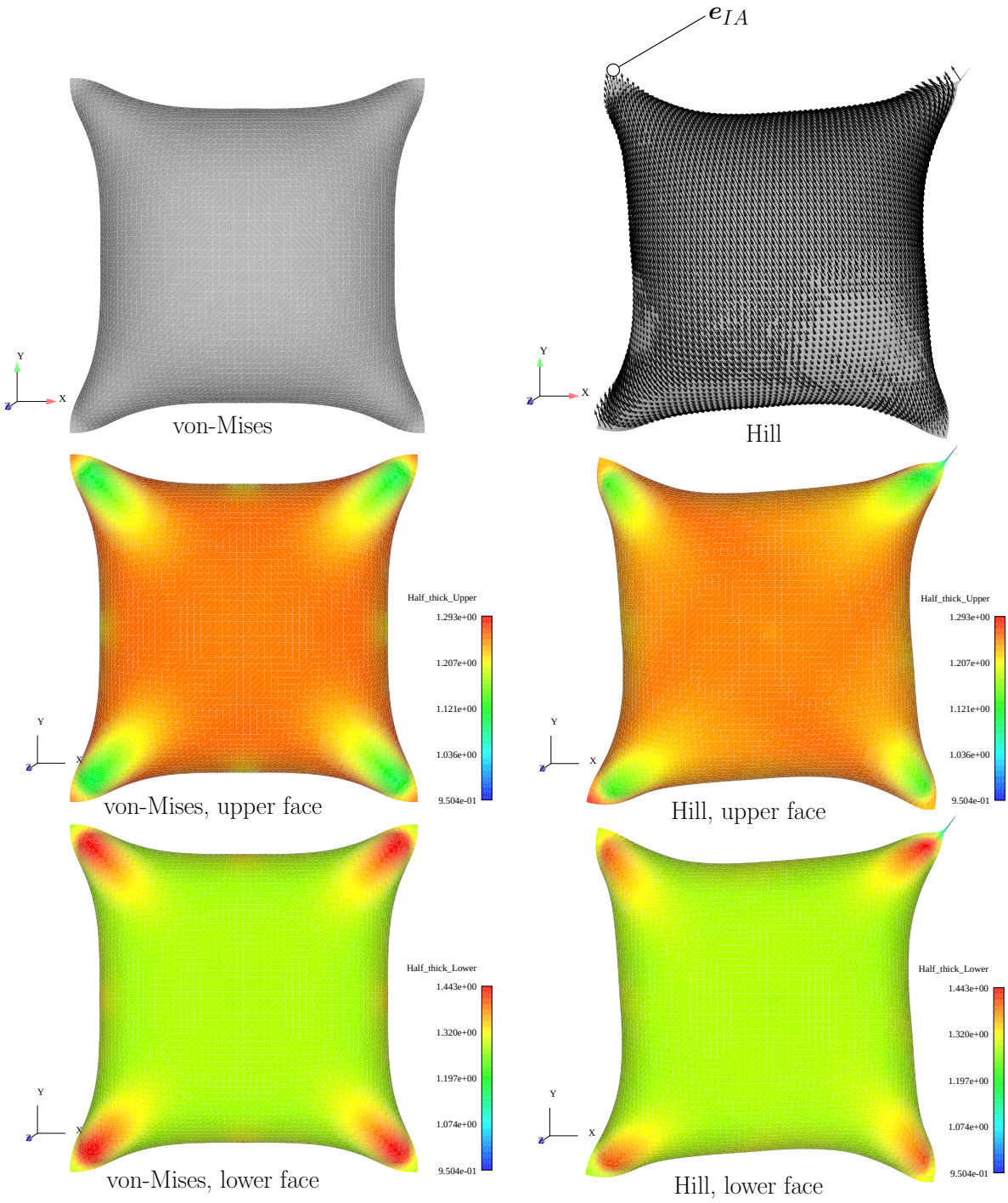


Figure 24: Square plate under pressure: von-Mises and Hill yield criterion contour plots.

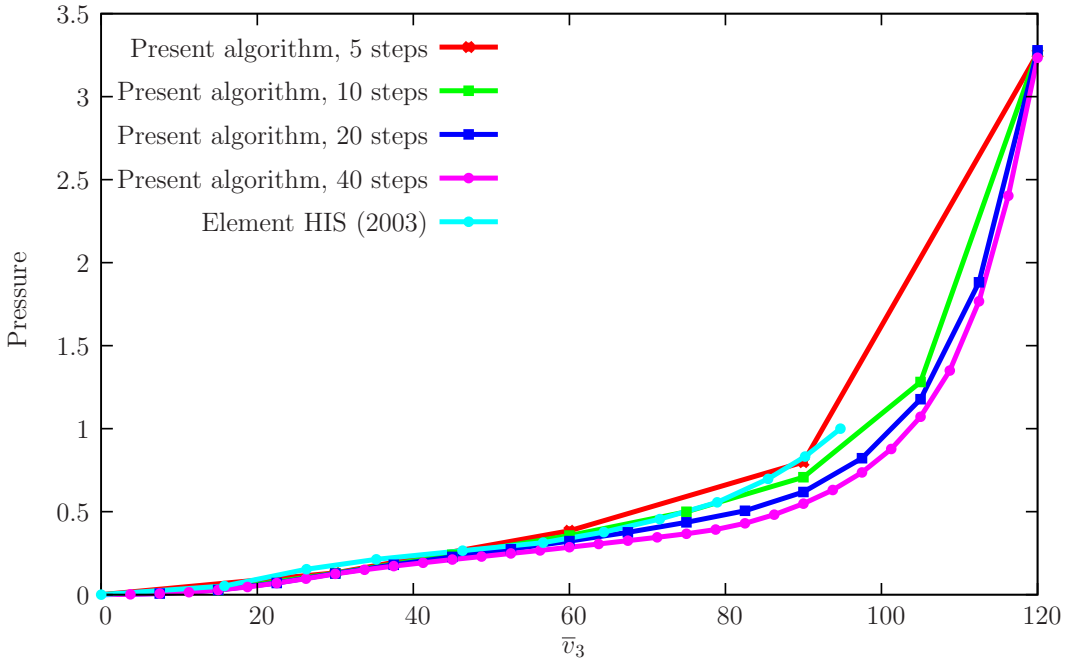


Figure 25: Square plate under pressure with von-Mises criterion: effect of the time steps. 3D EAS element HIS (cf. [6]) is used for comparison.

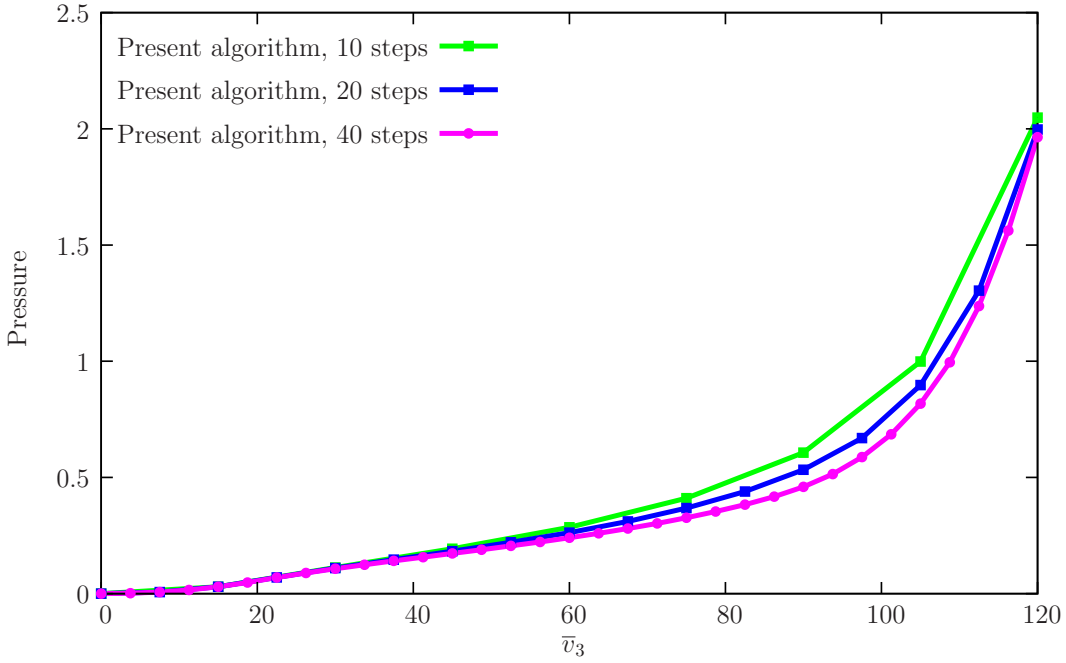


Figure 26: Square plate under pressure with Hill criterion: effect of the time steps.

In the second stage, $t \in]0.2, 1.5]$, the die moves 4 units in the longitudinal direction of the slab. Reaction results are shown in Figure 37 where good agreement with Reference [30] can be observed for the normal reactions. However, the tangential reactions are slightly lower in magnitude for our case.

6.5 Compressed blocks

We now reproduce the benchmark proposed by Temizer [36] which introduces sharp corners and edges in 3D and finite strains (with the neo-Hookean constitutive model). We represent two compressed blocks using one-quarter of the geometry, as depicted in Figure 38. Temizer uses two distinct meshes (a coarse mesh with $5 \times 5 \times 3$ elements for the lower block and a fine mesh with $8 \times 8 \times 4$ elements). The present method requires a finer mesh so that severe spurious inter-penetration (edge/edge crossing) does not occur. Often, contact problems are apparently successfully solved with the node-to-face algorithm but in close inspection there are large edge crossing or wrong target face selections. Also, this fact is known to cause oscillations in the load response. Although there are ad-hoc remedies to this, we usually employ two sufficiently refined meshes. Our coarser mesh consists of:

- Top block with $6 \times 6 \times 2$ elements.
- Bottom block with $9 \times 9 \times 5$ elements.

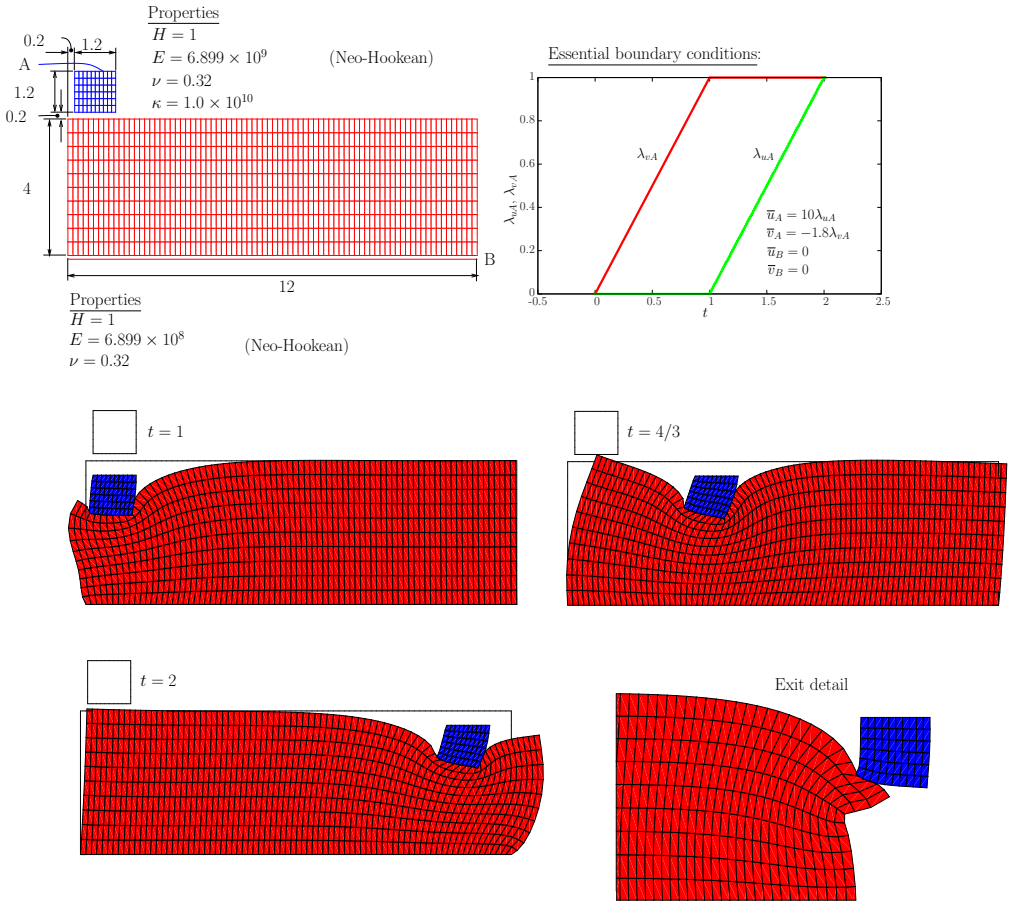


Figure 27: Ironing problem: relevant data and deformed mesh snapshots.

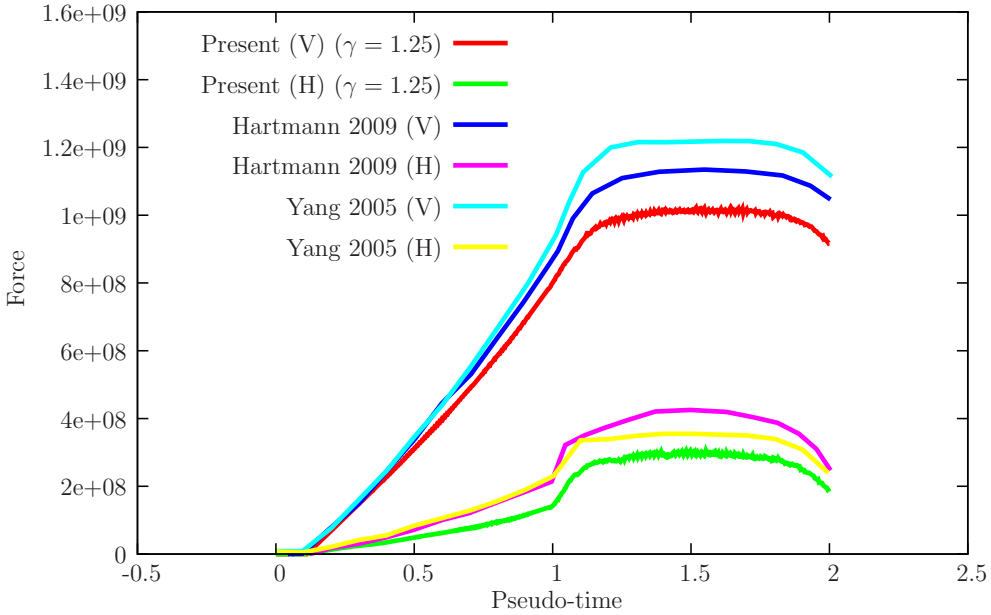


Figure 28: Ironing problem: results for the load in terms of pseudo-time, compared with the values reported by Yang *et al.* [42] and Hartmann *et al.* [19].

A finer mesh with a bottom block containing a $12\times12\times8$ arrangement is also adopted for comparison purposes. Although Temizer considered frictionless contact, we here adopt $\mu = 0.2$ for application with our projection algorithm. The load/deflection results obtained are compared with the results of Temizer (which are scaled by 1/100) and shown in Figure 39, where a good agreement can be observed.

6.6 Compressed concentric spheres

The work by Puso and Laursen [29] presents the concentric sphere problem indicating premature failure of the node-to-face algorithm (even in the frictionless case). Meshes are purposively incompatible to force gap discontinuities to occur during sliding of contact faces. Figure 40 presents the relevant data for this problem (we explicitly represent the two obstacles which are not visible in the original paper). We solve this problem to a slightly lower imposed displacement and inspect the effect of friction in the interface between the two hollow spheres (contact with the obstacles has zero friction coefficient). Note that, in [29], no friction was present. Figure 41 shows the reaction results, compared with the values reported by Puso and Laursen [29]. We note that for the highest value of friction coefficient ($\mu = 2.0$) some oscillations appear. For $\mu = 0.0$ there are also some irregularities in the $u_Z - R_Z$ curve since frictionless sliding induces more active face changes during iteration. The evolution of the energy error during Newton iteration is shown in Figure 42. We can observe that for $\mu = 0.2$, more iterations are required for

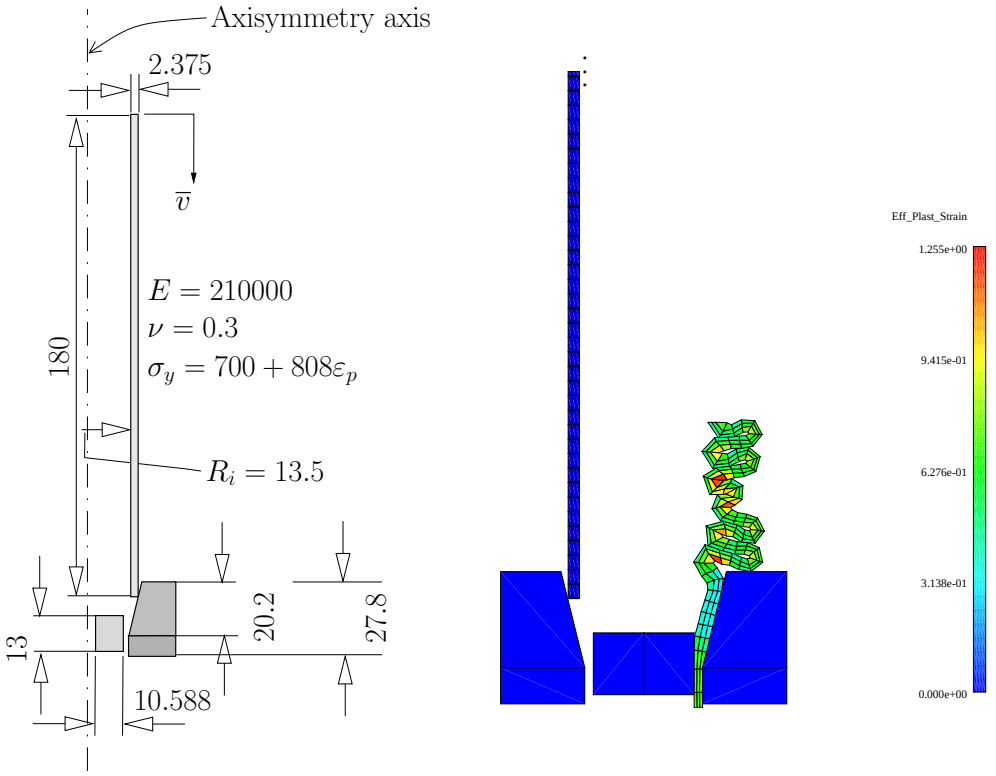


Figure 29: Post buckling of an elasto-plastic cylinder. A picture corresponding to the 5 wrinkles (cf. 4 were reported by Laursen [24]) is also shown.

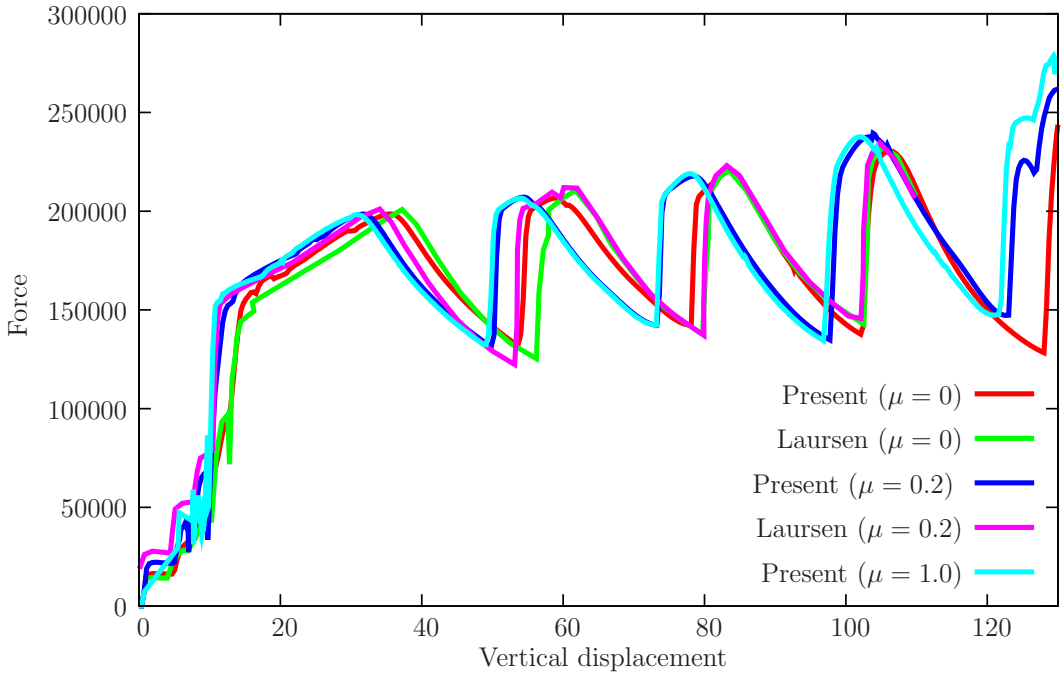


Figure 30: Post buckling of an elasto-plastic cylinder: compression/reaction results. A comparison with the results reported by Laursen [24] is performed.

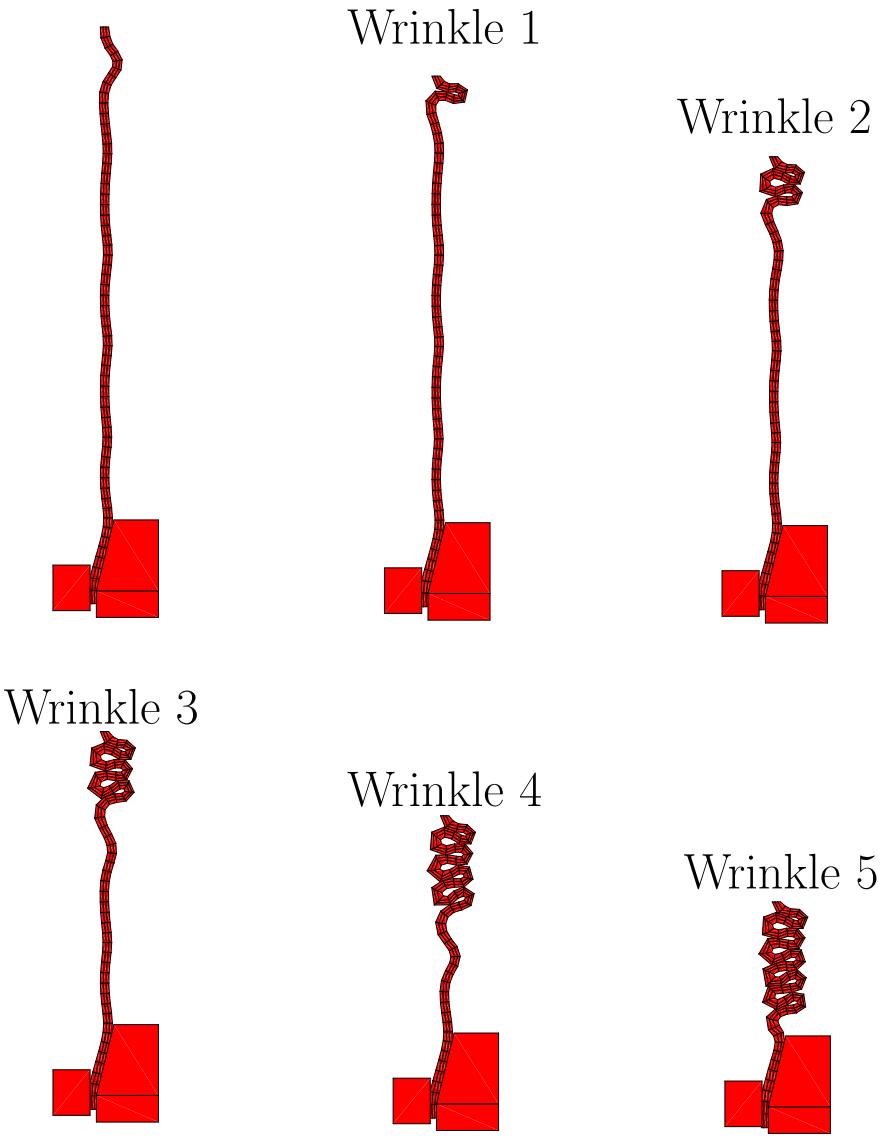


Figure 31: Post buckling of an elasto-plastic cylinder: sequence of wrinkle formation.

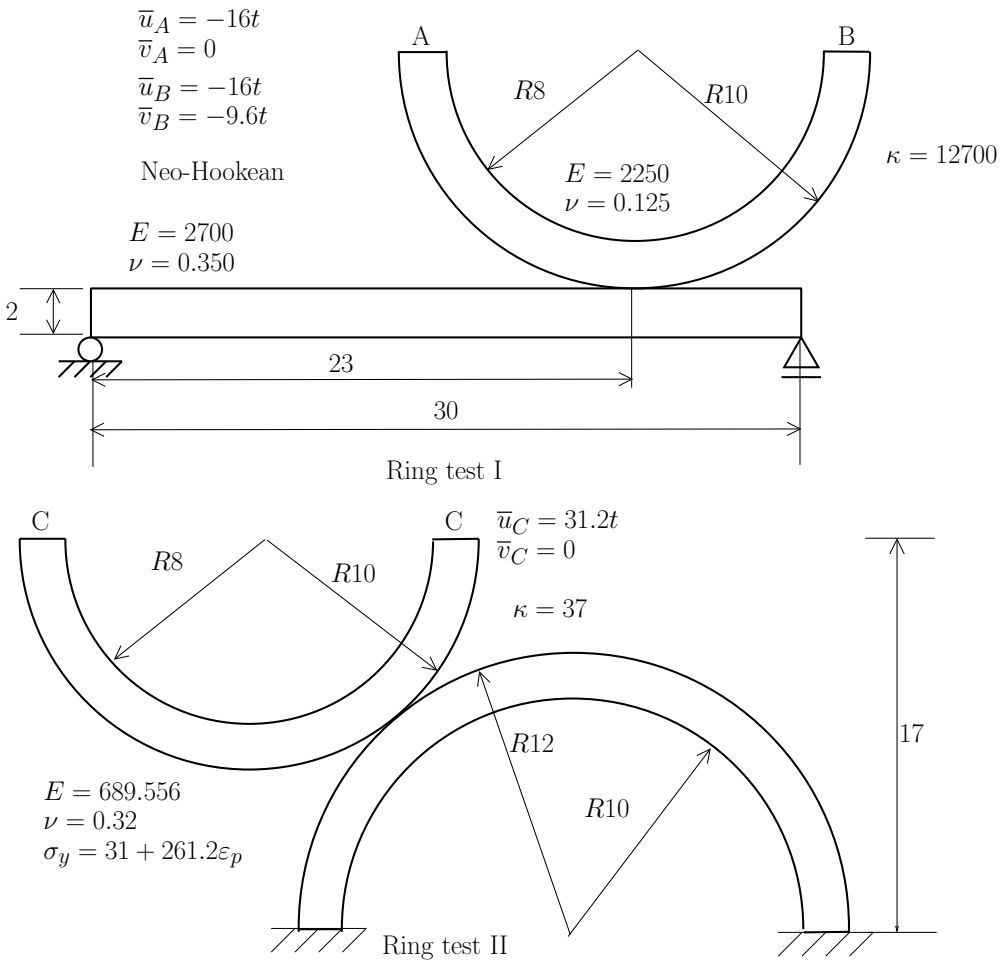
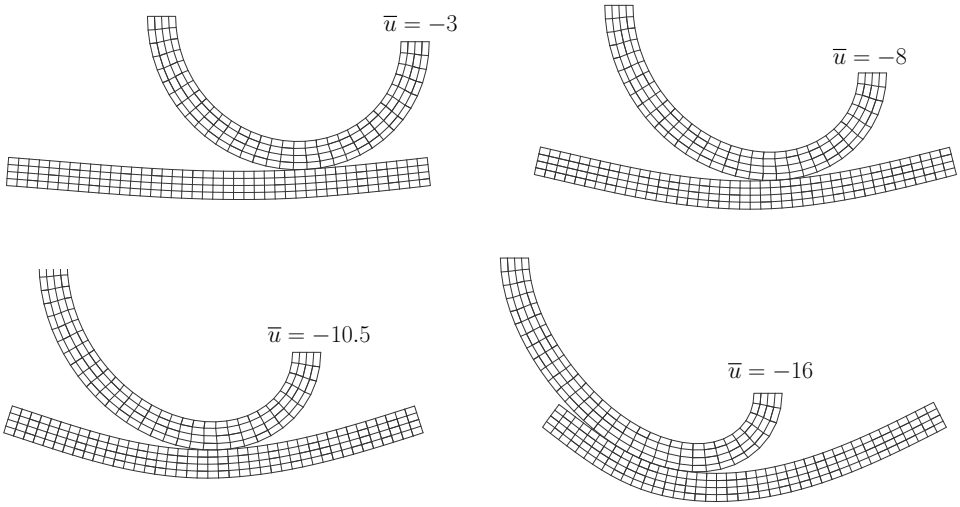
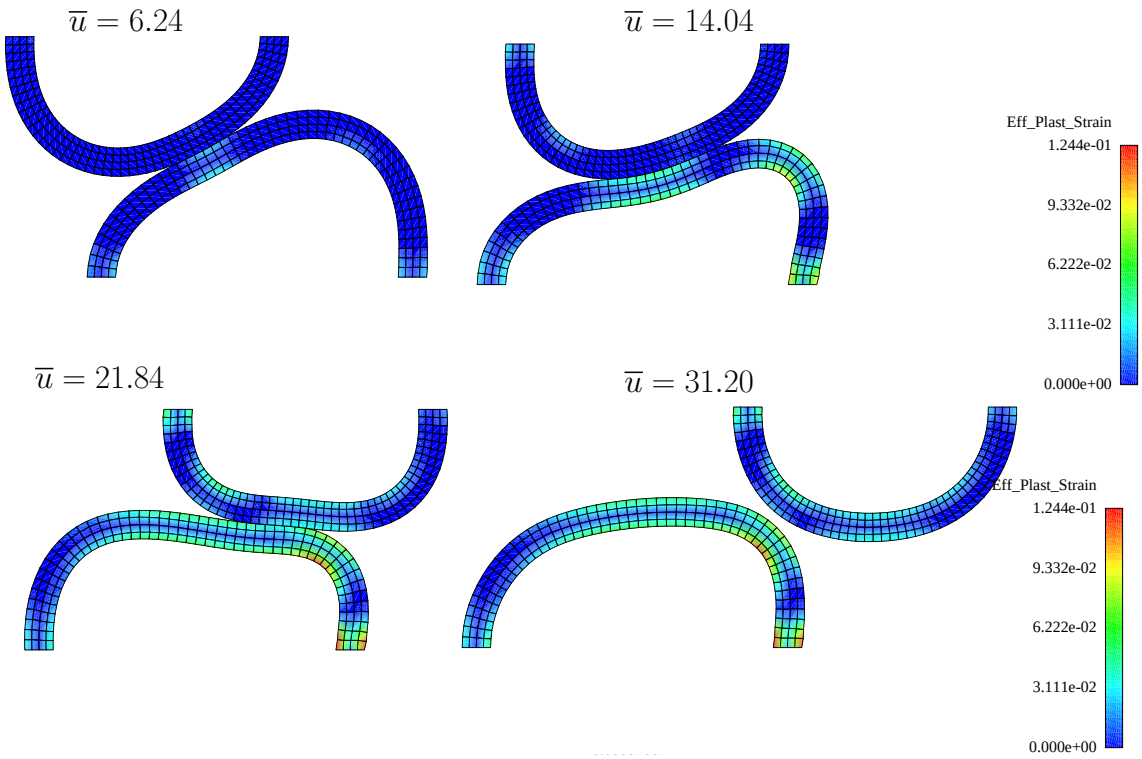


Figure 32: Geometrical, boundary conditions and constitutive data for the ring frictional tests.

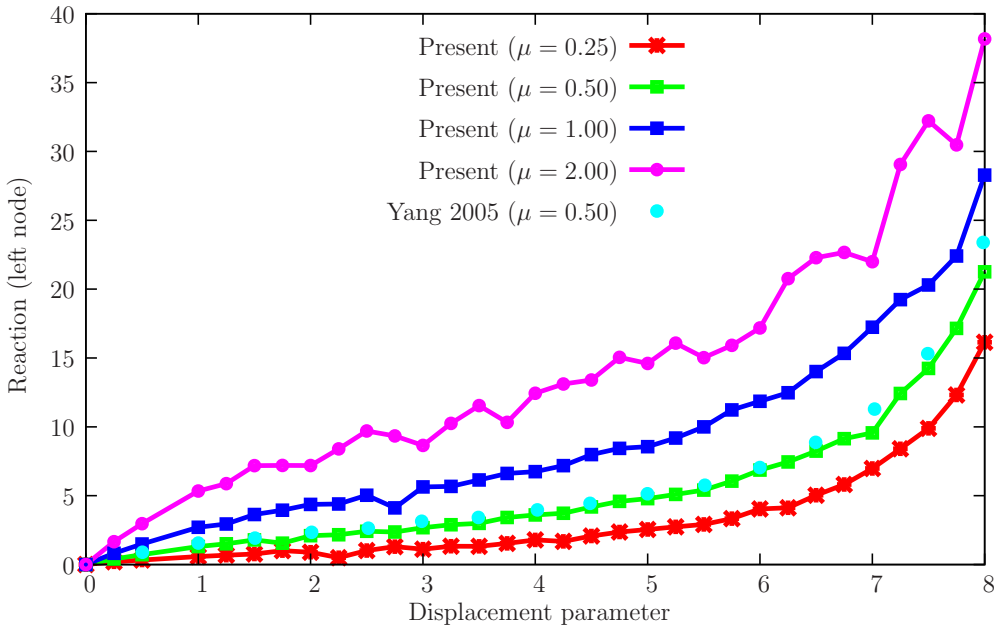


(a) Ring test I: sequence of deformed meshes.

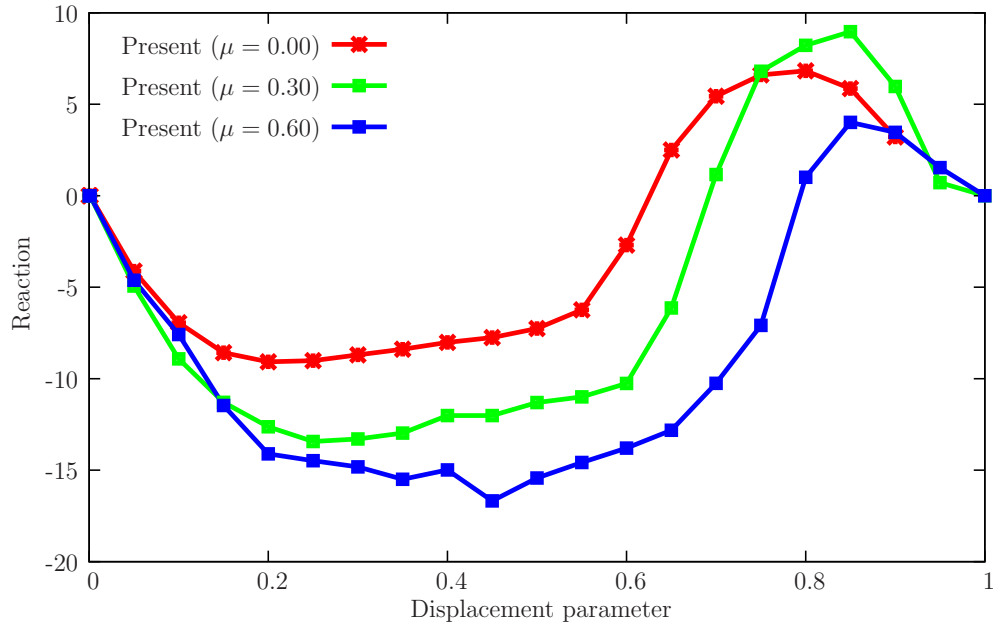


(b) Ring test II: contour plots for effective plastic strain.

Figure 33: Sequence of configurations for the two ring tests.



(a) Ring test I.



(b) Ring test II.

Figure 34: Results for the two frictional tests involving rings.

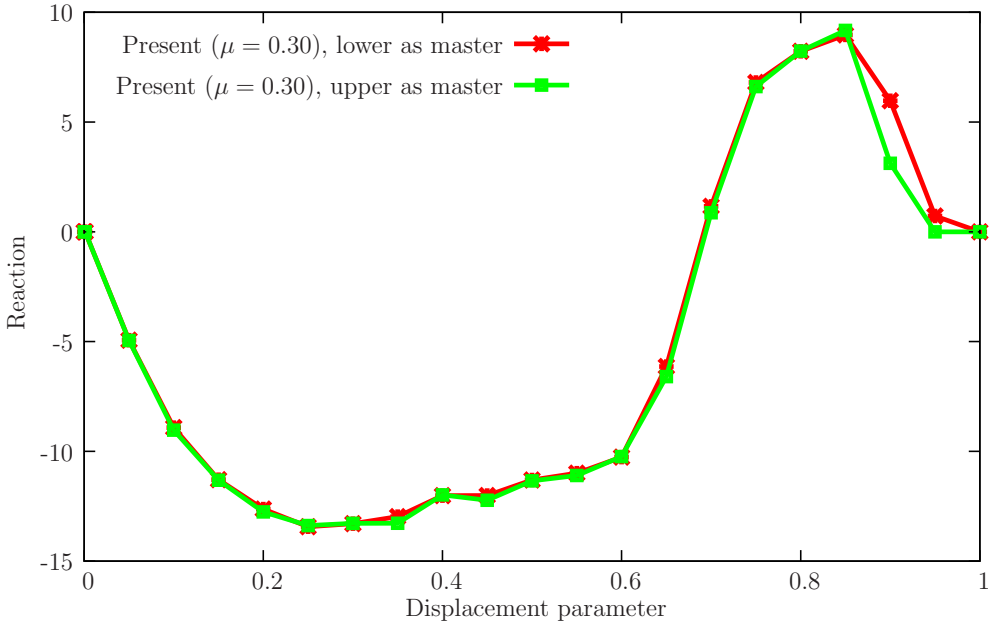


Figure 35: Effect of choice of master surface.

convergence in later stages.

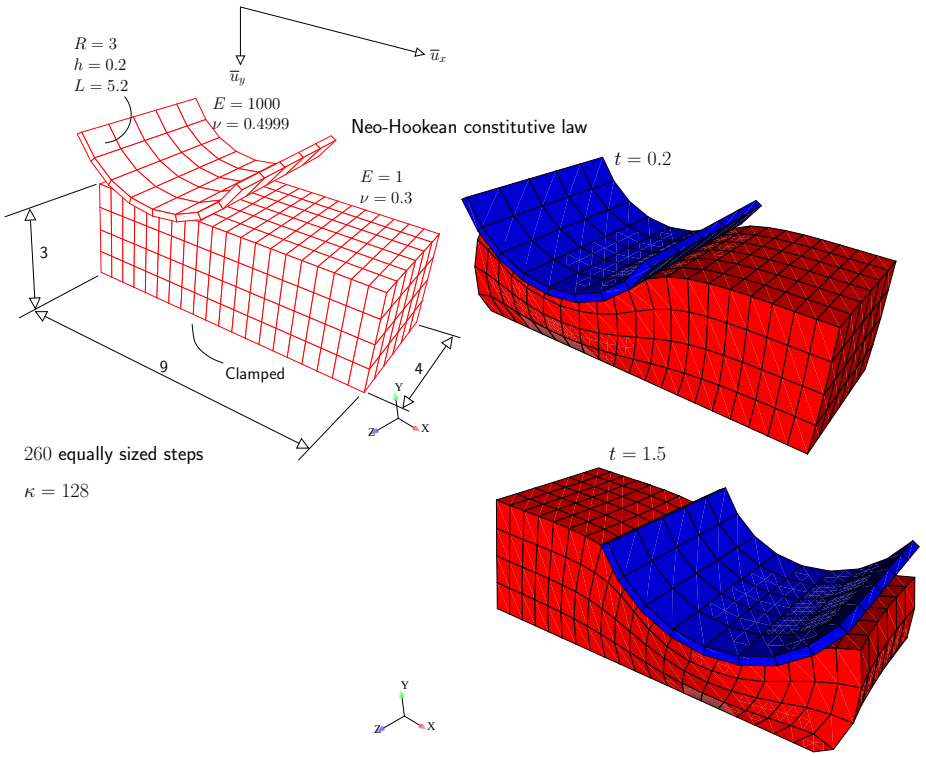


Figure 36: 3D ironing problem: relevant data and deformed meshes.

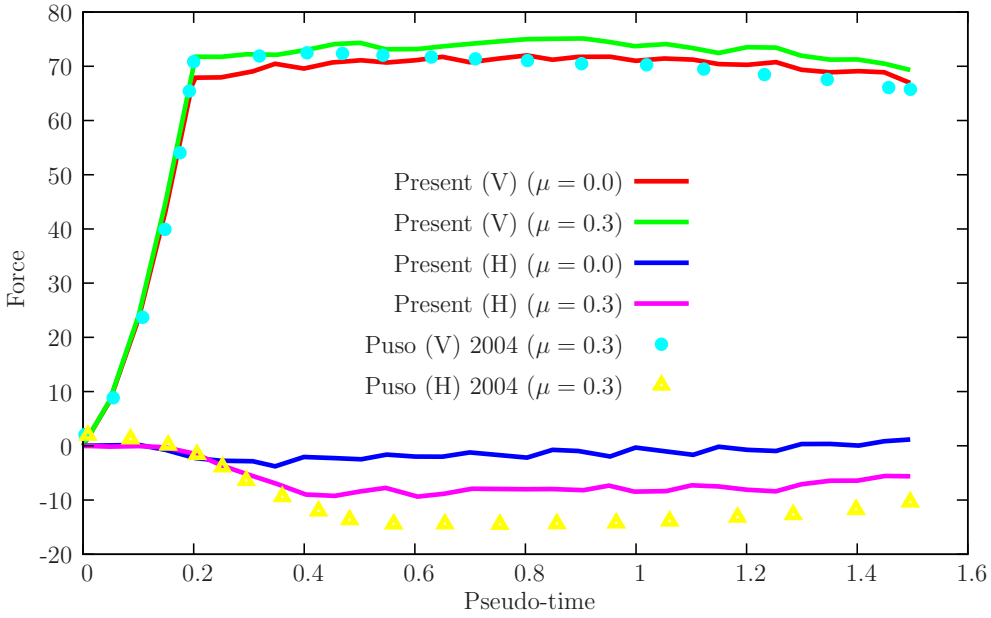


Figure 37: Reactions and comparison with the results of Puso and Laursen (cf. [30]).

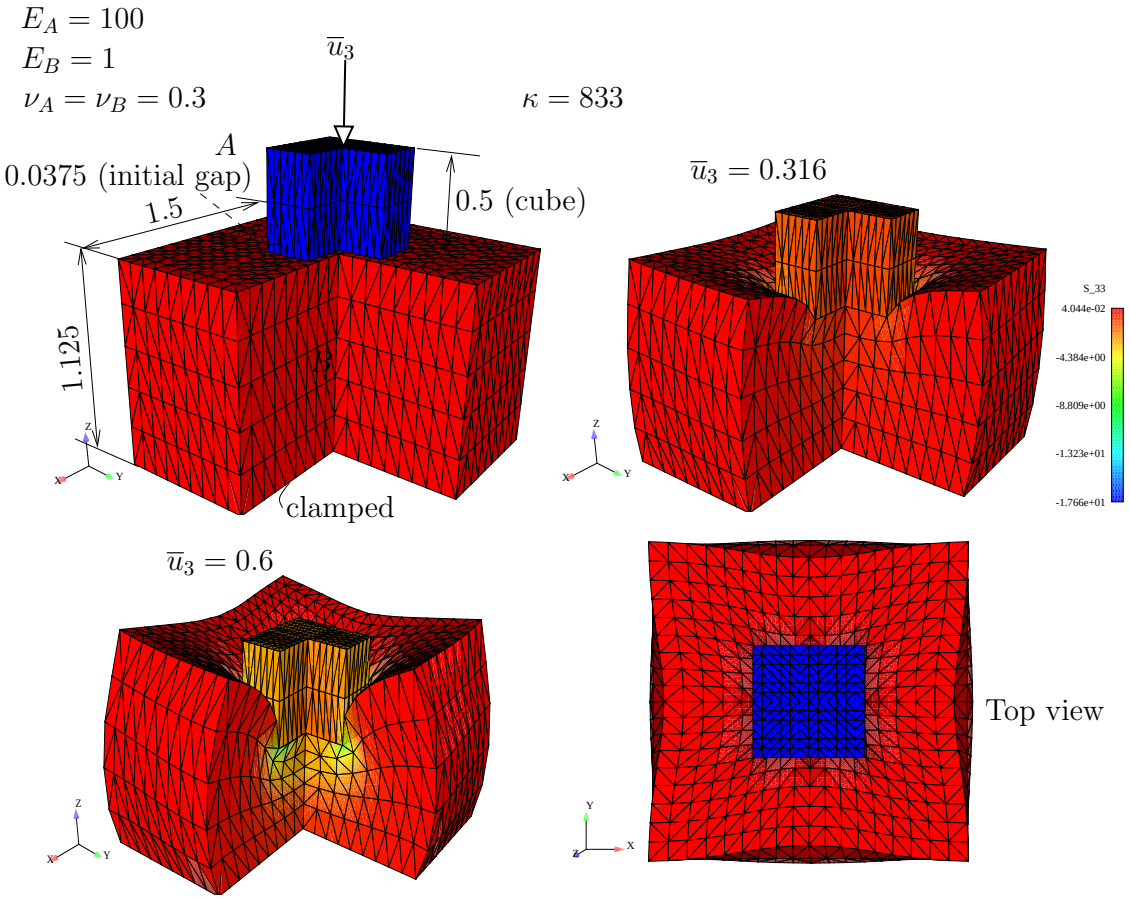


Figure 38: Two compressed blocks, data from Temizer (cf. [36]). Consistent units are used.

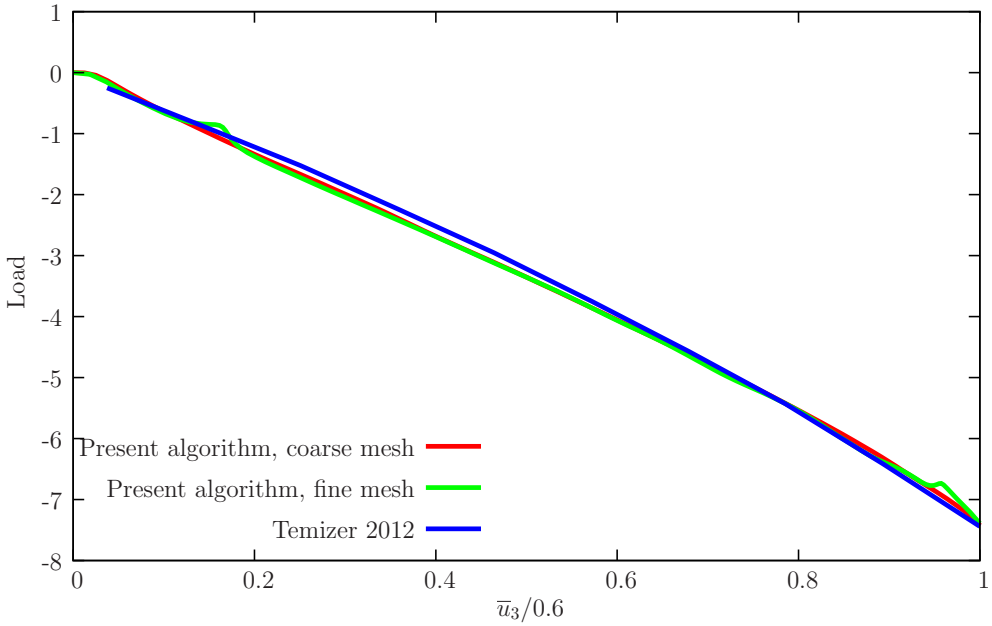


Figure 39: $\overline{u}_3$ vs reaction compared with Reference [36].

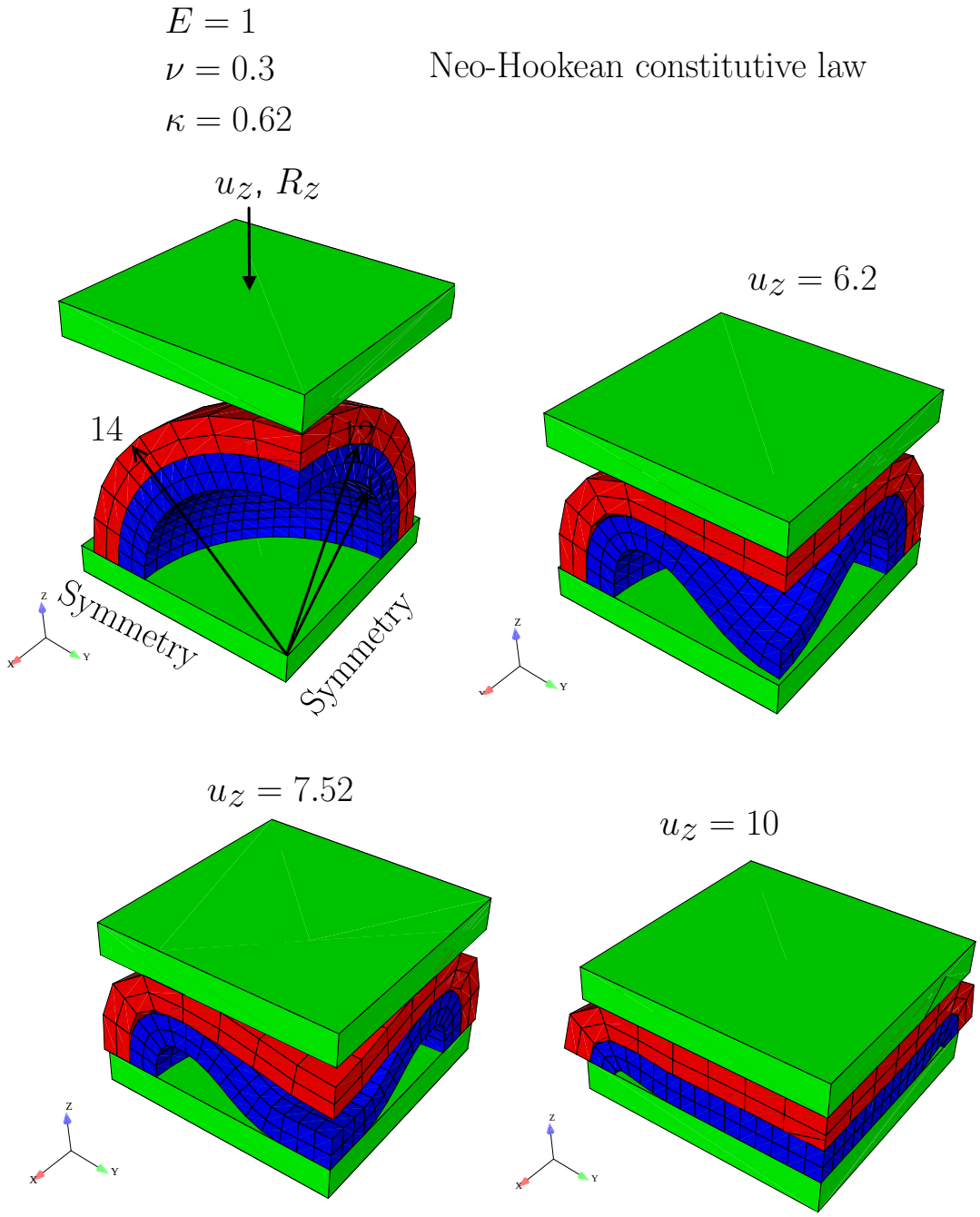


Figure 40: Compressed concentric spheres: relevant data and sequence of configurations.

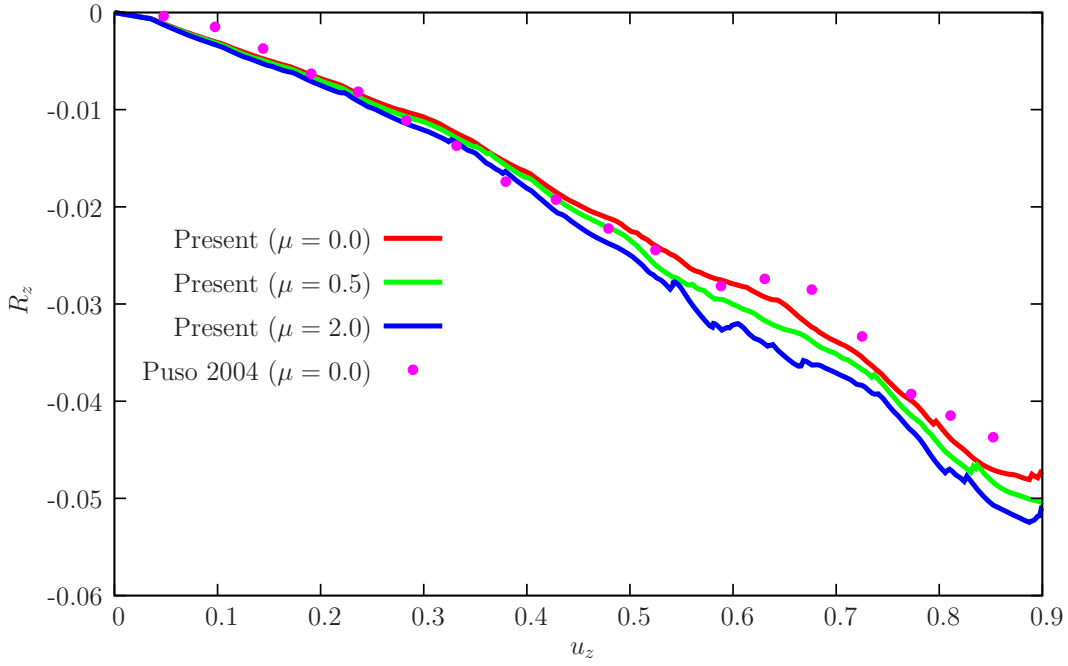


Figure 41: Compressed concentric spheres: displacement/reaction results, compared with [29]. We test $\mu = 0.0$, $\mu = 0.5$ and $\mu = 2.0$

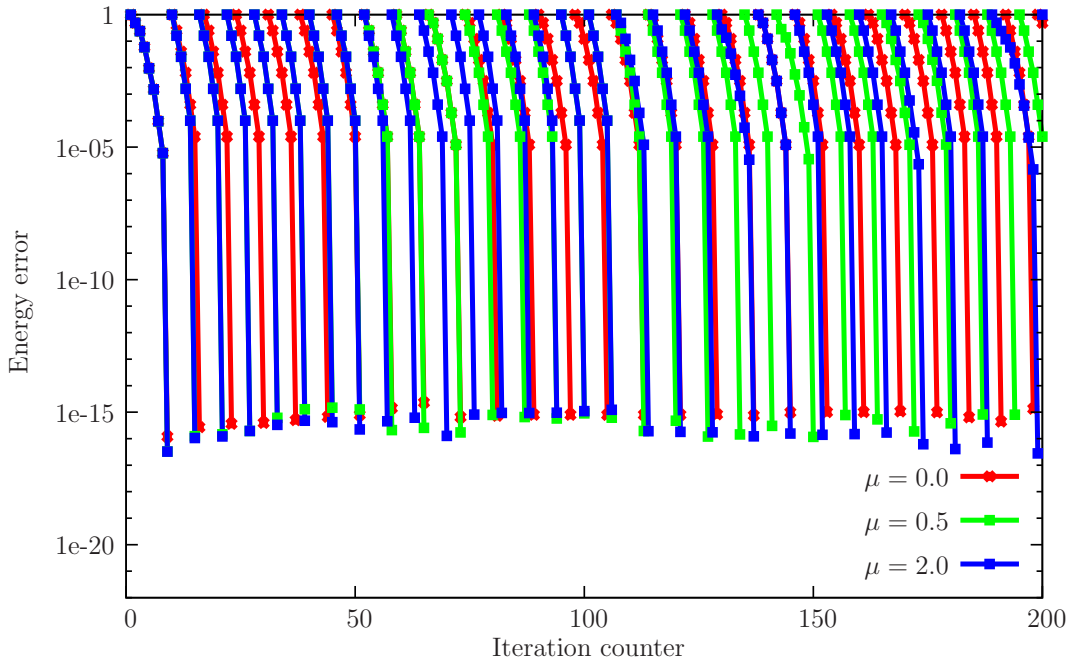


Figure 42: Compressed concentric spheres: Evolution of energy error in Newton iteration for $\mu = 0.0$, $\mu = 0.5$ and $\mu = 2.0$ (scaled to be 1 for each first iteration). First 200 iterations (with history updating counting as an extra iteration) are shown.

REFERENCES

[1] S.S. Antman. *Nonlinear problems of elasticity*. Springer, Second edition, 2005.

[2] S.S. Antman and R.S. Marlow. Material constraints, Lagrange multipliers, and compatibility. *Arch Ration Mech An*, 116:257–299, 1991.

[3] P. Areias. Simplas. https://ssm7.ae.uiuc.edu:80/simplas.

[4] P. Areias. *Finite element technology, damage modeling, contact constraints and fracture analysis*. Doutoramento, FEUP - Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias s/n 4200-465 Porto, Portugal, 2003. www.fe.up.pt.

[5] P. Areias, J.M.A. César de Sá, and C.A. Conceição António. Algorithms for the analysis of 3D finite strain contact problems. *Int J Numer Meth Eng*, 2004. In Press.

[6] P. Areias, J.M.A. César de Sá, C.A. Conceição António, and A.A. Fernandes. Analysis of 3D problems using a new enhanced strain hexahedral element. *Int J Numer Meth Eng*, 58:1637–1682, 2003.

[7] P. Areias, D. Dias-da-Costa, J. Alfaiate, and E. Júlio. Arbitrary bi-dimensional finite strain cohesive crack propagation. *Comput Mech*, 45(1):61–75, 2009.

[8] P. Areias, D. Dias-da-Costa, E.B. Pires, and J. Infante Barbosa. A new semi-implicit formulation for multiple-surface flow rules in multiplicative plasticity. *Comput Mech*, 49:545–564, 2012.

[9] P. Areias, D. Dias-da Costa, E.B. Pires, and N. Van Goethem. Asymmetric quadrilateral shell elements for finite strains. *Comput Mech*, 52(1):81–97, 2013.

[10] P. Areias, D. Dias-da Costa, J.M. Sargado, and T. Rabczuk. Element-wise algorithm for modeling ductile fracture with the Rousselier yield function. *Comput Mech*, 52:1429–1443, 2013.

[11] P. Areias, J. Garção, E.B. Pires, and J. Infante Barbosa. Exact corotational shell for finite strains and fracture. *Comput Mech*, 48:385–406, 2011.

[12] P. Areias, M. Ritto-Corrêa, and J.A.C. Martins. Finite strain plasticity, the stress condition and a complete shell model. *Comput Mech*, 45:189–209, 2010.

[13] D.N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, XXI(IV):337–344, 1984.

[14] Y. Basar and Y. Ding. Finite rotation shell elements for the analysis of finite rotation shell problems. *Int J Numer Meth Eng*, 34:165–169, 1992.

[15] K.-J. Bathe. *Finite Element Procedures*. Prentice-Hall, 1996.

[16] C. Chen and O.L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Comput Optim Appl*, 5:97–138, 1996.

[17] J. Chróścielewski, J. Makowski, and H. Stumpf. Genuinely resultant shell finite elements accounting for geometric and material non-linearity. *Int J Numer Meth Eng*, 35(1):63–94, 1992.

[18] J. Chroscielewski and W. Witkowski. Four-node semi-EAS element in six-field nonlinear theory of shells. *Int J Numer Meth Eng*, 68:1137–1179, 2006.

[19] S. Hartmann, J. Oliver, R. Weyler, J.C. Cante, and J.A. Hernández. A contact domain method for large deformation frictional contact problems. Part 2: Numerical aspects. *Comp Method Appl M*, 198:2607–2631, 2009.

[20] T.J.R. Hughes. *The finite element method*. Dover Publications, 2000. Reprint of Prentice-Hall edition, 1987.

[21] T.J.R. Hughes and E. Carnoy. Nonlinear finite element formulation accounting for large membrane stress. *Comp Method Appl M*, 39:69–82, 1983.

[22] V. Ivannikov. *A geometrically exact Kirchhoff-Love shell model: theoretical aspects and a unified approach for interpolative and non-interpolative approximations*. PhD thesis, Instituto Superior Técnico, Avenida Rovisco Pais 1049-001 Lisbon, Portugal, July 2014.

[23] J. Korelc. Multi-language and multi-environment generation of nonlinear finite element codes. 18(4):312–327, 2002.

[24] T.A. Laursen. *Computational contact and impact mechanics. Fundamentals of modeling interfacial phenomena in nonlinear finite element analysis*. Springer, 2002.

[25] D.M. Neto, M.C. Oliveira, L.F. Menezes, and J.L. Alves. Applying Nagata patches to smooth discretized surfaces used in 3D frictional contact problems. *Comp Method Appl M*, 271:296–320, 2014.

[26] R.W. Ogden. *Non-linear elastic deformations*. Dover Publications, Mineola, New York, 1997.

[27] T.H.H Pian and K. Sumihara. Rational approach for assumed stress finite elements. *Int J Numer Meth Eng*, 20:1685–1695, 1984.

[28] P. M. Pimenta, E. M. B. Campello, and P. Wriggers. A fully nonlinear multiparameter shell model with thickness variation and a triangular shell finite element. *Comput Mech*, 34(3):181–193, 2004.

[29] M.A. Puso and T.A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Comp Method Appl M*, 193:601–629, 2004.

[30] M.A. Puso and T.A. Laursen. A mortar segment-to-segment frictional contact method for large deformations. *Comp Method Appl M*, 193:4891–4913, 2004.

[31] Wolfram Research Inc. Mathematica, 2007.

[32] C. Sansour and F.G. Kollmann. Families of 4-node and 9-node finite elements for a finite deformation shell theory. an assessment of hybrid stress, hybrid strain and enhanced strain elements. *Comput Mech*, 24:435–447, 2000.

[33] J.C. Simo and F. Armero. Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes. *Int J Numer Meth Eng*, 33:1413–1449, 1992.

[34] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Springer, Corrected Second Printing edition, 2000.

[35] J.C. Simo, R.L. Taylor, and K.S. Pister. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Comp Method Appl M*, 51:177–208, 1985.

[36] I. Temizer. A mixed formulation of mortar-based frictionless contact. *Comp Method Appl M*, 223-224:173–185, 2012.

[37] C. Truesdell and W. Noll. *The non-linear field theories of mechanics*. Springer, Third edition, 2004.

[38] W. Wagner and F. Gruttmann. A robust non-linear mixed hybrid quadrilateral shell element. *Int J Numer Meth Eng*, 64:635–666, 2005.

[39] W. Wagner, S. Klinkel, and F. Gruttmann. Elastic and plastic analysis of thin-walled structures using improved hexahedral elements. *Comput Struct*, 80:857–869, 2002.

[40] P. Wriggers. *Computational Contact Mechanics*. John Wiley and Sons, New York, 2002.

[41] P. Wriggers and A. Haraldsson. A simple formulation for two-dimensional contact problems using a moving friction cone. *Commun Numer Meth Eng*, 19:285–295, 2003.

[42] B. Yang, T.A. Laursen, and X. Meng. Two dimensional mortar contact methods for large deformation frictional sliding. *Int J Numer Meth Eng*, 62:1183–1225, 2005.

[43] G. Zavarise and L. De Lorenzis. The node-to-segment algorithm for 2D frictionless contact: Classical formulation and special cases. *Comp Method Appl M*, 198:3428–3451, 2009.

[44] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part I: the recovery technique. *International Journal for Numerical Methods in Engineering*, 33:1331–1364, 1992.

SYMCOMP 2015
Faro, March 26-27, 2015

AN OVERVIEW ON THE MULTIDIMENSIONAL OPTIMAL ORDER DETECTION METHOD

Stéphane Clain[1,2], Jorge Figueiredo[1], Raphael Loubère[2], Steven Diot[2]

1: Centre of Mathematics
School of Sciences
University of Minho
Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: clain@math.uminho.pt

2: Institut de Mathématiques de Toulouse
Université Paul Sabatier
31062 Toulouse, France

Keywords: finite volume, polynomial reconstruction, hyperbolic problem, MOOD

Abstract. *Finite volume method is the usual framework to deal with numerical approximations for hyperbolic systems such as Shallow-Water or Euler equations due to its natural built-in conservation property. Since the first-order method produces too much numerical diffusion, popular second-order techniques, based on the MUSCL methodology, have been widely developed in the '80s to provide both accurate solutions and robust schemes, avoiding non-physical oscillations in the vicinity of the discontinuities. Although second-order schemes are accurate enough for the major industrial applications, they still generate too much numerical diffusion for particular situations (acoustic, aeronautic, long time simulation for Tsunami) and very high-order methods i.e. larger than third-order, are required to provide an excellent approximation for local smooth solution as well as an efficient control on the spurious oscillations deriving from the Gibbs' phenomenon. During the '90s and up to nowadays, two main techniques have been developed to tackle the accuracy issue. The ENO/WENO which can cast in the finite volume context mainly concerns structured grids since the unstructured case turns to be very complex with a huge computational cost. The Discontinuous Galerkin method handles very well accurate approximations but the computational cost and implementation effort are also very high. In 2010 was published a seminal paper that proposed a radically different method. The philosophy consists to use an a posteriori approach to prevent from creating oscillations whereas the traditional methods employ an a priori method which dramatically cuts the accuracy order. In this document, I shall briefly present the MOOD method, show its main advantages and give an overview of the current applications.*

1 A SMALL HISTORICAL INTRODUCTION

Numerical schemes for hyperbolic problems date back from the beginning of the '50s with the *Mathematical And Numerical Integrator And Calculator* Project (MANIAC project) to calculate the nuke fission or solve simple hydrodynamic problems [1, 2]. Finite difference was the unique framework to design numerical schemes and reaches the peak of its golden age with the book of Richtmyer and Morton [3]. Nevertheless, the method suffers of two major drawbacks: it is not conservative and continuity of the solution is required (at least) since one has to define punctual real values at the grid nodes. In a pioneer work, S. K. Godunov proposes in 1959 [4] a new method based on flux evaluations across the interfaces between cells. The main benefits are the built-in conservation property and the use of the mean values which enable discontinuous solutions discretization. The method takes advantage of the divergence form of the conservation laws and makes use of the divergence theorem (or Green theorem) on each cell, providing a set of semi-discrete equations associated to the constant piecewise unknowns. Such a method is very robust but suffers of a large amount of numerical diffusion providing at most a first-order scheme. Linear second-order methods such as the Lax-Wendroff method, *i.e.* the coefficients combining the unknown values do not depend on the solution, suit very well for smooth solutions but the so-called Gibbs' phenomenon occurs when dealing with discontinuities: oscillations characterized by local overshoots and undershoots create non physical approximations and should be eliminated. In the '70s, Van leer [5] introduces an important new concept: the Monotonic Upstream-Centered Scheme for Conservation Laws method (or MUSCL method) where a non-linear procedure is applied to eliminate the creation of new extrema still preserving a local linear representation in the smooth regions of the solution. Extensions for non-linear hyperbolic problems for two- or three-dimensional geometries give rise to a very important literature up to nowadays [6] and the technique is still very popular in the industrial context (most of the hydrodynamics codes use the MUSCL limiter) while almost all finite volume commercial codes implement the technique. The main drawbacks of the MUSCL method are its limitation to second-order schemes (with some rare exceptions) and the procedure strongly reduces the optimal accuracy. To provide higher order, a new method was proposed in the '90s based on the selection of several polynomial reconstructions associated to a given cell[7] such that, on the one hand, the accuracy is optimal for smooth solution and, on the other hand, the method reduces the creation of oscillations controlled by a smoothness indicator. The Essentially Not Oscillating method (ENO) and its Weighted version WENO were the state of the art of the very high-order (larger than three) finite volume methods at the beginning of the 2010s and the reader can refer to the recent review in [8]. In 2010, we have initiated a series of papers on the development of a new technology to suppress the oscillations while preserving the high accuracy for the smooth solutions. The Multidimensional Optimal Order Detection method radically differs from the other techniques since it is based on an *a posteriori* approach (or objective approach) whereas the traditional way addresses an

a priori procedure (or speculative approach) to determine whether or not a polynomial reconstruction is eligible. Moreover, unlike to the other methods, we can plug some physical constraints into the limiting (detection) procedures which turns out a very good advantage with respect to the traditional methods.
The rest of the paper is as following. We give the general framework of the finite volume method using very high-order approximations for hyperbolic systems. We address a short issue on the polynomial reconstructions and highlight the major difficulties to provide the optimal accuracy. Section three is dedicated to the MOOD method where we detail the algorithm and highlight the main advantages with respect to the traditional methods. We give in the fourth section some examples of numerical applications where we assess the accuracy and demonstrate the robustness of the method.

2 VERY HIGHT-ORDER FINITE VOLUME METHOD

For the sake of simplicity, we shall consider a hyperbolic scalar problem on a bounded domain Ω where one seek the function $\phi \stackrel{\Delta}{=} \phi(t,x)$ such as

$$\partial_t \phi + \partial_{x_1} F_1(\phi) + \partial_{x_2} F_2(\phi) = 0 \tag{1}$$

where $F_1(\phi)$ and $F_2(\phi)$ are the physical flux with $x = (x_1, x_2)$. We prescribe the Dirichlet condition $\phi = \phi_D$ on the inflow interface Γ^- given by

$$\Gamma^- = \{x \in \partial\Omega;\ F_1'(\phi_D)n_1 + F_2'(\phi_D)n_2 < 0\}, \quad \Gamma^+ = \{x \in \partial\Omega;\ F_1'(\phi_D)n_1 + F_2'(\phi_D)n_2 \geq 0\},$$

with $n = (n_1, n_2)$ the outward normal vector on the boundary while F_1' and F_2' stands for the derivatives of the physical flux with respect to ϕ. As an example, the linear convection equation $F_1(\phi) = u_1\phi$ $F_2(\phi) = u_2\phi$ cast in the general framework. Shallow water equations or the Euler system cast in a more general vector-values hyperbolic system where the boundary conditions turns to be more complex to be defined.

2.1 Mesh and notations

We introduce the following notations illustrated in Figure 1 to design the numerical scheme. The computational domain Ω is assumed to be a polygonal bounded set of $\mathbb{R}^2$ divided into polygonal cells c_i with m_i the cell centroid, $i \in \mathcal{E}_{el}$ the cell index set. For a given cell c_i, we denote by e_{ij} the edges of c_i such that

- $j \in \mathcal{E}_{el}$ if there exists an adjacent cell c_j with $e_{ij} = c_i \cap c_j$;
- $j = D$ if $e_{iD} = c_i \cap \Gamma_D$.

To avoid a specific treatment of the boundary edges we introduce $\widetilde{\mathcal{E}_{el}} = \mathcal{E}_{el} \cup \{D\}$, the cell index set augmented with index D for the Dirichlet condition and N for the reflection/transmission condition. We then define the set ν_i of all the indexes $j \in \widetilde{\mathcal{E}_{el}}$ such that

e_{ij} is an edge of c_i.
For each edge e_{ij}, $i \in \mathcal{E}_{el}$, $j \in \nu_i$, n_{ij} stands for the unit normal vector going from c_i to c_j and τ_{ij} is the unit tangent vector such that n_{ij}, τ_{ij} is a counter-clockwise oriented basis. We denote by m_{ij} the edge midpoint, while $\big(\xi_r, q_{ij,r}\big)$, $r = 1, ..., R$ stands for the quadrature rule for the numerical integration on e_{ij}, where ξ_r is the weight associated to the r^{th} quadrature point $q_{ij,r}$. If index $j = D$ (resp. $j = N$), n_{iD} and τ_{iD} represent the outward unit normal vector and unit tangent vector while m_{iD} and $q_{iD,r}$ are the edge midpoint and Gauss points.

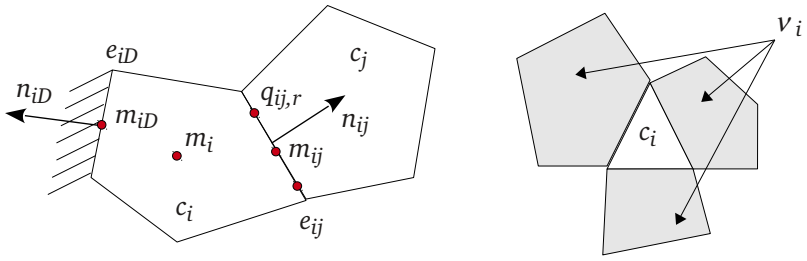


Figure 1: Mesh and notations (left). Definition of index set ν_i (right).

The generic first-order finite volume scheme writes

$$\phi_i^{n+1} = \phi_i^n - \Delta t \sum_{j \in \nu_i} \frac{|e_{ij}|}{|c_i|} \mathcal{F}_{ij}^n, \tag{2}$$

where ϕ_i^n is an approximation of the mean value of V at time t^n on cell c_i, Δt stands for the time step, $|e_{ij}|$ and $|c_i|$ are, respectively, the length of edge e_{ij} and the area of cell c_i. Vector $\mathcal{F}_{ij}$ represents a numerical approximation of the conservative flux across the interface e_{ij}. In the following, we shall denote by Φ the vector collecting all the approximation ϕ_i^n, $i \in \mathcal{E}_{el}$.

2.2 Polynomial reconstructions

To improve the accuracy of the scheme, polynomial reconstruction are providing in order to evaluate a very good approximation on both side of the interface we shall plug into the numerical flux. To achieve high-order approximations, polynomial reconstructions are involved to produce local representations of the approximation (see [9, 10, 12] for the conservative case and [13] for the extension to the diffusive flux case). We recall here the

fundamental lines of the reconstruction for the sake of consistency and to introduce the notations.
For a given cell c_i and a polynomial degree d, we associate the stencil $S(c_i, d)$ constituted of cells we pick-up around the reference cell c_i and we shall denote by $\boldsymbol{\phi}_i(\mathbf{x}; d)$ a local polynomial function of degree d associated to cell c_i with the following structure

$$\boldsymbol{\phi}_i(\mathbf{x}; d) = \phi_i \; + \sum_{1 \leq |\alpha| \leq d} \mathcal{R}_i^{\alpha} \Bigg((\mathbf{x} - m_i)^{\alpha} - M_i^{\alpha} \Bigg),$$

with ϕ_i an approximation of the ϕ mean value on cell c_i, $\alpha = (\alpha_1, \alpha_2)$ the multi-index, $|\alpha| = \alpha_1 + \alpha_2$ (see [12] for a detailed description) and

$$M_i^{\alpha} = \frac{1}{|c_i|} \int_{c_i} (\mathbf{x} - m_i)^{\alpha} \, d\mathbf{x},$$

such that the following conservativity property holds

$$\frac{1}{|c_i|} \int_{c_i} \boldsymbol{\phi}_i(\mathbf{x}; d) \, d\mathbf{x} = \phi_i.$$

To compute the reconstruction coefficients, we introduce the quadratic functional

$$E_i(\mathcal{R}_i) = \sum_{\ell \in S(c_i, d)} \left(\frac{1}{|c_\ell|} \int_{c_\ell} \boldsymbol{\phi}_i(\mathbf{x}; d) \, d\mathbf{x} - \phi_\ell \right)^2,$$

where ϕ_ℓ are approximated mean values on cells c_ℓ of the stencil and $\mathcal{R}_i = (\mathcal{R}_i^{\alpha})_{1 \leq |\alpha| \leq d}$ is the vector which gathers all the components. We seek for vector $\mathcal{R}_i$ which minimises the functional and denote by $\widehat{\phi}_i(\mathbf{x}; d)$ the associated polynomial. In [13], a detailed presentation of the method is given to provide the solution $\mathcal{R}_i$. An important point is the consistancy of the reconstruction process with all the polynomial of degree d to guarantee that we achieve a $d + 1$th-order of accuracy.

2.3 The generic finite volume scheme

Numerical flux $\mathcal{F}_{ij}^n$ in relation (2) is a first-order of approximation of the exact mean value of the flux across the interface

$$\frac{1}{|e_{ij}|} \int_{e_{ij}} \big(F_1(\phi(t, x)) n_1 + F_2(\phi(x, t)) n_2 \big) ds.$$

To provide a better approximation of the flux, one has to first use a high order quadrature rule for the flux integration along the edge using Gauss points $q_{ij,r}$, and secondly, an

accurate approximation at the Gauss points. Then the generic high order finite volume scheme writes

$$\phi_i^{n+1} = \phi_i^n - \Delta t \sum_{j \in \nu_i} \frac{|e_{ij}|}{|c_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}(\widehat{\phi}_i(q_{ij,r}; d), \widehat{\phi}_j(q_{ij,r}; d), n_{ij}), \tag{3}$$

with ξ_r the weights of the quadrature formulae and $\mathbb{F}(\cdot, \cdot, n_{ij})$ the numerical flux from c_i toward c_j. For example the upwind flux or the Lax friedrichs flux are commonly used in the case of the advection but other flux are proposedin [14].

3 THE MOOD METHOD

Relation (3) coupling with the polynomial reconstruction and assuming that Δ satisfies some stability CFL condition provides a very accurate approximation when dealing with smooth solution. unfortunately, it is well-known that even with smooth initial and boundary condition, solutions may present discontinuities and the stability no longer holds while non physical oscillations give rise. The main goal is to locally reduce the polynomial degree in domains where the solution is discontinuous while preserving the optimal order where the function is smooth.

3.1 The *a priori* limiting procedure versus the *a posteriori* detection

Limiters are non-linear procedures providing a reduction of the polynomial degrees to reinforce the stability. We refer to *a priori* limiting procedures in relation with the update step *i.e.* the stage which consists in assembling the flux contributions of each interface of the cell. Therefore, an *a priori* limiting procedure modifies the polynomial function used to evaluate $\widehat{\phi}_i(q_{ij,r}; d)$ and $\widehat{\phi}_j(q_{ij,r}; d)$.
As an example, the MUSCL technique is base on a local linear reconstruction $\widehat{\phi}_i(\mathbf{x}; 1) = \phi_i + \mathbf{a_i}(\mathbf{x} - m_i)$ where $\mathbf{a_i}$ stands for a first-order approximation of the gradient. To annihilate the Gibbs' effect, a limiter $\chi_i \in [0, 1]$ is introduced into the reconstruction and we set $\widehat{\phi}_i(\mathbf{x}; 1) = \phi_i + \chi_i \mathbf{a_i}(\mathbf{x} - m_i)$ such that $\chi_1 = 0$ provides the first-order scheme. The value of the limiter is obtained via an local analysis of the gradient in the vicinity of the cells. We sum-up the main drawback of the limiting procedure.

- No physical considerations are introduced in the computation of χ_i such as the positivity preserving principle.
- The limiter is almost activated even when not necessary. This is the main problem of the *a priori* procedure since it is based on the "precautionary principle" leading to an over-limitation which results into a strong reduction of accuracy.
- The *a priori* procedure is always carried out for each cell since is no simple way to detect if the limiting algorithm is really necessary. Unnecessary computational overheads result from this consideration.

- Mathematical properties such as the maximum principle are hard to achieved with second-order scheme and the CFL condition is usually more restrictive inducing more time steps to compute.

In other words, the *a priori* philosophy is blind and operates indiscriminately without indications on the real requirements and consequences on the updated approximations. The MOOD method is based on an *a posteriori* philosophy where, in short, the idea consists in using the maximum polynomial degree for the reconstruction, to evaluate the flux, to update the solution and then make some corrections only if it is really necessary, up to the user criteria. The main advantage is that we have new and objective informations deriving from a predictor called the candidate solution that we analyse and check to determine which cells or polynomial functions have to be really modified. The *a posteriori* concept enables to dramatically reduce the computational cost, integrate physical properties and better preserve the accuracy.

3.2 The MOOD loop

The main idea of the MOOD method is to determine, for each cell, the optimal degree that one can employ in the polynomial reconstruction that provides both the best accuracy and satisfies some stability conditions. In the following, we summarise the main ingredients of the method and refer to [10, 12] and the reader can find some extension in [15, 16, 17, 18]. The point is to compute an admissible and accurate solution Φ^{n+1} from Φ^n in a sense we shall present is the sequel. To this end, we introduce the Cell Polynomial Degree d_i (in short CellPD) as the degree of the polynomial function associated to cell c_i, while d_{ij} stands for the Edge Polynomial Degree (in short EdgePD) associated to edge e_{ij}. We deduce the EdgePD map from the CellPD map using the simple rule $\mathsf{d}_{ij}=\min(\mathsf{d}_i,\mathsf{d}_j)$ and compute the approximations $\phi_{ij,r}$, $\phi_{ji,r}$, $r=1,\ldots,R$ at point $q_{ij,r}$ on both sides of the edge using the polynomial reconstructions $\widehat{\phi}_i$ and $\widehat{\phi}_j$ of degree d_{ij}. The main problem is the determination of the CellPD map such that the solution $(Phi)^{n+1}$ is admissible. A fundamental assumption underlying the method is that the first-order scheme (also named the parachute scheme) will satisfy all the requirements of what we shall call an eligible solution. Consequently, one can reduce the polynomial degree until reaching the first-order scheme if necessary in the worst cases and then provide an admissible approximation. Parachute scheme are the usual upwind, Rusanov, HLL schemes which have very good properties from the stability point of view but generate a large amount of numerical diffusion.
Two independent mechanisms are involved in the MOOD method: the detection procedure and the limitation procedure. The detection stage is based on the notion of $\mathcal{A}$-eligible set, where we check each cell to determine whether the numerical solution is admissible or not. The limitation procedure mainly consists in reducing the polynomial degree where it is necessary to avoid the appearance of numerical instabilities.

3.2.1 $\mathcal{A}$-eligible set

The detection procedure is the core of the method. We establish criteria to determine whether the approximation of the mean values on cells correspond to an admissible solution or not. We here rephrase the abstract framework proposed in [10, 12] and denote by $\mathcal{A}$ the set of detection criteria (for example the positivity of the water height in the shallow-water context) that the numerical approximation has to respect on each cell. We say that a candidate solution is $\mathcal{A}$-eligible if it fulfils all the criteria of $\mathcal{A}$.
If the candidate solution is not $\mathcal{A}$-eligible on cell c_i, then we reduce the polynomial degree of the respective cell. However, the solution may not be $\mathcal{A}$-eligible regardless of the set $\mathcal{A}$ even if the polynomial degree is zero for the cell. Consequently, we shall consider the numerical solution *acceptable* on the cell if either it is $\mathcal{A}$-eligible or is a first-order approximation (*i.e.* the CellPD has been decremented to zero). Several techniques have been developed in [10, 12] to reduce the computational cost and avoid re-evaluation of all the fluxes on the whole domain. On the other hand, we extend the MOOD algorithm initially designed for a one-time step Euler scheme to the TVD-RK3 scheme by applying the MOOD procedure to each sub step of the TVD-RK3 procedure. Therefore, for sake of simplicity, in the following we shall present the MOOD procedure for just one step, bearing in mind that the TVD-RK3 scheme is a succession of sub steps. Other methods to compute the candidate solution in time use the ADER methodology.We refer to [18] for a presentation of the MOOD procedure in that case.

3.2.2 Candidate Solution and evaluation

In Figure 3.2.2, we display the principle of the MOOD loop. Assume that an approximation Φ^n is known. We first set the CellPD at the maximum polynomial degree value such that we carry out the reconstruction with the highest accuracy. We plug the values at the Gauss points into the numerical flux and then evaluate the candidate solution $\Phi^\star$. Each cell are analysed to check whether it is eligible or not. The problematic cells are then corrected by the reduction of the polynomial degree and new polynomial reconstructions, new flux are evaluated, providing a new candidate solution .

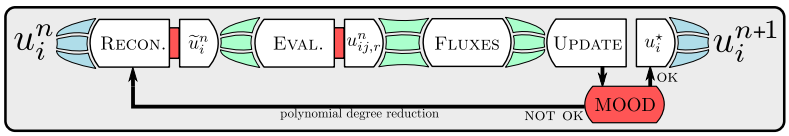


Figure 2: The MOOD loop

The loop stops when all the cell pass all the criteria of the $\mathcal{A}$-eligible set. Then the candidate solution $\Phi^\star$ turns to be the solution Φ^{n+1} at time t^{n+1}. It is important to notice

that only the interface which the cells have been modified have to be recomputed hence the computational cost associated to a re-evaluation of the cell is very low.

3.2.3 Detection criteria and detector chain

A detector is a criterion which enable to quantify or qualify the local candidate solution. From the detector, one can take a decision about the eligibility of the cell. Usually, several detectors are involved in a so-call detector chain where each detector evaluate a specific aspect of the solution such as the positivity, the smoothness, the oscillations and so on. Some detectors notice that the cell is or may potentially be problematic while other detectors release the potential problematic cells. We report to [11] and [19] for a list of detectors.
To highlight the method, we just present two simple detectors. The Physical Admissible Detector (PAD) requires that the candidate solution $\phi_i^\star > 0$ and is very important for the shallow water problem $\phi = h$ the water height or for the Euler system $\phi = \rho$. An other PAD could be $\phi_i^\star \in [0, 1]$ which is very important if we are dealing with a mass or a volume fraction. A second detector is the Extrema Detector (ED) which check if $\phi_i^\star$ is a local extremum with respect to the neighbour cells. Indeed, when an oscillations is created, new extrema are generated and the (ED) detects potential over- or under-shoots. The detector chain organization is very important to safe computational resource and take very quickly a decision. For example if we detect that the solution is negative with the PAD, we immediately state that the cell is problematic and no more evaluation for this cell is required. In the same way, if the (ED) is not activated (we do not have an extremum), we immediately mark the cell as clean and no more effort are necessary.It result that only a very small number of cells are really treated (less than 5% in practice) hence the detection procedure is very fast. Moreover, one can easily check that the detection procedure is highly parallelizable since the analysis of each cell is independent from one to each other. An other interesting point is that the choice of the variables used for the detection process may be different from the primitive or conservative ones. For instance, in [15] the entropy is used in the detector to keep the scheme from violating the entropy condition.

4 NUMERICAL TESTS AND EXAMPLES

We present several test cases where the efficiency, robustness and accuracy of the MOOD method are highlighted. Numerical errors are evaluated in the L^1 and L^∞ norms setting

$$L^1\text{-error: } \sum_{i=1}^{I} |\phi_i^N - \phi_i^{ex}|/I \quad \text{and} \quad L^\infty\text{-error: } \max_i |\phi_i^N - \phi_i^{ex}|,$$

where (ϕ_i^{ex}) and (ϕ_i^N) are respectively the exact and the approximated mean values of function ϕ on cell c_i at the final time $t^N = T$ and $I = \#\mathcal{E}_{el}$.

4.1 Linear convection

We first consider the simple convection of the double sine function on the academic square with periodic condition given by

$$\partial_t \phi + \nabla.((1,1)^t \phi) = 0$$

where the initial function is $\phi(0,x) = \sin(2\pi x_1)\sin(2\pi x_2)$. Figure 3 presents the convergence rate using two different detector chains. The first one (left panel) only detect the extrema and set the CellPD to 0 whether the extremum derives from an oscillation or is a real and smooth extremum. The second detector chain is supplemented with a smooth detector which release the (ED) when dealing with a smooth extrema. It results that the second chain enable the optimal error for all the reconstruction considered in the simulation.

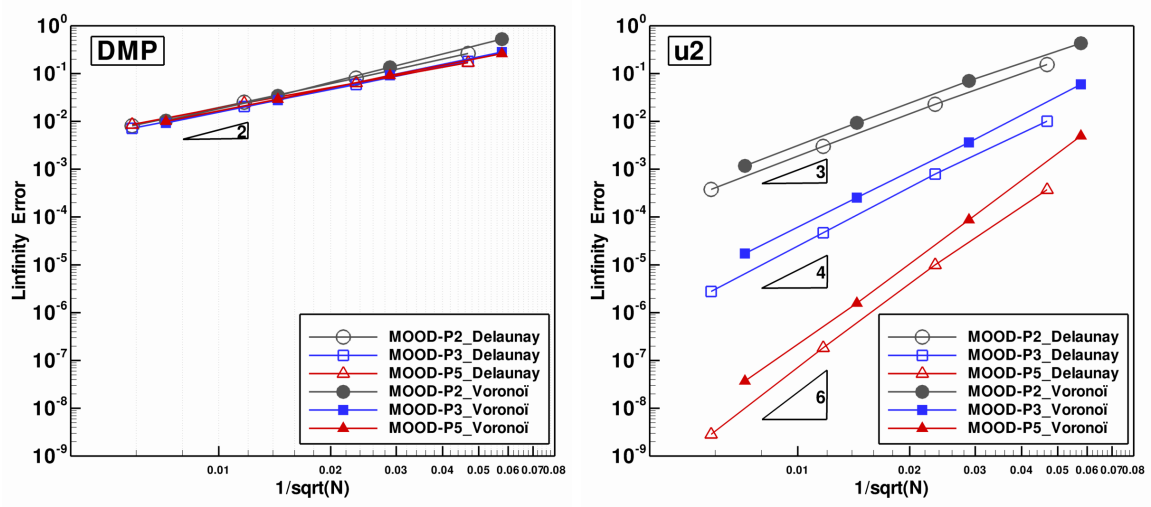


Figure 3: Advection of the double sine function with a constant velocity and periodic condition. Convergence curves with different detectors.

An other classical test is the slotted cylinder problem. The velocity is not constant but define a global rotation with respect to the origin. After a full revolution, Figure 4 left top panel gives an general view of the three shapes that rotate while the top central panel zooms out the slotted cylinder. After a full revolution, the slotted cylinder is smeared and the gap has almost disappeared when using the traditional MUSCL method as presented in the right top panel. The $\mathbb{P}_1$ reconstruction associated to the MOOD method provides a lightly better results (bottom left panel) but the $\mathbb{P}_3$ and $\mathbb{P}_5$ reconstructions manage to preserve the shape. Notice that after half of revolution, the cylinder crosses a portion of the mesh where the cells size are of the same order of the gap. In all the simulations, the maximum principle is strictly respected and no over- or under-shoots are reported.

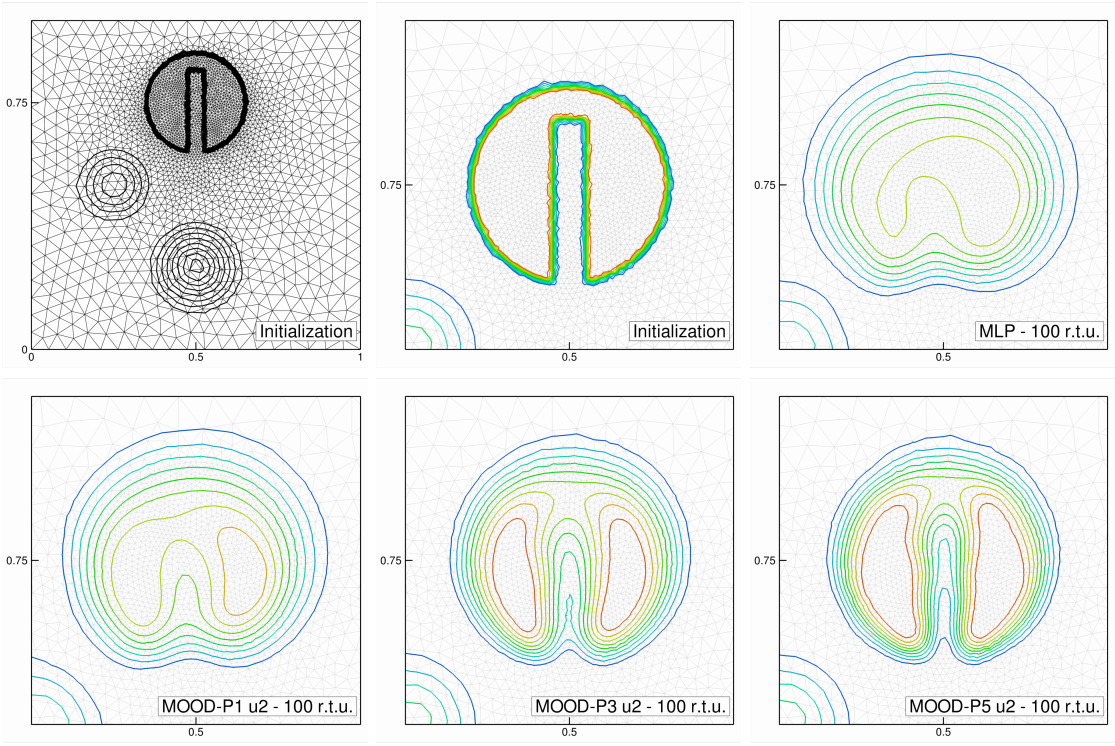


Figure 4: Full revolution of several geometrical figures. The slotted cylinder is well-preserved when using very high-order reconstruction. No maximum principle violation are reported in the simulations.

4.2 Shallow water equations

The shallow water is an important applications for the modelling of river, coast or Tsunami. In [11], the shallow-water system equipped with the non-conservative term deriving from the varying bathymetry is considered

$$\begin{aligned}
&\partial_t h + \nabla.(hU) = 0,\\
&\partial_t(hU) + \nabla.(hU \otimes U + \frac{1}{2}gh^2 I_2) = -gh\nabla b,
\end{aligned}$$

where h is the water height, $U = (u_1, u_2)^T$ the velocity, $U \otimes U$ the tensorial product, $Q = hU$ the mass flow, I_2 the $\mathbb{R}^2$ identity matrix, b the bathymetry with respect to a reference level and g the gravitational acceleration. A sophisticated finite volume method using the MOOD methodology was used to provide numerical approximations.
a steady-state vortex flow with varying bathymetry characterised by

$$H(x,y) = H_\infty - \frac{A^2}{4g}e^{2(1-r^2)}, \quad u(x,y) = A\hat{y}e^{(1-r^2)}, \quad v(x,y) = A\hat{x}e^{(1-r^2)},$$

with $\hat{x} = x - x_0$, $\hat{y} = y - y_0$, and $r^2 = \hat{x}^2 + \hat{y}^2$. We take $H_\infty = 1$, $A = 1$, and $x_0 = y_0 = 0$, while the bathymetry function is given by $b(r) = 0.2e^{(1-r^2)/2}$. Figure 5 depicts the geometry of the vortex as well as the velocity field for the square domain $\Omega = [-3, 3] \times [-3, 3]$.
The simulations are carried out until the final time $t_{\text{final}} = 1\,s$ where we test the MOOD procedure performance using different detectors, namely the DMP against DMP+u2. For that purpose, we consider four Delaunay meshes of 800, 3194, 12742 and 50958 triangles and perform simulations with $\mathbb{P}_2$, $\mathbb{P}_3$ and $\mathbb{P}_5$ for the conservative variables, while the reconstruction for the b function is exact with the $\mathbb{P}_2$ polynomial. Initial conditions are prescribed using the steady-state solution and the Dirichlet boundary conditions imposed on the Gauss points of the boundary edges. The convergence results obtained for the total height are presented in Table 1.

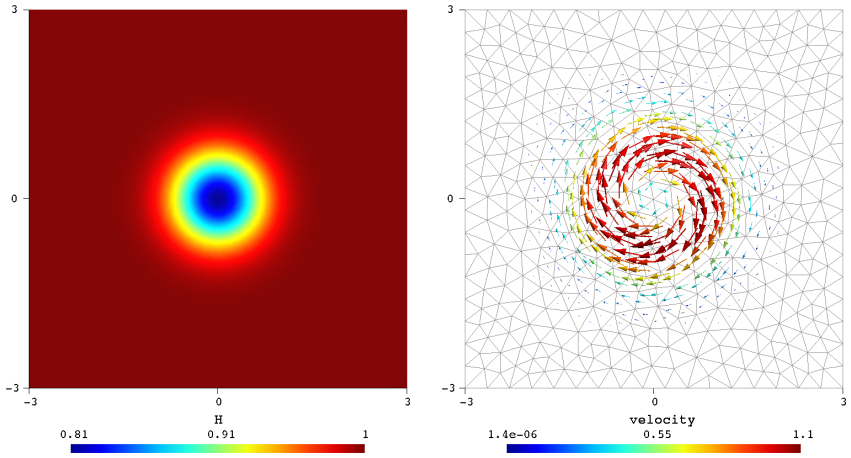


Figure 5: Free surface and velocity field for the static vortex with a 800 triangles mesh.

We report that we obtain the optimal order in all the cases and highlight the capacity of the MOOD method to deal with non-conservative problem.

Table 1: Total height L^1- and L^∞-errors and convergence order for the static vortex.

Nb of Cells	$\mathbb{P}_2$				$\mathbb{P}_3$				$\mathbb{P}_5$			
	err_1		err_∞		err_1		err_∞		err_1		err_∞	
800	4.85e-04	—	6.82e-03	—	8.69e-05	—	1.39e-03	—	3.89e-05	—	9.25e-04	—
3194	7.66e-05	2.7	9.99e-04	2.8	5.86e-06	3.9	8.16e-05	4.1	6.64e-07	5.9	1.41e-05	6.0
12742	1.02e-05	2.9	1.41e-04	2.8	3.67e-07	4.0	5.57e-06	3.9	1.05e-08	6.0	2.21e-07	6.0
50918	1.30e-06	3.0	1.86e-05	2.9	2.29e-08	4.0	4.26e-07	3.7	1.82e-10	5.9	3.55e-09	6.0

A more complex and realistic simulation test is an extension of the classical 2D partial

dam-break problem (see *e.g.* [20] and references therein). We assume that the reservoir (left part of the domain in Figure 6) is higher than the river (right part of the domain), the two entities being relied by a ramp with constant slope. We study the outflow just after the dam rupture until a final simulation time $t_{\text{final}} = 7\,s$. Several characteristic structures will be analysed to evaluate the scheme accuracy and robustness, namely numerical diffusion of the discontinuity, the vortexes deepness as an accuracy assessment and the oscillations around shocks generated by the outflow as a robustness assessment. The domain we

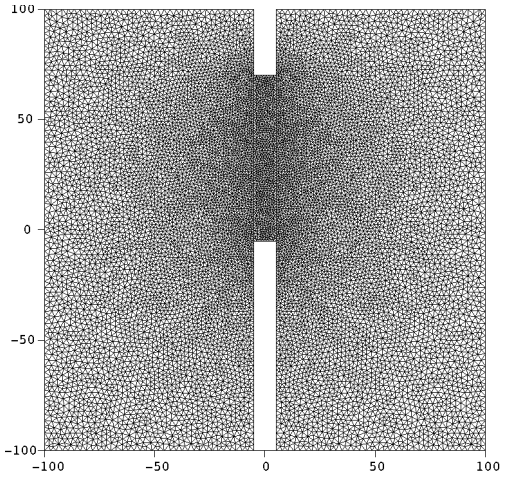


Figure 6: Partial dam-break geometry and the Delaunay mesh (24750 triangles).

consider has been proposed in [20] and the Delaunay mesh, composed of 24750 triangles, is depicted in Figure 6. The breach corresponds to the sub domain $[-5, 5] \times [-5, 70]$ and the bathymetry function is given by

$$b(x,y) = \begin{cases} 1 & , \quad -100 \leq x < -5, \\ 0.1(5-x) & , \quad -5 \leq x < 5, \\ 0 & , \quad 5 \leq x \leq 100, \end{cases}$$

while the initial free surface is given by

$$H(x,y,0) = \begin{cases} 10 & , \quad -100 \leq x < 5, \\ 5 & , \quad 5 \leq x \leq 100. \end{cases}$$

At the initial time $t = 0$ the system is assumed to be at rest and we prescribe reflection boundary conditions on the whole boundary. The bathymetry is characterised by a $\mathbb{P}_1$ polynomial reconstruction since the domain is flat or constituted of a linear ramp. Numerical simulations have been carried out a chain of detectors which include the PAD, the Extrema Detector and the u2 detector (see [10]) which evaluate the smoothness of the

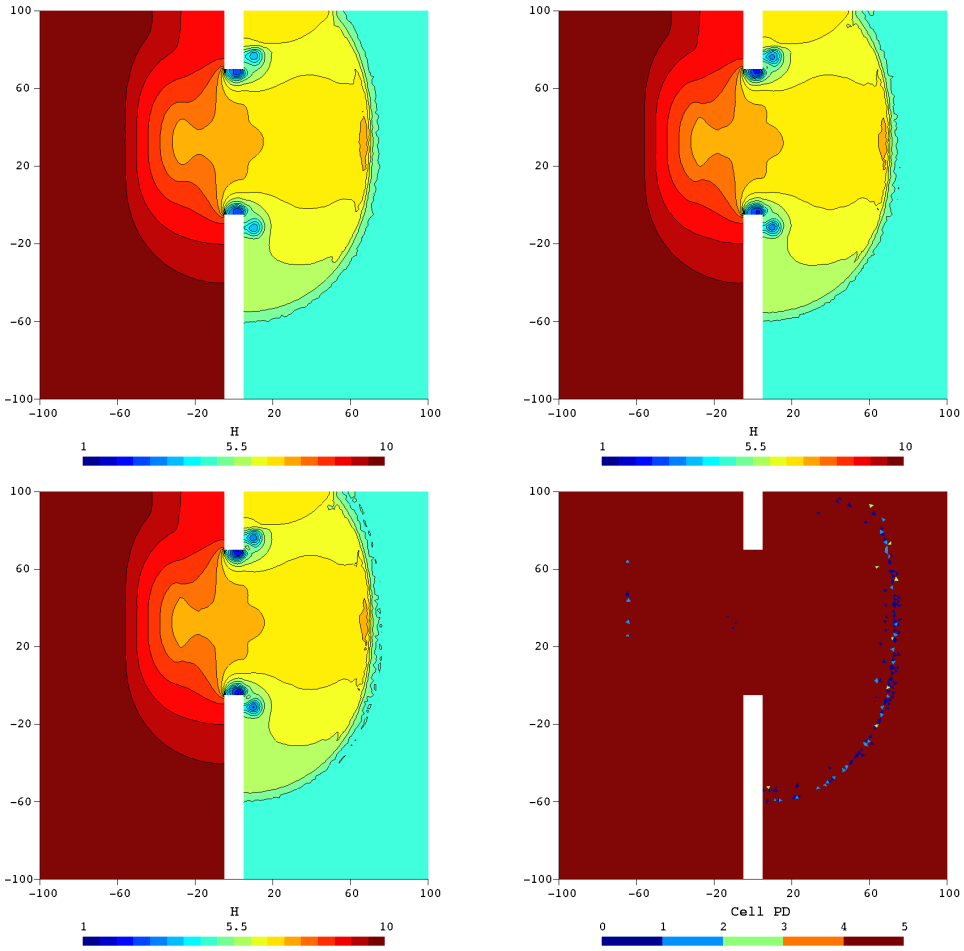


Figure 7: Total height at t_{final} using the DMP+u2$^{\nu}$ detector. Left top panel: $\mathbb{P}_2$. Right top panel: $\mathbb{P}_3$. Left bottom panel: $\mathbb{P}_5$. Right bottom panel: CellPD map with the $\mathbb{P}_5$ reconstruction at final time.

solution. we display in Figure 7 the total height at the final time using the new DMP+u2$^{\nu}$ detector for different polynomial reconstructions $\mathbb{P}_2$, $\mathbb{P}_3$ and $\mathbb{P}_5$. From the stability point of view, the oscillations nearby the shock wave are very well-contained for the $\mathbb{P}_2$ case and are small (below 0.6%) with the $\mathbb{P}_3$ reconstruction, mainly confined near the upper boundary. As for the $\mathbb{P}_5$ situation, oscillations are spread along a large part of the shock wave and represent up to 1.0% of the total height. The CellPD map (see Figure 7 right bottom) shows that the polynomial degree and we observe that in fact very few cells have to be cured.

4.3 Euler system

We end the series of numerical experiences with the Euler system which represents an excellent prototype of complex flow. We reproduce the equations

$$\partial_t \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ E \end{pmatrix} + \partial_{x_1} \begin{pmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(E+p) \end{pmatrix} + \partial_{x_2} \begin{pmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(E+p) \end{pmatrix} = 0,$$

with ρ the density, $U = (u_1, u_2)$ the velocity, P the pressure, E the total energy per unit volume

$$E = \rho \left(\frac{1}{2}(u_1^2 + u_2^2) + e \right),$$

and we assume the equation of state $e = \frac{p}{\rho(\gamma - 1)}$.

We aim to reproduce the propagation of a shock in a cylinder and determine all the interactions and reflections between the waves and the wall. Figure 8 displays a picture of the cavity where an initial strong shock travelling from left to right hits the cavity and develops complex structures.

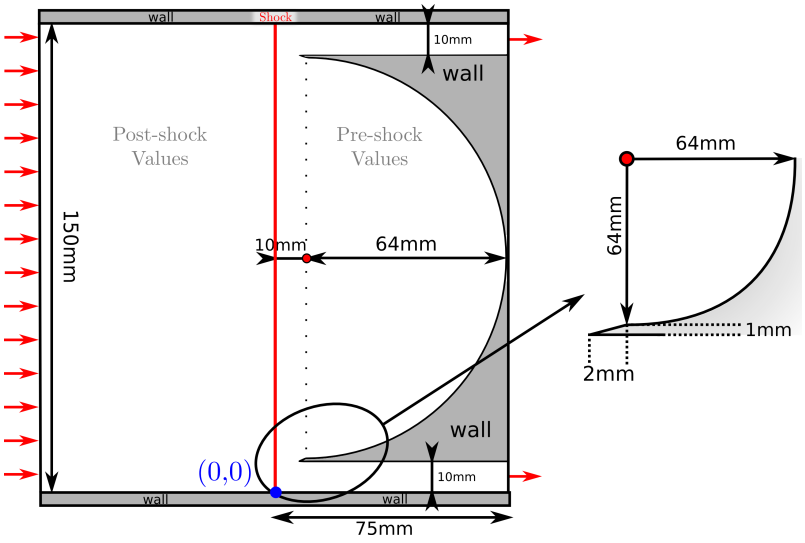


Figure 8: Design of the cavity. An incident pressure wave shocks with the curved cavity and generates complex reflection waves.

We give in Figure 9 the mesh used for the computation and underline that we mix different type of elements (triangle and quadrilateral cell) without any problem. Also notice that the mesh presents strong form factor that the reconstruction process and the MOOD method handle without any problem.

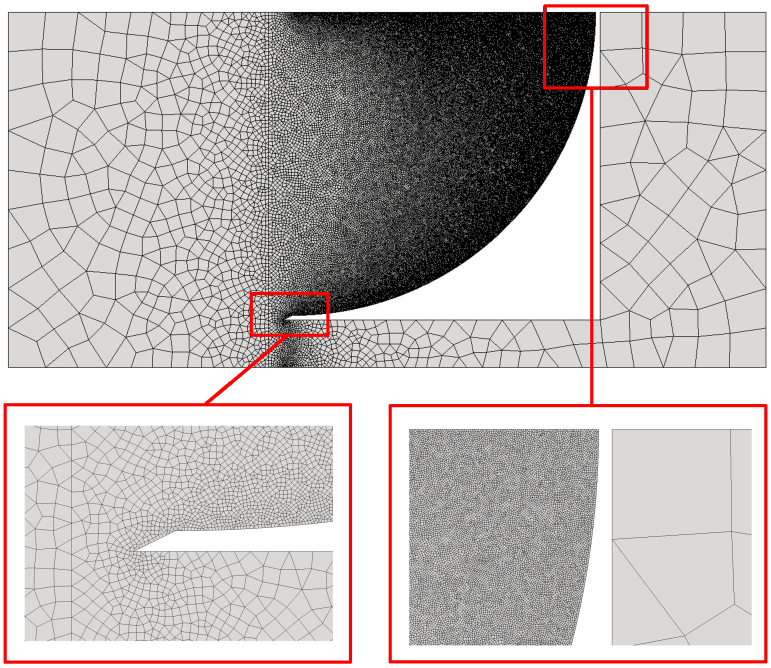


Figure 9: Mix 2D Mesh with 193.000 elements of the cavity

The simulations are carried out with the MOOD-P3 method (fourth-order) using the PAD and the u2 Detection Process. Figure 10 is rendered as a full mesh by symmetry even if the computation was done on a half-domain to easier compare with physical results of [21]. The simulation is clearly in agreement with the experience and demonstrate the high capacity of the MOOD method to handle complex shock structures and contact discontinuities.

5 CONCLUSIONS

In this document, we propose an overview of the MOOD method as a new technique to substitute the ENO/WENO or Discontinuous Galerkin framework. After five years of development, the methodology has grown up and becomes mature. In particular, the capacity to handle discontinuities, to track interfaces while preserving the accuracy for regular solutions have proved that the method turns to be an efficient alternative to the

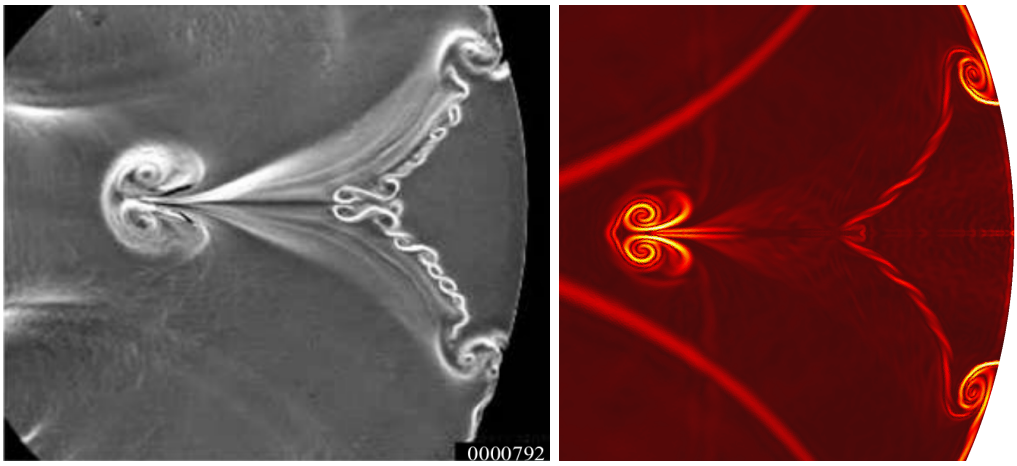


Figure 10: Experimental and simulation of the shock after hitting the cavity with the MOOD-$\mathbb{P}_3$ method. We observe that the structures are very well reproduced.

two other methods. In particular, the control of the polynomial degree is more efficient, the detection procedure faster and the physical restrictions of the solution are better included in the non-linear procedure. The capacity to handle unstructured meshes with different kind of cells and shapes associated to the low sensitivity to the form factor make the MOOD method a very versatile technology that, we expect, will be adopted in the next 10 years by a larger community.

ACKNOWLEDGEMENTS

This research was financed by FEDER Funds through Programa Operational Fatores de Competitividade — COMPETE and by Portuguese Funds FCT — Fundação para a Ciência e a Tecnologia, within the Projects PEst-C/MAT/UI0013/2014, PTDC/MAT/121185/2010 and FCT-ANR/MAT-NAN/0122/2012.

REFERENCES

[1] Dauxois, T, Peyrard, M., Ruffo, S. "The Fermi-Pasta-Ulam 'numerical experiment': history and pedagogical perspectives" Eur. J. Phys. Vol. **26** pp. 3-13, 2005.

[2] Von Neumann, R. D. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks. Journal of Applied Physics", Vol. **21**(3), pp. 232-237 (1950).

[3] Richtmyer R. D. and Morton K. W. "Difference Methods for Initial-Value Problems", second edition, Wiley-Interscience (1967).

[4] Godunov, S. K., "A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations", Math. Sbornik, Vol. **47**, pp. 271-306, 1959.

[5] van Leer, B. , Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method, J. Com. Phys., Vol. **32**, pp. 101-136, 1979.

[6] Buffard, T. , Clain, S., "Monoslope and multislope MUSCL methods for unstructured meshes", *J. Comput. Phys.* Vol. **229**, pp. 3745-3776, 2010.

[7] Casper J., Atkins, H. L., "A Finite-Volume High-Order ENO Scheme for Two-Dimensional Hyperbolic Systems", Journal of Computational Physics, Vol. **106**(1), pp. 62-76, 1993.

[8] Shu C.-W., "High order weighted essentially non-oscillatory schemes for convection dominated problems", SIAM Review, Vol. **51** pp. 82-126, 2009.

[9] Clain, S., Diot, S., Loubère, R., "A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD)", *J. Comput. Phys.* Vol. **230(10)**, pp. 4028-4050, 2011.

[10] Diot, S., Clain, S., Loubère, R., "Improved Detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials", *Comput. & Fluids* Vol. **64**, pp. 43-63, 2012.

[11] Clain, S., Figueiredo, J., "The MOOD method for the non-conservative shallow water system", preprint HAL hal-01077557 (2014), submitted.

[12] Diot, S., Loubère, R., Clain, S., "The MOOD method in the three-dimensional case: Very high-order finite volume method for hyperbolic systems", *Int. J. Numer. Meth. Fluids* Vol. **73**, pp. 362-392, 2013.

[13] S. Clain, G. Machado, J. M. Nóbrega, R. Pereira, A sixth-order finite volume method for multidomain convection-diffusion problem with discontinuous coefficients, Computer Methods in Applied Mechanics and Engineering, 267 (2013) 43–64.

[14] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3^{rd} revision, Springer-Verlag Berlin and Heidelberg GmbH & Co. K 2009.

[15] Berthon, C., Desveaux, V., "An entropy preserving MOOD scheme for the Euler equations", *Int. J. finite volumes* Vol. **11**, pp. 1-39, 2014.

[16] S. Diot1, M.M. François, E.D. Dendy, A higher-order unsplit 2D direct Eulerian finite volume method for two-material compressible flows based on the MOOD paradigms, Int. J. Numer. Meth. Fluids (2014), Vol. **76**(12), pp. 1064-1087, 2014.

[17] M. Dumber, O. Zanotti, R. Loubère, S. Diot, A posteriori subcell limiting for discontinuous Galerkin finite element method for hyperbolic system of conservation laws, J. Comput. Phys. 278 (2014) 47–75.

[18] R. Loubère, M. Dumbser, S. Diot, A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws, Communications in Computational Physics 16 (2014) 718–763.

[19] Clain, S., Figueiredo, J., Second-order finite volume mood method for the shallow water with dry/wet interface, proceeding of the SYMCOMP2015, Faro, March 26-27, 2015, ECCOMAS, Portugal

[20] I.K. Nikolos, A.I. Delis, An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography, Comput. Methods Appl. Mech. Engrg. 198 (2009) 3723–3750.

[21] Skews B.W., Kleine H. Flow features resulting from shock wave impact on a cylindrical cavity. J. Fluid Mech Vol. **580** pp.481-93, 2007.

SYMCOMP 2015
Faro, March 26-27, 2015

Static and Free Vibrations Analysis of Particulate Composite Plates using Radial Basis Functions

G.M.S. Bernardo[1,2]* and M.A.R. Loja[1,2]

1: GI-MOSM, Grupo de Investigação em Modelação e Optimização de Sistemas Multifuncionais
ISEL, IPL - Instituto Superior de Engenharia de Lisboa
Av. Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal

2: LAETA, IDMEC - Instituto Superior Técnico
Universidade de Lisboa,
Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal

e-mail: goncalo.bernardo@tecnico.ulisboa.pt , amelialoja@dem.isel.ipl.pt

Keywords: Particulate composites, functionally graded materials, volume fraction distributions, plates' mechanical behaviour.

Abstract: *Composite materials are known for their tailor-made properties, being the fiber reinforced laminate composites a commonly used type of composite. Other types of composites such as particulate composites have however an additional ability to vary in a continuous form the proportions of the phases involved in the composite manufacturing. This characteristic is an important feature as it enables the minimization of abrupt stresses transitions that always appear when laminates are used. The variation of phases' mixture in space can be specified to obey to a predetermined pattern.*

In the present work, one considers the possibility of the constituents of a dual-phase particulate composite plate, to vary either using an exponent power law or an exponential law, which in this last situation allows admitting a sandwich configuration. A set of illustrative cases considering moderately thick plates, is presented to allow for a comparative study concerning their static and free vibrations behavior.

1. Introduction

Multilayered materials are used in many structures of mechanical and civil engineering as well as in architecture and biomedical fields. In conventional laminated composite structures, homogeneous elastic laminae are bonded together to obtain enhanced mechanical and thermal properties [1]. The main inconvenience of such an assembly is that they create stress concentrations along interfaces. This situation can lead to delaminations, crackings, and another damage mechanisms which result from the abrupt change of the mechanical properties ate the interface between the layers. One way to overcome this problem is to use functionally graded materials (FGMs) within which material properties vary continuously.

In those materials the volume fractions of two or more materials are varied as a function of position along certain dimension(s) of the structure to achieve a required function [2]. For example, a barrier plate structures for high-temperature applications may form from a mixture of ceramic an a metal. The composition is varied from a ceramic-rich surface to a metal-rich surface, with a desired variation of the volume fractions of the two materials in between the two faces. The ceramic constituent of the material provides the high-temperature resistance due to its low thermal conductivity [3]. The gradual change of the material properties can be tailored to different applications ans working environments. This makes FGMs preferable in many applications. The continuous change in the microstructure of FGMs distinguish them from the fibre-reinforced laminated composite materials, which have the drawbacks mentioned before.

As related research and work considering those materials, one can be referred the work of Fukui and Yamanaka [4] in which they examined the effects of the gradition of components on the strength and deformation of thick-walled functionally graded material tubes under internal pressure. Fukui et al. [5] further extended their previous work by considering a thick-walled FGM tube under uniform thermal loading, and investigated the effect of graded components on residual stresses. They generated an optimum composition

of the FGM tube by minimizing the compressive circumferential stress at the inner surface. Fuchiyama and his colleagues [6] used an eight-node quadrilateral axisymmetric element to study transient thermal stresses and stress intensity factors of FGMs with cracks. In their analysis, they concluded that temperature-dependent properties should be considered in order to obtain more realistic results.

Recent work involving FGMs as been done, and we can refer Nguyen et al. [1,7], that used the first-order deformation plate models for modelling structures made of functionally graded materials to investigated the appropriates transverse shear factors values that should be used in the structural problems. In [1] they pontificated that factors by deriving it transverse shear stresses by using the energy considerations from the expression of membrane stresses. Then, using the obtained transverse shear factor, they performed a numerical analysis on a simply supported FGM square plate whose elastic properties were isotropic at each point and varied through the thickness according to the power law distribution. Further, in [7], they continued with that work, obtained additional results in order to compare with their previous works.

In 2012, Loja et al. [8] studied the influence of different FGM's properties homogenization schemes, namely the schemes due to Voigt, Hashin-Shtrikman and Mori-Tanaka, which can be used to obtain bounds estimates for the material properties of particulate composite structures. In that work, finite element models were used in order to achieved this goal and they concluded that, considering the studied schemes, the Mori-Tanaka and Hashin-Shtrikman estimes leads to less conservative results when compared to Voigt Rule of Mixtures. Tran et al. [9] presented a novel formulation based on isogeometric approach (IGA) and higher-order deformation plate theory (HDST) to study the behaviour of functionally graded material plates. They investigated the static, dynamic and buckling analysis of rectangular ans circular plates considering different boundary conditions and used the Mori-Tanaka technique and the Rule of Mixtures to predict the effective material properties of FGM plates.

The use of Multiquadric RBFs to study deformations of a simply supported functionally graded plate modeled by a third-order shear deformation theory was considered by Ferreira

and his colleagues [10]. In [11], the authors used a global collocation method, the first-order and the third-order shear deformations theories, the Mori-Tanaka technique to homogenize material properties, and approximated the trial solution with multiquadric radial basis functions to analyze free vibrations of FGMs. They founded that compute frequencies by the proposed method agree well with those from the other previous used analytical or numerical solutions based on the meshless local Petrov-Galerkin formulations. In 2005, [12] it was used the laywerwise deformation theory and multiquadric discretizations to performed the static and free vibration analysis of composite plates. They considered the radial basis functions as the approximation method for both the equations of motion and the boundary conditions and concluded that the multiquadrics discretization combined with layerwise theory allows a very accurate prediction of the natural frequencies.

A work published by [13] also used these functions as shape functions of a meshless method. Once the gradients and higher derivatives are determined analytically and are continuous and smooth, using RBF in the context of meshless methods [14], it is guaranteed that the formulation of the structural behavioral problem is not compromised, leading to results as good as provided for other methods, such as for example the finite element method, with the advantage of a significant computational effort reduction, as previously referred. Therefore it can be concluded that RBF can be used as approximating functions to model the behavior of a generic structure.

The remainder of this paper is organized as follows. In section 2, we present a briefly literature review related with the methods considered to calculate the distribution of materials volume fraction though the plates thickness, as well as the, FGMs properties homogenization schemes used to predict those properties in each point of the plates. Additionally, it is reviewed the main concepts of the first-order shear deformation theory in static and free vibration analysis. Section 3 presents our considerations on the radial basis interpolation, explaining the methods used to solve the problems considered in our present work. The Kansa Unsymmetric Collocation Method is presented to interpolate the functions on the inner points and boundary points of the plates. The presentation of numerical results of the static and dynamic problems and its analysis is developed within

Section 4 where a several studies are carried, considering different geometrical configurations for the plates, as well as, the distinct homogenization techniques presented in next sections. At last, in Section 5, final remarks and conclusions are withdrawn.

2. Functionally graded materials

2.1 Introduction

Functionally graded material, as referred in previous section, is a composite material which is created by mixing two distinct material phases. Two mixed materials are often ceramic at the top and metal at the bottom as shown in figure I. In this work, dual-phase FGM's. In the study carried out two different volume fraction laws will be considered, namely the Exponential Law [15] and the Power Exponent Law. Concerning to the homogenization schemes, the commonly known Rule of Mixtures [7,8,9], and Mori-Tanaka Technique [7,8,9,10,11] are considered.

The dual-phases composites used in this paper are constituted by aluminium and a ceramic which can be one of three different materials, which mechanical properties can be observed in table I.

Table I – Mechanical and inertia properties of metallic and ceramic materials.

Material	**Young Modulus** $E(GPa)$	**Poisson's Ratio** υ	**Density** $\rho(kg/m^3)$
Aluminum (Al)	70	0.3	2707
Zirconia (ZrO_2)	200	0.3	5700
Aluminium Oxide Al_2O_3	380	0.3	3800
Monotungsten Carbide (WC)	696	0.3	15600

2.1.1 Power Law Volume Fraction Distribution

The Power Law is one of the two volume fraction distribution laws considered in this study, and in the present case is used to calculate the metal volume fraction across the thickness z. This distribution law is given as:

$$V_c(z)=\left(\frac{1}{2}+\frac{z}{h}\right)^p \;;\; V_m=1-V_c \qquad \textbf{(2.1)}$$

where subscripts m and c refer to the metal and ceramic constituents, respectively. Equation **(2.1)** implies that the reference system is located at the middle surface of the composite plate, and the volume fraction varies accordingly through the thickness affected by the power index p.

In addition, in the present work two distinct configurations for the plate model are considered, being the one of them a full functionally graded model and the second a model that is able to consider three different layers, i.e., one on the top constituted by metal with a certain thickness, a intermediate layer which material properties will be calculated according to one of the homogenization schemes used. Figure I show a schematic representation of a FGM with such configuration.

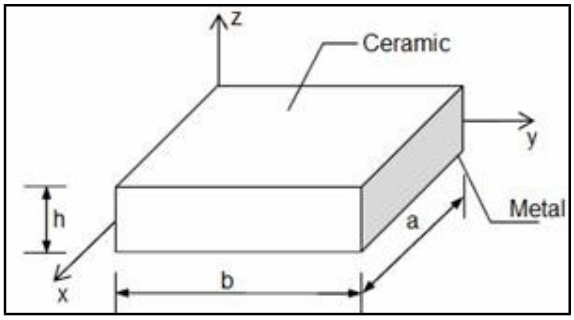


Figure I – Schematic representation of a functionally graded material composed by a ceramic phase on the top and a metallic phase on the bottom.

2.1.2 Exponential Law Volume Fraction Distribution

The other volume fraction distribution used is the Exponential Law. Let be an E-FGM plate of thickness $2t$, unit dimension in depth and infinitely long in x direction, fully ceramic at the bottom, changing to fully metal at the top surface. The material in intermediate region consists of varying proportions of those two materials, as considered in the previous distribution. The variation in z direction for example for the Young's modulus, is given as follows [15,16]

$$E(z)=E_c \quad \text{for} \quad -t\le z\le -e_c \tag{2.2a}$$

$$E(z)=A_1 e^{B_1(a-z)} \quad \text{for} \quad -e_c\le z\le e_c \tag{2.2b}$$

$$E(z)=E_m \quad \text{for} \quad e_c\le z\le t \tag{2.2c}$$

with $A_1=E_m$ and $B_1=\frac{1}{2e_c}\ln\frac{E_c}{E_m}$

Figure II shows a schematic representation of the FGM system considering the nomenclature considered in that approach.

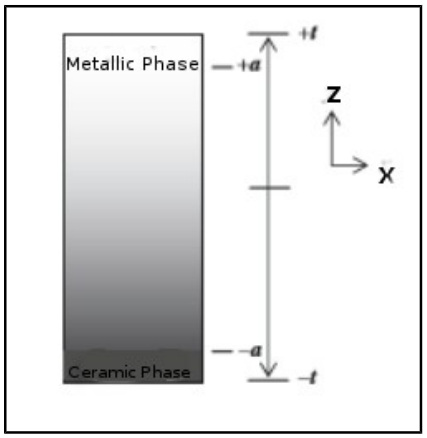


Figure II - Schematic of the functionally graded material (FGM) system considering the nomenclature considered in Schematic Exponent Law presented.

2.1.3 Rule of Mixtures Homogenization

The commonly designated Rule of Mixtures also called Voigt rule is one of the properties homogenization schemes used.
The effective material properties according to the Rule of Mixture are the given by:

$$P(z)=(P_c-P_m)V_c(z)+P_m \tag{2.3}$$

where P_c, Pm denote the material properties of the ceramic and the metal, respectively, including the Young's Modulus E, Poisson's ratio υ and density ρ.

2.1.4 Mori-Tanaka Homogenization

Another approach for properties homogenization, which contrarily to the Rule of Mixtures, consider the interactions among the constituents [10] is the Mori-Tanaka Technique that defines the effective bulk and shear modulus according to the relations:

$$\frac{K_e-K_m}{K_c-K_m}=\frac{V_c}{1+V_m\dfrac{K_c-K_m}{K_m-4/3\,\mu_m}} \tag{2.4}$$

$$\frac{\mu_e-\mu_m}{\mu_c-\mu_m}=\frac{V_c}{1+V_m\dfrac{\mu_c-\mu_m}{\mu_m+f_1}} \tag{2.5}$$

where

$$f_1=\frac{\mu_m(9K_m+8\mu_m)}{6(K_m+2\mu_m)}\ . \tag{2.6}$$

And the effective values of Young's Modulus E and Poisson's ratio υ are given by:

$$E=\frac{9K_e\mu_e}{3K_e+\mu_e}\ ;\ \upsilon=\frac{3K_e-2\mu_e}{2\left(3K_e+\mu_e\right)} \tag{2.7}$$

2.2 Governing equations for First-Order Shear Deformation Theory

Based on first-order shear deformation theory [17] (FSDT), the displacements of an arbitrary point in the plate are defined as:

$$\begin{aligned} u(x,y,z,t)&=u_0(x,y,t)+z\phi_x(x,y,t)\\ v(x,y,z,t)&=v_0(x,y,t)+z\phi_y(x,y,t)\\ w(x,y,z,t)&=w_0(x,y,t) \end{aligned} \tag{2.8a-c}$$

where u_0 and v_0 are the membrane displacements, φ_x and φ_y denote rotations about the x and y axes, respectively and w_0 is the deflection of the mid-plane.
The relationship between strains and displacements is described by:

$$\epsilon=\left[\epsilon_{xx}\ \epsilon_{yy}\ \gamma_{xy}\right]^T\epsilon_0+z\kappa_1\ ,\ \gamma=\left[\gamma_{xz}\ \gamma_{yz}\right]^T\epsilon_s \tag{2.9a-b}$$

where $\epsilon=\begin{bmatrix} u_{0,x}\\ v_{0,y}\\ u_{0,y}+v_{0,x}\end{bmatrix}$, $\kappa_1=\begin{bmatrix}\phi_{x,x}\\ \phi_{y,y}\\ \phi_{x,y}+\phi_{y,x}\end{bmatrix}$ and . $\epsilon_s=\begin{bmatrix}\phi_x+w_{0,x}\\ \phi_y+w_{0,y}\end{bmatrix}$

2.2.1 Static Analysis

A weak form of the static model for the plates under transverse loading q_0 can be briefly expressed as:

$$\int_{\Omega} \delta \epsilon^T \bar{D} \epsilon \, d\Omega + \int_{\Omega} \delta \gamma^T A^s \gamma \, d\Omega = \int_{\Omega} \delta w_0 q_0 \, d\Omega \qquad \textbf{(2.10)}$$

where $A^s = \int_{-h/2}^{h/2} G \, dz$, $\bar{D} = \begin{bmatrix} A & B \\ B & D \end{bmatrix}$,

in which $A = \int_{-h/2}^{h/2} Q \, dz$, $B = \int_{-h/2}^{h/2} Q \, z dz$ and $D = \int_{-h/2}^{h/2} Q \, z^2 dz$

The material matrices are given as:

$$Q = \frac{E(z)}{1-\upsilon^2(z)} \begin{bmatrix} 1 & \upsilon(z) & 0 \\ \upsilon(z) & 1 & 0 \\ 0 & 0 & (1-\upsilon(z))/2 \end{bmatrix}, \quad Q = \frac{E(z)}{2(1+\upsilon(z))} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \textbf{(2.11a-b)}$$

and the Poisson's ratio *υ(z)* and Young Modulus *E(z)* are the effective material properties and vary along the thickness direction.

2.2.2 Free Vibrations Analysis

For the free vibration analysis of the plates, weak form can be derived from the following dynamic equations

$$\int_{\Omega} \delta \epsilon^T \bar{D} \epsilon \, d\Omega + \int_{\Omega} \delta \gamma^T \bar{D}^s \gamma \, d\Omega = \int_{\Omega} \delta \tilde{u}^T m \ddot{\tilde{u}} \, d\Omega \qquad \textbf{(2.12)}$$

where *m* denotes the mass matrix and is calculated according to consistent form:

$$m = \begin{bmatrix} I_0 & 0 & 0 \\ 0 & I_0 & 0 \\ 0 & 0 & I_0 \end{bmatrix}, \qquad \textbf{(2.13)}$$

where $I_0=\begin{bmatrix} I_1 & I_3 \\ I_2 & I_3 \end{bmatrix}$, $(I_1, I_2, I_3)=\int_{-h/2}^{h/2} \rho(z)(1, z, z^2)\,dz$,

and $\tilde{u}=\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$, $u_1=\begin{bmatrix} u_0 \\ \phi_x \\ \phi_x+w_{0,x} \end{bmatrix}$, $u_2=\begin{bmatrix} v_0 \\ \phi_y \\ \phi_y+w_{0,y} \end{bmatrix}$, $u_3=\begin{bmatrix} w_0 \\ 0 \\ 0 \end{bmatrix}$.

2.2.3 Equilibrium equations

In FSDT the resultants {N}, {M} and {Q} are related with the deformations in terms of the Membrane Stiffness Matrix A, Membrane-Bending Matrix B and Bending Matrix D as follows:

$$\begin{Bmatrix} \{N\} \\ \{M\} \end{Bmatrix}=\begin{bmatrix} [A] & [B] \\ [B] & [A] \end{bmatrix}\begin{Bmatrix} \{\epsilon\} \\ \{\kappa_1\} \end{Bmatrix}\ ,\ \{Q\}=[A]\{\epsilon_s\} \qquad \textbf{(2.14a-b)}$$

where ϵ, κ_1 ϵ_s and are defined as before.

3. Radial Basis Functions Interpolation

A generic real valued function *f*, depending on *d* variables, *f*: $\mathbb{R}^d \rightarrow \mathbb{R}$,can generically be approximated by *s*: $\mathbb{R}^d \rightarrow \mathbb{R}$, given the values $\{f(x_i): i=1,2,...,n\}$, where $\{x_i: i=1,2,...,n\}$ is a set of distinct points in $\mathbb{R}$ which consist on the nodes of interpolation. In the present work we will consider approximations of the form:

$$s(x)=p_m(x)+\sum_i^n \lambda_i \Phi(\|x-x_i\|)\ ,\ x\in\mathbb{R}^d\ ,\ \lambda_i\in\mathbb{R} \qquad \textbf{(2.15)}$$

where p_m is a low-degree polynomial (not compulsory), ||.|| denotes the Euclidean norm

and Φ is a fixed function from $\mathbb{R}^+$ to $\mathbb{R}$. Then the coefficients, λ_i , of the approximation *s* are determined by requiring that *s* satisfy the interpolation and side conditions respectively:

$$s(x_j)=f(x_j)\ ,\ j=1,2,\ldots,n \tag{2.16}$$

$$\sum_{j=1}^{n}\lambda_j\, q(x_j)=0\ ,\ q\in\pi_m^d \tag{2.17}$$

The space of all polynomials of degree at most *m* in *d* variables is denoted by π^{d}_{m}. Some typical examples of Φ as well as their shape parameters are given in table II.

Table II – Types of different Radial Basis Functions.

RBF Type		Parameter
Multiquadric	$R_i(x,y)=\left[r_i^2+(\alpha_c\, d_c)^2\right]^q$	$\alpha_c\geq 0, q$
Gaussian	$R_i(x,y)=e^{-\alpha_c(r_i/d_c)^2}$	α_c
Thin Plate Spline	$R_i(x,y)=r_i^\eta$	η
Logarithm	$R_i(x,y)=r_i^\eta\log(r_i)$	η

We must note that the Gaussian and multiquadric functions do not have any restriction on nodes while in the other two, those nodes should not be coplanar [17]. The shape parameters exposed in table II, need to be determined for each type of problem in order to guarantee a good performance. This can be done by carrying out a preliminary numerical evaluation for each type of problem [18].

3.1 The Kansa Method (Multiquadrics)

In the present work there are considered the multiquadric radial basis functions, which

depend only on the distance to a center point x_j and has the form $\Phi(\|x-x_i\|)$ [20].

Consider a set of nodes $x_{1,\ldots},x_N \in \Omega \subset \mathbb{R}$. The radial basis functions centered at x_j are defined as:

$$\Phi_j(x) \equiv \Phi(\|x-x_j\|) \in R^d \ , \ j=1,\ldots,N \qquad \textbf{(2.18)}$$

where $\|x-x_j\|$ is the Euclidean norm. The multiquadric functions we use, are of the form showed in table II with q=0.5 and $c=1/\sqrt{N}$, in which N denote the number of nodes used in each edge of the plate grid.

One of the main advantages of radial basis functions is the insensibility to spatial dimension, making the implementation of this method much easier then, e.g., finite elements [20]. Pairwise distances between points are the only geometric properties as required by the method.

In this paper, it is proposed to use Kansa's unsymmetric collocation method [20].

Considering a boundary-valued problem with a domain $\Omega \subset \mathbb{R}^d$ and a linear elliptic partial differential equation of the form

$$Lu(x) = s(x) \ \text{ in } \ \Omega \qquad \textbf{(2.19)}$$

$$Bu(x)_{\delta\Omega} = f(x) \ \text{ on } \ \delta\Omega \qquad \textbf{(2.20)}$$

where $\delta\Omega$ represents the boundary of the problem. We use points along the boundary (x_j, $j=1,\ldots,N_B$) and in the interior (x_j, $j=N_{B+1},\ldots,N$).

Let the RBF interpolant to the solution $u(x)$ be as equation **(2.15)** without the $p_m(x)$ term.

Collocation with the boundary data at the boundary points and with PDE ate the interior leads to equations:

$$s_B(x)=p_m(x)+\sum_i^n \lambda_i B\Phi\left(\|x-x_i\|\right)=F(x_i)\ ,\ i=1,\ldots,N_B \qquad \textbf{(2.21)}$$

$$s_L(x)=p_m(x)+\sum_i^n \lambda_i L\Phi\left(\|x-x_i\|\right)=S(x_i)\ ,\ i=N_B+1,\ldots,N \qquad \textbf{(2.22)}$$

where $F(x_i)$ and $S(x_i)$ are prescribed values ate the boundary nodes and the function values ate the interior nodes respectively. This corresponds to a system of equations with an unsymmetric coefficient matrix, structured in matrix form as

$$\begin{pmatrix} B\Phi \\ L\Phi \end{pmatrix}[\lambda]=\begin{pmatrix} F \\ S \end{pmatrix} \qquad \textbf{(2.23)}$$

3.2 The eigenproblem

For a linear elliptic partial diferential operator L and a bounded region Ω in $\mathbb{R}^d$ with some boundary $\delta\Omega$, the eigenproblem seeks eigenvalues λ and u that satisfy

$$Lu+\lambda u=0 \ \text{ in } \ \Omega \qquad \textbf{(2.24)}$$

$$L_B u=0 \ \text{ in } \ \delta\Omega \qquad \textbf{(2.25)}$$

where L_B is a linear boundary operator. The eigenproblem of **(2.24)** and **(2.25)** is replaced by a finite-dimensional eigenvalue problem, based on RBF approximations.

The operator L is then approximated by a matrix that incorporates the boundary conditions. The eigenvalues and eigenvectors of this matrix are then evaluated by standard techniques.

To solve the those eigenprobems we consider N_I nodes in the interior of the domain and N_B nodes on the boundary, with $N=N_I+N_B$. We denote interpolation points by $x_i\in\delta\Omega$,

$i=1,...,N_B$ and $x_i \in \Omega$, $i=N_B+1,...,N$. For the boundary conditions, we have:

$$\sum_{i=1}^{N} \alpha_i L_B \Phi\left(\|x-x_j\|_2\right)=0 \ , \ j=1,\ldots,N_B \quad \text{or} \quad B\,\underline{\alpha}=0 \tag{2.26}$$

For the interior points, we have that:

$$\sum_{i=1}^{N} \alpha_i L \Phi\left(\|x-x_j\|_2\right)=\lambda \widetilde{u^T} \ , \ j=N_B+1,\ldots,N \tag{2.27}$$

or , $L^T \underline{\alpha}=\lambda \widetilde{u^T}$ where $L^I=\left[L\Phi\left(\|x-x_j\|\right)_2\right]_{N_I \times N}$.

Therefore, we can write a finite-dimensional problem as a generalized eigenvalue problem:

$$\begin{pmatrix} B \\ L^I \end{pmatrix}[\underline{\alpha}]=\lambda \begin{pmatrix} 0 \\ A^I \end{pmatrix} \underline{\alpha} \tag{2.28}$$

where $B=\left[L_B \Phi\left(\|x_{N_B}-x_j\|\right)_2\right]_{N_B \times N}$ and $A^I=\left[\Phi\left(\|x_{N_B+1}-x_j\|\right)_2\right]_{N_I \times N}$.

The generalized eigenvalues and eigenvectors of these matrices are intended solutions.

3.3 Boundary conditions interpolation

In the present work, we consider the simple multiquadric interpolation to impose the boundary conditions, i.e., for each boundary node one interpolates the function considering the Kansa's unsymmetrical collocation method [20]. In all the applications we consider simply supported plates along all the edges and these boundary condition are imposed as follows:

Edges where $x=0,x=a$: $M_{xx}=0$; $N_{xx}=0$; $\varphi_y=0$, $v_0=0$, $w_0=0$;
Edges where $y=0,x=b$: $M_{yy}=0$; $N_{yy}=0$; $\varphi_x=0$, $u_0=0$, $w_0=0$;

Hence, for the interpolation one can consider the equations **(2.14a-b)**.

4. Numerical examples/ Applications

In this section one presents the results obtained considering different studies. Firstly we performed a set of results in order to evaluate the methods implemented in this work.
Next, static and free vibrations studies are carried out for two different situations, one of them considering a continuous properties variation through all the thickness of the material and the other by using discrete layers though the plate thickness. In sub-section 4.3, the results are obtained in order to investigate the exponent p of power law volume fraction distributing that lead to a same bending stiffness if it is considered the power exponent law. Finally the stresses distributions are represented considering some of the cases studied in sub-section 4.2.
In all the cases carried out, it was considered a grid composed by 17x17 nodes, which it is known to provide sufficient accuracy [21]. For these studies, a simply-supported plate along all its edges with a unit length, is used, and in the static analyses a uniformly distributed load $p_0=10^4$ Pa is considered.

4.1 Validation Cases

The results carried out to evaluate the mid-plane deflections are shown in table III. Those results are compared with [1]. The values of the relative deviations (*Dev*) are also presented in different tables, and were calculated according to the expression: $Dev=\left(value_{present}-value_{REF}\right)/value_{REF}$, where the subscripts *present* and *REF*, refer to the values obtained in the present work and the ones obtained by the references, respectively.

In evaluation cases, it was considered Al/WC and Al/A_2O_3 for the mid-plane deflection and frequencies results, respectively.

Table III – Maximum deflections for Al/WC plate.

a/h	VALIDATIONS (w_{adm}=w/h)								
	w_{dam} (k=1)		Dev(%)	w_{dam} (k=5/6)		Dev(%)	w_{dam} (k=0.576)		Dev(%)
	Nguyen et al. [1]	Present*		Nguyen et al. [1]	Present*		Nguyen et al. [1]	Present*	
5	2.2177E-06	2.2078E-06	0.45%	2.2823E-06	2.2721E-06	0.45%	2.4554E-06	2.4482E-06	0.29%
10	3.1608E-05	3.1457E-05	0.48%	3.1866E-05	3.1714E-05	0.48%	3.2559E-05	3.2392E-05	0.51%
20	4.9023E-04	4.8837E-04	0.38%	4.9126E-04	4.8916E-04	0.43%	4.9406E-04	4.9086E-04	0.65%
50	0.01898	0.019	-0.11%	0.018986	0.0189	0.45%	0.019	0.019	0.00%

In the free vibration case considered for validation purposes, different aspect ratios were adopted and the results were obtained for different values of the exponent p of the power law volume fraction distribution. These results are presented in Table IVa-b and can be compared with those obtained by [9].

Table IVa – Fundamental frequency for Al/A_2O_3 plate.

a/h	VALIDATIONS								
	p=0		Dev(%)**	p=0.5		Dev(%)**	p=1		Dev(%)**
	Tran et al. (2013)	Present*		Tran et al. (2013)	Present*		Tran et al. (2013)	Present*	
5	2.112	2.1126	-0.03%	1.805	1.8106	-0.31%	1.631	1.6432	-0.75%
10	0.577	0.5769	0.02%	0.490	0.4928	-0.57%	0.442	0.4454	-0.77%
20	0.148	0.1476	0.27%	0.125	0.1256	-0.48%	0.113	0.1143	-1.15%

Table IVb – Fundamental frequency for Al/A_2O_3 plate (continued).

a/h	VALIDATIONS					
	p=2		Dev(%)**	p=10		Dev(%)**
	Tran et al. (2013)	Present*		Tran et al. (2013)	Present*	
5	1.397	1.4132	-1.16%	1.324	1.3316	-0.57%
10	0.382	0.3870	-1.31%	0.366	0.3680	-0.55%
20	0.098	0.0992	-1.22%	0.094	0.0947	-0.74%

As it can be seen, by tables III and IV, we can conclude that the results obtained in this work are very similar to those

4.2 Through thickness properties variation

Because the continuous variation of the phases mixture may arise some practical difficulties in its implementation, it is important to characterize/predict the differences that can appear if instead that continuous gradation a layer by layer is achieved.

To this purpose, in this case study, one has implemented those two situations considering the Mori-Tanaka homogenization scheme. In the layer by layer approach, each one has its own elastic coefficients that were calculated assuming a constant reinforcement agent volume fraction associated to the corresponding layer mid-plane thickness coordinate.

In tables V and VI, we can observe the results obtained considering a Al/WC FGM and using a shear correction factor k of 5/6 and 0.576 [1,7], respectively. The values of the deviations (*Dev*) shown in those tables were calculated according the equation: $Dev=\left(value_{CA}-value_{DA}\right)/value_{CA}$, where the subscripts *CA* and *DA* refer to the values obtained when it is used the continuum variation of the properties or a discrete approximation respectively.

Table V – Mid-plane deflection: Continuous (CA), discrete (DA) properties variation. (k=5/6)

p	a/h	Cont. App.	N_{cam}=1		N_{cam}=2		N_{cam}=5		N_{cam}=10		N_{cam}=20	
			w_{adm}	Dev(%)	w_{adm}	Dev(%)	w_{adm}	Dev(%)	w_{adm}	Dev(%)	w_{adm}	Dev(%)
0	5	4.7878E-07	4.7868E-07	0.02%	4.7868E-07	0.02%	4.7868E-07	0.02%	4.7868E-07	0.02%	4.7868E-07	0.02%
	10	6.6702E-06	6.6762E-06	-0.09%	6.6762E-06	-0.09%	6.6762E-06	-0.09%	6.6762E-06	-0.09%	6.6762E-06	-0.09%
	20	1.0291E-04	1.0260E-04	0.30%	1.0260E-04	0.30%	1.0260E-04	0.30%	1.0260E-04	0.30%	1.0260E-04	0.30%
0.5	5	1.2946E-06	1.2792E-06	1.19%	1.3078E-06	-1.02%	1.2940E-06	0.05%	1.2910E-06	0.28%	1.2855E-06	0.70%
	10	1.8883E-05	1.7873E-05	5.35%	1.8627E-05	1.36%	1.8313E-05	3.02%	1.8460E-05	2.24%	1.8758E-05	0.66%
	20	2.8283E-04	2.7558E-04	2.56%	2.7457E-04	2.92%	2.8684E-04	-1.42%	2.9594E-04	-4.64%	2.8274E-04	0.03%
1	5	1.7253E-06	2.0039E-06	-16.15%	1.9563E-06	-13.39%	1.7911E-06	-3.81%	1.7480E-06	-1.32%	1.7350E-06	-0.56%
	10	2.4545E-05	2.8008E-05	-14.11%	2.7914E-05	-13.73%	2.5639E-05	-4.46%	2.5030E-05	-1.98%	2.4709E-05	-0.67%
	20	3.8366E-04	4.3192E-04	-12.58%	4.3242E-04	-12.71%	4.0155E-04	-4.66%	3.7624E-04	1.93%	3.8934E-04	-1.48%
2	5	2.1736E-06	3.1448E-06	-44.68%	2.8616E-06	-31.65%	2.3511E-06	-8.17%	2.2297E-06	-2.58%	2.1894E-06	-0.73%
	10	3.0841E-05	4.4479E-05	-44.22%	3.9939E-05	-29.50%	3.3167E-05	-7.54%	3.1433E-05	-1.92%	3.1285E-05	-1.44%
	20	4.7548E-04	6.7994E-04	-43.00%	6.1069E-04	-28.44%	4.9651E-04	-4.42%	4.8606E-04	-2.23%	4.5775E-04	3.73%

Table VI – Mid-plane deflection: Continuous (CA), discrete (DA) properties variation. (k=0.576)

p	a/h	Cont. App.	N_{cam}=1		N_{cam}=5		N_{cam}=20	
			w_{adm}	Dev(%)	w_{adm}	Dev(%)	w_{adm}	Dev(%)
0	5	5.1530E-07	5.1575E-07	-0.09%	5.1575E-07	-0.09%	5.1553E-07	-0.04%
	10	6.8161E-06	6.8166E-06	-0.01%	6.8166E-06	-0.01%	6.8125E-06	0.05%
	20	1.0338E-04	1.0339E-04	-0.01%	1.0339E-04	-0.01%	1.0358E-04	-0.19%
0.5	5	1.3772E-06	1.3727E-06	0.33%	1.3779E-06	-0.05%	1.3744E-06	0.20%
	10	1.8750E-05	1.8248E-05	2.68%	1.8797E-05	-0.25%	1.8817E-05	-0.36%
	20	2.8133E-04	2.7716E-04	1.48%	2.9364E-04	-4.38%	2.8787E-04	-2.32%
1	5	1.8145E-06	2.1529E-06	-18.65%	1.9078E-06	-5.14%	1.8424E-06	-1.54%
	10	2.5039E-05	2.8611E-05	-14.27%	2.5913E-05	-3.49%	2.5165E-05	-0.50%
	20	3.8383E-04	4.3394E-04	-13.06%	3.9705E-04	-3.44%	3.8259E-04	0.32%
2	5	2.3362E-06	3.3892E-06	-45.07%	2.5084E-06	-7.37%	2.3477E-06	-0.49%
	10	3.1504E-05	4.4977E-05	-42.77%	3.3866E-05	-7.50%	3.1514E-05	-0.03%
	20	4.7327E-04	6.8149E-04	-44.00%	5.2206E-04	-10.31%	4.6013E-04	2.78%

By observing tables V and VI, we can easily verify that the results obtained considering a small number of layers lead to considerable deviations when compared with the ones obtained by a continuous approximation, in specially for bigger values of the exponent p. As it should be expected, when that parameter p is null, the consideration of different approaches have no effect in the results since, in practice, the plate is an isotropic homogeneous material composed only by the ceramic phase.

Additionally, from the tables, one can conclude that using a discrete approach, and twenty layers, is already a fair layer discretization, the results obtained are practically the same obtained by a continuous approximating for the material properties distribution along the plate thickness. The results show to be better when thick or moderately-thick plates are considered.

Table VII presents the fundamental frequency values calculated for the plate composed by Al/ A_2O_3.

Table VII – Fundamental frequency: Continuous (CA), discrete (DA) properties variation. (k=5/6)

p	a/h	Cont. App.	N_{cam}=1		N_{cam}=2		N_{cam}=5		N_{cam}=20	
			ω_{adm}	Dev(%)	ω_{adm}	Dev(%)	ω_{adm}	Dev(%)	ω_{adm}	Dev(%)
0	5	2.1126	2.1122	0.02%	2.1121	0.02%	2.1124	0.01%	2.1125	0.00%
	10	0.5769	0.5769	0.00%	0.5769	0.00%	0.5769	0.00%	0.5769	0.00%
	20	0.1476	0.1478	-0.14%	0.1477	-0.07%	0.1477	-0.07%	0.1477	-0.07%
0.5	5	1.5869	1.6116	-1.56%	1.5968	-0.62%	1.5911	-0.26%	1.5875	-0.04%
	10	0.4310	0.4396	-2.00%	0.4343	-0.77%	0.4320	-0.23%	0.4312	-0.05%
	20	0.1103	0.1126	-2.09%	0.1109	-0.54%	0.1106	-0.27%	0.1102	0.09%
1	5	1.4585	1.4020	3.87%	1.4136	3.08%	1.4499	0.59%	1.4604	-0.13%
	10	0.3973	0.3824	3.75%	0.3937	0.91%	0.3943	0.76%	0.3971	0.05%
	20	0.1017	0.0979	3.74%	0.0981	3.54%	0.1009	0.79%	0.1019	-0.20%
2	5	1.3755	1.2178	11.46%	1.2626	8.21%	1.3461	2.14%	1.3737	0.13%
	10	0.3764	0.3323	11.72%	0.3429	8.90%	0.3675	2.36%	0.3757	0.19%
	20	0.0965	0.0851	11.81%	0.0877	9.12%	0.0941	2.49%	0.0964	0.10%
10	5	1.2410	1.0749	13.38%	1.0892	12.23%	1.1627	6.31%	1.2295	0.93%
	10	0.3425	0.2935	15.67%	0.2973	13.20%	0.3190	6.86%	0.3392	0.96%
	20	0.0622	0.0548	11.90%	0.0551	11.41%	0.0591	4.98%	0.0615	1.13%

From table VII, we can also conclude as previously. The fundamental frequencies obtained through the discrete volume fraction approach are closer to the ones obtained via the continuous approach, when a greater number of layers is used. Concerning to this effect when associated to the exponent p value, it is possible to say that the deviations between these approaches increase with the increase of the exponent.

4.3 Volume fraction distributions

In this second study carried out, one wants to compare the effect of using each of the volume fraction distributions previously presented, i.e. the exponential law and the exponent power law, either concerning to the static deflection as well as with the free vibration response of the FGM plate. In the present case, it was used the Rule of Mixtures to obtain the equivalent properties. The exponent p of the power law distribution was determined in order to obtain the same bending stiffness as if it was estimated by using the exponential law volume fraction distribution. For the study of the free vibrations a similar approach was considered to relate both distributions, thus in this later case the exponent p of the power law was calculated in order to obtain the same second moment of mass as the obtained through the exponential law.

The FGM's used in the present study were Al/ZrO_2, Al/A_2O_3 and Al/WC. Different geometric configurations were also adopted for the cross section of the plates. The FGM were divided into three distinct layers, being the top and bottom layers composed entirely by metallic material and ceramic material, respectively, and the middle layers by a mixture of those materials. The thicknesses of the top and bottom layers are always identical and we make the middle plate thickness vary considering the relations e_c/h presented in next tables, where e_c and h denotes the thickness of mid-layer and h the total thickness of the plate.

Tables VIII to X show the results obtained on that study, for plates of Al/ZrO_2, Al/A_2O_3 and Al/WC, respectively. The values of the deviations (*Dev*) showed in the tables were calculate according the equation: $Dev=\left(value_{EL}-value_{PL}\right)/value_{PL}$, where the subscripts *EL* and *PL* refer to the values obtained when is used the exponential law and the power law, respectively.

Table VIII – Mid-plane deflection. PL and EL volume fraction distributions (E_c/E_m=2.86; k=5/6).

a/h	e_c/h	p_{eqv}	w_{adm}		Dev. (%)
			Exp. Law	Power Law	
5	1/3	0.9886	2.8846E-006	2.8637E-006	0.73%
	2/3	0.9135	2.8224E-006	2.7826E-006	1.43%
	7/9	0.8675	2.8238E-006	2.7641E-006	2.16%
10	1/3	0.9886	4.0852E-005	4.0702E-005	0.37%
	2/3	0.9135	3.9894E-005	3.9386E-005	1.29%
	7/9	0.8675	3.9683E-005	3.8935E-005	1.92%
20	1/3	0.9886	6.3320E-004	6.3069E-004	0.40%
	2/3	0.9135	6.1752E-004	6.1015E-004	1.21%
	7/9	0.8675	6.1903E-004	6.0649E-004	2.07%
50	1/3	0.9886	2.4600E-002	2.4600E-002	0.00%
	2/3	0.9135	2.3000E-002	2.3600E-002	-2.54%
	7/9	0.8675	2.3800E-002	2.3400E-002	1.71%

Table IX – Mid-plane deflection. PL and EL volume fraction distributions (E_c/E_m=5.43; k=5/6).

a/h	e_c/h	p_{eqv}	w_{adm}		Dev. (%)
			Exp. Law	Power Law	
5	1/3	0.982	2.1579E-006	2.0871E-006	3.39%
	2/3	0.8681	2.0788E-006	1.9475E-006	6.74%
	7/9	0.8014	2.0480E-006	1.9179E-006	6.78%
10	1/3	0.982	3.1570E-005	3.0330E-005	4.09%
	2/3	0.8681	2.9778E-005	2.7915E-005	6.67%
	7/9	0.8014	2.9267E-005	2.7408E-005	6.78%
20	1/3	0.982	4.8651E-004	4.7401E-004	2.64%
	2/3	0.8681	4.5848E-004	4.3478E-004	5.45%
	7/9	0.8014	4.5331E-004	4.2589E-004	6.44%

Table X – Mid-plane deflection. PL and EL volume fraction distributions (E_c/E_m=9.94; k=5/6).

a/h	e_c/h	p_{eqv}	w_{adm}		Dev. (%)
			Exp. Law	Power Law	
5	1/3	0.9763	1.7135E-006	1.5655E-006	9.45%
	2/3	0.8305	1.5467E-006	1.3348E-006	15.88%
	7/9	0.7486	1.5230E-006	1.3036E-006	16.83%
10	1/3	0.9763	2.4395E-005	2.2451E-005	8.66%
	2/3	0.8305	2.2366E-005	1.9475E-005	14.84%
	7/9	0.7486	2.1933E-005	1.8870E-005	16.23%
20	1/3	0.9763	3.7873E-004	3.4822E-004	8.76%
	2/3	0.8305	3.4809E-004	3.0331E-004	14.76%
	7/9	0.7486	3.4105E-004	2.9460E-004	15.77%

From these three tables, it can be observed a relevant difference in the deviations when we consider distinct ceramic-phases of the FGMs, leading to the conclusion that when the ratio between the Young' Modulus of the two material-phases is high, the deviations between the results become higher. Thus although providing a similar bending stiffness in both calculations, other elastic coefficients show to become more relevant namely concerning to shear. We also can conclude, from the previous tables, that the ratio *a/h* has not on general a predominant effect on the deviations calculated.

Tables XI and XII present the results obtained considering the goals of this study, for the fundamental frequencies, on Al/ZrO_2 and Al/A_2O_3 plates, respectively. The values $p_{eqvstiff}$ and $p_{eqvmass}$ in the third column of the table correspond to the exponents calculated in order to obtain the same bending stiffness and second moment of mass, respectively.

Table XI – Fundamental frequency. PL and EL V_f distributions (E_c/E_m=2.86; ρ_c/ρ_m=2.11; k=5/6).

a/h	e_c/h	$p_{eqvstiff}/p_{eqvmass}$	ω_{adm}				
			Exp. Law	Power Law	Dev. (%)	Power Law	Dev. (%)
5	1/3	0.9886/0.9918	1.8984	1.8940	-0.23%	1.8940	-0.23%
	2/3	0.9135/0.9372	1.9246	1.9263	0.09%	1.9252	0.03%
	7/9	0.8675/0.9028	1.9317	1.9400	0.43%	1.9386	0.36%
10	1/3	0.9886/0.9918	0.5146	0.5119	-0.52%	0.5173	0.52%
	2/3	0.9135/0.9372	0.5233	0.5222	-0.21%	0.5219	-0.27%
	7/9	0.8675/0.9028	0.5255	0.5266	0.21%	0.5261	0.11%
20	1/3	0.9886/0.9918	0.1314	0.1305	-0.68%	0.1308	-0.46%
	2/3	0.9135/0.9372	0.1338	0.1335	-0.22%	0.1334	-0.30%
	7/9	0.8675/0.9028	0.1343	0.1346	0.22%	0.1345	0.15%
50	1/3	0.9886/0.9918	0.0211	0.0209	-0.95%	0.0212	0.47%
	2/3	0.9135/0.9372	0.0215	0.0214	-0.47%	0.0214	-0.47%
	7/9	0.8675/0.9028	0.0216	0.0216	0.00%	0.0215	-0.46%

Table XII – Fundamental frequency. PL and EL Vf distributions (E_c/E_m=5.43; ρ_c/ρ_m=1.40; k=5/6).

a/h	e_c/h	$p_{eqvstiff}/p_{eqvmass}$	ω_{adm}				
			Exp. Law	Power Law	Dev. (%)	Power Law	Dev. (%)
5	1/3	0.982/0.9962	1.479	1.5018	1.54%	1.5018	1.54%
	2/3	0.9681/0.9705	1.5023	1.5536	3.41%	1.5611	3.91%
	7/9	0.8014/0.9538	1.5077	1.5684	4.03%	1.5843	5.08%
10	1/3	0.982/0.9962	0.399	0.4032	1.05%	0.4038	1.20%
	2/3	0.9681/0.9705	0.4074	0.4195	2.97%	0.421	3.34%
	7/9	0.8014/0.9538	0.4094	0.4244	3.66%	0.4276	4.45%
20	1/3	0.982/0.9962	0.1016	0.1029	1.28%	0.1028	1.18%
	2/3	0.9681/0.9705	0.1061	0.107	0.85%	0.1073	1.13%
	7/9	0.8014/0.9538	0.1047	0.1082	3.34%	0.1092	4.30%

The values obtained for the fundamental frequencies considering this study show to be better when compared with those obtained for the mid-plane deflection.

In fact, considering a Al/ZrO_2 plate, the results for fundamental frequencies obtained considering the two volume distribution laws with a equivalent exponent p showed to provide small deviations, and the ratios a/h and e_c/h seem to have not any effect on those deviations.

As observed in the mid-plane deflection calculation cases, when the ratios between the ceramic and metallic materials Young's Modulus are higher, the deviations follow the same trend.

4.4 Stresses distributions

This last study has the purpose of calculating and representing the stresses distributions through the thickness z of a thick plate (a/h=5), considering the same approximations taken into account in the first study. Table XIII show the results of the stresses (Pa) on the top of the plate at its plane center (x=y=$a/2$).

Table XIII – Stresses on the top of the plate for x=y=$a/2$.

k=0.576		σxx x10^6						σyy x10^6					
a/h	p	Cont	1	2	5	10	20	Cont	1	2	5	10	20
5	0	1.494	1.495	1.496	1.496	1.496	1.496	1.494	1.495	0.149	1.495	1.495	1.495
	1	5.594	1.473	2.576	3.721	4.517	4.930	5.594	1.473	2.576	3.723	4.517	4.930
	2	7.048	1.482	2.415	3.631	4.675	5.493	7.027	1.482	2.415	3.631	4.703	5.492
	10	9.379	1.495	1.561	2.097	2.991	4.292	9.382	1.495	1.561	2.079	2.991	4.294

Figure III – Representation of the stresses (Pa) through the thickness of the plate considering a continuous approximations (left) and discrete approximation (right) with one layer (p=0).

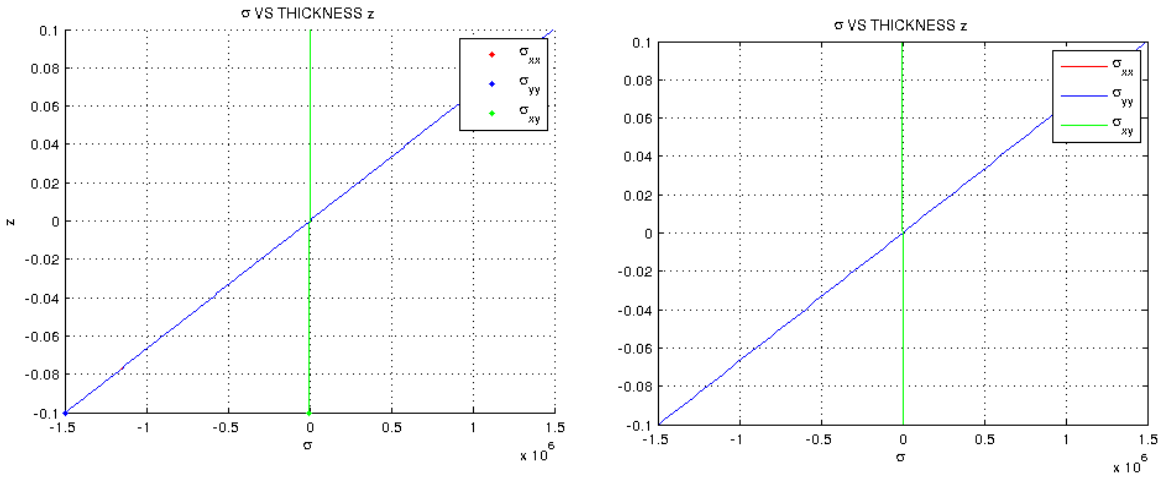


Figure IV – Representation of the stresses (Pa) through the thickness of the plate considering a continuous approximations (f)) and discrete approximation with one, two, five, ten and twenty layers – respectively from a) to e) (p=1).

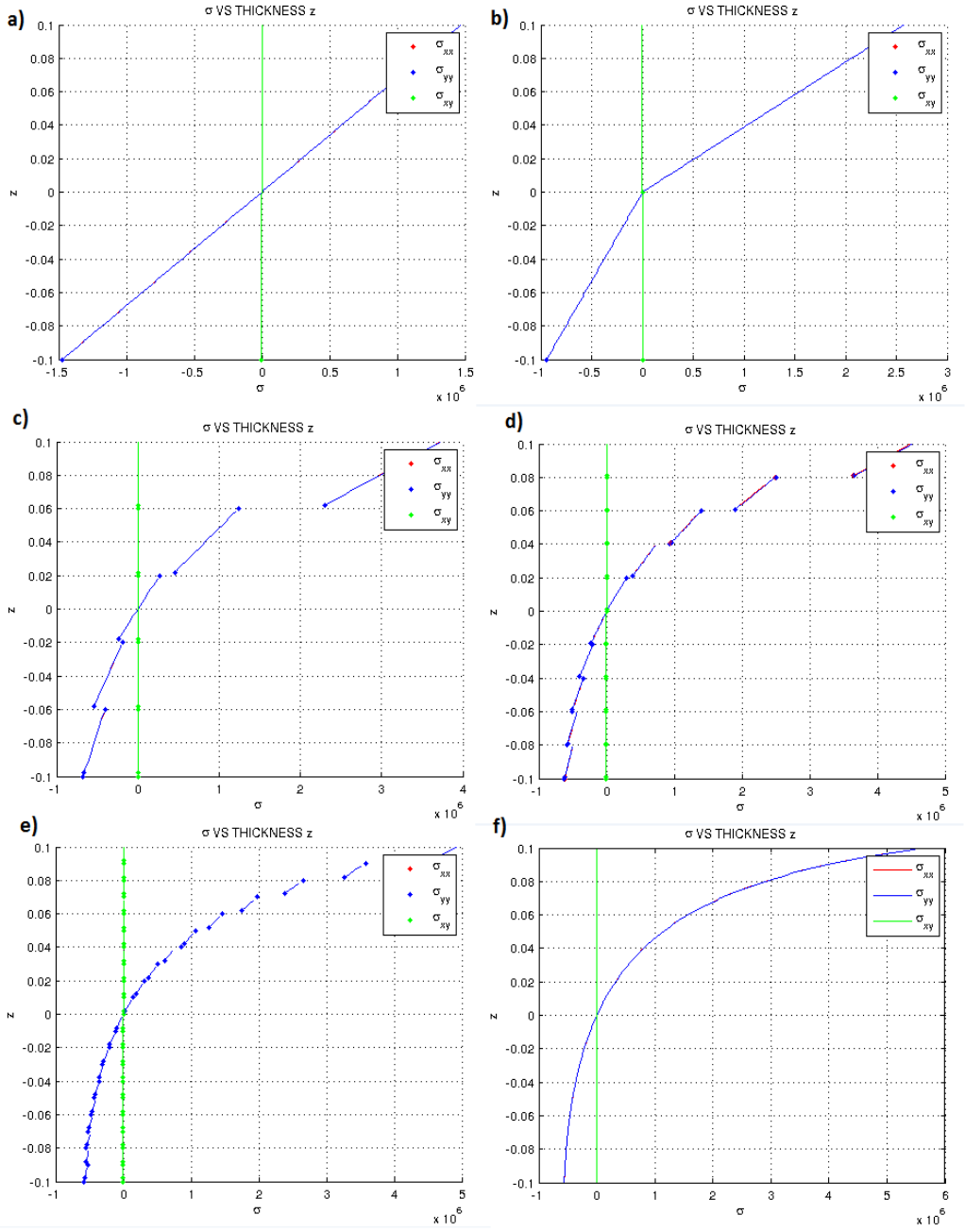
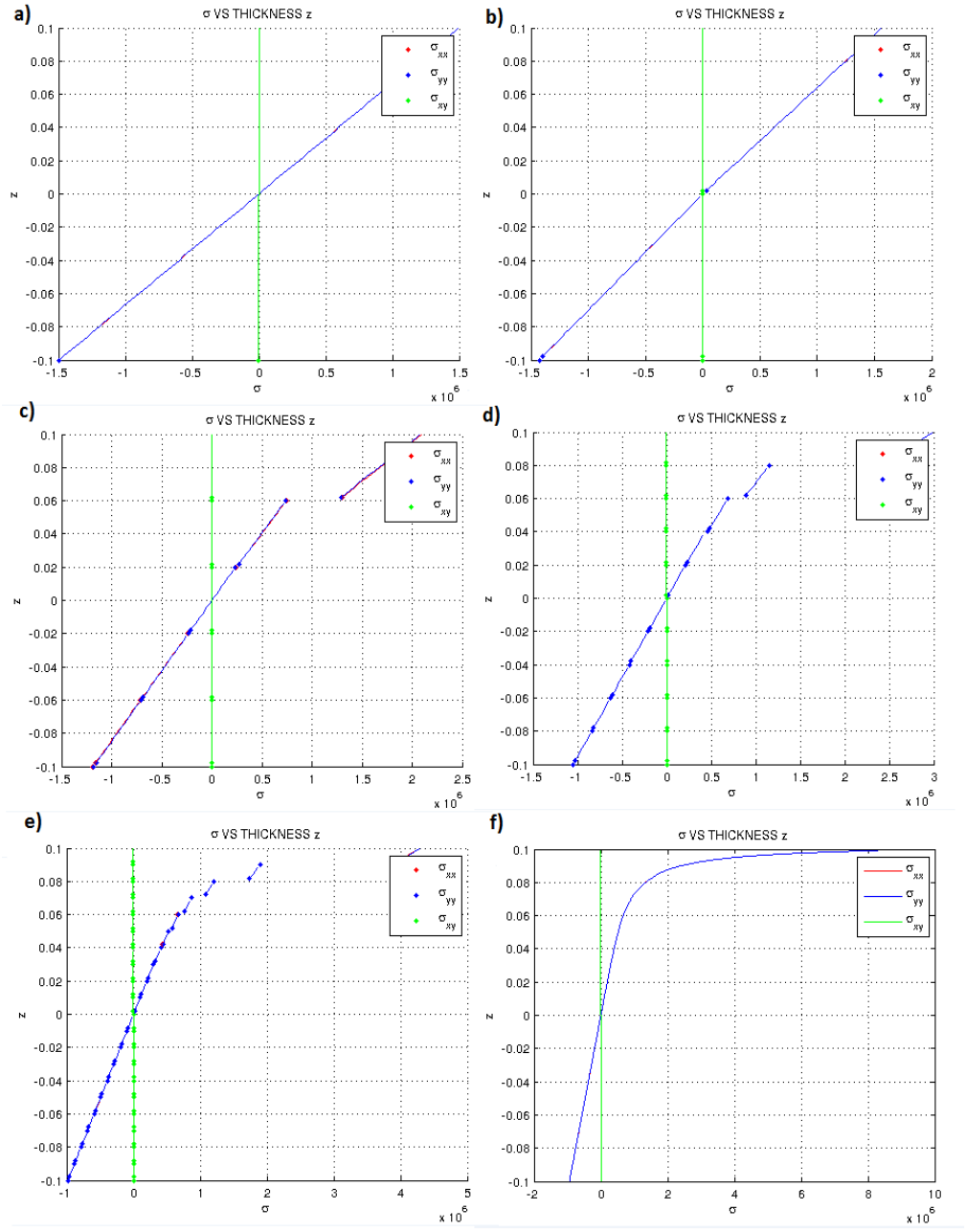


Figure V – Representation of the stresses (Pa) through the thickness of the plate considering a continuous approximations (f)) and discrete approximation with one, two, five, ten and twenty layers – respectively from a) to e) (p=10).



From these figures, and as it should be expected, in the case when exponent p=0, the different approaches considered have not any effect on the tensions distribution since in practice, the plate is entirely composed by an isotropic homogeneous material. For that reason, it was shown the results just considering the situations when is considered one layer.

Observing figures IV and V, we can conclude that the effect on the discontinuity of the stresses is related to the number of discrete layers used to approximate the FGM properties evolution in each point of the plate. In the cases where a small number of layers is used, it is obvious that these discontinuities are more relevant and by the values observed in table XIII, one can say that they are far from the values obtained if it is considered a continuous approximation through the thickness, which, in true, represents the effective conceptual evolution of the properties of an FGM. The values shown in the table also allow us to verify that for a smaller exponent p, the value of stresses converge faster to the results obtained in the continuous approximation, with the augment of the layers number considered on the discrete approach. Globally, by observing the sequence of figures it can be concluded that the stresses distributions converge with the increase of the number of layers used through the thickness.

These discontinuity results correspond to the pattern that one would be expecting as it is similar to what happens in fibre-reinforced laminated plates, namely concerning to the occurrence of abrupt discontinuities in the stresses at the interfaces of the layers.

5. Conclusions/Further Work

This work considered the use of a meshless approach based on multiquadric radial basis functions interpolation to calculate the transverse displacements on each point of a square functionally graded material plate under uniformly distributed load, considering different combinations of materials, as well as different homogenization techniques to predict their effective equivalent properties. Static and free vibrations analyses were carried out to enable

the characterization of the influence of different volume fraction distributions and homogenization schemes on the mechanical performance of the plate.

The results obtained in this work, allow to conclude that multiquadric method have a notable potential when used to interpolate functions in order to proceed to structural analysis on functionally graded materials, granting results that are very close from those obtained by other authors, which consider distinct approaches. The implemented package of procedures, also allows considering different expeditious methodologies to predict the equivalent elastic properties of the FGM.

A main area of future work, already in progress, is to implement higher shear deformation theories, as TSDT, and the further exploration of radial basis functions by considering different methods to discretize the functions, for example, by using RBFs with different natures as the compactly-supported radial basis functions.

Bibliography

[1] T.-K. Nguyen et al., Shear correction factors for functionally graded plates, Mechanics of Advanced Materials and Structures, 14:8, 567-575, 2007.

[2] J. N. Reddy, Analysis of functionally graded plates, International Journal for Numerical Methods in Engineering 47, 663-684, 2000.

[3] D. P. H. Hasselman and G. E. Youngblood, Enhcenced thermal stress resistence of structural ceramics with thermal conductivity gradient, Journal of the American Ceramic Society 61 (1,2):49-53, 1978.

[4] Y. Fukui and N. Yamanaka, Elastic analysis for thin-walled tubes of functionally graded material subjected to internal pressure, International Journal of Japan Society of Mechanical Engineers, Series A 1992, 35:379-385, 1992.

[5] Y. Fukui et al., The stress ans strains in a thin-walled tube of functionally graded materials under uniform thermal loading, International Journal of Japan Society of Mechanical Engineers, Series A 1993, 36:156-162, 1993.

[6] T. Fuchiyama et al., Analysis of thermal stress and stress intensity factor of functionally gradient materials, Ceramic Transitions, Functionally Gradient Materials 1993, 34:425-432, 1993.

[7] T.-K. Nguyen et al., First-order shear deformation plate models for functionally graded materials. Composite Structures 83 (2008) 25-36, 2008.

[8] M. A. R. Loja et al., A study on the modeling of sandwich functionally graded particulate composites, Composite Structures 94 (2012) 2209-2217, 2012.

[9] L. V. Tran et al., Isometric analysis of functionally graded plates using higher-order shear deformation theory, Composites: Part B 51 (2013), 368-383, 2013.
[10] A. J. M. Ferreira, R. C. Batra, C. M. C. Roque, L. F. Qian, P. A. L. S. Martins, Static analisys of functionally graded plates using third-order shear deformation theory and a meshless method, *Composite Structures 69:449-457*, 2005.
[11] A. J. M. Ferreira et al., Natural frequencies of functionally graded plates by a meshless method, Composite Structures 75 (2006) 593-600, 2006.
[12] A. J. M. Ferreira et al., Static deformations and vibration analysis of composite and sandwich plates using a layerwise theory and multiquadrics discretizations, Engineering Analysis with Boundary Elements 29 (2005) 1104-1114, 2005.
[13] Leitão, A meshless method for Kirchho- plate bending problems, *International Journal for Numerical Methods in Engineering 2001: 52:1107–1130*, May 2001.
[14] G. R. Liu and Y. T. Gu, An introduction to meshfree methods and their programming, pages 54-131; 310-359, *Springer*, 2005, ISBN-13 978-1-4020-3468-8.
[15] A. Bouchafa et al., Analytical modelling of thermal residual stresses in exponential functionally graded material system, Material and Design 91 (2010), 560-563, 2010.
[16] M. Bhandari and K. Purohit, Analysis of functionally graded material plate under tranverse load for various boundary conditions, IOSR Journal of Mechanical and Civil Engineering 10:46-55, Issue 5, 2014.
[17] J.N. Reddy, Mechanics of laminated composite plates and shells – Theory and Analysis, 2nd Edition, CRC Press, 2004.
[18] J C. Carr et al., "Surface interpolation with radial basis functions for medical imaging", IEEE Transactions Medical on Imaging, vol. 16(1), February 1997, pp. 96-107.
[19] J. G. Wang and G. R. Liu, On the optimal parameters of radial basis functions used for 2-D meshless methods, *Compt. Methods Mech. Engrg. 191:2611-2630*, 2002.
[20] E. J. Kansa, Multiquadrics – a scattered data approximation scheme with applications to computational fluid dynamics, I: surface approximations and partial derivative estimates. Computational Mathematical Applications, 19(8/9): 127-145, 1990.
[21] G. M. S. Bernardo and M. A. R. Loja, Optimization of structures modeled with a meshfree approach, 6th World Congress on Natural and Biologically Inspired Computing, 2014.
[22] V. P. Nguyen, T. Rabczuk, S. Bordas and M. Duflot, Meshless methods: a review and computer implementation aspects, *Mathematics and Computers in Simulation 79:763-813*, 2008.
[23] M. A. R. Loja, On the use of particle swarm optimization to maximize bending stiffness of functionally graded structures, *Journal of Symbolic Computation 61-62 (2014):12-30*, April 2014.
[24] M.A.R. Loja, C.M. Mota Soares, J.I. Barbosa, Optimization of magneto-electro-elastic composite structures using differential evolution, *Composite* Structures, 107 (2014): 276‑287, 2014.

SYMCOMP 2015
Faro, March 26-27, 2015

CONTRACTIONS OF PARTICULAR TYPES OF NILPOTENT LIE ALGEBRAS OF LOWER DIMENSIONS

J.M. Escobar[1*], J. Núñez[1] and P. Pérez-Fernández[2]

1: Dpto. de Geometría y Topología
Facultad de Matemáticas
Universidad de Sevilla
Calle Tarfia s/n. 41012-Sevilla (E)
e-mail: {pinchamate@gmail.com, jnvaldes@us.es}

2: Dpto. de Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Campus de La Cartuja. Sevilla
e-mail: pedropf@us.es

Keywords: Contractions, invariant functions ψ and φ; filiform Lie algebras.

Abstract. *In this paper, we use two invariant functions of Lie algebras, named ψ and φ function, respectively, to study contractions of certain particular types of nilpotent Lie algebras of lower dimensions, basically filiform Lie algebras.*

1 INTRODUCTION

At present, the study of certain physical concepts has significantly increased due to its significance in *limit processes* which allow us to relate Lie algebras between themselves. These processes were first investigated by Segal [12] in 1951 and two are the better known examples of them. The first of them involves the connection between classical mechanic and relativistic mechanic, with their respective Poincaré symmetry group and Galilean symmetry group. The second one is the limit process by which quantum mechanic is contracted to classical mechanic, when $\hbar \to 0$, which actually corresponds to a contraction of the Heisenberg algebra to the abelian algebra of the same dimension.

For these reasons physical or mathematical contractions are of great interest nowadays, not only for their applications but for the proper study of their algebraic properties. Indeed, after Segal, the concept of limit process between physical theories in terms of contractions of their associated symmetry groups was formulated by Erdal Inönü and Eugene Wigner [7, 8], who introduced the so-called *Inönü-Wingner contractions or IW- contractions*. Later, Saletan [11] studied a more general class of *one-parameter contractions*, for which the elements of the corresponding matrices are one-degree polynomials with respect the contraction parameter (in fact, WI-contractions are a subclass of Saletan contractions). Other extensions of the IW- contractions are, for instance, the *generalized Inönü-Wigner contractions*, introduced by Melsheiner [4], the *parametric degenerations* [2, 3, 13], very used in the Algebraic Invariants Theory, and the *singular contractions* [8].

Continuing with this research, the main goal of this paper is to study the proper contractions of filiform Lie algebras of lower dimensions by using the invariant functions ψ and φ, introduced in 2007 by Hrivnák and Novotný [9], as a tool.

The reason for dealing with filiform Lie algebras in particular (algebras introduced by M. Vergne in the late 60's of the past century [14]) is that these algebras are the most structured algebras within the nilpotent Lie algebras, which allows us to use and study them easier than other Lie algebras and later to try to generalize the properties obtained on them to the case of nilpotent Lie algebras (an historical evolution of Lie algebras in general can be checked in [1]).

The structure of the paper is as follows: in Section 1 we give some preliminaries on Lie algebras in general and on filiform Lie algebras, in particular. Section 2 is devoted to extend some results by Hrivnák and Novotný [9, 10] to the case of filiform Lie algebras of dimension 3. Proper contractions between Heisenberg algebras and filiform Lie algebras are studied in Section 4 and finally, we point out certain conclusions about the results that we have found throughout the paper.

2 Preliminaries

We show in this section some preliminaries on filiform Lie algebras, invariant functions in Lie algebras and proper contractions of Lie algebras.

2.1 Preliminaries on filiform Lie algebras.

In this subsection we show some preliminaries on Lie algebras in general and on filiform Lie algebras in particular. For a further review on these topics, the reader can consult [6]. An n-dimensional *Lie algebra* $\mathfrak{g}$ over a field K is an n-dimensional vector space over K endowed with a second inner law, named *bracket product*, which is bilinear and anti-commutative and which satisfies the *Jacobi identity*

$$J(u,v,w) = [u,[v,w]] + [v,[w,u]] + [w,[u,v]] = 0, \text{ for all } u,v,w \in \mathfrak{g}. \tag{1}$$

The *center* of $\mathfrak{g}$ is the set $Z(\mathfrak{g}) = \{u \in \mathfrak{g} \mid [u,v] = 0, \text{ for all } v \in \mathfrak{g}\}$. The Lie algebra is said to be *abelian* if $[u,v] = 0$, for all $u,v \in \mathfrak{g}$.
Two Lie algebras $\mathfrak{g}$ and $\mathfrak{h}$ are *isomorphic* if there exists a vector space isomorphism f between them such that

$$f([u,v]) = [f(u),f(v)], \text{ for all } u,v \in \mathfrak{g}. \tag{2}$$

It is denoted as $\mathfrak{g} \cong \mathfrak{h}$. The *lower central series* of a Lie algebra $\mathfrak{g}$ is defined as

$$\mathfrak{g}^1 = \mathfrak{g}, \ \mathfrak{g}^2 = [\mathfrak{g}^1,\mathfrak{g}], \ \ldots, \ \mathfrak{g}^k = [\mathfrak{g}^{k-1},\mathfrak{g}], \ \ldots \tag{3}$$

If there exists $m \in \mathbb{N}$ such that $\mathfrak{g}^m \equiv 0$, then $\mathfrak{g}$ is called *nilpotent*. The *nilpotency class* of $\mathfrak{g}$ if the smallest natural c such that $\mathfrak{g}^{c+1} \equiv 0$.
An n-dimensional nilpotent Lie algebra $\mathfrak{g}$ is said to be *filiform* if it is verified that

$$\dim \mathfrak{g}^k = n-k, \text{ for all } k \in \{2,\ldots,n\}. \tag{4}$$

The only n-dimensional filiform Lie algebra for $n < 3$ is the abelian. For $n \geq 3$, it is always possible to find an *adapted basis* $\{e_1,\ldots,e_n\}$ of $\mathfrak{g}$ such that

$$\begin{cases} [e_1,e_2] = 0, \\ [e_1,e_j] = e_{j-1}, \text{ for all } j \in \{3,\ldots,n\}, \\ [e_2,e_j] = [e_3,e_j] = 0, \text{ for all } j \in \{3,\ldots,n\}. \end{cases} \tag{5}$$

If $n \geq 4$, then the following two integers are invariants by isomorphism [5].

$$z_1 = min\{i \geq 4 | [e_i,e_n] \neq 0\}, \quad z_2 = min\{i \geq 4 | [e_i,e_{i+1}] \neq 0\}. \tag{6}$$

From the condition of filiformity and the Jacobi identity (1), the bracket product of $\mathfrak{g}$ is determined by (5) and the products

$$[e_i,e_j] = \sum_{k=2}^{\min\{i-1,n-2\}} c_{ij}^k e_k, \quad \text{for} \quad 4 \leq i < j \leq n, \tag{7}$$

where $c_{i,j}^k \in K$ are called *structure constants* of $\mathfrak{g}$. If all of them are zeros, then the filiform Lie algebra $\mathfrak{g}$ is called *model*. From the invariants (6), the model algebra is not isomorphic to any other algebra of the same dimension. From (5) and (7), every n-dimensional filiform Lie algebra $\mathfrak{g}$ having an adapted basis $\{e_1,\ldots,e_n\}$ verifies that

$$\mathfrak{g}^2 = \langle e_2,\ldots,e_{n-1}\rangle, \ \mathfrak{g}^3 = \langle e_2,\ldots,e_{n-2}\rangle, \ldots, \mathfrak{g}^{n-1} = \langle e_2\rangle, \ \mathfrak{g}^n = 0. \tag{8}$$

2.2 Invariant functions in Lie algebras

In this subsection we recall the definitions and main properties of invariant functions ψ and φ, obtained by Hrivnák and Novotný [9] in 2007.

2.2.1 The invariant function ψ

Definition 2.1 *Let $\mathfrak{g}$ be a Lie algebra. An endomorphism d of $\mathfrak{g}$ is said to be a* (α,β,γ)-derivation *of $\mathfrak{g}$ if there exist $\alpha,\beta,\gamma\in\mathbb{C}$ such that*

$$\alpha\, d[X,Y]=\beta\,[dX,Y]+\gamma\,[X,dY],\quad \forall X,Y\in\mathfrak{g}$$

The set of $(\alpha,\beta,\gamma)-$*derivations* of $\mathfrak{g}$ will be denoted by $Der_{(\alpha,\beta,\gamma)}\mathfrak{g}$.
Note that this definition is the extension of the usual definition of *derivation* of a Lie algebra when $\alpha=\beta=\gamma=1$.

Theorem 2.2 *Let $f:\mathfrak{g}\mapsto\widetilde{\mathfrak{g}}$ be an isomorphism between two complex Lie algebras $\mathfrak{g}$ and $\widetilde{\mathfrak{g}}$. Then, the mapping $\rho: End\,\mathfrak{g}\mapsto End\,\widetilde{\mathfrak{g}}$ defined as $\rho(d)=f\,d\,f^{-1}$ is an isomorphism between the corresponding vector spaces $Der_{(\alpha,\beta,\gamma)}\mathfrak{g}$ and $Der_{(\alpha,\beta,\gamma)}\widetilde{\mathfrak{g}}$, $\forall\alpha,\beta,\gamma\in\mathbb{C}$.*

Corollary 2.3 *The dimension of the vector space $Der_{(\alpha,\beta,\gamma)}\mathfrak{g}$ is an invariant of the Lie algebra $\mathfrak{g}$, $\forall\alpha,\beta,\gamma\in\mathbb{C}$.*

Definition 2.4 *The functions $\psi_{\mathfrak{g}},\psi^0_{\mathfrak{g}}:\mathbb{C}\mapsto\{0,1,2,...,(\dim\mathfrak{g})^2\}$ defined as*

$$(\psi_{\mathfrak{g}})(\alpha)=\dim Der_{(\alpha,1,1)}\mathfrak{g} \tag{9}$$

$$(\psi^0_{\mathfrak{g}})(\alpha)=\dim Der_{(\alpha,1,0)}\mathfrak{g} \tag{10}$$

are called $\psi_{\mathfrak{g}}$ and $\psi^0_{\mathfrak{g}}$ invariant functions *corresponding to the (α,β,γ)-derivations of $\mathfrak{g}$.*

Theorem 2.5 *Two 3-dimensional complex Lie algebras $\mathfrak{g}_1$ and $\mathfrak{g}_2$ are isomorphic if and only if $\psi_{\mathfrak{g}_1}=\psi_{\mathfrak{g}_2}$.*

2.2.2 The invariant function φ.

Definition 2.6 *Let (V,f) be a representation of the Lie algebra $\mathfrak{g}$, where V is a complex vector space. A V-*cochain *of dimension q is a q-linear mapping*

$$c:\underbrace{\mathfrak{g}\times\ldots\times\mathfrak{g}}_{q-times}\mapsto V$$

such that $c(x_1,\ldots,x_i,\ldots,x_j,\ldots,x_q)+c(x_1,\ldots,x_j,\ldots,x_i,\ldots,x_q)=0$, for all indices i,j, with $1\leq i<j\leq q$.

The vector space of all V-cochains of dimension q with $q \in \mathbb{N}$ and $C^0(\mathfrak{g}, V) = V$ will be denoted by $C^q(\mathfrak{g}, V)$. Now, we define the mapping $d : C^q(\mathfrak{g}, V) \mapsto C^{q+1}(\mathfrak{g}, V)$, with $q = 0, 1, 2, \ldots$ as

$$dc(x) = f(x)\, c, \text{ with } c \in C^0(\mathfrak{g}, V),$$

$$dc(x_1, \ldots, x_{q+1}) = \sum_{i=1}^{q+1} (-1)^{i+1}\, f(x_i)\, c(x_1, \ldots, \hat{x_i}, \ldots, x_{q+1})$$

$$= + \sum_{\substack{i,j=1 \\ i<j}}^{q+1} (-1)^{i+j}\, c([x_i, x_j], x_1, \ldots, \hat{x_i}, \ldots, \hat{x_j}, \ldots, x_{q+1}).$$

where $\hat{x_i}$ means that x_i has been omitted.

Under the same conditions as before, let $\kappa = (\kappa_{ij})$ be a symmetric complex matrix of dimension $(q+1) \times (q+1)$.

Definition 2.7 *A κ-twisted cocycle (or simply κ-cocycle) is any $c \in C^q(\mathfrak{g}, V)$, with $q \in \mathbb{N}$, verifying*

$$\begin{aligned} 0 = & \sum_{i=1}^{q+1} (-1)^{i+1} \kappa_{ii} f(x_i) c(x_1, \ldots, \hat{x_i}, \ldots, x_{q+1}) \\ & + \sum_{\substack{i,j=1 \\ i<j}}^{q+1} (-1)^{i+j} \kappa_{ij} c([x_i, x_j], x_1, \ldots, \hat{x_i}, \ldots, \hat{x_j}, \ldots, x_{q+1}). \end{aligned}$$

If the vector space V is identified with the algebra $\mathfrak{g}$, then the adjoint representation can be used as an action (see [6]). So, the equality of the definition 2.7 can be written as

$$\begin{aligned} 0 = & \sum_{i=1}^{q+1} (-1)^{i+1} \kappa_{ii}\ [x_i, c(x_1, \ldots, \hat{x_i}, \ldots, x_{q+1})] \\ & + \sum_{\substack{i,j=1 \\ i<j}}^{q+1} (-1)^{i+j} \kappa_{ij}\, c([x_i, x_j], x_1, \ldots, \hat{x_i}, \ldots, \hat{x_j}, \ldots, x_{q+1}). \end{aligned}$$

The set of all κ-cocycles of dimension q will be denoted by $Z^q(\mathfrak{g}, f, \kappa)$. Clearly, it is a vector subspace of $C^q(\mathfrak{g}, V)$.

If the following notation is considered

$$coc(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3) = Z^2 \left(\mathfrak{g}, ad_{\mathfrak{g}}, \begin{pmatrix} \beta_1 & \alpha_2 & \alpha_3 \\ \alpha_2 & \beta_3 & \alpha_1 \\ \alpha_3 & \alpha_1 & \beta_2 \end{pmatrix} \right),$$

it is easy to see that the vector space $coc(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ is constituted by the $B \in C^2(\mathfrak{g}, \mathfrak{g})$ such that $\forall X, Y, Z \in \mathfrak{g}$,

$$\begin{aligned} 0 = \alpha_1 B(X,[Y,Z]) + \alpha_2 B(Z,[X,Y]) + \alpha_3 B(Y,[Z,X]) \\ + \beta_1 [X, B(Y,Z)] + \beta_2 [Z, B(X,Y)] + \beta_3 [Y, B(Z,X)]. \end{aligned} \tag{11}$$

Theorem 2.8 *Let $g : \mathfrak{g} \mapsto \widetilde{\mathfrak{g}}$ be an isomorphism between Lie algebras $\mathfrak{g}$ and $\widetilde{\mathfrak{g}}$. Then, the mapping $\rho : C^q(\mathfrak{g}, \mathfrak{g}) \mapsto C^q(\widetilde{\mathfrak{g}}, \widetilde{\mathfrak{g}})$, for $q \in \mathbb{N}$, defined by $(\rho c)(x_1, \ldots, x_q) = gc(g^{-1}x_1, \ldots, g^{-1}x_q)$, $\forall c \in C^q(\mathfrak{g}, \mathfrak{g})$ and $\forall x_1, \ldots, x_q \in \widetilde{\mathfrak{g}}$, is an isomorphism between the vector spaces $C^q(\mathfrak{g}, \mathfrak{g})$ and $C^q(\widetilde{\mathfrak{g}}, \widetilde{\mathfrak{g}})$.*

Corollary 2.9 *The dimension of the vector space $Z^q(\mathfrak{g}, ad_{\mathfrak{g}}, \kappa)$ is an invariant of the Lie algebra $\mathfrak{g}$, for any $q \in \mathbb{N}$ and any complex $(q+1)$-square symmetric matrix κ.*

Definition 2.10 *The invariant functions φ and φ^0 corresponding to the n-dimensional Lie algebra $\mathfrak{g}$ are defined as*

$$\begin{aligned} \varphi &: \mathbb{C} \mapsto \{0, 1, \ldots, \frac{n^2(n-1)}{2}\} \\ (\varphi\mathfrak{g})(\alpha) &= \dim coc_{(1,1,1,\alpha,\alpha,\alpha)}\mathfrak{g} \end{aligned} \tag{12}$$

and

$$\begin{aligned} \varphi^0 &: \mathbb{C} \mapsto \{0, 1, \ldots, \frac{n^2(n-1)}{2}\} \\ (\varphi^0\mathfrak{g})(\alpha) &= \dim coc_{(0,1,1,\alpha,1,1)}\mathfrak{g}, \end{aligned} \tag{13}$$

respectively.

2.3 Proper contractions of Lie algebras

Let $\mathfrak{g} = (V, [,])$ be an n-dimensional *Lie algebra* and $U : (0,1] \mapsto \mathfrak{gl}(V)$ be an one-parameter mapping. If the limit

$$[X, Y]_0 = \lim_{\varepsilon \to 0^+} U^{-1}(\varepsilon)\,[U(\varepsilon)X, U(\varepsilon)Y]$$

exists for all $X, Y \in \mathfrak{g}$, we say that $\mathfrak{g}_0 = (V, [,]_0)$ is an one-parameter contraction of the algebra $\mathfrak{g}$ and we write $\mathfrak{g} \mapsto \mathfrak{g}_0$. The contraction $\mathfrak{g} \mapsto \mathfrak{g}_0$ is said to be *proper* if $\mathfrak{g}$ is not isomorphic to $\mathfrak{g}_0$.

The following results were shown in [10].

Theorem 2.11 *If $\mathfrak{g}_0$ is a proper contraction of the complex Lie algebra $\mathfrak{g}$, then*

1. $\dim Der(\mathfrak{g}) < \dim Der(\mathfrak{g}_0)$.
2. *$\psi\mathfrak{g} \leq \psi\mathfrak{g}_0$ and $\psi\mathfrak{g}(1) < \psi\mathfrak{g}_0(1)$.*
3. *$\varphi\mathfrak{g} \leq \varphi\mathfrak{g}_0$ and $\varphi^0\mathfrak{g} \leq \varphi^0\mathfrak{g}_0$.*

Moreover, it is satisfied that, in dimension 3, Condition 2 is a characterization of proper contractions of $\mathfrak{g}$.

3 Extending the definitions of the ψ and φ invariant functions to the case of filiform Lie algebras

In this section we extend the definitions of the invariant functions ψ and φ introduced by Hrivnák and Novotný [9] to the case of filiform Lie algebras of lower dimensions. On a sake of example, we only deal with the 3-dimensional case. Greater dimensions will be tackled in a similar way in future work.

3.1 The ψ invariant function for 3-dimensional filiform Lie algebras

Let $\mathfrak{f}_3$ the filiform Lie algebra of dimension 3 defined by the law $[e_1, e_3] = e_2$ (remember that, by agreement, all possible brackets not appearing in the expression of the law are considered null, that is, in this case, $[e_1, e_2] = [e_2, e_3] = 0$).
Let consider $d \in Der_{(\alpha,1,1)}\mathfrak{f}_3$. Then

$$\alpha d\big([X,Y]\big) = [d(X), Y] + [X, d(Y)] \quad \forall X, Y \in \mathfrak{f}_3. \tag{14}$$

Also, let

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

be the matrix associated with the endomorphism d.

We wish to obtain the elements of this matrix. To do this, for the pair of generators (e_1, e_2) the condition (14) is now

$$\alpha d\big([e_1, e_2]\big) = [d(e_1), e_2] + [e_1, d(e_2)],$$

and $d(e_i) = \sum_{h=1}^{3} a_{ih}\, e_h$. We can get the first condition to be fulfilled for the elements of this endomorphism according to

$$\begin{array}{rcl} \alpha d\big([e_1, e_2]\big) & = & \alpha d\big(0\big) = 0 \\ = [a_{11}e_1 + a_{12}e_2 + a_{13}e_3, e_2] & + & [e_1, a_{21}e_1 + a_{22}e_2 + a_{23}e_3] = a_{23}e_2, \end{array}$$

which implies $a_{23} = 0$.

Proceeding in the same way with the following pair (e_1, e_3), we have that from $\alpha d\big([e_1, e_3]\big) = [d(e_1), e_3] + [e_1, d(e_3)]$, we deduce, by taking into account the law of the algebra, that

$$\alpha\, d(e_2) = [a_{11}e_1 + a_{12}e_2 + a_{13}e_3, e_3] + [e_1, a_{31}e_1 + a_{32}e_2 + a_{33}e_3] = (a_{11} + a_{33})\, e_2.$$

So, $\alpha\, a_{21}e_1 + \alpha\, a_{22}e_2 + \alpha\, a_{23}e_3 = (a_{11} + a_{33})\, e_2$, and thus, according to the linear dependence, the following conditions are obtained

$$\alpha\, a_{21} = 0, \quad \alpha\, a_{22} = a_{11} + a_{33}, \quad \alpha\, a_{23} = 0.$$

Finally, by proceeding in the same way with the last pair (e_2, e_3), we have that

$$\alpha d\big([e_2, e_3]\big) = [d(e_2), e_3] + [e_2, d(e_3)] \equiv$$

$$0 = [a_{21}e_1 + a_{22}e_2 + a_{23}e_3, e_3] + [e_2, a_{31}e_1 + a_{32}e_2 + a_{33}e_3] = a_{21}e_2 = 0.$$

As a result, we find that $a_{21} = 0$.

Summarizing, we have obtained the following conditions

From pair (e_i, e_j)	Conditions
(e_1, e_2)	$a_{23} = 0.$
(e_1, e_3)	$\alpha a_{21} = 0, \quad \alpha a_{22} = a_{11} + a_{33}, \quad \alpha a_{23} = 0.$
(e_2, e_3)	$a_{21} = 0.$

which allow us to determine the vector space $Der_{(\alpha,1,1)}\mathfrak{f}_3$. Indeed

For $\alpha \neq 0$:

$$Der_{(\alpha,1,1)}\mathfrak{f}_3 = \mathrm{span}_{\mathbb{C}} \left\{ \begin{pmatrix} \alpha & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \right.$$
$$\left. \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right\}.$$

For $\alpha = 0$:

$$Der_{(0,1,1)}\mathfrak{f}_3 = \mathrm{span}_{\mathbb{C}} \left\{ \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \right.$$
$$\left. \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \right\}.$$

Therefore, $\psi_{\mathfrak{f}_3(\alpha)} = 6$, $\forall \alpha \in \mathbb{C}$. According to the notation used in [9], it is expressed in the following way

Note that this dimension is always the same independently of the value of α.

α	$\forall\, \alpha \in \mathbb{C}$
$\psi_{\mathfrak{f}_3(\alpha)}$	6

3.2 The φ invariant function for 3-dimensional filiform Lie algebras

The $\mathfrak{f}_3$-cochains $B_i \in C^2(\mathfrak{f}_3,\mathfrak{f}_3)$, $\forall i \in \{0,1,....,\frac{n^2(n-1)}{2}\}$, are defined starting from their non-null commutativity relations $B_r(e_s,e_t) = ke_u$, $s<t$ which verify the equality Eq. (11). The $\mathfrak{f}_3$-cochains which constitute a basis of the vector space $coc_{(1,1,1,\lambda,\lambda,\lambda)}\mathfrak{f}_3$ whose dimension determines the invariant function $\varphi_{\mathfrak{f}_3}$ are the following

For $\lambda = 0$:

$$\begin{array}{lll} B_1\colon\ B_1(e_1,e_2)=e_1 & B_2\colon\ B_2(e_1,e_2)=e_2 & B_3\colon\ B_3(e_1,e_2)=e_3 \\ B_4\colon\ B_4(e_2,e_3)=e_1 & B_5\colon\ B_5(e_2,e_3)=e_2 & B_6\colon\ B_6(e_2,e_3)=e_3 \\ B_7\colon\ B_7(e_1,e_3)=e_1 & B_8\colon\ B_8(e_1,e_3)=e_2 & B_9\colon\ B_9(e_1,e_3)=e_3 \end{array}$$

For $\lambda \neq 0$:

$$\begin{array}{lll} C_1\colon\ C_1(e_1,e_2)=e_1,\quad C_1(e_2,e_3)=e_3 & & C_2\colon\ C_2(e_2,e_3)=e_2 \\ C_3\colon\ C_3(e_2,e_3)=e_1 & & C_4\colon\ C_4(e_1,e_2)=e_2 \\ C_5\colon\ C_5(e_1,e_2)=e_3 & & C_6\colon\ C_6(e_1,e_3)=e_1 \\ C_7\colon\ C_7(e_1,e_3)=e_2 & & C_8\colon\ C_8(e_1,e_3)=e_3 \end{array}$$

Therefore, we have the following vector spaces

$$\begin{array}{ll} coc_{(1,1,1,0,0,0)}\mathfrak{f}_3 & = span_{\mathbb{C}}\{B_i, 1\leq i\leq 9\} \\ coc_{(1,1,1,\lambda,\lambda,\lambda)}\mathfrak{f}_3 & = span_{\mathbb{C}}\{C_i, 1\leq i\leq 8\}, \forall \lambda\neq 0. \end{array}$$

So, we have as a result

λ	0	$\forall \lambda \in \mathbb{C}\setminus\{0\}$
$\varphi_{\mathfrak{f}_3}(\lambda)$	9	8

4 Some examples of proper contractions between different types of algebras

In this section we study proper contractions from filiform Lie algebras of lower dimensions to different types of algebras. We will show two cases.

4.1 Proper contractions of 3-dimensional filiform Lie algebras

Theorem 2.11 allows us to know if there exists a proper contraction between 3-dimensional Lie algebras $\mathfrak{g}_1$, $\mathfrak{g}_{3,1}$, $\mathfrak{g}_{2,1}\oplus\mathfrak{g}_1$, $\mathfrak{g}_{3,2}$, $\mathfrak{g}_{3,3}$ and the 3-dimensional filiform Lie Algebras already studied in this paper. By using the corresponding invariant functions $\psi_{3\mathfrak{g}_1}$, $\psi_{\mathfrak{g}_{3,1}}$, $\psi_{\mathfrak{g}_{2,1}\oplus\mathfrak{g}_1}$, $\psi_{\mathfrak{g}_{3,2}}$ and $\psi_{\mathfrak{g}_{3,3}}$, already calculated by Novotný and Hrivnák in [10], we obtain that

$\mathfrak{g}_1$: Abelian Lie algebra

α	$\forall\alpha \in \mathbb{C}$
$\psi_{3\mathfrak{g}_1}(\alpha)$	9

$\mathfrak{g}_{3,1}$: $[e_2, e_3] = e_1$

α	$\forall\alpha \in \mathbb{C}$
$\psi_{\mathfrak{g}_{3,1}}(\alpha)$	6

$\mathfrak{g}_{2,1} \oplus \mathfrak{g}_1 : [e_1, e_2] = e_2$

α	$\forall\alpha \in \mathbb{C} \setminus \{0\}$	0
$\psi_{\mathfrak{g}_{2,1}\oplus\mathfrak{g}_1}(\alpha)$	4	6

$\mathfrak{g}_{3,2} : [e_1, e_3] = e_1, [e_1, e_3] = e_1 + e_2$

α	1	$\forall\alpha \in \mathbb{C} \setminus \{1\}$
$\psi_{\mathfrak{g}_{3,2}}(\alpha)$	4	3

$\mathfrak{g}_{3,3} : [e_1, e_3] = e_1, [e_2, e_3] = e_2$

α	1	$\forall\alpha \in \mathbb{C} \setminus \{1\}$
$\psi_{\mathfrak{g}_{3,3}}(\alpha)$	6	3

As $\psi_{\mathfrak{f}_3} \leq \psi_{3\mathfrak{g}_1}$ and $\psi_{\mathfrak{f}_3}(1) < \psi_{3\mathfrak{g}_1}(1)$, theorem 2.11 assures the existence of a proper contraction from $\mathfrak{f}_3$ to $3\mathfrak{g}_1$. Analogously, the same occurs between $\mathfrak{g}_{3,2}$ and $\mathfrak{f}_3$ since $\psi_{\mathfrak{g}_{3,2}} \leq \psi_{\mathfrak{f}_3}$ and $\psi_{\mathfrak{g}_{3,2}}(1) < \psi_{\mathfrak{f}_3}(1)$.

Moreover, note that $\psi_{\mathfrak{f}_3}(1) = 6$ and $\psi_{\mathfrak{g}_{3,1}}(1) = 6$. Therefore, the same theorem assures that there not exists any proper contraction between $\mathfrak{g}_{3,1}$ and $\mathfrak{f}_3$. Similarly, the same occurs between $\mathfrak{g}_{3,3}$ and $\mathfrak{f}_3$ due to that $\psi_{\mathfrak{f}_3}(1) = 6$ and $\psi_{\mathfrak{g}_{3,3}}(1) = 6$.

Besides, according to theorem 2.5, the algebras $\mathfrak{g}_{3,1}$ and $\mathfrak{f}_3$ are isomorphic. This implies that there is not any proper contraction between themselves.

4.2 Proper contractions between Heisenberg algebras and filiform Lie algebras

We have just seen that there not exists any proper contraction between $\mathfrak{g}_{3,1}$ and $\mathfrak{f}_3$. As $\mathfrak{g}_{3,1}$ is a 3-dimensional Heisenberg algebra, we ask ourselves if there exists a proper contraction between a Heisenberg algebra and a filiform Lie algebra in the case of dimension five.
To deal with this question, let us consider the Heisenberg algebra of dimension 5, defined by the brackets $[e_1, e_3] = e_5$ and $[e_2, e_4] = e_5$.

We want to obtain a basis of $Der_{(\alpha,1,1)}\mathbb{H}_5$ $\forall\alpha \in \mathbb{C}$. To do this, let consider $d \in Der_{(\alpha,1,1)}\mathbb{H}_5$ and the associated matrix with the endomorphism d

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{55} \end{pmatrix}$$

By proceeding in the same way as in the subsection 3.1, we obtain the following conditions for the elements of the matrix

From pair (e_i, e_j)	Conditions
(e_1, e_2)	$a_{14} = a_{23}$.
(e_1, e_3)	$\alpha\, a_{51} = 0, \quad \alpha\, a_{52} = 0, \quad \alpha\, a_{53} = 0,$ $\alpha\, a_{54} = 0, \quad \alpha\, a_{55} = a_{11} + a_{33}$.
(e_1, e_4)	$a_{12} = -\, a_{43}$.
(e_1, e_5)	$a_{53} = 0$.
(e_2, e_3)	$a_{21} = -a_{34}$.
(e_2, e_4)	$a_{11} + a_{33} = a_{22} + a_{44}$.
(e_2, e_5)	$a_{54} = 0$.
(e_3, e_4)	$a_{32} = a_{41}$.
(e_3, e_5)	$a_{51} = 0$.
(e_4, e_5)	$a_{52} = 0$.

This implies that

$$dim(Der_{(\alpha,1,1)}\mathbb{H}_5) = 15, \forall \alpha \in \mathbb{C}$$

and thus

α	$\forall \alpha \in \mathbb{C}$
$\psi_{\mathbb{H}_5}(\alpha)$	15

So, as $\psi_{\mathbb{H}_5} > \psi_{\mathfrak{f}_5}$, Theorem 2.11 proves that there not exists any proper contraction between a Heisenberg algebra and a filiform Lie algebra, both of dimension 5.

REFERENCES

[1] Boza, L., Fedriani, E.M., Núñez, J. and Tenorio, A.F. "A historical review of the classifications of Lie algebras" *Revista de la Unión Matemática Argentina* Vol. **54(2)** , pp. 75-99, 2013

[2] Burde, D. "Degenerations of nilpotent Lie algebras" *J. Lie Theory* Vol. **9** , pp. 193-202, 1999

[3] Burde, D. "Degenerations of 7-dimensional nilpotent" *Comm. Algebra* Vol. **33** , pp. 1259-1277, 2005

[4] Doebner, H. D. and Melsheimer O. "On a class of generalized group contractions" *Nuovo Cimento A* Vol. **49(10)** , pp. 306-311, 1967

[5] Echarte, F.J., Núñez, J. and Ramírez, F. "Description of some families of filiform Lie algebras" *Houston Journal of Mathematics* Vol. **34(1)** , pp. 19-32, 2008

[6] Humphreys, J.E. "Introduction to lie algebras and representation theory". Springer-Verlag, New York, 1972.

[7] Inönü, E. and Wigner, E. "On the contraction of groups and their representations" *Proc. Nat. Acad. Sci. U.S.A.* Vol. **39** , pp. 510-524, 1953

[8] Inönü, E. and Wigner, E. "On a particular type of convergence to a singular matrix" *Proc. Nat. Acad. Sci. U.S.A.* Vol. **40** , pp. 119-121, 1954

[9] Novotný, P. and Hrivnák, J. "On (α, β, γ)-derivations of Lie algebras and corresponding invariant functions" *Journal of Geometry and Physics* Vol. **58(2)** , pp. 208-217, 2008

[10] Novotný, P. and Hrivnák, J. "Twisted cocycles of Lie algebras and corresponding invariant functions" *Linear Algebra and its Applications* Vol. **430(4)** , pp. 1384-1403, 2009

[11] Saletan, E.J. "Contraction of Lie groups" *J. Math. Phys.* Vol. **2** , pp. 1-21, 1961

[12] Segal, I.E. "A class of operator algebras which are determined by groups" *Duke Math. J.* Vol. **18** , pp. 221-265, 1951

[13] Steinhoff, C. "Klassifikation und Degeneration von Lie Algebren Diplomarbeit" *Algebren Diplomarbeit*, Düsseldorf, 1997.

[14] Vergne, M. "Cohomologie des algèbres de Lie nilpotentes, Application à l'étude de la variété des algebres de Lie nilpotentes" *B*ull. Soc. Math. France Vol. **98** , pp. 81-116, 1970

SYMCOMP 2015
Faro, March 26-27, 2015

SOME APPLICATIONS OF SYMBOLIC COMPUTATION IN SPECTRAL THEORY

Ana C. Conceição[1] and José C. Pereira[2]

1: Center for Functional Analysis, Linear Structures and Applications (CEAFEL)
Departamento de Matemática
Faculdade de Ciências e Tecnologia, Universidade do Algarve
Campus de Gambelas 8005-139, Faro, Portugal
e-mail: aicdoisg@gmail.com

2: Center for Environmental and Sustainability Research (CENSE)
Center for Functional Analysis, Linear Structures and Applications (CEAFEL)
Departamento de Engenharia Electrónica e Informática
Faculdade de Ciências e Tecnologia, Universidade do Algarve
Campus de Gambelas 8005-139, Faro, Portugal
e-mail: unidadeimaginaria@gmail.com

Keywords: Spectral algorithms, factorization algorithms, paired singular integral operators, essentially bounded matrix functions, rational matrix functions, *Mathematica* software system

Abstract. *Spectral theory has many applications in several main scientific research areas (Structural Mechanics, Aeronautics, Quantum Mechanics, Ecology, Probability Theory, Electrical Engineering, among others) and the importance of its study is globally acknowledge. In recent years, several software applications were made available to the general public with extensive capabilities of symbolic computation. These applications, known as computer algebra systems (CAS), allow to delegate to a computer all, or a significant part, of the symbolic calculations present in many mathematical algorithms. In our work we use the CAS* Mathematica *to implement for the first time on a computer analytical algorithms developed by us and others within the Operator Theory. The main goal of this paper is to show how the symbolic computation capabilities of* Mathematica *allow us to explore the spectra of several classes of singular integral operators. For the one-dimensional case, nontrivial rational examples, computed with the automated process called [ASpecPaired-Scalar], are presented. For the matrix case, nontrivial essentially bounded and rational examples, computed with the analytical algorithms [AFact], [SInt], and [ASpecPaired-Matrix], are presented. In both cases, it is possible to check, for each considered paired singular integral operator, if a complex number (chosen arbitrarily) belongs to its spectrum.*

1 INTRODUCTION

Factorization Theory has a long and interesting history, closely related to the spectral theory, and its roots lie in the work of Plemelj (see [40]). Both theories have wide application in the study of Riemann-Hilbert boundary value problems, in the Fredholm theory of singular integral operators, in the theory of linear and non-linear differential equations, in linear transport theory, in the theory of diffraction of acoustic and electromagnetic waves, in the theory of scattering and of inverse scattering, in some branches of probability theory, among others (see, for instance, [1], [2], [17], [22], and [37]. One of the most important problems in factorization theory is the computation of the partial indices of factorable matrix functions. In turn, this problem is closely related to the theories of Wiener-Hopf systems of equations and of characteristic systems of singular integral equations with Cauchy kernel (see, for instance, [35] and [36]. The existing algorithms within the factorization theory show, in general, that it is possible to obtain some kind of factorization but are not designed to be implemented on a computer (see, for instance, [5], [6], [8], [9], [21], [23], [25], [30], [33], and [38]). Recently, we developed, and partially implemented on a computer, the generalized factorization algorithm [AFact] for special classes of essentially bounded matrix functions (see [16]). Due to its innovative character, the implementation of [AFact] potentiates the design of algorithms dedicated to specific domains of application (see, for instance, [14] and [19]). In [14] we have presented the [SInt] algorithm, a new calculation technique to compute some classes of Cauchy type singular integrals, which are important in the design of spectral algorithms. In [15] we described the explicit rational functions factorization algorithm [ARFact-Matrix] that computes explicit (left and right) factorizations of given non-singular rational matrix function defined on the unit circle. On the determination of the partial indices, some developments have also been made but, even in the rational case (and in recent publications), the methods are difficult to apply and were not designed to be implemented on a computer (see, for instance, [4], [8], [10], [24], [28], [30], [41], and [42]). In addition, the vast majority of explicit analytical factorization methods depend on the knowledge of the zeros and poles of scalar functions. As a consequence, in many applications in the real world, a numerical analysis of such methods is inevitable. However, due to many non-stability issues, such as the ones affecting the factorization partial indices (see, for instance, [25]), the numerical approach of Factorization Theory is a very difficult problem. Because of this fact, the design of new analytical methods, even if only for some restrict, special classes of matrix functions, is still very significant to the development of such theory. As an example, due to the symbolic and numeric capabilities of *Mathematica*, the [ARFact-Scalar] algorithm (see [15]) always computes the factorization index of the considered non-singular scalar rational function defined on the unit circle.

The development of the spectral theory is motivated by the need to solve problems emerging from several fields in Mathematics and Physics. At present time, some progress has been achieved (see, for instance, [3], [7], [26], [29], [31], and [32]) for some classes of singu-

lar integral operators whose properties allow the use of particular strategies in the study of the spectral problem but, despite several major developments, there is still no general and explicit method for obtaining the spectrum of any given singular integral operator. Similar to the case of factorization theory, the existing algorithms allow, in general, to study the spectrum of some kind of singular integral operators but they are not designed to be implemented on a computer. Recently, we designed (see [20]) two spectral analytical algorithms [ASpecPaired-Scalar] and [ASpecPaired-Matrix], that explore the spectra of some classes of paired singular integral operators, with rational coefficients, defined on the unit circle.

In our work we use the CAS *Mathematica*[1] to implement for the first time on a computer analytical algorithms developed by us and others authors within Operator Theory (see, for instance, [11], [12], [13], [14], [15], [16], [17], [18], [20], and [39]). In the last years we designed and/or implemented analytical algorithms for solving integral equations, analytical algorithms to factorize scalar and matrix functions, calculation techniques to compute singular integrals, and more recently analytical algorithms to study the spectrum and the kernel of several classes of singular integral operators. It is our belief that the construction and implementation on a computer of these kind of analytical algorithms is a very interesting line of research.

In this paper it is shown how the symbolic computation capabilities of *Mathematica* allow us to explore the spectra of several classes of singular integral operators. For the one-dimensional case, nontrivial rational examples, computed with the automated checking process called [ASpecPaired-Scalar], are presented. For the matrix case, nontrivial essentially bounded and rational examples, computed with the analytical algorithms [AFact], [SInt], and [ASpecPaired-Matrix], are given.

The paper also contains some final remarks about our current work and related lines of research that we find potentially interesting.

2 PAIRED SINGULAR INTEGRAL OPERATORS

Let $\mathbb{T}$ denote the unit circle in the complex plane. Let $\mathbb{T}_+$ and $\mathbb{T}_-$ denote the open unit disk and the exterior region of the unit circle (∞ included), respectively.

It is well known that the singular integral operator with Cauchy kernel, $S_{\mathbb{T}}$, defined almost everywhere on $\mathbb{T}$ by

$$S_{\mathbb{T}}\varphi(t) = \frac{1}{\pi i} \int_{\mathbb{T}} \frac{\varphi(\tau)}{\tau - t}\, d\tau,\ t \in \mathbb{T},$$

where the integral is understood in the sense of its principal value, represents a bounded linear operator in $L_2(\mathbb{T})$. In addition, $S_{\mathbb{T}}$ is a selfadjoint and unitary operator in the Lebesgue space $L_2(\mathbb{T})$ (see, for instance, [27] and [33]). Thus, we can associate with this

[1]All the research presented in this paper was done with *Mathematica* 9. At present time, we are using *Mathematica* 10, with no backward compatibility issues to report. For further information on the computer algebra system *Mathematica* visit the Wolfram's website at urlwww.wolfram.com.

operator two complementary Cauchy projection operators

$$P_{\pm} = (I \pm S_{\mathbb{T}})/2,$$

where I represents the identity operator.
The projectors $P_{\pm}$ allow us to decompose the space $L_2(\mathbb{T})$ into the topological direct sum

$$L_2(\mathbb{T}) = L_2^+(\mathbb{T}) \oplus L_2^{-,0}(\mathbb{T}),$$

where $L_2^+(\mathbb{T}) = \mathrm{im}P_+$ and $L_2^{-,0}(\mathbb{T}) = \mathrm{im}P_-$. We also consider $L_2^-(\mathbb{T}) = L_2^{-,0}(\mathbb{T}) \oplus \mathbb{C}$.
Let $L_\infty(\mathbb{T})$ be the space of all essentially bounded functions on the unit circle.
Let $\mathcal{R}(\mathbb{T})$ be the algebra of rational functions without poles on $\mathbb{T}$ and let $\mathcal{R}_{\pm}(\mathbb{T})$ denote the subsets of $\mathcal{R}(\mathbb{T})$ whose elements are without poles in $\mathbb{T}_{\pm}$.
There exist several numerical algorithms and approximation methods for evaluating some classes of singular integrals. Also, there are several analytical techniques that allow the exact computation of singular integrals for particular cases. However, the [SInt] and [SIntAFact] algorithms ([14]) are the only analytical algorithms, up to our knowledge, written and implemented for computing singular integrals with general essentially bounded functions on the unit circle.
Let $\varphi, \psi \in [L_\infty(\mathbb{T})]_{n,n}$. Operators of the form $T = \varphi I + \psi S_{\mathbb{T}}$ and $\widetilde{T} = \varphi I + S_{\mathbb{T}}\psi I$ are linear and bounded singular integral operators (see, for instance, [27]). In the following, these operators will be written in a more convenient form as

$$T_{\{a,b\}} = aP_+ + bP_- \tag{1}$$

and

$$\widetilde{T}_{\{a,b\}} = P_+aI + P_-bI, \tag{2}$$

where $a = \varphi + \psi$ and $b = \varphi - \psi$. We will call these operators, paired singular integral operators, with coefficients a and b.

2.1 Factorization of functions: scalar and matrix cases

Let us now introduce the concept of a generalized factorization for matrix functions (see, for instance, [10] and [37]): we say that a matrix function $r \in [L_\infty(\mathbb{T})]_{n,n}$, that is, a matrix function whose entries are essentially bounded functions on the curve $\mathbb{T}$, admits a left (right) generalized factorization in $L_2(\mathbb{T})$ if it can be represented as

$$r = r_+\Lambda r_- \qquad (r = r_-\Lambda r_+), \tag{3}$$

where

$$r_+^{\pm 1} \in \left[L_2^+(\mathbb{T})\right]_{n,n}, \quad r_-^{\pm 1} \in \left[L_2^-(\mathbb{T})\right]_{n,n}, \quad \Lambda(t) = diag\{t^{\varkappa_i}\}_{j=1}^n,$$

$\varkappa_j \in \mathbb{Z}$, $j = \overline{1,n}$, with $\varkappa_1 \geq \varkappa_2 \geq ... \geq \varkappa_n$, and $r_+P_+r_-I$ $(r_-P_+r_+I)$ represents a bounded linear operator in $[L_2(\mathbb{T})]_n$; the number $\varkappa = \sum_{j=1}^{n} \varkappa_j$ is called the factorization index of the

determinant of the matrix function r. The integers $\varkappa_j$ are called the left (right) partial indices of r. If $\varkappa_j = 0$, $j = \overline{1,n}$, then r is said to admit a left (right) canonical generalized factorization (can. gen. fact.).
Any non-singular rational matrix function $r \in [\mathcal{R}(\mathbb{T})]_{n,n}$ admits a left (right) generalized factorization of the form (3) (see, for instance, [25]), where

$$r_+^{\pm 1} \in [\mathcal{R}_+(\mathbb{T})]_{n,n}\,, \quad r_-^{\pm 1} \in [\mathcal{R}_-(\mathbb{T})]_{n,n}\,.$$

For the particular rational scalar case we note that

$$\varkappa = z_+ - p_+, \tag{4}$$

where z_+ is the number of zeros of r in $\mathbb{T}_+$ (with regard to their multiplicities) and p_+ is the number of poles of r in $\mathbb{T}_+$ (with regard to their multiplicities) (see, for instance, [15]).

Remark 1.

(*i*) A natural and nontrivial question arises concerning the relation of the left and right partial indices of a generalized factorization of a matrix function r. It is well known that the sum of the left partial indices and the sum of the right partial indices are equal, that is, the factorization index $\varkappa$ is uniquely determined by a given matrix function r. It was proved in [24] that this relation is the only existing one between the sets of the left and right partial indices.

(*ii*) The left (right) partial indices $\varkappa_i$ are uniquely determined by a given matrix function r, that is, in a factorization of the form (3), the matrix Λ is uniquely defined. However, this is not true for $r_\pm$ and the general relation between the factors of two distinct generalized factorizations of the same matrix function r is described, for instance, in [25].

2.2 On the spectra of paired singular integral operators

In this subsection it is explained how the study of the factorability of scalar and matrix functions is related to the study of the spectra of the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2), respectively.
The spectrum of a bounded linear operator T is a closed, bounded, and non-empty subset of $\mathbb{C}$, defined by

$$\sigma(T) = \{\lambda \in \mathbb{C} : \, T - \lambda I \text{ is not a bounded invertible operator}\}.$$

In [20] we proved the following spectral result related with the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$:

Theorem 1. *Let $a, b \in [L_\infty(\mathbb{T})]_{n,n}$. If $ab = ba$, then*

$$\sigma(T_{\{a,b\}}) = \sigma(\widetilde{T}_{\{a,b\}}). \tag{5}$$

Remark 2. In the case $ab \neq ba$, the equality (5) is not necessarily satisfied. In fact, due to the symbolic computation capabilities of *Mathematica* and our analytical factorization algorithm [ARFact-Matrix], several interesting examples where $\sigma(T_{\{a,b\}}) \neq \sigma(\widetilde{T}_{\{a,b\}})$ were easily constructed (see Examples 4 and 5). Obviously, in the scalar case the equality (5) is always satisfied (see Subsection 3.1).

It is obvious that if a or b are constant functions, i.e., $a(t) \equiv c$ or $b(t) \equiv c$, for $c \in \mathbb{C}$, then $c \in \sigma(T_{\{a,b\}})$ and $c \in \sigma(\widetilde{T}_{\{a,b\}})$.
Using Theorem 2, Theorem 3, Remark 2, and Remark 3, the following result on the spectra of $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ can be formulated (see [20]).

Theorem 2. Let $a, b \in [L_\infty(\mathbb{T})]_{n,n}$.

(*i*) If $det(a(t) - \lambda_1 e) \equiv 0$ ($\lambda_1 \in \mathbb{C}$) and $det(b(t) - \lambda_2 e) \equiv 0$ ($\lambda_2 \in \mathbb{C}$), then

$$\sigma(T_{\{a,b\}}) = \sigma(\widetilde{T}_{\{a,b\}}) = \{\lambda_1, \lambda_2\}.$$

(*ii*) If $det(a(t) - \lambda e) \not\equiv 0$ ($\forall \lambda \in \mathbb{C}$), then

$$\sigma(T_{\{a,b\}}) = \{\lambda \in \mathbb{C} : (a - \lambda e)^{-1}(b - \lambda e) \text{ does not admit a left can. gen. fact.}\}.$$

$$\sigma(\widetilde{T}_{\{a,b\}}) = \{\lambda \in \mathbb{C} : (b - \lambda e)(a - \lambda e)^{-1} \text{ does not admit a left can. gen. fact.}\}.$$

(*iii*) If $det(b(t) - \lambda e) \not\equiv 0$ ($\forall \lambda \in \mathbb{C}$), then

$$\sigma(T_{\{a,b\}}) = \{\lambda \in \mathbb{C} : (b - \lambda e)^{-1}(a - \lambda e) \text{ does not admit a right can. gen. fact.}\}.$$

$$\sigma(\widetilde{T}_{\{a,b\}}) = \{\lambda \in \mathbb{C} : (a - \lambda e)(b - \lambda e)^{-1} \text{ does not admit a right can. gen. fact.}\}.$$

Remark 3. For the scalar case, by using the [ARFact-Scalar] algorithm (see subsection 3.1), the calculation of the factorization index is always possible in the rational case. As a consequence, for one-dimensional paired singular integral operators of classes (1) and (2) with rational coefficients, it is also always possible to check if a complex number (chosen arbitrarily) belongs to the their spectra.

3 EXPLORING THE SPECTRA OF PAIRED SINGULAR INTEGRAL OPERATORS WITH SYMBOLIC COMPUTATION TECHNIQUES

In this section it is described how the symbolic computation allow us to explore the spectra of several classes of singular integral operators. For the one-dimensional case, nontrivial rational examples, computed with the automated checking process called [ASpecPaired-Scalar], are presented. For the matrix case, nontrivial essentially bounded and rational examples, computed with the analytical algorithms [AFact] and [ASpecPaired-Matrix], are given.

3.1 [ASpecPaired-Scalar] algorithm

In this subsection it is shown how the symbolic and numeric computation capabilities of the computer algebra system *Mathematica* can be used to design and implement a spectral algorithm to explore the spectra of one-dimensional paired singular integral operators of the form

$$T_{\{a,b\}} = aP_+ + bP_- \quad \text{and} \quad \widetilde{T}_{\{a,b\}} = P_+aI + P_-bI,$$

defined in (1) and (2), respectively, with $a, b \in \mathcal{R}(\mathbb{T})$.
The [ASpecPaired-Scalar] algorithm [20] checks if a complex number (chosen arbitrarily) belongs to the spectra of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$. The implementation of this spectral algorithm with the *Mathematica* software system makes the results of lengthy and complex calculations available in a simple way.
This spectral algorithm has a rather simple structure since the knowledge of the factorization index $\varkappa$ of a non-singular scalar rational function, that can be determined by formula (4), is the only information the algorithm needs to determine if a complex number is in the spectra of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ (see Theorem 5). In addition, the symbolic computation capabilities of *Mathematica*, and the *pretty-print* functionality[2], allow the [ASpecPaired-Scalar] code to be very simple and syntactically similar to its analytical counterpart.
The [ASpecPaired-Scalar] algorithm can be applied to any given one-dimensional paired singular integral operator of classes (1) and (2), with rational coefficients. There are two options to input $a(t)$ and $b(t)$:

1. Insert $a(t)$ and $b(t)$ directly.

2. Insert zeros, poles (and multiplicities) of $a(t)$ and $b(t)$.

For each pair of inputed functions $a, b \in \mathcal{R}(\mathbb{T})$, and complex value λ, chosen arbitrarly, the [ASpecPaired-Scalar] algorithm gives one of the following *outputs* (see Remark 2):

$$[\textit{Output 1}] \quad \lambda \in \sigma(T_{\{a,b\}}) \text{ and } \lambda \in \sigma(\widetilde{T}_{\{a,b\}}) \tag{6}$$

[2]The *pretty-print* functionality allows to write on the computer screen scientific formulas in the traditional format, as if one was using pencil and paper.

$$[\textit{Output 2}] \quad \lambda \notin \sigma(T_{\{a,b\}}) \text{ and } \lambda \notin \sigma(\widetilde{T}_{\{a,b\}}) \tag{7}$$

[ASpecPaired-Scalar]

```
1  Input rational functions a(t) and b(t), and complex number λ.
2  if a(t) ≡ λ or b(t) ≡ λ
3     then λ ∈ σ(T_{a,b})                                   ▷ λ ∈ σ(T̃_{a,b})
4     else  r(t) ← (a(t)−λ)/(b(t)−λ)
5           {z_i}_{i←1}^m ← list of zeros of function r(t)
6           {p_j}_{j←1}^n ← list of poles of function r(t)
7           if at least one zero or pole of r(t) lies in 𝕋
8              then λ ∈ σ(T_{a,b})                          ▷ λ ∈ σ(T̃_{a,b})
9              else  z_+ ← number of zeros of r that lie in 𝕋_+
                           with regard to their multiplicities
10                   p_+ ← number of poles of r that lie in 𝕋_+
                           with regard to their multiplicities
11                   ϰ ← z_+ − p_+
12                   if ϰ = 0
13                      then λ ∉ σ(T_{a,b})                 ▷ λ ∉ σ(T̃_{a,b})
14                      else  λ ∈ σ(T_{a,b})                ▷ λ ∈ σ(T̃_{a,b})
```

Figure 1: Pseudo code of [ASpecPaired-Scalar] algorithm. The symbol ▷ denotes a comment.

Figure 1 presents the pseudo code of the [ASpecPaired-Scalar] algorithm.
The analysis of the pseudo code reveals that one main step in this algorithm is the computation of the zeros and poles (with regard to their multiplicities), of the rational function

$$r(t) = (a(t) - \lambda)(b(t) - \lambda)^{-1} \tag{8}$$

and whether they lie in $\mathbb{T}$, $\mathbb{T}_+$, or $\mathbb{T}_-$.
We note that, since the zeros and poles of function r defined in (8) are a crucial information for this spectral algorithm, the success of the [ASpecPaired-Scalar] algorithm depends on the possibility of finding those zeros and poles by solving polynomial equations. This can be a serious limitation when working with polynomials of the fifth degree or higher. However, even in this case, thanks to the symbolic and numeric capabilities of *Mathematica*, it is still possible to check if a complex number λ (chosen arbitrarily) belongs to the spectra of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2).

Mathematica uses `Root` objects to represent solutions of algebraic equations in one variable, when it is impossible to find explicit formulas for these solutions. The `Root` object is not a mere denoting symbol but rather an expression that can be symbolically manipulated and numerically evaluated with any desired precision. In particular, it is still possible to know if any given `Root` lies in $\mathbb{T}$, $\mathbb{T}_+$, or $\mathbb{T}_-$ (see Figure 3). In pratical terms, this means that the factorization index of r (when it exists) is always obtained explicitly by the spectral algorithm, and this is all the information the [ASpecPaired-Scalar] algorithm requires to conclude if a given complex number is in the spectra of the operators.

3.1.1 [ASpecPaired-Scalar] examples

In this subsection we present some nontrivial examples computed by the automated checking process [ASpecPaired-Scalar]. For each pair of inputed functions $a, b \in \mathcal{R}(\mathbb{T})$, and complex value λ, chosen arbitrarly, the [ASpecPaired-Scalar] algorithm gives the *output* (6) or (7).
All the examples were computed on a MacBook Pro with a 2.5 GHz Intel Core i5 processor and 4 GB of DDR3 RAM, running Mac OS X 10.9.5 (Mavericks) in single user mode.

Example 1. Let us consider now the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) with rational scalar coefficients

$$a(t) = t^{21} + 3it + 1 + i \quad \text{and} \quad b(t) = t^6 - it^5 + 3t^2 + (1-3i)t.$$

We want to check if the complex number $\lambda = i$ belongs to the spectra of those operators.

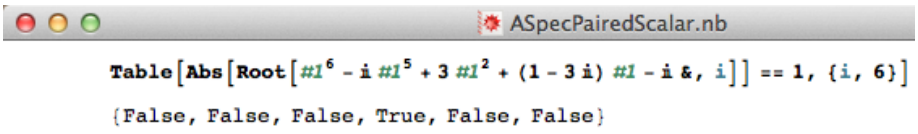


Figure 2: Snippet of the calculations made by the [ASpecPaired-Scalar] algorithm for studying the poles of the auxiliary function r in Example 1. In this case, the pole `Root`$[\#1^6 - i\#1^5 + 3\#1^2 + (1-3i)\#1 - i\&, 4]$ lies in $\mathbb{T}$.

The [ASpecPaired-Scalar] algorithm constructs the auxiliary function

$$r(t) = \frac{t^{21} + 3it + 1}{t^6 - it^5 + 3t^2 + (1-3i)t - i}$$

computes its zeros and poles, with regard to their multiplicities and determines whether they lie in $\mathbb{T}$, $\mathbb{T}_+$, or $\mathbb{T}_-$. Since one of the poles of r lies in $\mathbb{T}$ (see Figure 2) the rational function is not factorable and the algorithm concludes that

$$i \in \sigma(T_{\{a,b\}}) \quad \text{and} \quad i \in \sigma(\widetilde{T}_{\{a,b\}}).$$

Example 2. Let us consider the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) with rational scalar coefficients

$$a(t) = 3t^3 - 5t^2 - 3 + i \quad \text{and} \quad b(t) = t^6 - 3t^4 + t^3 - 2t^2 + 2t + i.$$

We want to check if the complex number $\lambda = 1 + i$ belongs to the spectra of those operators.
The [ASpecPaired-Scalar] algorithm constructs the auxiliary function

$$r(t) = \frac{3t^3 - 5t^2 - 4}{t^6 - 3t^4 + t^3 - 2t^2 + 2t - 1}$$

computes its zeros and poles, with regard to their multiplicities and determines whether they lie in $\mathbb{T}$, $\mathbb{T}_+$, or $\mathbb{T}_-$ (see Figure 3). The factorization index is computed as $\varkappa = 2 - 4 = -2$. Since $\varkappa \neq 0$ the *output* is

$$1 + i \in \sigma(T_{\{a,b\}}) \quad \text{and} \quad 1 + i \in \sigma(\widetilde{T}_{\{a,b\}}).$$

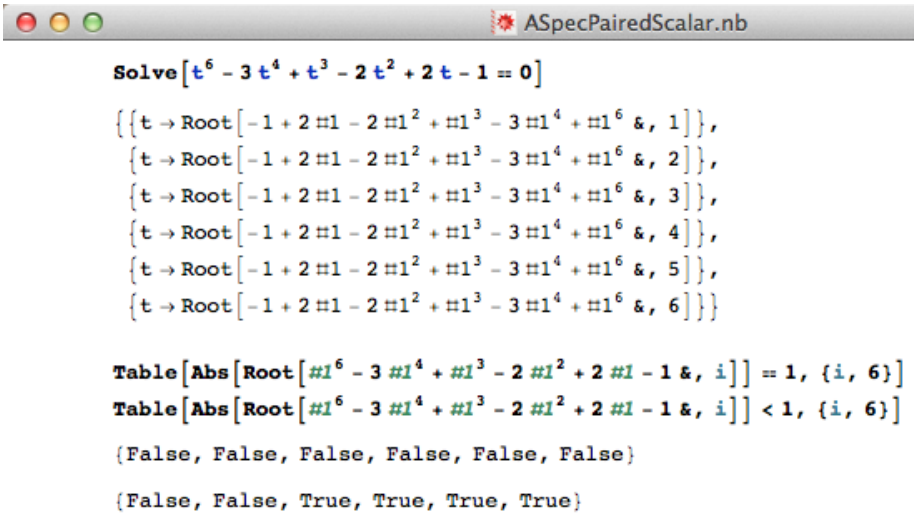


Figure 3: Snippet of the calculations made by the [ASpecPaired-Scalar] algorithm for studying the poles of the auxiliary function r in Example 2. This case shows that, in spite of the impossibility of computing, in an explicit way, the roots of the sixth degree polynomial $t^6-3t^4+t^3-2t^2+2t-1$ it is still possible to know if they lie in $\mathbb{T}$, $\mathbb{T}_+$, or $\mathbb{T}_-$. Here *Mathematica* uses the objects `Root`$[\#1^6-3\#1^4+\#1^3-2\#1^2+2\#1-1\&, i]$ to represent the solutions of $t^6 - 3t^4 + t^3 - 2t^2 + 2t - 1 = 0$. In this case, there are 4, out of 6, poles lying inside the unit circle.

Example 3. Let us consider the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) with rational scalar coefficients

$$a(t) = (t^9 + 5t^2 - 1 - i)t^{-2} \quad \text{and} \quad b(t) = t^9 + t^3 + 2t - 10 - i.$$

Let us check if the complex number $\lambda = 0$ belongs to the spectra of these operators. The [ASpecPaired-Scalar] algorithm constructs the auxiliary function

$$r(t) = \frac{t^9 + 5t^2 - 1 - i}{t^{11} + t^5 + 2t^3 - (10 + i)t^2}$$

computes its zeros and poles, with regard to their multiplicities and determines whether they lie in $\mathbb{T}$, $\mathbb{T}_+$, or $\mathbb{T}_-$ (see Figure 4). The factorization index is computed as $\varkappa = 2 - 2 = 0$ and the algorithm concludes that

$$0 \notin \sigma(T_{\{a,b\}}) \quad \text{and} \quad 0 \notin \sigma(\widetilde{T}_{\{a,b\}}).$$

ASpecPairedScalar.nb

```
(* Zeros of r *)
Table[Abs[Root[#1^9 + 5 #1^2 - 1 - i &, i]] == 1, {i, 9}]
Table[Abs[Root[#1^9 + 5 #1^2 - 1 - i &, i]] < 1, {i, 9}]

{False, False, False, False, False, False, False, False, False}

{False, False, False, True, False, False, True, False, False}

(* Poles of r *)
Table[Abs[Root[#1^11 + #1^5 + 2 #1^3 - (10 + i) #1^2 &, i]] == 1, {i, 11}]
Table[Abs[Root[#1^11 + #1^5 + 2 #1^3 - (10 + i) #1^2 &, i]] < 1, {i, 11}]

{False, False, False, False, False, False, False, False, False, False, False}

{False, False, False, False, True, True, False, False, False, False, False}
```

Figure 4: Snippet of the calculations made by the [ASpecPaired-Scalar] algorithm for studying the zeros and poles of the auxiliary function r in Example 3. In this case there are 2 zeros and 2 poles lying in the interior of $\mathbb{T}$.

3.2 [ASpecPaired-Matrix] algorithm

In this subsection it is explained how the symbolic and numeric computation capabilities of the computer algebra system *Mathematica* can be used to design and implement a spectral algorithm to explore the spectra of paired singular integral operators of the form

$$T_{\{a,b\}} = aP_+ + bP_- \quad \text{and} \quad \widetilde{T}_{\{a,b\}} = P_+aI + P_-bI,$$

defined in (1) and (2), respectively, with $a, b \in [\mathcal{R}(\mathbb{T})]_{n,n}$, $n \geqslant 2$.

The [ASpecPaired-Matrix] algorithm [20] checks if a complex number (chosen arbitrarily) belongs to the spectra of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$. As in the scalar case, the implementation of this spectral algorithm with *Mathematica* makes the results of lengthy and complex calculations available in a simple way.
In the design of this spectral algorithm we used the factorization algorithm [ARFact-Matrix] (see [15]), that computes explicit factorizations for given factorable rational matrix function defined on the unit circle. For this reason, the success of the [ASpecPaired-Matrix] algorithm depends on the possibility of finding solutions of polynomial equations. However, due to the complexity of the matrix case, it is not as feasible as before to use the `Root` objects to obtain an explicit matrix function factorization when working with polynomials of a high degree. In fact, one crucial step of this algorithm is finding the zeros of the determinant of a rational matrix function defined through the matrix functions a and b. This means that the dimension of the matrix is also a limiting factor, even when its entries are rational functions with low degree polynomials. We also note that, although the final factorization may have relatively simple entries, if we were to use the traditional pencil and paper tools the intermediate calculations would take typically many working hours, up to the point of infeasibility, even for low matrix orders.
The [ASpecPaired-Matrix] algorithm can be used to explore the spectra of a given paired singular integral operator of classes (1) or (2).

As in the case of the [ASpecPaired-Scalar] algorithm, for the [ASpecPaired-Matrix] algorithm there are two options to input the entries of $a(t)$ and $b(t)$:

1. Insert the entries of $a(t)$ and $b(t)$ directly.
2. Insert, for each entry, the numerator and the poles (and its multiplicities).

Note that, due to the non-commutativity of matrix function multiplication and the usage of the [ARFact-Matrix] algorithm, the code of [ASpecPaired-Matrix] is ignificantly more complex than the code of the [ASpecPaired-Scalar] algorithm.
For each pair of input functions $a, b \in [\mathcal{R}(\mathbb{T})]_{n,n}$ and complex number λ, the [ASpecPaired-Matrix] algorithm gives one of the following *outputs*:

$$[\textit{Output 1}] \quad \lambda \notin \sigma(T_{\{a,b\}}) \text{ and } \lambda \notin \sigma(\widetilde{T}_{\{a,b\}}) \tag{9}$$

$$[\textit{Output 2}] \quad \lambda \notin \sigma(T_{\{a,b\}}) \text{ and } \lambda \in \sigma(\widetilde{T}_{\{a,b\}}) \tag{10}$$

$$[\textit{Output 3}] \quad \lambda \in \sigma(T_{\{a,b\}}) \text{ and } \lambda \in \sigma(\widetilde{T}_{\{a,b\}}) \tag{11}$$

$$[\textit{Output 4}] \quad \lambda \in \sigma(T_{\{a,b\}}) \text{ and } \lambda \notin \sigma(\widetilde{T}_{\{a,b\}}) \tag{12}$$

In Figure 5 is shown the flowchart of the [ASpecPaired-Matrix] algorithm.

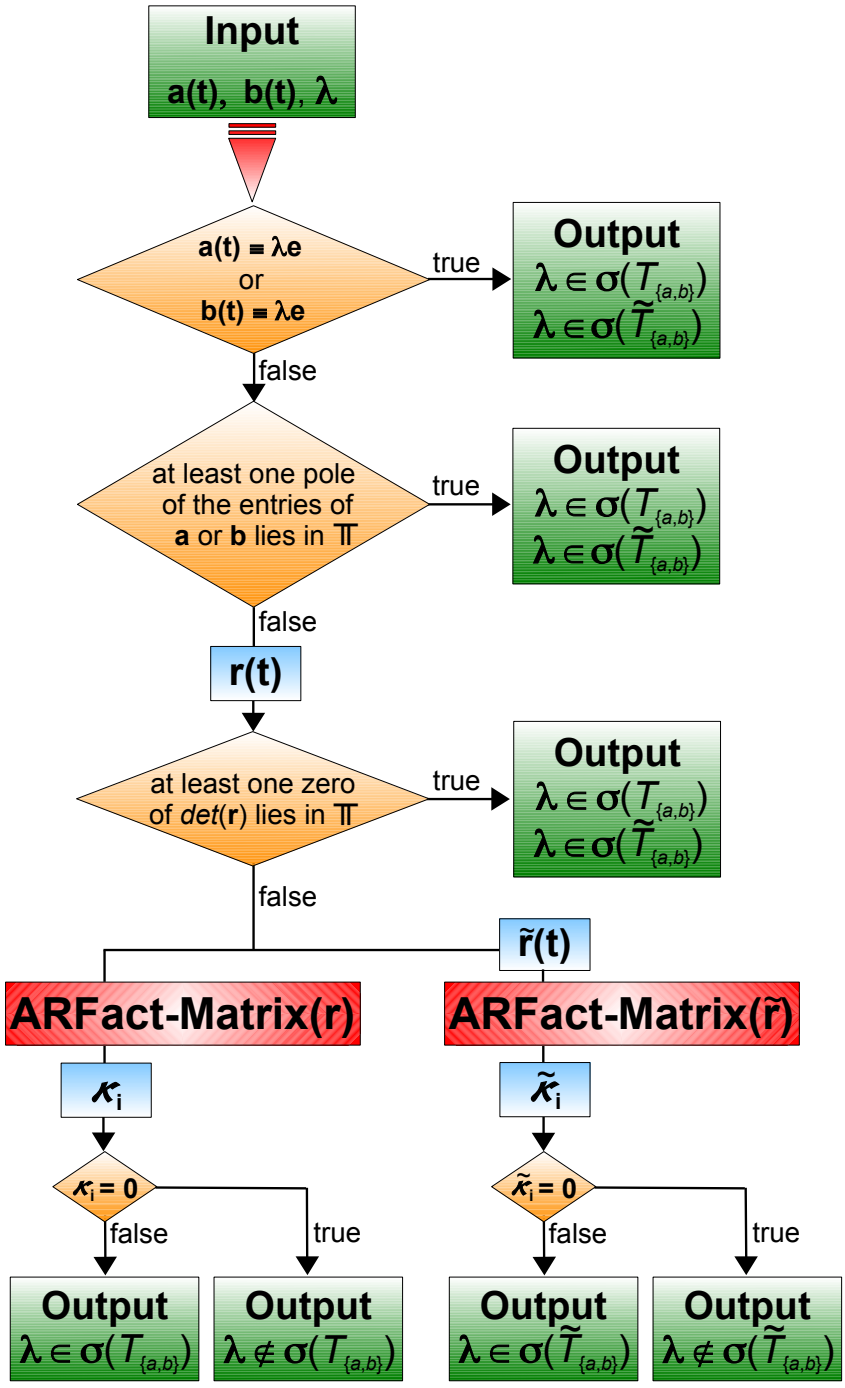


Figure 5: Flowchart of the [ASpecPaired-Matrix] algorithm.

3.2.1 [ASpecPaired-Matrix] examples

In this subsection we present some nontrivial examples computed with the automated checking process [ASpecPaired-Matrix]. For each pair of input functions $a, b \in [\mathcal{R}(\mathbb{T})]_{n,n}$ and complex value λ, the [ASpecPaired-Matrix] algorithm gives the *output* (9), (10), (11), or (12).
Appendix A contains some of the factors $r_\pm$ and $\widetilde{r}_\pm$, computed with the factorization algorithm [ARFact-Matrix]. Note that, although these factors are not used directly by the [ASpecPaired-Matrix] algorithm to study the spectra of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$, they are necessary for the computation of the left partial indices $\varkappa_i$ and $\widetilde{\varkappa}_i$.
All the examples were computed on a MacBook Pro with a 2.5 GHz Intel Core i5 processor and 4 GB of DDR3 RAM, running Mac OS X 10.9.5 (Mavericks) in single user mode.

Example 4. Let us consider the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) with rational matrix coefficients

$$a(t) = \begin{pmatrix} (1-it)t^{-1} & -i \\ -2t^{-1} & t+i \end{pmatrix} \quad \text{and} \quad b(t) = \begin{pmatrix} -2i & t^{-1} \\ t+2i & (-2-it)t^{-1} \end{pmatrix}.$$

Since in this case $ab \neq ba$, then the equality (5) is not necessarily satisfied (see Remark 2).
Let us now check if the complex number $\lambda = -i$ belongs to the spectra of the operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$. There are no poles of the entries of matrix functions a and b lying on $\mathbb{T}$ and therefore, the [ASpecPaired-Matrix] algorithm constructs the auxiliary matrix function

$$r(t) = (a(t) + ie)^{-1}(b(t) + ie) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and computes the determinant of r, $det(r) = -1$. Since there are no zeros of $det(r)$ that lie on $\mathbb{T}$, the [ASpecPaired-Matrix] algorithm constructs

$$\widetilde{r}(t) = (b(t) + ie)(a(t) + ie)^{-1} = \begin{pmatrix} (-it^3 + 2t^2 + 2)t^{-2} & t^{-2} + 1) \\ (t^4 + 4it^3 - 4t^2 - 4)t^{-2} & (it^3 - 2t^2 - 2)t^{-2} \end{pmatrix}$$

and determines the left partial indices $\varkappa_i$ and $\widetilde{\varkappa}_i$ of r and $\widetilde{r}$ as $\varkappa_1 = \varkappa_2 = 0$ and $\widetilde{\varkappa}_1 = 1$ and $\widetilde{\varkappa}_2 = -1$, respectively. As such, the algorithm concludes that

$$-i \notin \sigma(T_{\{a,b\}}) \quad \text{and} \quad -i \in \sigma(\widetilde{T}_{\{a,b\}}).$$

Example 5. Let us consider the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) with rational matrix coefficients

$$a(t)=\begin{pmatrix} -1 & 0 & t \\ 0 & (1-t)t^{-1} & 0 \\ 1 & 0 & -1 \end{pmatrix} \quad \text{and} \quad b(t)=\begin{pmatrix} 0 & 0 & 0 \\ 0 & (1-t)t^{-1} & 0 \\ 0 & 0 & t-1 \end{pmatrix}.$$

In this case we also have $ab \neq ba$. As a consequence, the equality (5) is not necessarily satisfied (see Remark 2).
Let us now check if the complex number $\lambda=-1$ belongs to the spectra of the operators. There are no poles of the entries of matrix functions a and b lying on $\mathbb{T}$ and so, the [ASpecPaired-Matrix] algorithm constructs the auxiliary matrix function

$$r(t)=(a(t)+e)^{-1}(b(t)+e)=\begin{pmatrix} 0 & 0 & t \\ 0 & 1 & 0 \\ t^{-1} & 0 & 0 \end{pmatrix}$$

and computes the determinant of r, $det(r)=-1$. Since there are no zeros of $det(r)$ in $\mathbb{T}$, the [ASpecPaired-Matrix] algorithm computes

$$\widetilde{r}(t)=(b(t)+e)(a(t)+e)^{-1}=\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

and determines the left partial indices $\varkappa_i$ and $\widetilde{\varkappa}_i$ of r and $\widetilde{r}$ as $\varkappa_1=1$, $\varkappa_2=0$, $\varkappa_3=-1$ and $\widetilde{\varkappa}_1=\widetilde{\varkappa}_2=\widetilde{\varkappa}_3=0$, respectively. As such, the algorithm concludes that

$$-1 \in \sigma(T_{\{a,b\}}) \quad \text{and} \quad -1 \notin \sigma(\widetilde{T}_{\{a,b\}}).$$

Example 6. Let us consider the paired singular integral operator $T_{\{a,b\}}$ defined in (1) with rational matrix coefficients

$$a(t)=\begin{pmatrix} (t+1)t^{-1} & 0 & (2t-1)t^{-2} & 0 \\ 0 & 1 & 0 & -i \\ 0 & (t+2i)^{-1} & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

and

$$b(t)=\begin{pmatrix} 1 & \frac{i\left(4t^4-4t^3+18t^2-18t+4\right)}{t^2(t^2+4)} & (1-t)t^{-2} & 2(3t-1)t^{-1} \\ -i(t+1/2)t^{-1} & 2t & i & -3it \\ \frac{t+1/2}{t(t+2i)} & 2i(t+2i)^{-1} & (2t+1+2i)(t+2i)^{-1} & \frac{4t-1}{t(t+2i)} \\ 0 & i(2t-1) & -1 & 3t \end{pmatrix}.$$

We want to check if the complex number $\lambda = 1$ belongs to the spectra of this operators. There are no poles of the entries of matrix functions a and b lying on $\mathbb{T}$ and the [ASpecPaired-Matrix] algorithm constructs the auxiliary matrix function

$$r(t) = (a(t) - e)^{-1}(b(t) - e) = \begin{pmatrix} 0 & i(-2+t)(4+t^2)^{-1} & 1 & (3t-1)t^{-1} \\ (t+1/2)t^{-1} & 2i & t+1 & (4t-1)t^{-1} \\ 0 & i(2t-1) & -1 & 3t-1 \\ (t+1/2)t^{-1} & i(2t-1) & -1 & 3t \end{pmatrix}$$

and computes the determinant of r, $det(r) = \frac{i(1+2t)(-1+3t)\left(2t^4+11t^2+5t+10\right)}{2t^2(4+t^2)}$. Since there are no zeros of $det(r)$ that lie on $\mathbb{T}$, the [ASpecPaired-Matrix] algorithm computes the left partial indices $\varkappa_i$ of r as $\varkappa_1 = \varkappa_2 = \varkappa_3 = \varkappa_4 = 0$. As such, the algorithm concludes that

$$1 \notin \sigma(T_{\{a,b\}}).$$

Example 7. Let us consider the paired singular integral operator $T_{\{a,b\}}$ defined in (1) with rational matrix coefficients

$$a(t) = \begin{pmatrix} \frac{1-2i+it}{t-2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1-i+4it}{4t-1} & 0 & 0 & 0 & 0 & 0 \\ i(t-2)^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & t^{-1} & 0 \\ 0 & 0 & 0 & 0 & 1 & i & 0 \\ -\frac{1}{t(t-2)} & 0 & 0 & 0 & 0 & 0 & 1+i \end{pmatrix}$$

and

$$b(t) = \begin{pmatrix} i & \frac{i}{t-7i} & 0 & \frac{1}{t-2} & 0 & 0 & \frac{1}{t-2} \\ 0 & \frac{i(1+4t)}{4t-1} & \frac{t+1}{4t-1} & 0 & \frac{1}{t} & 0 & 0 \\ 0 & -\frac{1}{t-7i} & i & \frac{i(t-1)}{t-2} & 0 & 0 & \frac{i}{t-2} \\ 0 & 0 & -1 & i & 1 & 0 & 0 \\ \frac{1}{t} & 0 & 0 & 1 & \frac{1+it^2}{t^2} & 0 & 0 \\ 1 & \frac{1}{t} & 0 & 0 & 0 & i & 0 \\ 0 & -\frac{i}{t(t-7i)} & 0 & -\frac{1}{t(t-2)} & 0 & 1 & \frac{it^2-2it-1}{t(t-2)} \end{pmatrix}.$$

We want to check now if the complex number $\lambda = i$ belongs to the spectra of this operators. There are no poles of the entries of matrix functions a and b lying on $\mathbb{T}$ and the [ASpecPaired-Matrix] algorithm constructs the auxiliary matrix function

$$r(t) = (a(t) - ie)^{-1}(b(t) - ie)$$

$$= \begin{pmatrix} 0 & (i(-2+t))(-7i+t)^{-1} & 0 & 1 & 0 & 0 & 1 \\ 0 & 2i & 1+t & 0 & (4t-1)t^{-1} & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & t^{-1} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & t & t^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

and computes the determinant of r, $det(r) = \frac{t^2+(5+2i)t-1}{t^2}$. Since there are no zeros of $det(r)$ that lie on $\mathbb{T}$, the [ASpecPaired-Matrix] algorithm computes the left partial indices $\varkappa_i$ of r as $\varkappa_1 = \varkappa_2 = \varkappa_3 = \varkappa_4 = \varkappa_5 = \varkappa_6 = 0$ and $\varkappa_7 = -1$. As such, the algorithm concludes that

$$i \in \sigma(T_{\{a,b\}}).$$

3.3 Special classes of paired singular integral operators

This subsection is devoted to special classes of paired singular integral operators with essentially bounded coefficients, defined on the unit circle. For some classes of such paired singular integral operators, due the specificity and complexity of the operators, in general, it is very hard to automate the exploration process. However, it is still possible to use, in a step-by-step manner, the symbolic computation capabilities of *Mathematica*, to explore the corresponding spectra.

3.3.1 On the factorability of essentially bounded Hermitian matrix functions

In general, it is possible to show that the study of the factorability of essentially bounded Hermitian second-order matrix functions with negative determinant and definite diagonal elements, can be reduced (see, for instance, [11], [16], and [37]) to the study of the factorability of matrix functions of the form

$$A_\gamma(\varphi) = \begin{pmatrix} 1 & \varphi \\ \overline{\varphi} & |\varphi|^2 + \gamma \end{pmatrix} \tag{13}$$

where φ is an essentially bounded function on the unit circle, that is, $\varphi \in L_\infty(\mathbb{T})$, $\overline{\varphi}$ denotes the complex conjugate of φ in the unit circle, and $\gamma \in \mathbb{C}$. In addition, a canonical generalized factorization of matrix functions of the class (13) has applications in several scientific research areas (see, for instance, [1], [14], [17], [19], [22], [34], [36], and [37]).
Although we have theoretical results for $\varphi \in L_\infty(\mathbb{T})$, we can always assume, without loss of generality, that $\varphi \in L_\infty^+(\mathbb{T})$, that is, φ is an essentially bounded function, holomorphic and bounded on $\mathbb{T}_+$ (see, for instance, [13], [16], [17], and [18]).
In [11], [13], [15], and [16] we have stated necessary and sufficient conditions for the existence of a canonical generalized factorization $A_\gamma(\varphi) = A_\gamma^+ A_\gamma^-$.

Let $H_{r,\theta}(\mathbb{T})$ denote the set of all bounded and analytic functions in $\mathbb{T}_+$ that can be represented as the product of a rational outer function r and an inner function θ (i.e., θ is a bounded analytic function on the interior of the unit circle such that its modulus is equal to one a.e. on $\mathbb{T}$). For the case when $\varphi \in H_{r,\theta}(\mathbb{T})$, we designed the generalized factorization algorithm [AFact] (see [11] and [16]) that allows us to know if a matrix function of the class (13) admits, or not, a left canonical generalized factorization of the form (3).

3.3.2 On the spectra of special classes of paired singular integral operators

Let us now consider the special classes of paired singular integral operators of the form

$$T_{\{a,b\}} = aP_+ + bP_-,$$

defined in (1) with $a, b \in [L_\infty(\mathbb{T})]_{2,2}$ and $\lambda \in \mathbb{C}$, such that $(a - \lambda e)^{-1}(b - \lambda e)$ belongs to class (13), for a matrix function $\varphi \in L_\infty(\mathbb{T})$ and $\gamma \in \mathbb{C}$.
Let us also consider the special classes of paired singular integral operators of the form

$$\widetilde{T}_{\{a,b\}} = P_+ aI + P_- bI \ ,$$

defined in (2) with $a, b \in [L_\infty(\mathbb{T})]_{2,2}$ and $\lambda \in \mathbb{C}$, such that $(b - \lambda e)(a - \lambda e)^{-1}$ belongs to class (13), for a matrix function $\varphi \in L_\infty(\mathbb{T})$ and $\gamma \in \mathbb{C}$.
In [20], based directly on the ideas and concepts that were presented in [16], we formulated new results[3] that relate the spectra of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$with the spectra of the special class of self-adjoint singular integral operators

$$P_+ \varphi P_- \overline{\varphi} P_+ \ ,$$

where $\varphi \in L_\infty(\mathbb{T})$.
Let $T_{\gamma\{a,b\}}$ be a singular integral operator of class (1), with $a, b \in [L_\infty(\mathbb{T})]_{2,2}$ and $\lambda \in \mathbb{C}$, such that $(a - \lambda e)^{-1}(b - \lambda e)$ belongs to class (13), for a matrix function $\varphi \in L_\infty(\mathbb{T})$ and $\gamma \in \mathbb{C}$.

Theorem 3. $\lambda \in \sigma\left(T_{\{a,b\}}\right) \Leftrightarrow -\gamma \in \sigma\left(P_+ \varphi P_- \overline{\varphi} P_+\right)$

Theorem 4. If $P_- \varphi = \varphi$ and $\gamma \in \mathbb{C} \setminus \{0\}$, then $\lambda \notin \sigma\left(T_{\{a,b\}}\right)$.

Theorem 5. If $\gamma \in \mathbb{C} \setminus \mathbb{R}_0^-$, then $\lambda \notin \sigma\left(T_{\{a,b\}}\right)$.

[3] Similar results to Theorems 6, 7, and 8 can be obtained for the case when the transpose of the matrix function $(b - \lambda e)^{-1}(a - \lambda e)$ belongs to class (13).

Remark 4. Similar results to Theorems 3, 4, and 5 were obtained for the spectrum of the singular integral operator $\widetilde{T}_{\{a,b\}}$ defined in (2).

As a consequence, based on Theorem 2.1 of [16], the [AFact] algorithm allows to check if a complex number λ (chosen arbitrarily) belongs to the spectra of the singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) , for the case when $(a-\lambda e)^{-1}(b-\lambda e)$ or $(b-\lambda e)(a-\lambda e)^{-1}$, respectively, belong to class (13). Note that the computations of the [AFact] algorithm do not depend on the degree of the polynomials that may eventually be part of inner function θ. Therefore, for some subclasses of operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$, whose spectra cannot be studied with the [ASpecPaired-Matrix] algorithm due to the many zeros and poles present in the entries of the corresponding matrix coefficients, it may still be possible to use [AFact] to perform this analysis. In addition, our [SInt] algorithm (see [14]) computes the singular integral $S_{\mathbb{T}}\varphi$ for an essentially bounded function φ, and can be used to check, in some particular cases (see Theorem 4), if a complex number λ (chosen arbitrarily) belongs to the spectra of the singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) .

3.3.3 Essentially bounded examples

In this subsection we present nontrivial examples to show how the symbolic capabilities of the computer algebra system *Mathematica*, the generalized factorization algorithm [AFact] ([16]), and the calculation technique [SInt] (see [14]) can be used to explore the spectra of some particular classes of paired singular integral operators related with the class of essentially bounded matrix functions (13). All the examples were computed on a MacBook Pro with a 2.5 GHz Intel Core i5 processor and 4 GB of DDR3 RAM, running Mac OS X 10.9.5 (Mavericks) in single user mode.

We consider the paired singular integral operators $T_{\{a,b\}}$ and $\widetilde{T}_{\{a,b\}}$ defined in (1) and (2) with essentially bounded matrix coefficients

$$a(t)=\begin{pmatrix} i & 0 \\ -\overline{\varphi}\,\gamma^{-1} & [1+(i-1)\gamma]\,\gamma^{-1} \end{pmatrix} \quad \text{and} \quad b(t)=\begin{pmatrix} i & \varphi \\ 0 & i \end{pmatrix},$$

where $\varphi \in L_\infty(\mathbb{T})$, $\gamma \in \mathbb{C}\setminus\{0\}$, and the overline in $\overline{\varphi}$ denotes the complex conjugate of φ defined over the unit circle.
Once more, in this general case we have $ab \neq ba$. So, the equality (5) is not necessarily satisfied (see Remark 2).

Example 8. Let us now check if the complex number $\lambda = i-1$ belongs to the spectra of operator $T_{\{a,b\}}$ defined in (1).
Based on Theorem 2, we define the matrix function

$$r(t) = (a(t) - (i-1)e)^{-1}(b(t) - (i-1)e) = \begin{pmatrix} 1 & \varphi \\ \overline{\varphi} & |\varphi|^2 + \gamma \end{pmatrix}.$$

In this case $r(t)$ is a matrix function of class (13), and therefore Theorems 6, 7, and 8 can be used to check if λ belongs to the spectrum of operator $T_{\{a,b\}}$.

Example 8.1. Let us consider $\varphi(t)$ represented as $\varphi(t) = x(t)y_-(t)$, where $\overline{y_-}$ is an arbitrary essentially bounded function on the unit circle, analytic in the interior of the unit circle, x is a rational function without poles on $\mathbb{T}$, and γ is an arbitrary complex value. In this case the [SInt] algorithm can be used to check if λ belongs to the spectrum of operator $T_{\{a,b\}}$.

(i) Let

$$x(t) = \frac{3it^6 - (8+2i)t - 8}{(i-10)t^9 + 2t^7 + t^6 + 1}$$

The calculation technique [SInt] concludes that $S_{\mathbb{T}}\varphi = -\varphi$, that is, $P_-\varphi = \varphi$. Therefore, by Theorem 4,

$$i - 1 \notin \sigma(T_{\gamma\{a,b\}}).$$

(ii) Let

$$x(t) = \frac{t^{10} + \sqrt{2}it}{-10t^{15} + t^{13} - 2it^{12} + 1}$$

The [SInt] algorithm gives the *output* that $S_{\mathbb{T}}\varphi = -\varphi$, that is, $P_-\varphi = \varphi$. Therefore, by Theorem 4,

$$i - 1 \notin \sigma(T_{\gamma\{a,b\}}).$$

In both cases (i) and (ii), if y_- is a rational function, the study of the spectrum of operator $T_{\{a,b\}}$ can also be done by the [ARFact-Matrix] algorithm. However, due to the complexity of the matrix case, it is not as feasible as before to use the `Root` objects to obtain an explicit matrix function factorization when working with polynomials of such a high degree.

Example 8.2. Let φ be an arbitrary essentially bounded function and $\gamma \in \mathbb{C} \setminus \mathbb{R}_0^-$. By Theorem 8 we can conclude immediately, without using the symbolic computation capabilities of *Mathematica*, that

$$i - 1 \notin \sigma(T_{\gamma\{a,b\}}).$$

Example 8.3. Let us consider

$$\varphi(t) = \frac{\theta(t)}{t-2},$$

where θ is an arbitrary inner function and $\gamma \in \mathbb{R}^-$. In this case the [AFact] algorithm can be used to check if λ belongs to the spectrum of operator $T_{\{a,b\}}$.

(i) Let θ be an arbitrary inner function, differentiable in a neighborhood of $t = 1$, defined on $\mathbb{T}$, and $\gamma = -1$. The factorization algorithm [AFact] finds that the matrix function $r(t)$ admits a canonical generalized factorization in $L_2(\mathbb{T})$. Therefore, we can conclude that

$$1 \notin \sigma\left(P_+\varphi P_-\overline{\varphi}P_+\right).$$

By Theorem 3,

$$i-1 \notin \sigma(T_{\gamma\{a,b\}}).$$

(ii) Let θ be an arbitrary inner function, differentiable in a neighborhood of $t = -1$, defined on $\mathbb{T}$, such that satisfies the condition[4] $\theta'(-1) = 0$ and $\gamma = -\frac{1}{9}$. The [AFact] algorithm finds that the matrix function $r(t)$ admits a non-canonical generalized factorization in $L_2(\mathbb{T})$. Thus, we can conclude that

$$\frac{1}{9} \in \sigma\left(P_+\varphi P_-\overline{\varphi}P_+\right).$$

By Theorem 3,

$$i-1 \in \sigma(T_{\gamma\{a,b\}}).$$

Remark 5. Using Theorem 5, we construct the matrix function

$$\widetilde{r}(t) = (b(t)-(i-1)e)(a(t)-(i-1)e)^{-1} = \begin{pmatrix} 1+|\varphi|^2 & \gamma\varphi \\ \overline{\varphi} & \gamma \end{pmatrix}.$$

Since it is possible to rewrite $\widetilde{r}(t)$ as

$$\widetilde{r}(t) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & \overline{\varphi} \\ \varphi & 1+|\varphi|^2 \end{pmatrix}\begin{pmatrix} 0 & \gamma \\ 1 & 0 \end{pmatrix},$$

the study of the factorability of this matrix function can be reduced to the study of the factorability of

$$\begin{pmatrix} 1 & \overline{\varphi} \\ \varphi & 1+|\varphi|^2 \end{pmatrix}$$

[4]This condition is provided explicitly in the output of the [AFact] algorithm. It arises from the construction of a homogeneous linear system which we know to be uniquely solvable when $-\gamma \in \sigma\left(P_+\varphi P_-\overline{\varphi}P_+\right)$.

which is a matrix function of class (13). From here, Theorem 5.6 of [20] can be used to conclude immediately, without the need to use the symbolic computation capabilities of *Mathematica*, that

$$i - 1 \notin \sigma(\widetilde{T}_{\gamma\{a,b\}}),$$

for all $\varphi \in L_\infty(\mathbb{T})$ and $\gamma \in \mathbb{C} \setminus \{0\}$.

4 CONCLUSIONS

The design of our analytical algorithms is focused on the possibility of implementing on a computer all, or a significant part, of the extensive symbolic and numeric calculations present in the algorithms. The methods developed so far rely on innovative techniques of Operator Theory and have a great potential to be extended of extension to ever more complex and general problems. Also, by implementing these methods on a computer, new and powerful tools are created for exploring that same potential, making the results of lengthy and complex calculations available in a simple way to researchers of different areas.

- We are considering the design and implementation of other factorization, spectral and kernel algorithms. In particular, we hope to publish in the near future some results concerning algorithms that allow to explore the spectra and compute the kernels of singular integral operators related with Hankel and commutator operators.
- We hope that our work within the Operator Theory, and with *Mathematica*, will help in the path to the future design and implementation of several other analytical algorithms, with numerous applications in many areas of research and technology.
- We also hope that, going forward, these analytical methods, and their implementation using a computer algebra system with large symbolic and numeric computation capabilities, may contribute to the development of the numerical approach in Operator Theory.

Acknowledgements

This research was partially supported by Center for Functional Analysis, Linear Structures and Applications (CEAFEL), Instituto Superior Técnico (Portugal).

APPENDIX A

Example 4.

Left canonical factorization of r

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Left non-canonical factorization of $\widetilde{r}$

$$\begin{pmatrix} (-it^3+2t^2+2)t^{-2} & t^{-2}+1 \\ (t^4+4it^3-4t^2-4)t^{-2} & (it^3-2t^2-2)t^{-2} \end{pmatrix}$$

$$= \begin{pmatrix} -i & t \\ t+2i & i(t+i)^2 \end{pmatrix} \begin{pmatrix} t & 0 \\ 0 & t^{-1} \end{pmatrix} \begin{pmatrix} 1+2it^{-3} & it^{-3} \\ 2 & 1 \end{pmatrix}$$

Example 5.

Left non-canonical factorization of r

$$\begin{pmatrix} 0 & 0 & t \\ 0 & 1 & 0 \\ t^{-1} & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} t & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & t^{-1} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Left non-canonical factorization of $\widetilde{r}$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Example 6.

Left canonical factorization of r

$$\begin{pmatrix} 0 & i(-2+t)(4+t^2)^{-1} & 1 & (3t-1)t^{-1} \\ (t+1/2)t^{-1} & 2i & t+1 & (4t-1)t^{-1} \\ 0 & i(2t-1) & -1 & 3t-1 \\ (t+1/2)t^{-1} & i(2t-1) & -1 & 3t \end{pmatrix}$$

$$= \begin{pmatrix} 3 & -3 & 1 & \frac{i(-2+t)}{4+t^2} \\ 4 & -3 & 1+t & 2i \\ 3t & -3t & -1 & i(-1+2t) \\ 1+3t & -3t & -1 & i(-1+2t) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1+(2t)^{-1} & 0 & 0 & 1 \\ 1+(2t)^{-1} & 0 & 0 & (3t)^{-1} \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Example 7.

Left non-canonical factorization of r

$$\begin{pmatrix} 0 & (i(-2+t))(-7i+t)^{-1} & 0 & 1 & 0 & 0 & 1 \\ 0 & 2i & 1+t & 0 & (4t-1)t^{-1} & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & t^{-1} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & t & t^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = r_+(t)\Lambda(t)r_-(t)$$

$$r_+(t) = \begin{pmatrix} i - \frac{28-98i}{k_1(t-7i)} & \frac{-2(7+2i)k_2t}{k_1(t-7i)} & 0 & 1 & 0 & 1 & 0 \\ k_3 & -1-k_2t & 1+t & 0 & 0 & 0 & -1+4t \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & t \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & t & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\Lambda(t) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t^{-1} \end{pmatrix}$$

$$r_-(t) = \begin{pmatrix} k_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & t^{-1} & 0 & 0 & 0 & 0 & 0 \\ k_2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ k_4 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$k_1 = -5 - 16i + \sqrt{25+20i}$,
$k_2 = -\frac{1}{2}\left(5 + 2i - \sqrt{25+20i}\right)$
$k_3 = \frac{1}{2}\left(3 + 2i + \sqrt{25+20i}\right)$
$k_4 = \left(2i\left[9 + 2i - \sqrt{25+20i} + \left((-33-24i) + \sqrt{565+1600i}\right)t\right]\right)(k_1k_5)^{-1}$
$k_5 = 5 + 2i + \sqrt{25+20i} + 2t$

REFERENCES

[1] Ablowitz, M.J., Clarkson, P.A. *Solitons, Nonlinear Evolution Equations and Inverse Scattering*, London Mathematical Society: Lecture Note Series **149**, Cambridge University Press, 1991.

[2] Aktosun, T., Klaus, M., van der Mee, C. "Explicit Wiener-Hopf factorization for certain non-rational matrix functions", *Integr. Equ. Oper. Theory*, Birkhäuser-Verlag, Vol. **15(6)**, pp. 879-900, 1992.

[3] Asch, M., Lebeau, G. "The spectrum of the damped wave operator for a bounded domain in R^2", *Experiment. Math.*, Euclid publishers, Vol. **12(2)**, pp. 227-241, 2003.

[4] Ball, J.A., Clancey, K.F. "An elementary description of partial indices of rational matrix functions", *Integr. Equ. Oper. Theory*, Birkhäuser-Verlag, Vol. **13(3)**, pp. 316-322, 1990.

[5] Bastos, M.A., Bravo, A., Karlovich, Yu.I., Spitkovsky, I. M. "On the factorization of some block triangular almost periodic matrix Functions", *Oper. Theory Adv. Appl.*, Birkhäuser Basel, Vol. **242**, pp. 25-52, 2014.

[6] Bastos, M.A., Karlovich, Yu., Spitkovsky, I. M., Tishin, P. M. "On a new algorithm for almost periodic factorization", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **103**, pp. 53-74, 1998.

[7] Böttcher, A., Karlovich, Yu. I. "Cauchy's singular integral operator and Its beautiful spectrum", *Oper. Theory Adv. Appl.*, Birkhäuser Basel, Vol. **129**, pp. 109-142, 2001.

[8] Câmara, M.C., dos Santos, A.F. "Generalized factorization for a class of $n \times n$ matrix unctions - Partial indices and explicit formulas", *Integr. Equ. Oper. Theory*, Birkhäuser Verlag, Vol. **20(2)**, pp. 198-230, 1994.

[9] Câmara, M.C., dos Santos, A.F., Carpentier, M. "Explicit Wiewer-Hopf factorization and non-linear Riemann-Hilbert problems", *Proc. Roy. Soc. Edinburgh Sect.A*, Vol. **132(1)** (A), pp. 45-74, 2002.

[10] Clancey, K., Gohberg, I. *Factorization of Matrix Functions and Singular Integral Operators*, Operator Theory: Advances and Applications, Birkhäuser Verlag, Vol. **3**, 1981.

[11] Conceição, A.C., *Factorization of Some Classes of Matrix Functions and its Applications (in portuguese)*, PhD Thesis, Universidade do Algarve (Portugal), 2007.

[12] Conceição, A.C., Kravchenko, V.G. "Factorization algorithm for some special matrix functions", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **181**, pp. 173-185, 2008.

[13] Conceição, A.C., Kravchenko, V.G. "About explicit factorization of some classes of non-rational matrix functions", *Math. Nachr.*, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, Vol. **280(9-10)**, pp. 1022-1034, 2007.

[14] Conceição, A.C., Kravchenko, V.G., Pereira, J.C. "Computing some classes of Cauchy type singular integrals with *Mathematica* software", *Adv. Comput. Math.*, Springer US, Vol. **39**, pp. 273-288, 2013.

[15] Conceição, A.C., Kravchenko, V.G., Pereira, J.C. "Rational functions factorization algorithm: a symbolic computation for the scalar and matrix cases", *Proceedings of the 1st National Conference on Symbolic Computation in Education and Research (CSEI2012)*, Lisboa, Portugal, P02, 2012.

[16] Conceição, A.C., Kravchenko, V.G., Pereira, J.C. "Factorization algorithm for some special non-rational matrix functions", *Oper. Theory Adv. Appl.*, Birkhäuser Basel, Vol. **202**, pp. 87-109, 2010.

[17] Conceição, A.C., Kravchenko, V.G., Teixeira, F.S. "Factorization of some classes of matrix functions and the resolvent of a Hankel operator", *FSORP2003 Factorization, Singular Operators and Related Problems*, Kluwer Academic Publishers, Ed. S. Samko, A. Lebre, A. F. dos Santos, Funchal, Portugal, pp. 101-110, 2003.

[18] Conceição, A.C., Kravchenko, V.G., Teixeira, F.S. "Factorization of matrix funtions and the resolvents of certain operators", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **142**, pp. 91-100, 2003.

[19] Conceição, A.C., Marreiros, R.C. "On the kernel of a singular integral operator with non-Carleman shift and conjugation", *Oper. Matrices, 24 pp., to appear.*

[20] Conceição, A.C., Pereira, J.C. "Exploring the spectra of some classes of paired singular integral operators: the scalar and matrix cases", *Libertas Mathematica (new series), 35 pp., to appear.*

[21] Ehrhardt, T., Speck, F.-O. "Transformation techniques towards the factorization of non-rational 2×2 matrix functions", *Linear Algebra Appl.*, Elsevier Science Inc, Vol. **353(1-3)**, pp. 53-90, 2002.

[22] Faddeev, L.D., Tkhatayan, L.A. *Hamiltonian Methods in the Theory of Solitons*, Springer-Verlag, 1987.

[23] Feldman, I., Gohberg, I., Krupnik, N. "An explicit factorization algorithm", *Integr. Equ. Oper. Theory*, Birkhäuser Verlag, Vol. **49(2)**, pp. 149-164, 2004.

[24] Feldman, I., Marcus, A. "On some properties of factorization indices", *Integr. Equ. Oper. Theory*, Birkhäuser Verlag, Vol. **30(3)**, pp. 326-337, 1998.

[25] Gohberg, I., Kaashoek, M.A., Spitkovsky, I.M. "An overview of matrix factorization theory and operator applications", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **141**, pp. 1-102, 2003.

[26] Gohberg, I., Krupnik, N. "The spectrum of singular integral operators in L_p Spaces", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **206**, pp. 111-125, 2010.

[27] Gohberg, I., Krupnik, N. "One-Dimensional Linear Singular Integral Equations", *Operator Theory: Advances and Applications*, Birkhäuser Verlag, Vol. **53**, 1992.

[28] Gohberg, I., Lerer, L., Rodman, L. "Factorization indices for matrix polynomials", *Bulletin of the American Mathematical Society*, Vol. **84(2)**, pp. 275-277, 1978.

[29] Isgur, A., Spitkovsky, I.M. "On the spectra of some Toeplitz and Wiener-Hopf operators with almost periodic matrix symbols", *Oper. Matrices*, Vol. **2(3)**, pp. 371-383, 2008.

[30] Janashia, G., Lagvilava, E. "On factorization and partial indices of unitary matrix-functions of one class", *Georgian Math. J.*, Kluwer Academic Publishers-Plenum Publishers, Vol. **4(5)**, pp. 439-442, 1997.

[31] Kravchenko, V.G., Lebre, A.B., Litvinchuk, G.S. "Spectrum problems for singular Integral operators with Carleman Shift", *Math. Nachr.*, **226(1)**, pp. 129-151, 2001.

[32] Kravchenko, V.G. , Lebre, A.B., Rodríguez, J. S. "Factorization of singular integral operators with a Carleman shift and spectral problems", *J. Integral Equations Appl.*, Vol. **13(4)**, pp. 339-383, 2001.

[33] Kravchenko, V. G., Litvinchuk, G. S. *Introdution to the Theory of Singular Integral Operators with Shift*, Mathematics and its Applications", Kluwer Academic Publishers, **289**, 1994.

[34] Kravchenko, V.G., Migdal'skii, A.I. "A regularization algorithm for some boundary-value problems of linear conjugation", *Dokl. Math.*, Vol. **52**, pp. 319-321, 1995.

[35] Kravchenko, V.G., Nikolaichuk, A.M. "On partial indices of the Riemann problem for two pair of functions", *Soviet Math. Dokl.*, Vol. **15** , pp. 438-442, 1974.

[36] Litvinchuk, G.S. *Solvability Theory of Boundary Value Problems and Singular Integral Equations with Shift*, Mathematics and its Applications, Kluwer Academic Publishers, Vol. **523**, 2000.

[37] Litvinchuk, G.S., Spitkovskii, I.M. *Factorization of Measurable Matrix Functions*, Operator Theory: Advances and Applications, Birkhäuser Verlag, Vol. **25**, 1987.

[38] Mikhlin, S.G., Prssdorf, S. *Singular Integral Operators*, Springer- Verlag, 1986.

[39] Pereira, J.C., Conceição, A.C. "Exploring the spectra of singular integral operators with rational coefficients", *Proceedings of the 1st International Conference on Algebraic and Symbolic Computation (SYMCOMP2015 - ECCOMAS Thematic Conference)*, Ed. A. Loja, J. I. Barbosa, and J. A. Rodrigues, Lisboa, Portugal, pp. 175-194, 2013.

[40] Plemelj, J. "Riemannshe Funktionenscharen mit gegebener Monodromiegruppe", *Monat. Math. Phys.*, Vol. **19**, pp. 211-245, 1908.

[41] Voronin, A.F. "A method for determining the partial indices of symmetric matrix functions", *Sib. Math. J.*, Vol. **52**, pp. 41-53, 2011.

[42] Voronin, A.F. "Partial indices of unitary and hermitian matrix functions", *Sib. Math. J.*, Vol. **51**, pp. 805-809, 2010.

SYMCOMP 2015
Faro, March 26-27, 2015

SYMBOLIC COMPUTATION APPLIED TO THE SINGULAR INTEGRAL OPERATORS WITH NON-CARLEMAN SHIFT AND CONJUGATION

Ana C. Conceição[1], Rui C. Marreiros[1], and José C. Pereira[2]

1: Center for Functional Analysis, Linear Structures and Applications (CEAFEL)
Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade do Algarve
Campus de Gambelas 8005-139, Faro, Portugal
e-mail: aicdoisg@gmail.com, rmarrei@ualg.pt

2: Center for Environmental and Sustainability Research (CENSE)
Center for Functional Analysis, Linear Structures and Applications (CEAFEL)
Departamento de Engenharia Electrónica e Informática
Faculdade de Ciências e Tecnologia
Universidade do Algarve
Campus de Gambelas 8005-139, Faro, Portugal
e-mail: unidadeimaginaria@gmail.com

Keywords: Singular integral operators with non-Carleman shift, kernel dimension, factorization algorithms, *Mathematica* software system

Abstract. *In recent years, several software applications were made available to the general public with extensive capabilities of symbolic computation. These applications, known as computer algebra systems (CAS), allow to delegate to a computer all, or a significant part of, the symbolic calculations present in many mathematical algorithms. The main goal of this paper is to show how the symbolic computation capabilities of the CAS* Mathematica *can be used to explore the dimension of the kernel of some classes of singular integral operators with non-Carleman shift and conjugation, defined on the unit circle. The new analytical algorithm [ADimKer-NonCarleman] is presented. Several nontrivial examples computed with the symbolic computation techniques [AFact], [ARFact-Scalar], [ARFact-Matrix], and [SInt] are given.*

1 INTRODUCTION

The history of the study of singular integral operators (SIOs) with shift is rich. Equally interesting are the related histories of the study of singular integral equations with shift and of boundary value problems with shift. All these problems were studied during the second half of the last century, and continue to be studied at present time. Ilya Vekua's book [22] (first edition in 1959) played a key role in this process; in this and in other similar books (see e.g. [23]), it has been shown how some mathematical physics problems lead to the solvability of boundary value problems with shift. The Fredholm theory of SIOs with Carleman shift was constructed in the sixties and the seventies of the XX century (see [18]). For the case of non-Carleman shift, the theory was completed in the eighties (see [14]). However, more interesting questions about the solvability of boundary value problems with shifts, have been considered only with very restrictive conditions on the corresponding coefficients (see [18]). Recent progress in the study of the spectral properties of SIOs with linear fractional Carleman shift and conjugation (see [9], [12], [13], and [21]) makes it possible to study the solvability of the related boundary value problems (see [17]). For non-Carleman shift, the question about the solvability of this type of problems remains open (see [1], [15], and [20]).

In [16] we studied a generalized Riemann boundary value problem with a non-Carleman shift and conjugation on the real line.

In [7] we considered a SIO with non-Carleman shift and conjugation on the unit circle. Some estimates for the dimension of its kernel were obtained.

In recent years, several software applications with extensive capabilities of symbolic computation were made available to the general public. These computer algebra systems (CAS) allow to delegate to a computer all, or a significant part of, the symbolic and numeric calculations present in many mathematical algorithms. In our work we use the CAS *Mathematica*[1] to implement, on a computer, some of our analytical algorithms within operator theory. In [4] it was presented a calculation technique, [SInt], that allows to compute some classes of Cauchy type singular integrals on the unit circle. The factorization algorithm [AFact] for special classes of factorable essentially bounded hermitian matrix functions, was presented in [6]. Paper [5] contains the description of the analytical algorithms [ARFact-Scalar] and [ARFact-Matrix] that compute explicit factorizations for factorable rational functions defined on the unit circle.

In the present paper it is shown how symbolic computation can be used to explore the dimension of the kernel of some classes of SIOs with non-Carleman shift and conjugation, with rational coefficients, defined on the unit circle. Section 2 is dedicated to the theoretical results (see [7]) on the dimension of the kernel of some classes of SIOs with non-Carleman shift and conjugation. The most general case is considered in Subsection

[1]All the research presented in this paper was done with *Mathematica* 9. At present time, we are using *Mathematica* 10, with no backward compatibility issues to report. For further information on the computer algebra system *Mathematica* visit the Wolfram's website at `www.wolfram.com`.

2.1. Special cases are described in Subsection 2.2. In Section 3 we describe how the symbolic computation techniques [ARFact-Scalar], [ARFact-Matrix], and [SInt] designed and implemented with *Mathematica* can be used to compute the constants that appear in the estimates obtained in [7]. These three analytical algorithms allow us to design a new algorithm called [ADimKer-NonCarleman], that estimates the dimension of the kernel of some classes of SIOs with non-Carleman shift and conjugation, defined on the unit circle. Several nontrivial examples computed with the symbolic computation technique [ADimKer-NonCarleman] are presented. This paper also contains some final remarks about our current work and related lines of research that we find potentially interesting.

2 SINGULAR INTEGRAL OPERATORS WITH NON-CARLEMAN SHIFT AND CONJUGATION

Let $\mathbb{T}$ denote the unit circle in the complex plane. Let $\mathbb{T}_+$ and $\mathbb{T}_-$ denote the open unit disk and the exterior region of the unit circle (∞ included), respectively.
It is well known that the SIO with Cauchy kernel, $S_{\mathbb{T}}$, is defined almost everywhere on $\mathbb{T}$ by

$$S_{\mathbb{T}}\varphi(t) = \frac{1}{\pi i} \int_{\mathbb{T}} \frac{\varphi(\tau)}{\tau - t}\, d\tau,\ t \in \mathbb{T}, \tag{1}$$

where the integral is understood in the sense of its principal value and represents a bounded linear operator in $L_2(\mathbb{T})$. In addition, $S_{\mathbb{T}}$ is a selfadjoint and unitary operator in the Lebesgue space $L_2(\mathbb{T})$ (see, for instance, [11] and [14]). Thus, we can associate with this operator two complementary projection operators

$$P_\pm = (I \pm S_{\mathbb{T}})/2, \tag{2}$$

where I represents the identity operator.
The projectors $P_\pm$ allow us to decompose the space $L_2(\mathbb{T})$ in the topological direct sum

$$L_2(\mathbb{T}) = L_2^+(\mathbb{T}) \oplus L_2^{-,0}(\mathbb{T}),$$

where $L_2^+(\mathbb{T}) = \mathrm{im}P_+$ and $L_2^{-,0}(\mathbb{T}) = \mathrm{im}P_-$. We also consider $L_2^-(\mathbb{T}) = L_2^{-,0}(\mathbb{T}) \oplus \mathbb{C}$.
As usual, $L_\infty(\mathbb{T})$ denotes the space of all essentially bounded functions on $\mathbb{T}$.
Let $\mathcal{C}(\mathbb{T})$ denote the algebra of all continuous functions on $\mathbb{T}$, $\mathcal{R}(\mathbb{T})$ denote the algebra of rational functions without poles on $\mathbb{T}$, and $\mathcal{R}^\pm(\mathbb{T})$ denote the subsets of $\mathcal{R}(\mathbb{T})$ whose elements are without poles in $\mathbb{T}_\pm$, respectively.
Let us now introduce the concept of matrix function generalized factorization (see, for instance, [2] and [19]): we say that a matrix function $c \in L_\infty^{n\times n}(\mathbb{T})$ admits a right (left) generalized factorization in $L_2(\mathbb{T})$ if it can be represented as

$$c = c_-\Lambda c_+ \quad (c_+\Lambda c_-), \tag{3}$$

where

$$c_-^{\pm 1} \in \left[L_2^-(\mathbb{T})\right]^{n\times n}, \quad c_+^{\pm 1} \in \left[L_2^+(\mathbb{T})\right]^{n\times n}, \quad \Lambda(t) = \mathrm{diag}\{t^{\kappa_j}\},$$

$\kappa_j \in \mathbb{Z}$, $j = \overline{1,n}$, with $\kappa_1 \geq \kappa_2 \geq ... \geq \kappa_n$, and $c_- P_+ c_+ I$ $(c_+ P_+ c_- I)$ represents a bounded linear operator in $L_2^n(\mathbb{T})$; the number $\kappa = \sum_{j=1}^{n} \kappa_j$ is called the factorization index of the determinant of the matrix function c. The integers κ_j are uniquely defined by the matrix function c and are called its right (left) partial indices. If $\kappa_j = 0$, $j = \overline{1,n}$, then c is said to admit a right (left) canonical generalized factorization in $L_2(\mathbb{T})$.
Any non-singular continuous matrix function $c \in \mathcal{C}^{n\times n}(\mathbb{T})$ admits a generalized factorization of the form (3) in $L_2(\mathbb{T})$ (see, for instance, the above cited [2] and [19]).
Any non-singular rational matrix function $c \in \mathcal{R}^{n\times n}(\mathbb{T})$ admits a factorization of the form (3) (see, for instance, [10]), where

$$c_-^{\pm 1} \in \left[\mathcal{R}^-(\mathbb{T})\right]^{n\times n}, \quad c_+^{\pm 1} \in \left[\mathcal{R}^+(\mathbb{T})\right]^{n\times n}.$$

For the particular scalar case we note that $\kappa = \operatorname{ind} c$ if $c \in \mathcal{C}(\mathbb{T})$; as usual, $\operatorname{ind}\varphi$ denotes the Cauchy index of a continuous function $\varphi \in \mathcal{C}(\mathbb{T})$, i.e.,

$$\operatorname{ind}\varphi = \frac{1}{2\pi}\{\arg\varphi(t)\}_{t\in\mathbb{T}}.$$

In the rational scalar case $c^{\pm} \in \mathcal{R}(\mathbb{T})$, then

$$\kappa = z_+ - p_+, \tag{4}$$

where z_+ is the number of zeros of c in $\mathbb{T}_+$ (with regard to their multiplicities) and p_+ is the number of poles of c in $\mathbb{T}_+$ (with regard to their multiplicities) (see, for instance, [5])
Let ω be a homeomorphism of $\mathbb{T}$ onto itself, which is differentiable on $\mathbb{T}$ and whose derivative does not vanish there. The function $\omega : \mathbb{T} \to \mathbb{T}$ is called a shift function or simply a shift on $\mathbb{T}$. By

$$\omega_k(t) \equiv \omega[\omega_{k-1}(t)], \quad \omega_1(t) \equiv \omega(t), \quad \omega_0(t) \equiv t, \quad t \in \mathbb{T},$$

we denote the k-th iteration of the shift, $k \geq 2$, $k \in \mathbb{N}$.
A shift ω is called a (generalized) Carleman shift of order $n \in \mathbb{N} \setminus \{1\}$ if $\omega_n(t) \equiv t$, but $\omega_k(t) \not\equiv t$ for $k = \overline{1, n-1}$. Otherwise, if ω is not a Carleman shift, it is called a non-Carleman shift.
In what follows we will consider a linear fractional non-Carleman shift preserving the orientation on $\mathbb{T}$

$$\alpha(t) = \frac{\mu t + \nu}{\overline{\nu} t + \overline{\mu}}, \quad t \in \mathbb{T}, \tag{5}$$

where $\mu, \nu \in \mathbb{C}$: $|\mu|^2 - |\nu|^2 = 1$. This shift has two fixed points, τ_1 and τ_2, given by the formula

$$\tau_{1,2} = \frac{\mu - \overline{\mu} \pm \sqrt{(\mu + \overline{\mu})^2 - 4}}{2\overline{\nu}}.$$

Obviously $\tau_1 \neq \tau_2$ if $|\operatorname{Re}\mu| \neq 1$.
The rational shift function α admits the factorization of the form (3)

$$\alpha(t) = \alpha_+(t)t\alpha_-(t),$$

where

$$\alpha_+(t) = \frac{1}{\overline{\nu}t + \overline{\mu}}, \quad \alpha_-(t) = \frac{\mu t + \nu}{t} = (\overline{\alpha_+(t)})^{-1}.$$

We see that the functions $\alpha_\pm, \alpha_\pm^{-1}$ are analytic in $\mathbb{T}_\pm$ and continuous in the closure of $\mathbb{T}_\pm$, respectively.
Let $\sigma(\xi)$ and $\|\xi\|_2$ denote the spectrum and the spectral norm of a matrix $\xi \in \mathbb{C}^{n\times n}$, respectively. Later we will make use of the following result from [15].

Lemma 1. *For any continuous matrix function $f \in \mathcal{C}^{n\times n}(\mathbb{T})$ such that*

$$\sigma[f(\tau_j)] \subset \mathbb{T}_+, \quad j = 1, 2,$$

there exists a polynomial matrix s satisfying the conditions

$$\max_{t\in\mathbb{T}} \left\| s(t)f(t)s^{-1}(\alpha(t)) \right\|_2 < 1$$

and

$$P_+ s^{\pm 1} P_+ = s^{\pm 1} P_+.$$

For a continuous matrix function f satisfying the conditions of Lemma 1, let R denote the set of all such polynomial matrices s,

$$l_1(s) = \sum_{i=1}^{n} \max_{j=\overline{1,n}} l_{i,j},$$

where $l_{i,j}$ is the degree of the element $s_{i,j}(t)$ of the polynomial matrix s and

$$l = \min_{s\in R}\{l_1(s)\}. \tag{6}$$

As usual, let $\widetilde{L}_2(\mathbb{T})$ denote the real space of all Lebesgue measurable square summable complex valued functions on $\mathbb{T}$. On $\widetilde{L}_2(\mathbb{T})$, associated with the shift α, we consider the shift operator U defined by

$$(U\varphi)(t) = \alpha_+(t)\varphi[\alpha(t)].$$

The shift operator U satisfies the properties:
i) U is isometric, i.e., $\|U\varphi\| = \|\varphi\|$;
ii) $US = SU$.

We also consider the following two operators on $\widetilde{L}_2(\mathbb{T})$: the bounded linear involutive operator of complex conjugation C,

$$(C\varphi)(t) = t^{-1}\overline{\varphi(t)},$$

and the functional operator

$$A = \sum_{j=0}^{m} a_j U^j,\ a_j \in \mathcal{C}(\mathbb{T}).$$

The operators $P_\pm$, U and C, verify the properties

$$CU = UC, \quad UP_\pm = P_\pm U, \quad CP_\pm = P_\mp C.$$

In this paper we will consider the SIO with non-Carleman shift and conjugation defined on the unit circle

$$K = P_+ + (aI + AC)P_-, \tag{7}$$

with coefficients $a, a_0, a_1, ..., a_m \in \mathcal{C}(\mathbb{T})$.

2.1 The case $\mathcal{C}(\mathbb{T})$

In this subsection we present some estimates for the dimension of the kernel of the SIO K defined in (7).
In what follows, the continuous function a is invertible on $\mathbb{T}$; then we can construct the matrix functions

$$b_0 = \begin{pmatrix} a^{-1} & a^{-1}a_0 \\ -a^{-1}\overline{a_0} & \overline{a} - a^{-1}\left|a_0\right|^2 \end{pmatrix}, \tag{8}$$

$$b_1 = \begin{pmatrix} 0 & a^{-1}a_1 \\ -a^{-1}(\alpha)\overline{a_1} & -a^{-1}\overline{a_0}a_1 - a^{-1}(\alpha)\overline{a_1}a_0(\alpha) \end{pmatrix},$$

$$b_2 = \begin{pmatrix} 0 & a^{-1}a_2 \\ -a^{-1}(\alpha_2)\overline{a_2} & -a^{-1}\overline{a_0}a_2 - a^{-1}(\alpha)\overline{a_1}a_1(\alpha) - a^{-1}(\alpha_2)\overline{a_2}a_0(\alpha_2) \end{pmatrix},$$

$$\dots,$$

$$b_m = \begin{pmatrix} 0 & a^{-1}a_m \\ -a^{-1}(\alpha_m)\overline{a_m} & -a^{-1}\overline{a_0}a_m - \dots - a^{-1}(\alpha_m)\overline{a_m}a_0(\alpha_m) \end{pmatrix},$$

$$b_{m+1} = \begin{pmatrix} 0 & 0 \\ 0 & -a^{-1}(\alpha)\overline{a_1}a_m(\alpha) - \dots - a^{-1}(\alpha_m)\overline{a_m}a_1(\alpha_m) \end{pmatrix},$$

$$\dots,$$

$$b_{2m} = \begin{pmatrix} 0 & 0 \\ 0 & -a^{-1}(\alpha_m)\overline{a_m}a_m(\alpha_m) \end{pmatrix}.$$

Let us consider the matrix function b_0 defined in (8). Note that $\det b_0(t) \neq 0$ for all $t \in \mathbb{T}$. Therefore, the non-singular continuous matrix function b_0 admits a right generalized factorization of the form (3) in $L_2(\mathbb{T})$

$$b_0 = b_- \Lambda b_+. \tag{9}$$

Let us consider

$$\Lambda_\pm = \operatorname{diag}\{t^{\kappa_1^\pm}, t^{\kappa_2^\pm}\},\ \kappa_j^\pm = \frac{1}{2}(\kappa_j \pm |\kappa_j|), \quad j = 1, 2. \tag{10}$$

It is assumed that

$$b_\pm^{\pm 1} \in C^{2\times 2}(\mathbb{T}).$$

For the continuous function a_0 let us denote its projections by

$$(a_0)_\pm := P_\pm(a_0).$$

Considering the decomposition $a_0 = (a_0)_+ + (a_0)_-$ we start by analysing the particular case (see [7]) when the function $(a_0)_-$ is the null function. This is an interesting case, since the matrix function b_0 admits the right generalized factorization of the form (9), where

$$b_- = \begin{pmatrix} a_-^{-1} & 0 \\ -a_-^{-1}\overline{(a_0)_+} & \overline{a_+} \end{pmatrix}, \quad b_+ = \begin{pmatrix} a_+^{-1} & a_+^{-1}(a_0)_+ \\ 0 & \overline{a_-} \end{pmatrix},$$

and

$$\Lambda(t) = \operatorname{diag}\{t^{-\kappa}, t^{-\kappa}\}.$$

Obviously, we get the following result (see [7]).

Proposition 1. *Let a be an invertible continuous function on $\mathbb{T}$, and let*

$$a(t) = a_-(t)t^\kappa a_+(t), \quad \kappa = \operatorname{ind} a,$$

be a generalized factorization of the form (3) *of a in $L_2(\mathbb{T})$. Then the right partial indices of the matrix function b_0 are $\kappa_1 = -\kappa$ and $\kappa_2 = -\kappa$.*

For the general case, we must consider the function

$$u := (a_0)_-(\overline{a_-}a_+)^{-1},$$

its projections

$$u_\pm := P_\pm u,$$

and the Hankel operator (acting in the Hardy class $H_2(\mathbb{T})$) with symbol $\varphi \in L_\infty(\mathbb{T})$

$$H_\varphi = P_- \overline{\varphi} P_+.$$

On the right partial indices of the matrix function b_0 we get the following result (see [7]).

Proposition 2. *Let a be an invertible continuous function on $\mathbb{T}$, and let*

$$a(t) = a_-(t)t^{\kappa}a_+(t), \quad \kappa = \operatorname{ind} a,$$

be a generalized factorization of the form (3) *of a in $L_2(\mathbb{T})$. Then the right partial indices of the matrix function b_0 are*

$$\kappa_1 = -\kappa + k, \quad \kappa_2 = -\kappa - k,$$

where

$$k = \dim \ker(H^*_{\overline{u_-}} H_{\overline{u_-}} - I). \tag{11}$$

Let e_n denote the $(n \times n)$ identity matrix and, for simplicity, $e \equiv e_2$.
To obtain an estimate for the dimension of the kernel of the SIO with non-Carleman shift and conjugation K defined in (7) we need to consider the $(4m \times 4m)$ matrix function

$$f = \operatorname{diag}\{\Lambda_-^{-1} b_-^{-1}, e_{4m-2}\} c \operatorname{diag}\{b_+^{-1}(\alpha)\Lambda_+^{-1}(\alpha), e_{4m-2}\}, \tag{12}$$

where $b_\pm$ are factors of a right generalized factorization of b_0 in $L_2(\mathbb{T})$, $\Lambda_\pm$ are given in (10), α is the linear fractional non-Carleman shift defined in (5), and c is the $(4m \times 4m)$ matrix function

$$c = \begin{pmatrix} b_1 & b_2 & \cdots & b_{2m-1} & b_{2m} \\ -e & 0 & \cdots & 0 & 0 \\ 0 & -e & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & -e & 0 \end{pmatrix}.$$

In [7] we established our main result on the estimate of the dimension of the kernel of K.

Theorem 1. *Let K be the SIO with non-Carleman shift and conjugation defined in* (7). *Let a be an invertible function on $\mathbb{T}$, $\kappa = \operatorname{ind} a$, and k be the number defined in* (11); *let f be the matrix function defined in* (12), *and l_K be the number defined in* (6) *for the matrix function f. Then the following estimate holds*

$$\dim \ker K \leq l_K + \max(\kappa - k, 0) + \max(\kappa + k, 0) + 1. \tag{13}$$

The estimate (13) depends on the integer constants l_K, κ, and k. In Section 3 we will see that some symbolic computation techniques (see, for instance, [4], [5], and [6]) can be used to determine the constants κ and k. In addition, it is important to note that the three constants can be equal zero, one or more at a time.

Since the right partial indices of the matrix function b_0 defined in (8) are $\kappa_1 = -\kappa + k$ and $\kappa_2 = -\kappa - k$ (see Proposition 2) we can get the estimate (13) by computing κ_1 and κ_2 instead. Therefore, Theorem 1 can be reformulated as

Theorem 2. *Let K be the SIO with non-Carleman shift and conjugation defined in* (7). *Let a be an invertible function on $\mathbb{T}$. Moreover, let κ_1 and κ_2 be the right partial indices of the matrix function b_0 defined in* (8). *Then the following estimate holds*

$$\dim\ker K \leq l_K + \max(-\kappa_1, 0) + \max(-\kappa_2, 0) + 1.$$

Remark 1. This last result is very important since it is possible, for some classes of the matrix functions (see, for instance, [5] and [6]) to compute directly the right partial indices, κ_1 and κ_2, of the matrix function b_0.

2.2 The case $H_\infty(\mathbb{T}) \cap \mathcal{C}(\mathbb{T})$

In this subsection we present some estimates for the dimension of the kernel of the SIO K defined in (7) for the case when $a^{\pm 1} \in \mathcal{C}(\mathbb{T})$ and some of the coefficients $a_0, a_1, ..., a_m$ are bounded analytic functions in $\mathbb{T}_+$, that is, are functions that belong to the class $H_\infty(\mathbb{T})$. It is easy to see that for the case when the function a_0 is a continuous, bounded, analytic function in $\mathbb{T}_+$, that is, $a_0 \in H_\infty(\mathbb{T}) \cap \mathcal{C}(\mathbb{T})$, the selfadjoint SIO $H^*_{\overline{u}_-} H_{\overline{u}_-} - I$ has a trivial kernel. Thus, according to Theorem 1, we get the following result

Theorem 3. *Let K be the SIO with non-Carleman shift and conjugation defined in* (7). *Let a be an invertible function on $\mathbb{T}$, $\kappa = \operatorname{ind} a$. If $P_-(a_0) \equiv 0$, then*

$$\dim\ker K \leq l_K + 2\max(\kappa, 0) + 1.$$

Let us now consider the case when $a_j \in H_\infty(\mathbb{T}) \cap \mathcal{C}(\mathbb{T})$, $j = \overline{1, m}$.

Theorem 4. *Let K be the SIO with non-Carleman shift and conjugation defined in* (7). *Let a be an invertible function on $\mathbb{T}$, $\kappa = \operatorname{ind} a$, and k be the number defined in* (11). *If $P_-(a_j) \equiv 0$, $j = \overline{1, m}$, then*

$$\dim\ker K \leq \max(\kappa - k, 0) + \max(\kappa + k, 0) + 1 \tag{14}$$

Based on Proposition 2, we can get estimate (14) only by computing κ_1 and κ_2, which is to say that Theorem 4 can be reformulated as

Theorem 5. *Let K be the SIO with non-Carleman shift and conjugation defined in* (7). *Let a be an invertible function on $\mathbb{T}$. Moreover, let κ_1 and κ_2 be the right partial indices of the matrix function b_0 defined in* (8). *Then the following estimate holds*

$$\dim\ker K \leq \max(-\kappa_1, 0) + \max(-\kappa_2, 0) + 1.$$

Remark 2. This last result is very important since it is possible, for some classes of the matrix functions (see, for instance, [5] and [6]) to compute directly the right partial indices, κ_1 and κ_2, of the matrix function b_0 (see Examples 2 and 3).

Corollary 1. *Let K be the SIO with non-Carleman shift and conjugation defined in* (7). *Let a be an invertible function on $\mathbb{T}$, $\kappa = \operatorname{ind} a$. If $P_-(a_j) \equiv 0,\ j = \overline{0,m}$, then*

$$\dim \ker K \leq 2 \max(\kappa, 0) + 1. \tag{15}$$

3 THE [ADimKer-NonCarleman] ALGORITHM

This Section is dedicated to the formal description of the [ADimKer-NonCarleman] algorithm.
It is explained how the symbolic computation techniques [ARFact-Scalar], [ARFact-Matrix], and [SInt], designed to be implemented with *Mathematica*, can be used to compute the constants that appear in the estimates (14) e (15), for the rational case.

For a given SIO with non-Carleman shift and conjugation

$$K = P_+ + (aI + AC)P_-, \tag{16}$$

with coefficients $a, a_0, a_1, ..., a_m \in \mathcal{R}(\mathbb{T})$, the algorithm [ADimKer-NonCarleman] gives as output an estimate for the dimension of its kernel.

In the design of this new analytical algorithm we used the calculation technique [SInt] (see [4]) to compute Cauchy type singular integrals of the form (1), with given integrand factors $a_j,\ j = \overline{0,m}$, represented as $a_j(t) = r(t)\,[x_+(t) + y_-(t)]$, where $x_+, \overline{y_-},\ r \in \mathcal{R}(\mathbb{T})$. Up to our knowledge, this calculation technique is one of the only two analytical algorithms (see also [SIntAFact] algorithm in [5]) written and implemented to compute SIOs with general essentially bounded functions. This technique uses extensively the properties of the projection operators $P_\pm$ defined in (2), and explores the rationality of $r(t)$ to reduce all possible analytic situations to a few basic cases. The [SInt] algorithm was implemented on a computer with the *Mathematica* software system, thus automating the extensive symbolic and numeric calculations needed for computing the SIOs. The analysis of its source code[2] reveals that the crucial steps are the decomposition of $r(t)$ and the computation of projections. The calculations involved in these two steps can become quite lengthy as the expressions of the functions $r(t)$, $x_+(t)$, and $y_-(t)$ grow larger and more complex. However, based on our experiments with the algorithm, it is reasonable to expect total execution times in the order of a few seconds for most inputs.

[2]The source code of the [SInt] algorithm is available for download with the online edition of [5].

The [SInt] algorithm can be applied to particular functions $x_+(t)$ and $y_-(t)$ or it can compute the closed form of (1) as a general expression in $x_+(t)$ and $y_-(t)$.

The [ADimKer-NonCarleman] algorithm also uses the factorization algorithm [ARFact-Scalar] (see [5]), that computes explicit factorizations for any given factorable rational function defined on the unit circle, to determine the factorization index of the scalar function a. The symbolic computation capabilities of *Mathematica*, and the *pretty-print* functionality[3], allow this part of the code to be very simple and syntactically similar to its analytical counterpart. We note that, since the zeros and poles of the rational function $a(t)$ are a crucial information for this calculation technique, the success of the [ARFact-Scalar] algorithm and, consequently, of the [ADimKer-NonCarleman] algorithm, depends on the possibility of finding those zeros and poles by solving polynomial equations. This can be a serious limitation when working with polynomials of the fifth degree or higher. However, even in this case, thanks to the symbolic and numeric capabilities of *Mathematica*, it is still possible to obtain an explicit, and for all purposes exact, rational factorization (see [5] and [8]). In fact, *Mathematica* uses `Root` objects to represent solutions of algebraic equations in one variable, when it is impossible to find explicit formulas for these solutions. The `Root` object is not a mere denoting symbol but rather an expression that can be symbolically manipulated and numerically evaluated. In particular, it is still possible to know if any given `Root` lies in the unit circle, in the interior or in the exterior of the unit circle (see, for instance, Figure 3 in [8]), which is all the information the [ARFact-Scalar] algorithm needs to construct the factors $a_\pm(t)$ of a given factorable rational function a, and to compute its factorization index κ. Thus, in the rational case, the constant κ that appears in the estimates (14) e (15), can always be computed by using formula (4) (see, for instance, Figure 5 in [8]).
In the design of the [ADimKer-NonCarleman] algorithm we also used the analytical factorization algorithm [ARFact-Matrix] (see [5]) to compute the right partial indices of the matrix function b_0 defined in (8). One crucial step of the [ARFact-Matrix] algorithm, implemented using the CAS *Mathematica*, is finding the zeros of the determinant of the rational matrix function b_0. Therefore, as in the scalar case, the success of the algorithms depends on the possibility of finding solutions of polynomial equations. However, due to the complexity of the matrix case, it is not as feasible as before to use the `Root` objects to obtain an explicit matrix function factorization when working with high degree polynomials. Nonetheless, even when working with polynomials of relative low degree, we note that, although the final factorization may have relatively simple entries, if we were to use the traditional pencil and paper tools the intermediate calculations would take typically many working hours, up to the point of infeasibility and therefore, there is still something to gain with the implementation of the [ARFact-Matrix] algorithm.
The [ADimKer-NonCarleman] algorithm can be applied to given SIOs with non-Carleman

[3]The *pretty-print* functionality allows to write on the computer screen scientific formulas in the traditional format, as if one was using pencil and paper.

shift and conjugation K defined in (16).
The flowchart of the [ADimKer-NonCarleman] algorithm is shown in Figure 1.

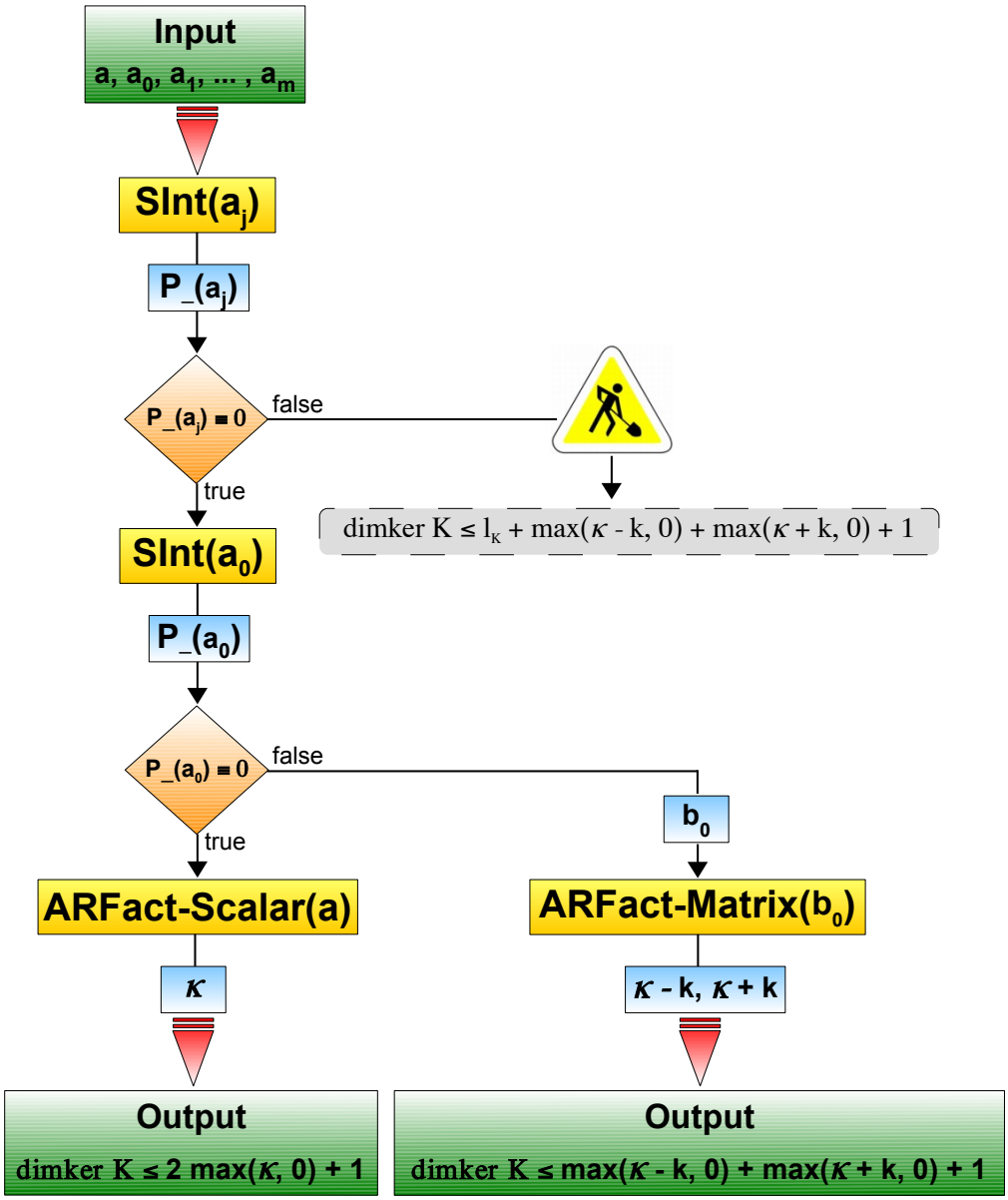


Figure 1: Flowchart of [ADimKer-NonCarleman] algorithm.

Remark 3. The algorithm (in its current state) can also be generalized and applied to the non-rational functions $a_j(t),\ j=\overline{0,m}$, which can be represented as the product of a rational function by a function whose conjugate is a bounded analytic function in the interior of the unit circle (see Remark 4 in Example 1).

There are three options to input functions $a(t)$ and $a_j(t),\ j=\overline{0,m}$:

1. Insert $a(t)$ and $a_j(t),\ j=\overline{0,m}$ directly.
2. Insert $a(t)$ directly; insert zeros, poles (and multiplicities) of $a_j(t),\ j=\overline{0,m}$.
3. Insert $a(t)$ directly; insert numerators, poles (and multiplicities) of $a_j(t),\ j=\overline{0,m}$.

For each chosen set of input functions $a, a_j(t),\ j=\overline{0,m}\in\mathcal{R}(\mathbb{T})$ and complex number λ, the [ADimKer-NonCarleman] algorithm gives one of the following *outputs*:

$$[\textit{Output 1}]\quad \dim\ker K\le 2\max(\kappa,0)+1 \tag{17}$$

$$[\textit{Output 2}]\quad \dim\ker K\le \max(\kappa-k,0)+\max(\kappa+k,0)+1 \tag{18}$$

3.1 [ADimKer-NonCarleman] examples

This Subsection contains several nontrivial rational examples computed with the [ADimKer-NonCarleman] algorithm.
All the examples were computed on a MacBook Pro with a 2.5 GHz Intel Core i5 processor and 4 GB of DDR3 RAM, running Mac OS X 10.9.5 (Mavericks) in single user mode.
The factors $b_\pm$ computed with the factorization algorithm [ARFact-Matrix] for examples 2 and 3 are presented in Appendix A. Note that, although these factors are not used directly by the [ADimKer-NonCarleman] algorithm to estimate the dimension of the kernel of the SIO K defined in (16), they are necessary for the computation of the right partial indices $\kappa-k$ and $\kappa+k$.

Example 1. Let us consider the SIO K defined in (16) with rational coefficients

$$a(t)=\frac{t^9+5t^2-1-i}{t^{11}+t^5+2t^3-(10+i)t^2},\ a_0(t)=\frac{(7+\sqrt{5}i)t^{30}-1}{t^2+it+6},\ a_1(t)=\frac{2t+3i+\sqrt{2}}{t^{15}+2it^3+t^2-10}.$$

The [ADimKer-NonCarleman] algorithm computes the singular integral $P_-(a_1)$ and concludes that $P_-(a_1)\equiv 0$. As a consequence, the singular integral $P_-(a_0)$ is computed. Since $P_-(a_0)\equiv 0$, the algorithm computes the Cauchy index of function a, $\kappa=0$. Thus, the [ADimKer-NonCarleman] algorithm gives the *output*

$$\dim\ker K\le 1.$$

Remark 4. Based on Remark 3 the previous example can be generalized for

$$a_0(t) = \frac{(7+\sqrt{5}i)t^{30}-1}{t^2+it+6}X_{1,+}(t) \text{ and } a_1(t) = \frac{2t+3i+\sqrt{2}}{t^{15}+2it^3+t^2-10}X_{2,+}(t),$$

where $X_{1,+}(t)$ and $X_{1,+}(t)$ are bounded and analytical continuous functions in the interior of the unit circle. The *output* is the same as in Example 1.

Example 2. Let us consider the SIO K defined in (16) with rational coefficients

$$a(t) = t, \qquad a_0(t) = \frac{2t^2}{2t-i}, \qquad a_1(t) = \frac{6t^{14}-it+3i}{t^7+2it^2-13}, \qquad a_2(t) = \frac{\sqrt{2}it^3+2t-3}{6t^{14}-it+3i}.$$

The [ADimKer-NonCarleman] algorithm computes the singular integrals $P_-(a_j)$, $j = \overline{1,2}$ and concludes that $P_-(a_j) \equiv 0$, $j = \overline{1,2}$. As a consequence, the singular integral $P_-(a_0)$ is computed. Since $P_-(a_0) \not\equiv 0$, the algorithm constructs the auxiliary matrix function

$$b_0(t) = \begin{pmatrix} t^{-1} & \dfrac{2t}{2t-i} \\ \dfrac{2i}{t^2(t-2i)} & \dfrac{-2t^2+it+2}{-2t^2+5it+2} \end{pmatrix},$$

and computes its right partial indices, $\kappa_1 = -1$ and $\kappa_2 = -1$. Thus, the [ADimKer-NonCarleman] algorithm gives the *output*

$$\dim \ker K \leq 3.$$

Example 3. Let us consider the SIO K defined in (16) with rational coefficients

$$a(t) \equiv 1, \qquad a_0(t) = \frac{t}{1-2t}, \qquad a_j(t) \in H_\infty(\mathbb{T}) \cap \mathcal{R}(\mathbb{T}),\ j = \overline{1,m}.$$

The [ADimKer-NonCarleman] algorithm computes the singular integrals $P_-(a_j)$, $j = \overline{1,m}$ and concludes that $P_-(a_j) \equiv 0$, $j = \overline{1,m}$. As a consequence, the singular integral $P_-(a_0)$ is computed. Since $P_-(a_0) \not\equiv 0$, the algorithm constructs the auxiliary matrix function

$$b_0(t) = \begin{pmatrix} 1 & \dfrac{t}{1-2t} \\ \dfrac{1}{2-t} & \dfrac{2(t-1)^2}{(2t-1)(t-2)} \end{pmatrix},$$

and computes its right partial indices, $\kappa_1 = 0$ and $\kappa_2 = 0$. Thus, the [ADimKer-NonCarleman] algorithm gives the *output*

$$\dim \ker K \leq 1.$$

Remark 5. In the previous example $u_- = P_-(a_0) = \dfrac{1}{2(1-2t)}$. Since $\kappa_1 = -\kappa + k$, $\kappa_1 = -\kappa - k$, and $\kappa = 0$, it follows that $k = \dim\ker(H^*_{\overline{u_-}} H_{\overline{u_-}} - I) = 0$.

Example 4. Let us consider the SIO K defined in (16) with rational coefficients

$$a(t) \equiv 1, \qquad a_0(t) = \frac{t}{1-2t}\overline{\theta(t)}, \qquad a_j(t) \in H_\infty(\mathbb{T}) \cap \mathcal{R}(\mathbb{T}),\ j = \overline{1,m},$$

where θ is an arbitrary rational inner function.
The [ADimKer-NonCarleman] algorithm computes the singular integrals $P_-(a_j)$, $j = \overline{1,m}$ and concludes that $P_-(a_j) \equiv 0$, $j = \overline{1,m}$. As a consequence, the singular integral $P_-(a_0)$ is computed. Since $P_-(a_0) \not\equiv 0$, $\forall \theta(t)$, the algorithm constructs the auxiliary matrix function

$$b_0(t) = \begin{pmatrix} 1 & \dfrac{t}{1-2t}\overline{\theta(t)} \\ \dfrac{1}{2-t}\theta(t) & \dfrac{2(t-1)^2}{(2t-1)(t-2)} \end{pmatrix}.$$

Since the right partial indices of b_0 are $\kappa_1 = 0$ and $\kappa_2 = 0$, for any inner function $\theta(t)$, differentiable in a neighboord[4] of $t = 1$ in $\mathbb{T}$, that might be considered by the user, the [ADimKer-NonCarleman] algorithm (when it works[5]) gives the *output*

$$\dim\ker K \leq 1.$$

Remark 6. This estimate can be confirmed for any inner function (rational or non-rational), differentiable in a neighboord of $t = 1$ in $\mathbb{T}$ since, in this case, it is possible to obtain an explicit generalized factorization of the form (3) for the matrix function b_0 by using the [AFact] algorithm (see [6]). One of the factors of this generalized factorization is

$$\Lambda(t) = \mathrm{diag}\{t^k, t^{-k}\}$$

where k is precisely the number defined in (11).

4 CONCLUSIONS

The design of our analytical algorithms is focused on the possibility of implementing on a computer all, or a significant part of, the extensive symbolic and numeric calculations present in the algorithms. The methods developed rely on innovative techniques of Operator Theory and have a great potential of extension to ever more complex and general

[4]This condition is provided explicitly in the *output* of the [AFact] algorithm.

[5]When function $\theta(t)$ contains polynomials with a high degree it is not feasible to use the [ARFact-Matrix] to compute the right partial indices of b_0, κ_1 and κ_1.

problems. Also, by implementing these methods on a computer, new and powerful tools are created for exploring that same potential, making the results of lengthy and complex calculations available in a simple way to researchers of different areas.

- We note that most of all the concepts and results established for the unit circle within Operator Theory can be generalized for the real line. It is our opinion that the design and implementation of analytical algorithms that work with singular integral operators defined in the real line is a very interesting new line of research. Currently, we are attempting to generalize [ADimKer-NonCarleman] to other types of curves (namely the real line) and to other classes of singular integrals.

- We hope that our work within the Operator Theory, and with *Mathematica*, will help in the path to the future design and implementation of several other analytical algorithms, with numerous applications in many areas of research and technology.

Acknowledgements

This research was partially supported by Center for Functional Analysis, Linear Structures and Applications (CEAFEL), Instituto Superior Técnico (Portugal).

APPENDIX A

Example 2.

Right factorization of b_0:

$$b_0(t) = \begin{pmatrix} t^{-1} & \dfrac{2t}{2t-i} \\ \dfrac{2i}{t^2(t-2i)} & \dfrac{-2t^2+it+2}{-2t^2+5it+2} \end{pmatrix} = b_-(t)\Lambda(t)b_+(t)$$

$$b_-(t) = \begin{pmatrix} \dfrac{3(2-3i)t}{13(i-2t)} & 1+\dfrac{3(8-15i)}{289(i-2t)} \\ \dfrac{3(2-3i)+26t}{13(2t-i)} & \dfrac{4[(6+61i)-3(38+i)t]}{289t(2t-i)} \end{pmatrix}, \Lambda(t) = \begin{pmatrix} t^{-1} & 0 \\ 0 & t^{-1} \end{pmatrix},$$

$$b_+(t) = \begin{pmatrix} \dfrac{6(38+i)}{289(t-2i)} & 1+\dfrac{4(73+116i)-4(54-29i)t-289t^2}{289(2i-t)} \\ 1-\dfrac{3(2-3i)}{26(2i-t)} & \dfrac{(2-3i)t[6(2-i)+(7+6i)t]}{26(t-2i)} \end{pmatrix}$$

Example 3.

Right factorization of b_0:

$$b_0(t) = \begin{pmatrix} 1 & \dfrac{t}{1-2t} \\ \dfrac{1}{2-t} & \dfrac{2(t-1)^2}{(2t-1)(t-2)} \end{pmatrix} = b_-(t)\Lambda(t)b_+(t)$$

$$b_-(t) = \begin{pmatrix} 1+\dfrac{3}{8(2t-1)} & \dfrac{3}{2} \\ \dfrac{1}{4(2t-1)} & 1 \end{pmatrix}, \Lambda(t) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$b_+(t) = \begin{pmatrix} 1+\dfrac{3}{2(t-2)} & -2+\dfrac{1}{2-t} \\ \dfrac{9}{8(2-t)} & 1+\dfrac{3}{4(t-2)} \end{pmatrix}$$

REFERENCES

[1] Baturev, A.A., Kravchenko, V.G., Litvinchuk, G.S. "Approximate methods for singular integral equations with a non-Carleman shift", *J. Integral Equations Appl.* Vol. **8(1)**, pp. 1-17, 1996.

[2] Clancey, K., Gohberg, I. *Factorization of Matrix Functions and Singular Integral Operators*, Operator Theory: Advances and Applications, Birkhäuser Verlag, Vol. **3**, 1981.

[3] Conceição, A.C. *Factorization of Some Classes of Matrix Functions and its Applications (in portuguese)*, PhD Thesis, Universidade do Algarve (Portugal), 2007.

[4] Conceição, A.C., Kravchenko, V.G., Pereira, J.C. "Computing some classes of Cauchy type singular integrals with *Mathematica* software", *Adv Comput Math*, Springer US, Vol. **39**, pp. 273-288, 2013.

[5] Conceição, A.C., Kravchenko, V.G., Pereira, J.C. "Rational Functions Factorization Algorithm: a symbolic computation for the scalar and matrix cases", *Proceedings of the 1st National Conference on Symbolic Computation in Education and Research (CSEI2012)*, Lisboa, Portugal, P02, 2012.

[6] Conceição, A.C., Kravchenko, V.G., Pereira, J.C. "Factorization Algorithm for Some Special Non-rational Matrix Functions", *Oper. Theory Adv. Appl.*, Birkhäuser Basel, Vol. **202**, pp. 87-109, 2010.

[7] Conceição, A.C., Marreiros, R.C. "On the kernel of a singular integral operator with non-Carleman shift and conjugation", *Oper. Matrices, 24 pp., to appear.*

[8] Conceição, A.C., Pereira, J.C. "Exploring the spectra of some classes of paired singular integral operators: the scalar and matrix cases", *Libertas Mathematica (new series), 35 pp., to appear.*

[9] Ehrhart, T. "Invertibility theory for Toeplitz plus Hankel operators and singular integral operators with flip", *J. Funct. Anal.* Vol. **208(1)**, pp. 64-106, 2004.

[10] Gohberg, I., Kaashoek, M.A., Spitkovsky, I.M. "An Overview of Matrix Factorization Theory and Operator Applications", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **141**, pp. 1-102, 2003.

[11] Gohberg, I., Krupnik, N. "One-Dimensional Linear Singular Integral Equations", *Operator Theory: Advances and Applications*, Birkhäuser Verlag, Vol. **53**, 1992.

[12] Kravchenko, V.G., Lebre, A.B., Litvinchuk, G.S. "Spectrum Problems for Singular Integral Operators with Carleman Shift", *Math. Nachr.*, **226(1)**, pp. 129-151, 2001.

[13] Kravchenko, V.G. , Lebre, A.B., Rodríguez, J.S. "Factorization of Singular Integral Operators with a Carleman Shift and Spectral Problems", *J. Integral Equations Appl.*, Vol. **13(4)**, pp. 339-383, 2001.

[14] Kravchenko, V.G., Litvinchuk, G.S. *Introdution to the Theory of Singular Integral Operators with Shift*, Mathematics and its Applications", Kluwer Academic Publishers, **289**, 1994.

[15] Kravchenko, V.G., Marreiros, R.C. "On the dimension of the kernel of a singular integral operator with shift", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **242**, pp. 197-220, 2014.

[16] Kravchenko, V.G., Marreiros, R.C., Rodriguez, J.S. "An estimate for the number of solutions of an homogenous generalized Riemann boundary value problem with shift", *Oper. Theory Adv. Appl.*, Birkhäuser Verlag, Vol. **220**, pp. 163-178, 2012.

[17] Litvinchuk, G.S. *Solvability Theory of Boundary Value Problems and Singular Integral Equations with Shift*, Mathematics and its Applications, Kluwer Academic Publishers, Vol. **523**, 2000.

[18] Litvinchuk, G.S. *Boundary Value Problems and Singular Integral Equations with Shift*, Nauka, Moscow, (in russian), 1977.

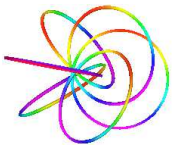
[19] Litvinchuk, G.S., Spitkovskii, I.M. *Factorization of Measurable Matrix Functions*, Operator Theory: Advances and Applications, Birkhäuser Verlag, Vol. **25**, 1987.

[20] Marreiros, R.C. *On the kernel of singular integral operators with non-Carleman shift*, Ph.D thesis, University of Algarve, Faro, (in portuguese), 2006.

[21] Spitkovskii, I.M., Tashbaev, A.M. "Factorization of Certain Piecewise Constant Matrix Functions and its Applications", *Math. Nachr.* Vol. **151**, pp. 241-261, 1991.

[22] Vekua, I.N. *Generalized Analytic Functions*, Nauka, Moscow, (in russian), 1988.

[23] Vekua, N.P. *Systems of Singular Integral Equations*, Nauka, Moscow, (in russian), 1970.



SYMCOMP 2015
Faro, March 26-27, 2015

MODELLING AND NUMERICAL SIMULATION OF THE RECENT OUTBREAK OF EBOLA

Amira Rachah[1] and Delfim F. M. Torres[2]

[1]Mathématiques pour l'Industrie et la Physique
Institut de Mathématiques de Toulouse
Université Paul Sabatier
F-31062 Toulouse Cedex 9, France
e-mail: amira.rachah@math.univ-toulouse.fr

[2]Center for Research and Development in Mathematics and Applications (CIDMA)
Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal
e-mail: delfim@ua.pt

Keywords: Ebola modelling, SEIR epidemic model, Matlab simulations.

Abstract. *We present a mathematical model that describes one of the most virulent pathogens for humans—the Ebola virus. The spread of this lethal virus is investigated by the SEIR (Susceptible, Exposed, Infectious, Recovered) model. We discuss and simulate the model and then control the evolution of the Ebola virus with the goal to study the impact of vaccination on its spread.*

1 INTRODUCTION

The Ebola virus is one of the most deadliest pathogens for humans. The latest major outbreak occurred in West Africa in 2014 [1,2]. The extremely rapid increase of this lethal virus and the high mortality rate, make it a major problem for public health. Signs start between two days and three weeks after contracting the virus. Symptoms typically begin with fever, sore throat, muscle pain, and headaches. Then, vomiting, diarrhea and rash usually follow, along with decreased function of the liver and kidneys. At this time some people begin to bleed both internally and externally. The disease has a high risk of death, killing between 25 and 90 percent of those infected, which typically happens six to sixteen days after symptoms appear [3–9].
To understand the spread of the latest outbreak, it is crucial to model the virus by using parameters estimated from recent data of World Health Organization (WHO) and simulate it. Real-time estimates of the virus parameters and mathematical modelling are important for predicting the evolution of this Ebola outbreak and investigate the

effects of ongoing and new interventions [10], e.g., vaccination as a precaution to the epidemic [11–14].
The paper is organized as follows. In Section 2 we describe the 2014 Ebola virus outbreak using the SEIR (Susceptible, Exposed, Infectious, Recovered) compartimental model for the spread of the disease, which is described by a system of four ordinary differential equations (ODE). After transmission of the virus, the susceptible individuals S enter the exposed (infected but not infectious) class E before they become infectious I, that either recover and survive (R) or die [15–19] (Section 2.1). To solve numerically the system modeling the recent Ebola outbreak (by using parameters estimated from real-data), we use an ODE solver of Matlab (Section 2.2). Then, in order to predict the effect of vaccination on the infected individuals along time, we improve the model by adding a vaccination term (Section 3.1). We discuss and simulate the model in case of vaccination, controlling the evolution of the virus with the goal to study the impact of vaccination on the spread of Ebola (Section 3.2). We end with Section 4 of conclusion and an appendix with our Matlab code.

2 MODELLING AND SIMULATION OF THE EBOLA VIRUS

We begin by modeling and simulating the evolution of the spread of Ebola virus, using the Susceptible–Exposed–Infectious–Recovered (SEIR) model. A simpler but not so realistic epidemic SIR model has been recently considered in [10], with no incubation period included in the study. Here, in contrast, the dynamics of the model is given in three stages: susceptible to exposed (infected but not infectious), exposed to infectious, and infectious to recovered. The novelties now considered are important from the medical point of view, since the length of time between exposure to the Ebola virus and the development of symptoms (incubation period) always exists, between 2 to 21 days [20,21], being typically between 4 to 10 days [22].

2.1 Mathematical model

In our description of the transmission of Ebola virus, we model the total infections that occur during the outbreak by subdividing the population into four classes. The susceptible individuals at time t, denoted by $S(t)$, enter the exposed class $E(t)$ before they become infectious. The infectious class at time t, denoted by $I(t)$, represents the individuals that are infected with the disease and are suffering the symptoms of Ebola. Finally, we have the recovered class, which at time t is denoted by $R(t)$. The total population, assumed constant during the short period of time under study, is given by $N = S(t)+E(t)+I(t)+R(t)$ at any instant of time t. The transmission of Ebola virus is then described by the

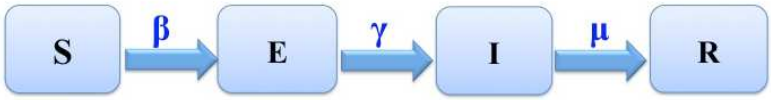


Figure 1: Diagram of the Susceptible–Exposed–Infectious–Recovered (SEIR) model (1).

following set of nonlinear ordinary differential equations (ODEs):

$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta S(t)I(t), \\ \dfrac{dE(t)}{dt} = \beta S(t)I(t) - \gamma E(t), \\ \dfrac{dI(t)}{dt} = \gamma E(t) - \mu I(t), \\ \dfrac{dR(t)}{dt} = \mu I(t). \end{cases} \tag{1}$$

The transitions between different states are described by the following parameters:

- the transmission rate β,
- the infectious rate γ,
- the recovered rate μ.

Figure 1 shows the relationship between the variables of our SEIR model.

2.2 Numerical simulation

We simulate model (1) by using the parameters estimated on November 2014 by Kaurov and Althaus [18, 23], who studied statistically the recent data of the World Health Organisation (WHO) [24]. In particular, in the statistical study [23], Kaurov studies the outbreak by modeling it with Wolfram's Mathematica language. The parameters obtained by Kaurov and Althaus are $\beta = 0.2$, $\gamma = 0.1887$ and $\mu = 0.1$, and are based on the fact that 88% of population is susceptible, 7% of population is exposed (infected but not infectious) and 5% of population is infectious [18, 23]. In agreement, the initial susceptible, exposed, infectious and recovered populations are given respectively by $S(0) = 0.88$, $E(0) = 0.07$, $I(0) = 0.05$ and $R(0) = 0$.
Our numerical simulations were done with the ode45 solver of Matlab (see Appendix A.1). The numerical solution of the system of differential equations, which represents mathematically the SEIR model of the spread of Ebola, is shown in Figure 2. To interpret the results, we describe the evolution of each group along time and the links between them. Figure 2 shows that the susceptible group begins to plummet due to how infectious the

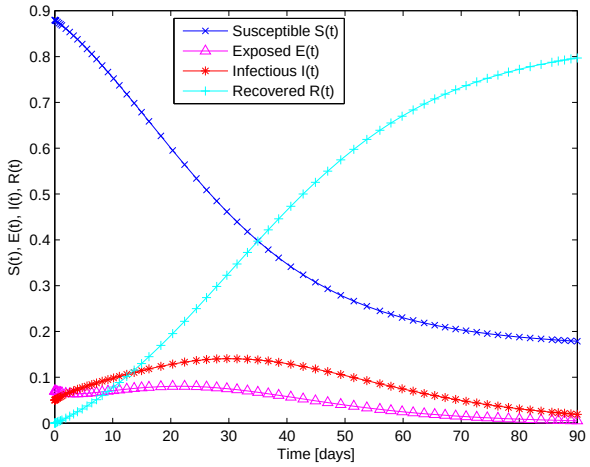


Figure 2: Solution to the Susceptible–Exposed–Infectious–Recovered (SEIR) model (1) with $S(0) = 0.88$, $E(0) = 0.07$, $I(0) = 0.05$, $R(0) = 0$, $\beta = 0.2$, $\gamma = 0.1887$ and $\mu = 0.1$.

virus is and, simultaneously, the exposed and infectious individuals begin to rise. Note that our results are in agreement with the value of the basic reproduction number R_0 that, in this case, is given by $R_0 = \beta/\mu = 2$, meaning that the infection spreads fast in the population [25].

3 THE EBOLA VACCINATION MODEL

Infectious diseases have tremendous influence on human life. In last decades, controlling infectious diseases has been an increasingly complex issue. A strategy to control infectious diseases is through vaccination [26]. Now our idea is to study the effect of vaccination in practical Ebola situations. The French nurse cured of Ebola with the help of an experimental vacine, after contracting the virus in Liberia, is a proof of the possibility of treatment, by vaccinating infected individuals [27].

3.1 Dynamics of the vaccination model

We now improve the SEIR model described in Section 2.1 by introducing vaccination. The system of equations that describe the Susceptible–Exposed–Infectious–Recovery (SEIR)

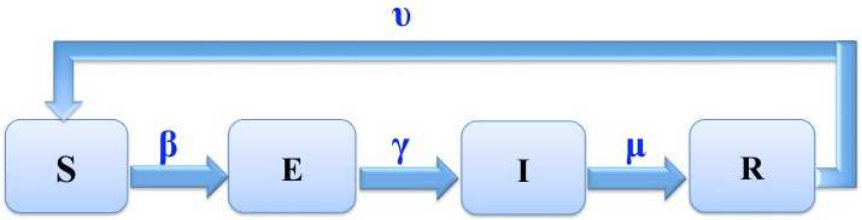


Figure 3: The scheme of the Susceptible–Exposed–Infectious–Recovered model with vaccination (2).

model with vaccination is:

$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta S(t)I(t) + \upsilon S(t), \\ \dfrac{dE(t)}{dt} = \beta S(t)I(t) - \gamma E(t), \\ \dfrac{dI(t)}{dt} = \gamma E(t) - \mu I(t), \\ \dfrac{dR(t)}{dt} = \mu I(t) - \upsilon S(t), \end{cases} \tag{2}$$

where β is the rate of infection, μ is the rate of recovery, and υ is the percentage of individuals vaccinated every day [28]. Note that for $\upsilon = 0$ (i.e., no vaccination) system (2) reduces to (1). The relationship between the variables of the SEIR model with vaccination (2) is shown in Figure 3.

3.2 Simulation of the vaccination model

We simulate, with the help of Matlab (see Appendix A.2), the SEIR model with vaccination (2) in order to predict the evolution of every class of individuals in case of vaccination. In our study, comparison is based in the test of different rates of vaccinations and their effect on the curve of each group. The results are shown in Figures 4 and 5. These figures present different rates of vaccination and their effect on the peak of the curve of infectious individuals along time. The curve of infectious individuals shows that when the percentage of vaccinated individuals increase, the peak of the curve of infectious individuals is less important, the period of infection (the corresponding number of days) is shorter and the number of recovered is more important. In fact, in case of infection without vaccination, the curve of the infectious class goes to zero in about 90 days (see Figure 4), where in presence of vaccination the curve of the infectious class goes to zero in about 35 days (see Figure 5). This shows the efficiency of using vaccination in controlling Ebola. Table 1 presents the maximum percentage of infectious individuals corresponding to different rates of vaccinations and the respective number of days for the peak of infectious. The evolution of the spread of the Ebola virus is also modelled and simulated, by

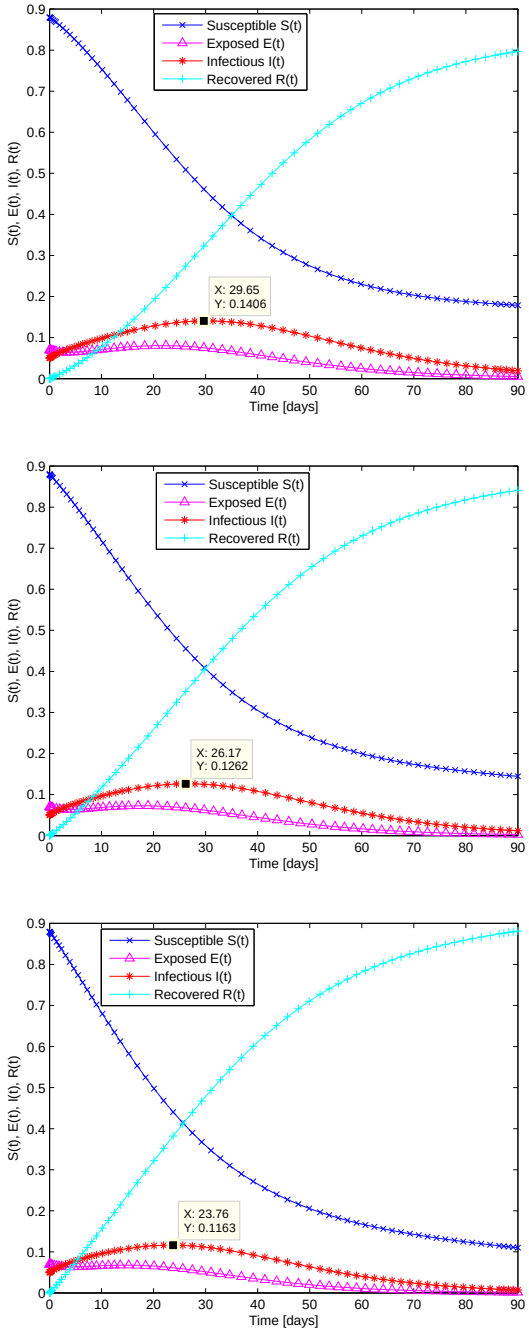


Figure 4: Solution to the vaccination SEIR model (2) with $\beta = 0.2$, $\gamma = 0.1887$, $\mu = 0.1$, $S(0) = 0.88$, $E(0) = 0.07$, $I(0) = 0.05$ and $R(0) = 0$, and different rates of vaccination: $\upsilon = 0$ (up), $\upsilon = 0.005$ (middle), and $\upsilon = 0.01$ (down).

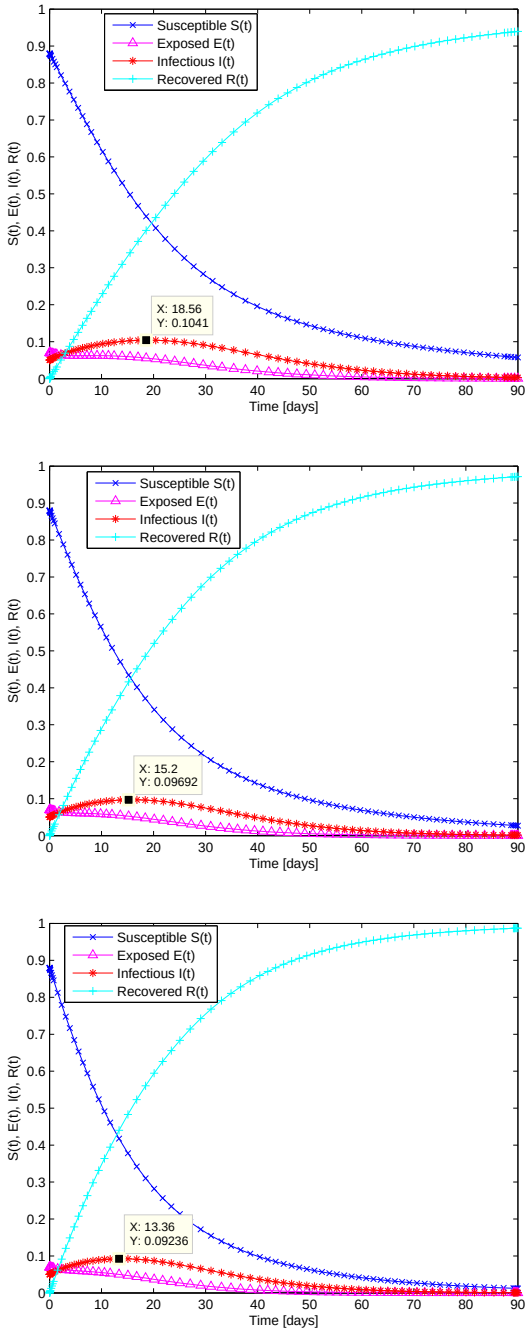


Figure 5: Solution to the vaccination SEIR model (2) with $\beta = 0.2$, $\gamma = 0.1887$, $\mu = 0.1$, $S(0) = 0.88$, $E(0) = 0.07$, $I(0) = 0.05$ and $R(0) = 0$, and different rates of vaccination: $\upsilon = 0.02$ (up), $\upsilon = 0.03$ (middle) and $\upsilon = 0.04$ (down).

Rate of vaccination	Maximum percentage of infectious	Days to the peak of $I(t)$
0.000	14.0%	29.65
0.005	12.6%	26.17
0.010	11.6%	23.76
0.020	10.4%	18.56
0.030	09.6%	15.20
0.040	09.2%	13.36

Table 1: Maximum percentage of infectious individuals I corresponding to different rates υ of vaccination and respective number of days for the peak of I.

Rate of vaccination	Maximum % of I of SEIR	Maximum % of I of SIR
0.000	14.0%	17%
0.005	12.6%	15%
0.010	11.6%	13%

Table 2: Comparison between the SIR model studied in [10] and the SEIR model investigated here: maximum percentage of infectious individuals I corresponding to different rates υ of vaccination.

using the more basic SIR (Susceptible–Infectious–Recovery) epidemic model, in [10]. In Table 2 we present a comparison between the numerical results of the SIR model of [10] and the numerical results of the SEIR model here obtained. Precisely, Table 2 shows the maximum percentage of infectious individuals corresponding to different rates of vaccination, both in case of the SIR model [10] and in the case of the SEIR model considered here. One may conclude that the model now proposed for the 2014 Ebola outbreak in West Africa is an improvement of the SIR model studied before in [10].

4 CONCLUSION

We discussed a SEIR mathematical model that provides a good description of the 2014 Ebola outbreak. Our study of the SEIR model improves the recent results obtained in [10] for the more basic SIR model. The SEIR model flows from SIR model, describing better the spread of Ebola of the 2014 outbreak in West Africa by being closer to the data provided by the World Health Organization and to the reality of the epidemics. Our results show that a vaccination strategy greatly helps to reduce the number of infected and increases the number of recovered individuals, in a short period of time.
As future work, we plan to investigate the usefulness of optimal control to minimize the number of infected individuals and the cost of vaccination, when subject to the SEIR model with vaccination here considered.

A MATLAB CODE

In this appendix we provide the Matlab code used in our simulations. The scripts are a direct implementation of the differential equations that describe the considered models. Only standard Matlab features are used.

A.1 Matlab code for the SEIR model (1) without vaccination

```
function dy = eq_model_SEIR(t,y)
dy = zeros(4,1);
dy(1) =  -beta.*y(1).*y(3);
dy(2) =  beta.*y(1).*y(3) - gamma.*y(2);
dy(3) =  gamma.*y(2) - mu.*y(3);
dy(4) =  mu.*y(3);

y0=[S_0;E_0;I_0;R_0];
t0= [0 100];
[time,y]= ode45(@eq_model_SEIR,t0,y0);
S=y(:,1);
E=y(:,2);
I=y(:,3);
R=y(:,4);
h1=figure(1)
plot(temps,S,'b-x',temps,E,'m-^',temps,I,'r-*',temps,R,'c-+')
xlabel('Time [days]');
ylabel('S(t), E(t), I(t), R(t)');
legend('Susceptible','Exposed','Infectious','Recovered');
```

A.2 Matlab code for the SEIR model (2) with vaccination

```
function dy = eq_model_SEIR_vac(t,y)
dy = zeros(4,1);
dy(1) =  -beta.*y(1).*y(3)- v.*y(1);
dy(2) =  beta.*y(1).*y(3) - gamma.*y(2);
dy(3) =  gamma.*y(2) - mu.*y(3);
dy(4) =  mu.*y(3)+ v.*y(1);

y0=[S_0;E_0;I_0;R_0];
t0= [0 100];
[time,y]= ode45(@eq_model_SEIR_vac,t0,y0);
S_vac=y(:,1);
E_vac=y(:,2);
I_vac=y(:,3);
R_vac=y(:,4);
h1=figure(1)
plot(temps,S_vac,'b-x',temps,E_vac,'m-^',temps,I_vac,'r-*',temps,R_vac,'c-+')
xlabel('Time [days]');
```

```
ylabel('S(t), E(t), I(t), R(t)');
legend('Susceptible','Exposed','Infectious','Recovered');
```

Acknowledgments

This research was supported by the *Institut de Mathématiques de Toulouse* (Rachah) and the *Portuguese Foundation for Science and Technology* (FCT), within CIDMA project UID/MAT/04106/2013 and OCHERA project PTDC/EEI-AUT/1450/2012, co-financed by FEDER under POFC-QREN with COMPETE reference FCOMP-01-0124-FEDER-028894 (Torres).

REFERENCES

[1] M. Barry, F. A. Traoré, F. B. Sako, D. O. Kpamy, E. I. Bah, M. Poncin, S. Keita, M. Cisse, A. Touré. "Ebola outbreak in Conakry, Guinea: Epidemiological, clinical, and outcome features". *Médecine et Maladies Infectieuses* 44 (2014), no. 11–12, 491–494.

[2] J. A. Lewnard, M. L. Ndeffo Mbah, J. A. Alfaro-Murillo, F. L. Altice, L. Bawo, T. G. Nyenswah, A. P. Galvani. "Dynamics and control of Ebola virus transmission in Montserrado, Liberia: a mathematical modelling analysis". *The Lancet Infectious Diseases* 14 (2014), no. 12, 1189–1195.

[3] "WHO, Report of an International Study Team. Ebola haemorrhagic fever in Sudan 1976". *Bull. World Health Organ.* 56 (1978), no. 2, 247–270.

[4] "Report of an International Commission. Ebola haemorrhagic fever in Zaire, 1976". *Bull. World Health Organ.* 56 (1978), no. 2, 271–293.

[5] "Uganda Ministry of Health. An outbreak of Ebola in Uganda". *Trop. Med. Int. Health.* 7 (2002), no. 12, 1068–1075.

[6] J. Legrand, R. F. Grais, P. Y. Boelle, A. J. Valleron, A. Flahault. "Understanding the dynamics of Ebola epidemics". *Epidemiol. Infect.* 135 (2007), no. 4, 610–621.

[7] S. F. Dowell, R. Mukunu, T. G. Ksiazek, A. S. Khan, P. E. Rollin, C. J. Peters. "Transmission of Ebola hemorrhagic fever: a study of risk factors in family members, Kikwit, Democratic Republic of the Congo, 1995. Commission de Lutte contre les Epidémies à Kikwit". *J. Infect. Dis.* 179 (1999), Suppl. 1, S87–S91.

[8] L. Borio et al. [Working Group on Civilian Biodefense; Corporate Author]. "Hemorrhagic fever viruses as biological weapons: medical and public health management". *Journal of the American Medical Association* 287 (2002), no. 18, 2391–2405.

[9] C. J. Peters, J. W. LeDuc. "An introduction to Ebola: the virus and the disease". *Journal of Infectious Diseases* 179 (Suppl. 1), 1999.

[10] A. Rachah, D. F. M. Torres. "Mathematical modelling, simulation and optimal control of the 2014 Ebola outbreak in West Africa". *Discrete Dyn. Nat. Soc.* 2015 (2015), Art. ID 842792.

[11] C. P. Farrington, M. N. Kanaan. "Branching process models for surveillance of infectious diseases controlled by mass vaccination". *Biostatistics* 4 (2003), no. 2, 279–295.

[12] Forum on Medical and Public Health Preparedness for Catastrophic Events. The 2009 H_1N_1 Influenza Vaccination Campaign, Summary of a Workshop Series, The National Academies Press, Washington, D.C., 2010.

[13] W. Robert. "Tindle Vaccines for Human Papillomavirus Infection and Disease Medical Intelligence Unit". Medical Intelligence Unit, R. G., Landes Company, Austin, Texas, 1999.

[14] H. S. Rodrigues, M. T. T. Monteiro, D. F. M. Torres. "Vaccination models and optimal control strategies to dengue". *Math. Biosci.* 247 (2014), 1–12.

[15] G. Chowell, J. M. Hayman, L. M. A. Bettencourt, C. Castillo-Chavez. *Mathematical and Statistical Estimation Approaches in Epidemiology*, Springer, Dordrecht, 2009.

[16] D. Zeng, H. Chen, C. Castillo-Chavez, W. B. Lober, M. Thurmond. "Infectious Disease Informatics and Biosurveillance". *Integrated Series in Information Systems*, Vol. 27, Springer, New York, 2011.

[17] G. Chowell, N. W. Hengartner, C. Castillo-Chavez, P. W. Fenimore, J. M. Hyman. "The basic reproductive number of Ebola and the effects of public health measures: the cases of Congo and Uganda". *Journal of Theoretical Biology* 229 (2004), no. 1, 119–126.

[18] C. L. Althaus. "Estimating the Reproduction Number of Ebola Virus (EBOV) During the 2014 Outbreak in West Africa". PLOS Currents Outbreaks, September 2, 2014. `http://dx.doi.org/10.1371/currents.outbreaks.91afb5e0f279e7f29e7056095255b288`

[19] J. Astacio, D. Briere, M. Guilléon, J. Martinez, F. Rodriguez, N. Valenzuela-Campos. "Mathematical models to study the outbreaks of Ebola". Report BU-1365-M, 1996.

[20] WHO. "Ebola virus disease Fact sheet No. 103", World Health Organization, September 2014. `http://www.who.int/mediacentre/factsheets/fs103/en`

[21] WHO. "Ebola Hemorrhagic Fever Signs and Symptoms". CDC. 28 January 2014. `http://www.cdc.gov/vhf/ebola/symptoms/index.html`

[22] M. Goeijenbier, J. J. van Kampen, C. B. Reusken, M. P. Koopmans, E. C. van Gorp. "Ebola virus disease: a review on epidemiology, symptoms, treatment and pathogenesis". *Neth. J. Med.* 72 (2014), no. 9, 442–448.

[23] V. Kaurov. "Modeling a Pandemic like Ebola with the Wolfram Language". Technical Communication & Strategy, November 4, 2014. http://blog.wolfram.com/2014/11/04/modeling-a-pandemic-like-ebola-with-the-wolfram-language

[24] WHO, World Health Organization. Ebola response roadmap – Situation report update. Last time accessed: October 2014. http://www.who.int/csr/disease/ebola/situation-reports/en

[25] M. A. Lewis, M. Dietel, P. Scriba, W. K. Raff. *Biologie und Epidemiologie der Hormonersatztherapie-Biology and Epidemiology of Hormone Replacement Therapy.* Springer-Verlag Berlin, Heidelberg, 2006.

[26] H. S. Rodrigues, M. T. T. Monteiro, D. F. M. Torres. "Dengue in Cape Verde: vector control and vaccination". *Math. Popul. Stud.* 20 (2013), no. 4, 208–223.

[27] A. J. Valleron, D. Schwartz, M. Goldberg, R. Salamon. Collectif Lépidémiologie humaine, Conditions de son développement en France, et rôle des mathématiques. Institut de France Académie des Sciences, Vol. 462, 2006.

[28] M. Jérôme Yon. Modélisation de la propagation d'un virus. INSA Rouen, 2010.

SYMCOMP 2015
Faro, March 26-27, 2015

SECOND-ORDER FINITE VOLUME MOOD METHOD FOR THE SHALLOW WATER WITH DRY/WET INTERFACE

Jorge Figueiredo[1*], Stéphane Clain[1,2]

1: Centre of Mathematics
School of Sciences
University of Minho
Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: {jmfiguei,clain}@math.uminho.pt

2: Institut de Mathématiques de Toulouse
Université Paul Sabatier
31062 Toulouse, France

Keywords: finite volume, dry/wet interface, shallow water equation, hydrostatic reconstruction

Abstract. *The shallow water system is a fundamental work-piece for tsunami or flooding simulations. One of the major difficulties is the correct location of the dry/wet interface to evaluate accurate approximations of the velocity and kinetic energy. On the other hand, the MOOD method has been recently proposed to provide more efficient schemes in the framework of the Euler system. We propose to compare two second-order methods, namely the MUSCL and the MOOD techniques, and draw comparisons on accuracy shock capturing and dry/wet interface.*

1 INTRODUCTION

Shallow water equations with varying bathymetry is a challenging topic due to the wide number of applications such that tsunami, flooding, coastal erosion [14]. A large number of numerical schemes has been proposed and studied and, in particular, very high-order finite volume methods (fifth- and sixth-order for instance) have received great attention in order to provide very accurate numerical solutions [10, 11, 7]. Nevertheless, the design of new second-order methods is still an important objective since most of the engineering or environmental applications are developed with such a technique due to its simplicity and computational efficiency [1, 9, 15]. Recently, a new limiting technology, named the Multi-dimensional Optimal Order Detection (MOOD), has been proposed and tested for the Euler system [5, 6, 8]. An extension version has been proposed for the non-conservative shallow water equations where a sixth-order scheme was tested on two-dimensional unstructured meshes [7]. We here tackle the question of comparing the efficiency of the MOOD and MUSCL second-order methods. The two techniques are fundamentally different since the MUSCL is based on *a priori* criteria, whereas the MOOD uses *a posteriori* detectors to prevent the solution from oscillating in the vicinity of discontinuities.
This work proposes a comparison study between the two methods for the simple one-dimensional case in order to assess their accuracy and shock capturing capacity, as well as dry/wet interfaces location. After developing the key ingredients of the discretization, we introduce the MUSCL and MOOD methods and highlight their differences. Numerical tests are then carried out to draw the comparisons using four relevant simulations: the lake at rest to check the C-property, a regular case for the accuracy, a discontinuous case for the shock capturing, and finally a set of two dry/wet tests to evaluate both methods in this specific but important configuration (flooding, dam break or tsunami).

2 SECOND-ORDER FINITE VOLUME SCHEME

The classical shallow water system with varying bathymetry writes

$$\begin{aligned} \partial_t h + \partial_x(hu) &= 0, \\ \partial_t(hu) + \partial_x\left(hu^2 + \frac{g}{2}h^2\right) &= -gh\partial_x b, \end{aligned}$$

where h denotes the water height, u the velocity, b the bathymetry, $g = 9.81$ the gravity acceleration, and $\eta = h + b$ the free surface. Vector $U = (h, hu, b)$ represents the conservative quantities.

2.1 Discretization

Domain $\Omega = [0, L]$ is decomposed into non-overlapping cells $c_i = [x_{i-1/2}, x_{i+1/2}]$ with centroid x_i, $i = 1, \cdots, I$. For a final time T, $0 = t^0 < t^1 < \cdots < t^n < \cdots < t^N = T$ is a regular subdivision with time step $\Delta t = \frac{T}{N}$. We denote by $\Delta x_i = |c_i|$ the length of the cell, while α_i^n represents an approximation of the mean value over cell c_i for function α

($\alpha = h, \eta, hu, b$) at time t^n. We recall that for regular functions (say C^2) over the cell c_i, the point-wise value at x_i is a second-order approximation of the mean value. In the same way, $\alpha^n_{i+1/2,L}$ and $\alpha^n_{i+1/2,R}$ represent approximations on the left and right side of $x_{i+1/2}$.

2.2 Hydrostatic reconstruction

We recall the hydrostatic reconstruction introduced by Audusse *et al.* [1]. We denote by $b^n_{i+1/2} = \max(b^n_{i+1/2,L}, b^n_{i+1/2,R})$ and set

$$h^{*,n}_{i+1/2,L} = \max(0, h^n_{i+1/2,L} - b^n_{i+1/2} + b^n_{i+1/2,L}), \quad \eta^{*,n}_{i+1/2,L} = h^{*,n}_{i+1/2,L} + b^n_{i+1/2},$$
$$h^{*,n}_{i+1/2,R} = \max(0, h^n_{i+1/2,R} - b^n_{i+1/2} + b^n_{i+1/2,R}), \quad \eta^{*,n}_{i+1/2,R} = h^{*,n}_{i+1/2,R} + b^n_{i+1/2}.$$

For the sake of consistency, we also set $u^{*,n}_{i+1/2,L} = u^n_{i+1/2,L}$ and $u^{*,n}_{i+1/2,R} = u^n_{i+1/2,R}$.

2.3 Generic second-order scheme

We use the Audusse *et al.* methodology [1, 9], where the following scheme has been proposed

$$U^{n+1}_i = U^n_i - \frac{\Delta t}{\Delta x_i}\left[\mathcal{F}^n_{i+1/2} + \varepsilon^n_{i+1/2,L} - F^n_{i-1/2} - \varepsilon^n_{i-1/2,R}\right] + \Delta t \mathcal{S}^n_i,$$

with $\mathcal{F}^n_{i-1/2} = \mathbb{F}(U^{*,n}_{i-1/2,L}, U^{*,n}_{i-1/2,R})$ the numerical flux for the conservative contribution (Rusanov or HLL for example [12]) with

$$U^{*,n}_{i-1/2,L} = \begin{pmatrix} h^{*,n}_{i-1/2,L} \\ h^{*,n}_{i-1/2,L} u^{*,n}_{i-1/2,L} \\ b^n_{i-1/2} \end{pmatrix}, \quad U^{*,n}_{i-1/2,R} = \begin{pmatrix} h^{*,n}_{i-1/2,R} \\ h^{*,n}_{i-1/2,R} u^{*,n}_{i-1/2,R} \\ b^n_{i-1/2} \end{pmatrix}.$$

We introduce the non-conservative numerical flux to deal with the discontinuous part of the non-conservative source term

$$\varepsilon^n_{i+1/2,L} = \frac{g}{2}\left[(h^{*,n}_{i+1/2,L})^2 - (h^n_{i+1/2,L})^2\right], \quad \varepsilon^n_{i-1/2,R} = \frac{g}{2}\left[(h^{*,n}_{i-1/2,R})^2 - (h^n_{i-1/2,R})^2\right],$$

while the discretization of the regular part of the non-conservative source term writes

$$\mathcal{S}^n_i = -g\frac{h^n_{i+1/2,L} + h^n_{i-1/2,R}}{2} \times \frac{b^n_{i+1/2,L} - b^n_{i-1/2,R}}{\Delta x_i}.$$

3 MUSCL versus MOOD

To provide a second-order scheme, local linear reconstructions are required, and therefore we compute slopes to provide an approximation of the first derivatives (see [4] for an overview of the MUSCL method). For any function $\alpha = h, \eta, hu, b$, we define the slopes

$$p^n_{i-1/2}(\alpha) = 2\frac{\alpha^n_i - \alpha^n_{i-1}}{\Delta x_{i-1} + \Delta x_i}, \quad p^n_{i+1/2}(\alpha) = 2\frac{\alpha^n_{i+1} - \alpha^n_i}{\Delta x_i + \Delta x_{i+1}},$$
$$p^n_i(\alpha) = 2\frac{\alpha^n_{i+1} - \alpha^n_{i-1}}{\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}}.$$

The first-order scheme corresponds to take $\alpha^n_{i+1/2,L} = \alpha^n_{i-1/2,R} = \alpha^n_i$, that is $p^n(\alpha) = 0$. Notice that for that case, we have $\mathcal{S}^n_i = 0$ and the contribution of bathymetry variations is concentrated on the interfaces. It is well-known that a non-limited linear reconstruction will give rise to oscillations in the vicinity of a discontinuity due to the Gibbs phenomenon and non-linear limiting procedures have to be implemented to locally reduce the accuracy and preserve the monotonicity. Based on linear reconstructions, the traditional MUSCL approach consists in reducing the slopes such that some stability criterion is achieved. Thus, such a method is *a priori* since the corrections are made before updating the solution. On the contrary, the MOOD method assumes that the solution is smooth enough. A candidate solution is computed without altering the slopes and then, based on the candidate solution, corrections of the slopes are provided to satisfy some stability criterion. The *a posteriori* method has the advantage to perform corrections only for problematic cells and therefore this technique is less intrusive and provides better accuracy.

3.1 MUSCL method

The MUSCL second-order scheme corresponds to define the reconstructed values on the left and right side of the interfaces by

$$\alpha^n_{i+1/2,L} = \alpha^n_i + q^n_i(\alpha)\frac{\Delta x_i}{2}, \quad \alpha^n_{i-1/2,R} = \alpha^n_i - q^n_i(\alpha)\frac{\Delta x_i}{2},$$

where the limited slope

$$q^n_i(\alpha) = \phi\left(p^n_{i-1/2}(\alpha), p^n_{i+1/2}(\alpha)\right)$$

is computed using a limiter function ϕ such as the min-mod, the van-Alabada or the van-Leer limiters. An important point is that the reconstruction cannot be performed with h, η and b at the same time for compatibility reasons. It has been proved that the good choice is to carry out the MUSCL procedure on h and η, and then deduce the values for b on the interfaces (this is the reason why b depends on time) setting

$$b^n_{i+1/2,L} = \eta^n_{i+1/2,L} - h^n_{i+1/2,L}, \quad b^n_{i-1/2,R} = \eta^n_{i-1/2,R} - h^n_{i-1/2,R}.$$

Notice that a second-order method in space also requires a second-order method in time to be effective. The usual TVD-RK2 (Heun method) is employed to guarantee a global second-order method for smooth solutions.

3.2 MOOD method

We give here a short introduction to the MOOD method, but a detailed description is given in [5, 6, 8, 7, 2]. The Cell Polynomial Degree (CPD) map corresponds to a vector associated to the cells which indicates the degree of the polynomial reconstruction, while the Edge Polynomial Degree (EPD) is a vector associated to the edges indicating the degree of the polynomial used to evaluate the reconstruction on both sides of the edges for the flux and source term computation. In practice, we take $\mathrm{EPD}(i+1/2) =$

$\min(\text{CPD}(i), \text{CPD}(i+1))$. In our specific case, a cell c_i may have a $\text{CPD}(i) = 1$ if we use the slope (second-order approximation) or $\text{CPD}(i) = 0$ if the slope is null (first-order approximation).
The MOOD method is based on the following loop. Assume that we known an approximation $U_h^n = (U_i^n)_{i=1,\cdots,I}$ at time t^n and initialise the CPD to 1, *i.e.* we use a second-order approximation for each cell.

1. We build a candidate solution $U_h^\star$ based on the CPD map. In practice, we use the reconstruction indicated by the corresponding EPD map to compute the reconstructed values used to evaluate the numerical fluxes and the source term.

2. We look at each value $U_i^\star$ of the candidate solution to check if it satisfies a set of conditions (or detectors).

3. If all the cells are valid, then the candidate solution turns out to be the approximation at time t^{n+1}, *i.e.* $U_h^{n+1} = U_h^\star$. Otherwise, we modify the CPD map reducing the polynomial degree from 1 to 0 for the problematic cells and go back to step 1.

The MOOD method assumes that the first-order scheme (the CPD map is zero everywhere) fulfils the set of conditions. Hence, in the worst case, the scheme becomes a first-order one.

3.2.1 Basic detectors

Several detectors may be defined to determine whether a cell is eligible or not as proposed in [7]. The main point is to detect if the candidate solution is physically admissible and to prevent the appearance of oscillations characterised by creation of local extrema.

Physical Admissible Detector (PAD) The candidate solution satisfies the PAD condition on cell c_i if $h_i^\star \geq 0$. Such a condition is crucial since negative water height values are non-physical.

Maximum Principle Detector (MPD) The candidate solution satisfies the MPD condition on cell c_i if

$$\min(h_{i-1}^n, h_i^n, h_{i+1}^n) \leq h_i^\star \leq \max(h_{i-1}^n, h_i^n, h_{i+1}^n),$$

which implies that the candidate value remains between the local minimum and local maximum at time t^n. Such condition enables to detect potential oscillation since the Gibbs phenomenon induces the creation of local extrema.

Extrema Detector (ED) In the case where the solution does not depend on time, the MPD condition does not make sense and therefore it is useful to consider the new following criterion

$$\min(h^\star_{i-1}, h^\star_{i+1}) \leq h^\star_i \leq \max(h^\star_{i-1}, h^\star_{i+1}),$$

which allows to detect the extrema of the discrete candidate solution.

3.2.2 Relaxation detectors

Extrema may derive from local oscillations associated to the Gibbs phenomenon, but may also be smooth extrema corresponding to real ones. Hence, we have to distinguish these two situations in order to set the CPD $= 0$ for the oscillation case, whereas in the other case we preserve the CPD $= 1$ to provide a second-order of approximation. To this end, we introduce a new tool. For any function $\alpha = h, \eta, hu$, we set

$$C_i(\alpha) = \frac{\alpha_{i+1} + \alpha_{i-1} - 2\alpha_i}{(\Delta x)^2},\ i = 2, \ldots, I-1, \quad \text{and} \quad C_1(\alpha) = C_2(\alpha),\ C_I(\alpha) = C_{I-1}(\alpha),$$

where for the sake of simplicity we assume $\Delta x_i = \Delta x$, and compute the following local curvature indicators

$$\chi_{m,i} = \min(C_{i-1}, C_i, C_{i+1}), \quad \chi_{M,i} = \max(C_{i-1}, C_i, C_{i+1}),$$

for $i = 2, \ldots, I-1$, where we omit the function α for the sake of simplicity. Notice that one can have $|\chi_{m,i}| > |\chi_{M,i}|$. We then define the following relaxation limiters.

Small Curvature Detector (SCD) or Plateau Detector Let ε_C be a given tolerance parameter. Then, CPD$(i) = 1$ if

$$\max(|\chi_{m,i}|, |\chi_{M,i}|) \leq \varepsilon_C.$$

Such condition means that the curvature is so small that the numerical solution is locally linear and therefore the reconstruction should not be limited.

Local Oscillation Detector (LOD) We must enforce CPD$(i) = 0$ if one has

$$\chi_{m,i}\chi_{M,i} \leq 0.$$

This condition detects a local oscillation due to the variation of the curvatures sign.

Smoothness Detector(SD) Let ε_S be a given tolerance parameter. The numerical solution is considered locally smooth if

$$1 \geq \frac{\min(|\chi_{m,i}|, |\chi_{M,i}|)}{\max(|\chi_{m,i}|, |\chi_{M,i}|)} \geq 1 - \varepsilon_S.$$

If that is the case we set CPD$(i) = 1$. This detector determines if the minimum and the maximum curvatures are close enough with respect to the threshold parameter and the numerical solution is considered locally smooth.

3.2.3 Condition for solution eligibility

Since the MOOD procedure can involve up to six detectors, we now detail how the detectors are linked one to each to other and are used to determine if the CPD is 0 (non-admissible candidate solution) or 1 (admissible). The detector chain is given in Figure 1 where for each cell, the algorithm detection provides the new value of the CPD.

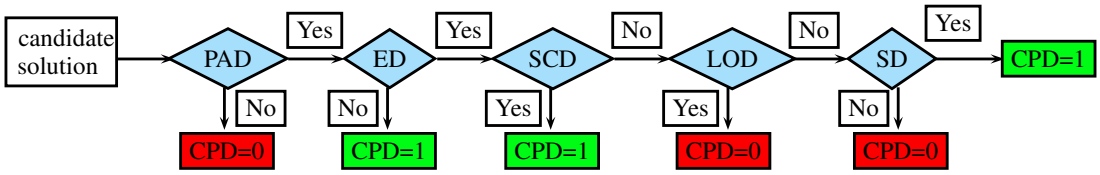


Figure 1: The chain detectors algorithm for the MOOD method.

4 NUMERICAL SIMULATIONS

Numerical tests are carried out to assess the performance of the two schemes. The time step Δt is controlled by the CFL condition reduced by a factor 0.4 with respect to the maximum admissible time step of the first-order scheme. All meshes are constituted by cells having equal length $\Delta x = L/I$. The HLL flux scheme is used in all numerical simulations since it is less diffusive than the Rusanov one. For the MUSCL method the limiter procedure is applied to h, η, and $q = hu$, whereas for the MOOD method only the water height h is considered in the detector procedures. In the MOOD case the relaxation parameters used within the detector scheme are $\varepsilon_C = \delta^3$ and $\varepsilon_S = 0.5$, where $\delta = \Delta x/L = 1/I$ (see [7]). Finally, when dry/wet interfaces are present, after each integration step we perform a clipping where we set the water height equal to zero if $h < 10^{-6}$.
To assess the convergence, we introduce the L^1- and L^∞-errors as

$$L^1\text{-error:} \quad \sum_{i=1}^{I} |\alpha_i^N - \alpha_i^{ex}|/I \quad \text{and} \quad L^\infty\text{-error:} \quad \max_i |\alpha_i^N - \alpha_i^{ex}|,$$

where (α_i^{ex}) and (α_i^N) are respectively the exact and the approximated mean values on cell c_i at the final time $t^N = T$.

4.1 Lake at rest

Lake at rest simulation is the first sanity check experience to test the C-property, *i.e.* the preservation of the steady-state situation with null velocity [3]. On domain $[0, 1]$, we assume that the fluid is initially at rest, *i.e.* $q = u = 0$, while the bathymetry b and its first derivative present some discontinuities and the total height $\eta = \max(2, b)$ has various

dry/wet interfaces (see Figure 2). We consider successive meshes with 50, 100, and 200 cells, and compute the solution until the final time $T = 1$ corresponding to 554, 1108 and 2215 time steps, respectively. In the simulations, reflection conditions are prescribed at the boundary.

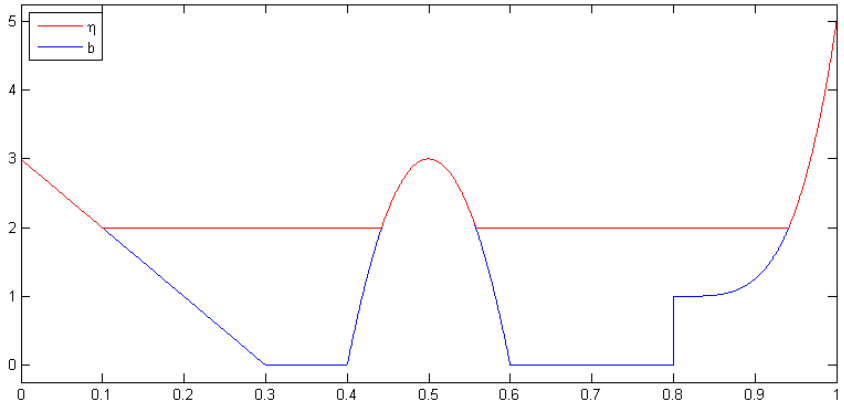


Figure 2: Bathymetry function and total height for the lake at rest.

After performing all the numerical tests, we report that the L^1- and L^∞-errors for η, h, q and u stand below 10^{-15} and 10^{-14}, respectively. Since the calculations are performed in double precision, we conclude that both MUSCL and MOOD implementations satisfy the C-property.

4.2 Smooth solution

We now turn to the regular case where we assess the schemes accuracy. We want to evaluate the impact of the limiting/detecting procedures when an optimal second-order approximation should be achieved. We intend to draw some comparisons between the MUSCL and MOOD schemes and determine which scheme provides the best performance. For this purpose, we consider a steady-state supercritical flow and evaluate the approximation which intends to preserve the stationary regime. The flow considered has the upstream boundary located at $x = 0$ and the downstream one at $x = L$ (we consider a channel with length $L = 10$). The stationary solution is given by

$$q(x) = q_0, \quad \frac{q_0^2}{2gh^2(x)} + h(x) + b(x) = \frac{q_0^2}{2gh^2(0)} + h(0) + b(0).$$

where we take $q_0 = 13.29$, and $h(0) = \eta(0) - b(0)$ with $\eta(0) = 2$ (see e.g. [13]). The bathymetry function is the exponential bump $b(x) = 0.2\exp(-5(x-4)^2))$ plotted in Figure 3 together with the free surface.

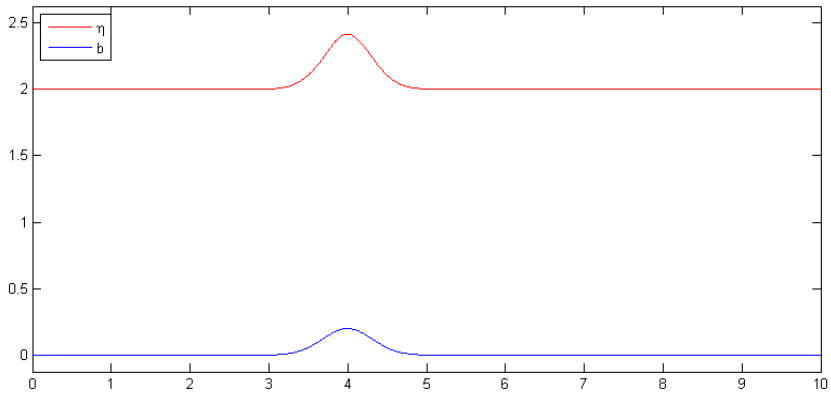


Figure 3: Bathymetry function and free surface for the supercritical stationary flow.

Given the nature of the flow - supercritical with a Froude number larger than 1.25 - Dirichlet boundary conditions are prescribed at $x = 0$, whereas transmission conditions hold at the downstream boundary. The initial condition is the steady-state solution and the numerical simulations are carried out until $t = 10$ for meshes with 100, 200, 400, 800 and 1600 cells, involving up to 44298 time steps.
We test the MOOD method with two different detectors chains: the first one omits the Small Curvature Detector (SCD or plateau detector), whereas the second one uses the full set of detectors as presented in Figure 1. The goal to skip the SCD is to draw a comparison with the MUSCL scheme in similar conditions, since the latter method cannot distinguish between very small variations deriving from the real number truncation and the non-physical oscillations. Notice that the proposed simulation involves a solution which is essentially flat far away from the bump, hence the SCD deactivation could be decisive. In that particular simulation, the MUSCL limiter only involves the conservative variable h since it provides the best results.

Table 1: Total height and velocity L^1- and L^∞-errors and convergence order for the supercritical case: MUSCL scheme.

Nb of Cells	η				u			
	err_1		err_∞		err_1		err_∞	
100	2.44e-03	—	4.76e-02	—	5.40e-03	—	7.91e-02	—
200	5.20e-04	2.2	2.05e-02	1.2	1.12e-03	2.3	3.38e-02	1.2
400	1.12e-04	2.2	5.77e-03	1.8	2.38e-04	2.2	9.52e-03	1.8
800	2.30e-05	2.3	1.47e-03	2.0	4.89e-05	2.3	2.43e-03	2.0
1600	5.43e-06	2.1	3.73e-04	2.0	1.16e-05	2.1	6.15e-04	2.0

Tables 1, 2 and 3 provide the L^1- and L^∞-errors and convergence order for the MUSCL

Table 2: Total height and velocity L^1- and L^∞-errors and convergence order for the supercritical case: MOOD scheme without SCD. The percentage of cells having maximum CPD at $t = T$ is also shown.

Nb of Cells	η				u				Cells with CPD = 1
	err_1		err_∞		err_1		err_∞		
100	1.81e-03	—	2.93e-02	—	5.30e-03	—	8.33e-02	—	73%
200	7.02e-05	4.7	1.22e-03	4.6	1.74e-04	4.9	4.23e-03	4.3	91%
400	8.43e-06	3.1	1.29e-04	3.2	2.10e-05	3.1	2.73e-04	4.0	88%
800	1.45e-06	2.5	2.21e-05	2.5	3.81e-06	2.5	5.12e-05	2.4	100%
1600	2.86e-07	2.3	4.26e-06	2.4	7.84e-07	2.3	1.07e-05	2.3	100%

Table 3: Total height and velocity L^1- and L^∞-errors and convergence order for the supercritical case: full detectors chain. The percentage of cells having maximum CPD at $t = T$ is also shown.

Nb of Cells	η				u				Cells with CPD = 1
	err_1		err_∞		err_1		err_∞		
100	1.91e-03	—	3.15e-02	—	5.06e-03	—	6.88e-02	—	91%
200	5.23e-05	5.2	8.19e-04	5.3	1.24e-04	5.4	1.61e-03	5.4	100%
400	8.49e-06	2.6	1.29e-04	2.7	2.13e-05	2.5	2.73e-04	2.6	100%
800	1.45e-06	2.5	2.21e-05	2.5	3.81e-06	2.5	5.12e-05	2.4	100%
1600	2.86e-07	2.3	4.26e-06	2.4	7.84e-07	2.3	1.07e-05	2.3	100%

scheme, the MOOD scheme without the SCD detector, the MOOD scheme with the full detectors chain, respectively, while we report in the two last tables the percentage of cells that have CPD = 1 at $t = T$ for the MOOD method.
Clearly the MOOD method provides smaller errors and the better convergence order of accuracy. We note that the deactivation of the SCD does not affect the global error since we are dealing with a plateau, *i.e.* the first-order and the second-order schemes provide the same results in that zone. The top of the bump is relaxed by the MOOD Smooth Detector providing the optimal CPD, *i.e.* the approximation is a second-order one in the vicinity of the extremum, whereas the MUSCL strongly cuts the slope leading to significant negative impact on the accuracy.

4.3 Dam break on a wet bed

Having tackled the smooth solution case, we now consider the dam break problem since it involves a shock and therefore comparisons between the two methods can be performed following two criteria: the presence (or not) of oscillations and the number of intermediate cells in the shock. For that, we consider the domain $[0, 50]$ and assume the initial configuration: $\eta(x) = 5$ for $0 \leq x \leq 25$, $\eta(x) = 1$ for $25 < x \leq 50$, $u(x) = 0$ on the whole domain and the bathymetry is flat with $b(x) = 0$. The simulations are carried out for a 100 cells mesh using reflection boundary conditions, and we evaluate the approximation at the final simulation time $T = 3$.

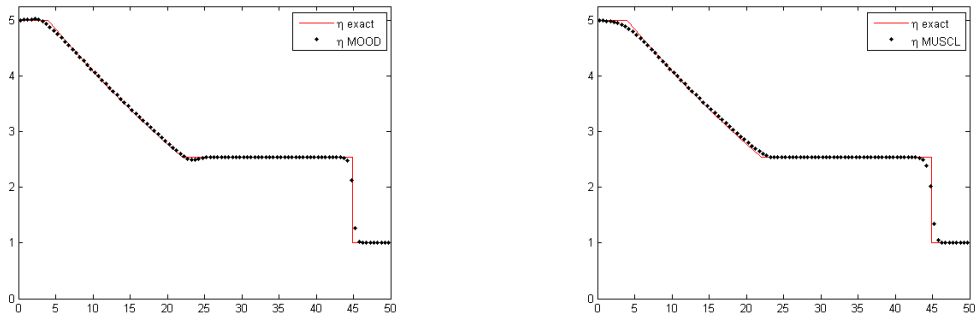


Figure 4: Exact and approximated total height for the dam break case for the MOOD (left) and MUSCL (right) methods with 100 cells.

Figures 4 and 5 present the free surface and the velocity for the MOOD (left) and the MUSCL (right) methods. In both cases, no oscillations are reported and we observe that there are 3 cells in the η-shock for the MUSCL case, whereas the MOOD technique manages to capture the shock within only 2 cells. In the MOOD case, we notice a small numerical artefact at the end of the rarefaction ($x = 22$) since the transition presents a discontinuity of the derivative whereas the CPD map remains equal to one in the vicinity of the transition. The CPD map should be zeroed at that point and the detector fails to see such discontinuity. A new detector should be provided to correctly perform the treatment of such a transition.

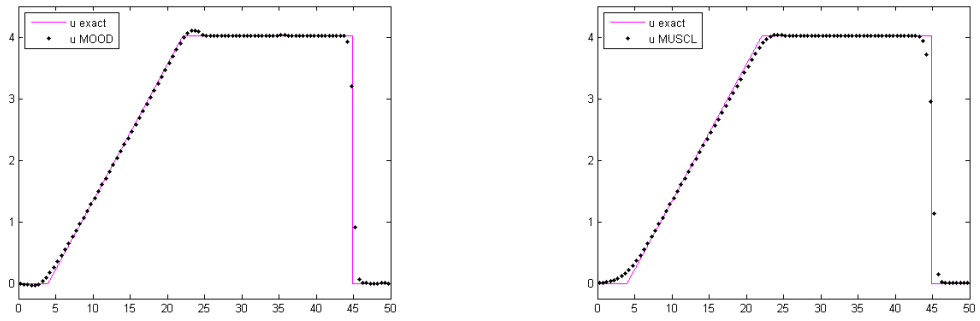


Figure 5: Exact and approximated velocity for the dam break case for the MOOD (left) and MUSCL (right) methods with 100 cells.

When dealing with rough solutions, the convergence order of the errors is not relevant (almost equal to one in our case), but the multiplicative constant is of crucial importance. Indeed, if one assumes a convergence order of the form $\text{err} = C(\Delta x)^{\beta}$, for rough solutions one has $\beta = 1$, but the choice of the scheme may affect the value of C. In Figure 6

we display the L^1-norm convergence curves for the total height and the velocity and observe that the corresponding multiplicative constants are lower for the MOOD method ($C_\eta = 10^{-1.43}$, $C_u = 10^{-1.14}$) with respect to the MUSCL one ($C_\eta = 10^{-1.26}$, $C_u = 10^{-0.97}$).

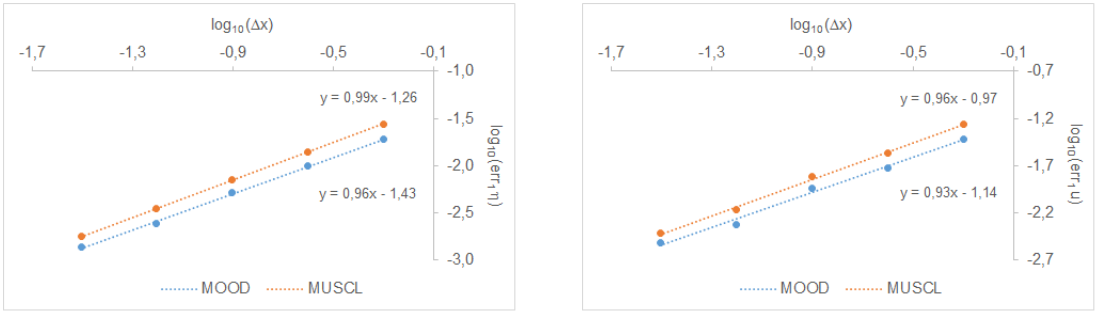


Figure 6: Comparison of the total height (left) and velocity (right) L^1-errors for the dam break case for the MOOD and MUSCL methods.

4.4 Dry/wet simulation

Dry/wet interface approximation is a fundamental issue to be addressed when dealing with coastal problems or flooding. The capacity to provide good approximations of the velocity close to the interface is crucial for the applications since the impact of waves or flooding is deeply linked to the kinetic energy or the friction force associated to the flow velocity. We propose here two representative test cases, namely a smooth bathymetry situation which corresponds to a coastal problem and a discontinuous bathymetry representing a wave impact on a wall.

4.4.1 Smooth bathymetry

In this simulation we consider the domain $[0, 50]$ and choose a smooth bathymetry function given by $b(x) = 0$ for $0 \leq x \leq 20$, $b(x) = 0.15(x - 20)$ for $20 < x \leq 50$. The initial configuration concerning the total height is $\eta(x) = 5$ for $0 \leq x \leq 25$, $\eta(x) = b(x)$ for $25 < x \leq 50$. The initial velocity, the boundary conditions and the number of cells are the same considered in the previous test. The final simulation time is $T = 1.5$.
We plot in Figures 7 and 8, respectively, the free surface and the velocity, for the first-order scheme with 10 000 cells as a reference solution, as well as for the approximations with the MOOD and MUSCL methods. We report that the MOOD method has a small over-estimated water height close to the dry/wet interface, but the velocity is very well approximated. On the contrary, the MUSCL method provides smoother water height

close to the interface but the velocity is over-estimated. The figures show that the MOOD technique manages to correctly handle the interface.

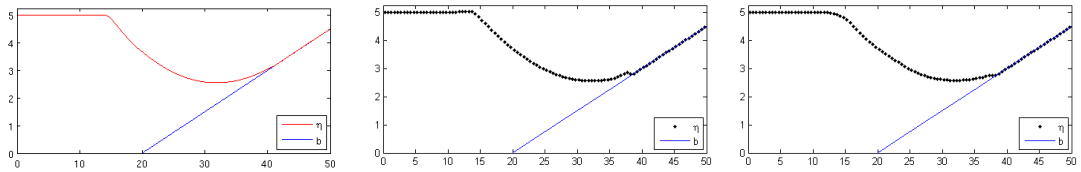


Figure 7: Total height with the first-order (left with 10 000 cells), the MOOD (middle) and MUSCL (right) method for the dry/wet case with smooth bathymetry (100 cells).

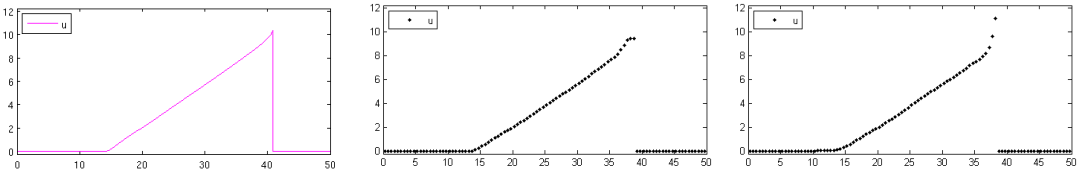


Figure 8: Velocity with the first-order (left with 10 000 cells), the MOOD (middle) and MUSCL (right) method for the dry/wet case with smooth bathymetry (100 cells).

4.4.2 Discontinuous bathymetry

In this simulation we consider the domain $[0, 50]$ and choose a discontinuous bathymetry function given by $b(x) = 0$ for $0 \leq x \leq 35$, $b(x) = 3 + 0.125(x - 35)$ for $35 < x \leq 50$. The initial configuration concerning the total height and velocity, as well as the boundary conditions and the number of cells are the same considered in dam break test. The final simulation time is $T = 2.75$. Numerical simulations are carried out where the first-order case is calculated with 10 000 cells to provide a reference solution. Figures 9 and 10 present, respectively, the free surface and the velocity for the first-order scheme and the approximations with the MOOD and MUSCL methods. We report that the MOOD method provides a slightly sharper shock in the transition with respect to the MUSCL method. Moreover, the velocity is well approximated with the MOOD case and over-estimated with the MUSCL technique. As in the smooth case, a small over-estimation of the water height occurs with the MOOD method.

5 CONCLUSIONS

We propose in the present work a comparison between the MOOD and the MUSCL methods to achieve second-order approximation of the shallow water equations. We report

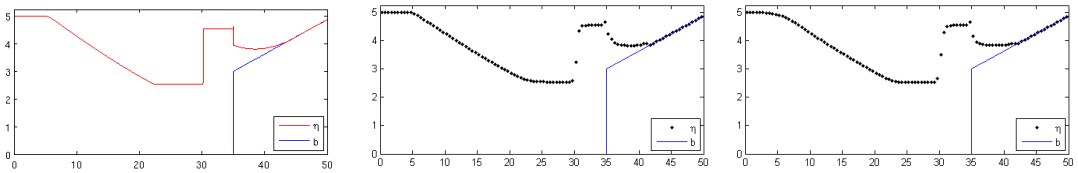


Figure 9: Total height with the first-order (left with 10 000 cells), the MOOD (middle) and MUSCL (right) method for the dry/wet case with discontinuous bathymetry (100 cells).

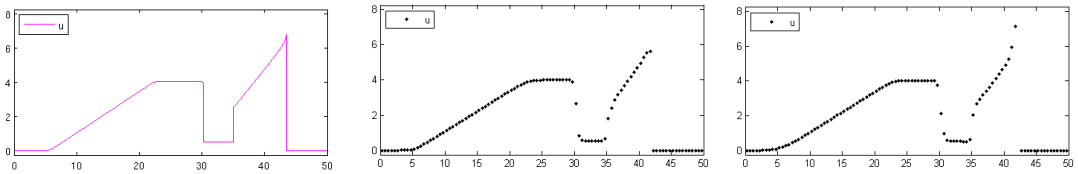


Figure 10: Velocity with the first-order (left with 10 000 cells), the MOOD (middle) and MUSCL (right) method for the dry/wet case with discontinuous bathymetry (100 cells).

that the MOOD method is less diffusive regarding to the MUSCL one and manages to treat very well the non-conservative term. Critical situations such as dry/wet interface with smooth or discontinuous bathymetry are also well treated by the MOOD methodology. Nevertheless, new detectors have to be proposed to overcome the small over-estimation of the water height in the case of smooth ramps.

ACKNOWLEDGEMENTS

This research was financed by FEDER Funds through Programa Operational Fatores de Competitividade — COMPETE and by Portuguese Funds FCT — Fundação para a Ciência e a Tecnologia, within the Projects PEst-C/MAT/UI0013/2014, PTDC/MAT/121185/2010 and FCT-ANR/MAT-NAN/0122/2012.

REFERENCES

[1] Audusse, E., Bouchut, F., Bristeau, M. O., Klein, R., Perthame, B., "A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows", *SIAM J. Sci. Comput.* Vol. **25**, pp. 2050-2065, 2004.

[2] Berthon, C., Desveaux, V., "An entropy preserving MOOD scheme for the Euler equations", *Int. J. finite volumes* Vol. **11**, pp. 1-39, 2014.

[3] Bermúdez, A., Vázquez , M. E., "Upwind methods for hyperbolic conservation laws with source terms", *Computers & Fluids* **24**, pp. 1049-1071, 1994.

[4] Buffard, T. , Clain, S., "Monoslope and multislope MUSCL methods for unstructured meshes", *J. Comput. Phys.* Vol. **229**, pp. 3745-3776, 2010.

[5] Clain, S., Diot, S., Loubère, R., "A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD)", *J. Comput. Phys.* Vol. **230(10)**, pp. 4028-4050, 2011.

[6] Diot, S., Clain, S., Loubère, R., "Improved Detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials", *Comput. & Fluids* Vol. **64**,pp. 43-63, 2012.

[7] Clain, S., Figueiredo, J., "The MOOD method for the non-conservative shallow water system", preprint HAL hal-01077557 (2014), submitted.

[8] Diot, S., Loubère, R., Clain, S., "The MOOD method in the three-dimensional case: Very high-order finite volume method for hyperbolic systems", *Int. J. Numer. Meth. Fluids* Vol. **73**, pp. 362-392, 2013.

[9] Duran, A., Liang, Q., Marche, F., "On the well-balanced numerical discretization of shallow water equations on unstructured meshes", *J. compt. phys.* Vol. **235**, pp. 565-586, 2013.

[10] Noelle, S., Pankratz, N., Puppo, G., Natvig, J. R., "Well-balanced inite volume schemes of arbitrary order of accuracy for shallow water flows", *J. Comput. Phys.* Vol **213**, pp. 474-499, 2006.

[11] Noelle, S., Xing, Y., Shu, C.-W., "High-order well-balanced finite volume WENO schemes for shallow water equations with moving water", *J. Comput. Phys.* Vol **226**, pp. 29-58, 2007.

[12] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3^{rd} revision, Springer-Verlag Berlin and Heidelberg GmbH & Co. K 2009.

[13] Delestre, O., Lucas, C., Ksinant, P.-A., Darboux, F., Laguerre, C., Vo,T.-N.-T., James, F., Cordier, S., "SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies", *Int. J. Numer. Meth. Fluids* Vol. **72** pp. 269-300, 2013.

[14] Wijetunge, J. J., "Numerical simulation of the 2004 indian ocean tsunami: case study of effect of sand dunes on the spatial distribution of inundation in Hambantota, Sri Lanka", *J. Appl. Fluid Mech.* Vol. **3**, pp. 125-135, 2010.

[15] Zhou, J. G., Causon, D. M., Mingham, C. G., Ingram, D. M., "Numerical solutions of the shallow water equations with discontinuous bed topography", *Int. J. Numer. Meth. Fluids* Vol. **38**, pp. 769-788, 2002.

SYMCOMP 2015
Faro, March 26-27, 2015

A LANCZOS TAU METHOD SOFTWARE LIBRARY FOR THE SOLUTION OF DIFFERENTIAL EQUATIONS

M. S. Trindade[1*], José M. A. Matos[2] and Paulo B. Vasconcelos[3]

1: Departamento de Matemática, Faculdade de Ciências da Universidade do Porto
Porto, Portugal
e-mail: marcelo.trindade@fc.up.pt

2: Instituto Superior de Engenharia do Porto and Centro de Matemática da Universidade do Porto
Porto, Portugal
e-mail: jma@isep.ipp.pt

3: Faculdade de Economia and Centro de Matemática da Universidade do Porto
Porto, Portugal
e-mail: pjv@fep.up.pt

Keywords: Tau method, preconditioners, orthogonal polynomials, diferential equations

Abstract. *This work aims to build a numerical software library based on the Tau method to solve ordinary, partial and integro-differential equations, using MATLAB. The Tau method can be very effective in the solution of certain type of problems and, therefore, the existence of a numerical library to disseminate it is of major importance. Furthermore, the method as been used for the solution of particular problems but has not yet been explored as a technique to solve general problems. Focus on this paper will be on stability issues, namely those issued from the solution of linear systems required for the process. Preconditioners for the solution with the Tau method will be tackled, with emphasizes on incomplete LU factorizations and more simple block diagonal ones along with nonstationary iterative solvers, such as GMRES or BICGSTAB. Numerical results will enlighten the reduction in condition number and thus on better approximations without compromising the time for computation. Indeed, quite often, a reduction in computation time is also achieved.*

1 Introduction

Many natural phenomenons, physical, chemical, biological, as well as social, economics for instance, relate changes rate between one or more variables, and they can be interpreted with mathematical models founded on differential equations. Usually a closed solutions does not exist or it is very difficult to develop, so one must resort to numerical methods. In this work we explore the Tau method, a numerical technique to approximate, by orthogonal polynomials, the solution of a differential system of equations. Since it is a spectral method, its efficiency results from the spectral convergence, which is characteristic of this class of methods.

The Tau method was originally proposed by C. Lanczos in 1938 [4] and Ortiz [12] has made an important contribution by developing a matricial version, the operational formulation, enabling to write the relations between the boundary conditions together with the differential equations, expressed in an orthogonal polynomial basis. The solution, that is the coefficients of the polynomials, can be obtained solving a truncated (infinite) algebraic linear system. Since then, several studies applying the Tau method have been performed to approximate the solution of differential linear and non-linear equations [1, 3, 10], partial differential equations [8, 9, 11] and integro-differential equations [5]. Nevertheless, in these works the Tau method is tuned for the approximation of a specific problem.

The method attempts to solve $Dy(x) = f(x)$ by including a perturbation applied to the second member. Thus, this perturbation results by truncating an infinity linear algebraic system which expresses the relations between the polynomial solution coefficients and the differential equation. Both an approximate solution and the second member perturbation are a linear combinations of orthogonal polynomials P_n.

If we are interested in solving an ordinary or partial differential equation to high accuracy on a simple domain, and if the data defining the problem are smooth, then spectral methods are usually the best tool. Spectral methods, with respect to the more usual finite differences or finite elements, allows to achieve high order accuracy [14].

In the present paper, the spectral Tau method is offered as an automatic solver to compute an approximate solution to linear ordinary differential equations. In section 2, the operational formulation is presented; in section 3, the preconditioning techniques applied to the associated linear systems are explored; in section 4, a framework in MATLAB is introduced and some numerical results illustrating the method are shown; finally the last section concludes.

2 Operational formulation

Let us begin by showing how the Tau method works, using a generic problem (1) on a finite interval $J = [a, b]$, where D is the differential operator, m is the higher order derivative, $p_r(x)$ are the polynomials coefficients, $f(x)$ is a polynomial or a polynomial approximation of a function in $[a, b]$, $y(x)$ is the solution function sought and (3) are the boundary conditions.

$$Dy(x) = f(x), \quad a < x < b, \tag{1}$$

$$D = \sum_{r=0}^{m} p_r(x) \frac{d^r}{dx^r}, \tag{2}$$

$$g_j(x_j) = \phi_j, \quad j = 1, \ldots, m. \tag{3}$$

To build polynomials approximations $y_n(x)$ of the solution $y(x)$ of the problem (1) Lanczos' Tau method, as interpreted by [7] and [12], starts by considering η and μ matrices, such that both lead to respectively

$$a\eta P = \frac{d}{dx} y \tag{4}$$

and

$$a\mu P = xy, \tag{5}$$

where P is a sequence of orthogonal polynomials and $y = aP$ is the polynomial whose coefficients in the base P are the entries of the vector a. If P are Chebyshev (7) or Legendre (9) polynomials, respectively

$$\begin{cases} T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \\ T_0(x) = 1, \quad T_1(x) = x \end{cases} ; \tag{6}$$

$$\mu_T = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots \\ 1/2 & 0 & 1/2 & 0 & \ldots \\ 0 & 1/2 & 0 & 1/2 & \ldots \\ 0 & 0 & 1/2 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{and} \quad \eta_T = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots \\ 1 & 0 & 0 & 0 & \ldots \\ 0 & 4 & 0 & 0 & \ldots \\ 3 & 0 & 6 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{7}$$

and

$$\begin{cases} L_{n+1}(x) = \frac{(2n+1)x}{n+1} L_n(x) - \frac{n}{n+1} L_{n-1}(x) \\ L_0(x) = 1, \quad L_1(x) = x \end{cases} ; \tag{8}$$

$$\mu_L = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots \\ 1/3 & 0 & 2/3 & 0 & \dots \\ 0 & 2/5 & 0 & 3/5 & \dots \\ 0 & 0 & 3/7 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{and} \quad \eta_L = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & \dots \\ 0 & 3 & 0 & 0 & \dots \\ 1 & 0 & 5 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{9}$$

Matrix η is related with the derivatives of P, and the multiplication of μ matrix with P results in an increase of one polynomial degree in all of P terms. Let $n \in \mathbb{N}$ be a positive integer ans suppose that we are looking for y_n, an n degree polynomial approximation of y, the exact solution of (1), the operational formulation leads to the solution of the linear system (10)

$$\Gamma a = b, \tag{10}$$

where

$$\Gamma = \left[B; \Pi\right]^T, \tag{11}$$

$$\Pi = \sum_{r=0}^{m} \eta^r p_r(\mu), \qquad p_r(\mu) = \sum_{k=0}^{n_r} p_{r,k}\mu^k, \quad \text{if} \quad p_r(x) = \sum_{k=0}^{n_r} p_{r,k}x^k. \tag{12}$$

Dirichlet, Neumann or mixed boundary conditions can be used as well. For illustration purposes, if B follows Neumann conditions (to increasing derivatives), d_i is the P derivative of order i, then

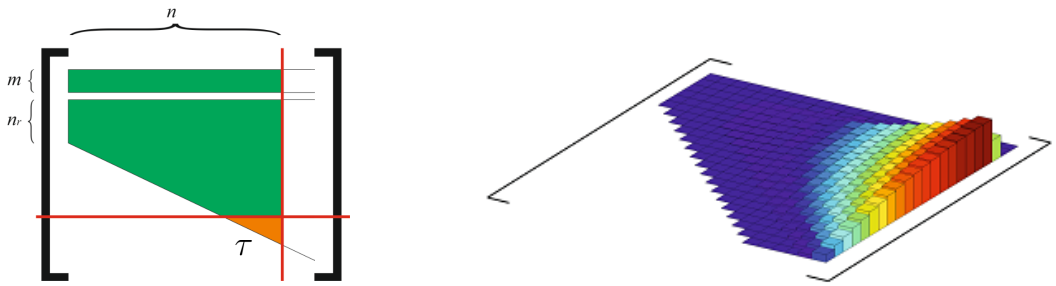
$$B = [\beta_{ij}]^{(n+1)\times m} = \begin{bmatrix} P_0^{(d_0)}(x_1) & P_0^{(d_1)}(x_2) & \dots & P_0^{(d_{m-1})}(x_m) \\ P_1^{(d_0)}(x_1) & P_1^{(d_1)}(x_2) & \dots & P_1^{(d_{m-1})}(x_m) \\ \vdots & \vdots & \ddots & \vdots \\ P_n^{(d_0)}(x_1) & P_n^{(d_1)}(x_2) & \dots & P_n^{(d_{m-1})}(x_m) \end{bmatrix}, \tag{13}$$

where a and b are, respectively, the sought polynomial coefficients in $y_n(x) = aP$, which is an approximate solution of (1), and the independent vector written in the same polynomial basis:

$$a = [a_0, a_1, \cdots, a_n]^T \qquad \text{and} \quad b = [\phi_1, \phi_2, \cdots, \phi_m, f_0, f_1, \cdots, f_l, 0, 0]^T, \tag{14}$$

where l is the deegre of f in (1) and $n \geq m + l$.
The Γ matrix, indicated in figure 1, has a trapezoidal form.
The above formulation is the "entire" way to apply the Tau method, i.e. getting an approximate polynomial solution to the whole domain. However there is another way to do it, the called "step-by-step" version, which consists in to define k subintervals of the interval $J = [a, b]$, not necessarily of equal length, obtaining

$$j_i = [a_i, b_i], \quad i = 1, ..., k, \tag{15}$$



(a) Γ matrix format (b) Γ matrix values

Figure 1: Γ matrix

such that

$$a = a_1 < b_1 = a_2 < \ldots < b_k = b. \tag{16}$$

Thus, the entire way is applied k times and each one give us one polynomial solution in j_i subinterval. Step-by-step Tau version can give betters results, once that with its is possible to use lower polynomial order than with the entire way.

3 Preconditioning of linear systems

This section has the goal to highlight on the importance of preconditioning, since matrix $\Gamma = [B; \Pi]^T$ is usually ill-conditioned.

Definition 1: Condition number of a matrix A is a scalar such that

$$cond(A) = ||A|| \cdot ||A^{-1}|| = \left(\max_{x \neq 0} \frac{||Ax||}{||x||} \right) \cdot \left(\min_{x \neq 0} \frac{||Ax||}{||x||} \right)^{-1}. \tag{17}$$

where $||A||$ is a matrix norm defined by

$$||A||_p = \max_{x \neq 0} \frac{||Ax||_p}{||x||_p}. \tag{18}$$

The condition number of a matrix measures how close it is to singularity (for rectangular matrices it measures closeness to rank deficiency). High condition numbers lead to ill-conditioned problems while lower ones lead to well-conditioned problems (being 1 the best possible) [2].

Preconditioning consists at making changes in the original system $\Gamma a = b$, multiplying by matrices M and Q, called preconditioners, while keeping the solution a unchanged. Often, without preconditioning a numerical method may not be able to deliver an approximate solution. Furthermore, quite often, preconditioned linear system of equations can deliver faster convergence. To this end, the possibilities for preconditioning are shown in equation (19)

$$(M^{-1}\Gamma Q)(Q^{-1}a) = M^{-1}b, \tag{19}$$

where this modified system can now have $Q = I$ (I is identity matrix) and $M \approx \Gamma$. Then the idea is to find M such that

$$cond(M^{-1}\Gamma) << cond(\Gamma). \tag{20}$$

Two class of preconditioners are considered in this work:

3.1 ILU factorization

The incomplete LU factorization (ILU) is based in the scheme L (Lower triangular) and U (Upper triangular) but rejecting some $l_{i,j}$ or $u_{i,j}$ fills according to a dual-dropping strategy: a dropping threshold and a limit on the amount of fill. This process can be performed keeping the sparsity format of Γ. For apply the ILU was used the following MATLAB code (Algorithm 3.1).

Algorithm 3.1: ILU factorization

```
---------------------------- ILU FACTORIZATION ----------------------------
setup.type = 'nofill';                                %TYPE OF FACTORIZATION
setup.droptol = '0.1';                %THE DROP TOLERANCE OF INCOMPLETE LU
setup.milu = 'off';                                  %MODIFIED INCOMPLETE LU
setup.udiag = '0';                     %REPLACE ZEROS ON THE DIAGONAL OF U
setup.thresh = '1';                                    %THE PIVOT THRESHOLD
[L,U]=ilu(sparse(GAMMA),setup);                           %BUILDING L AND U
I=eye(dim);
%---------------- BUILDING M^-1 FROM L AND U MATRICES -------------------%
for exe=1:dim
    B=I(:,exe);
    %------------------- PROGRESSIVE CHANGING FOR L ---------------------%
    d=zeros(dim,1); d(1)=B(1)/L(1,1);                    %FOR THE FIRST ONE
    for i=2:dim
        d(i)=(B(i)-L(i,1:i-1)*d(1:i-1))/L(i,i);              %FOR ANOTHERS
    end
    %------------------- RETROATIVE CHANGING FOR U ----------------------%
    X=zeros(dim,1); X(dim)=d(dim)/U(dim,dim);             %FOR THE LAST ONE
    for i=dim-1:-1:1
        X(i)=(d(i)-U(i,i+1:end)*X(i+1:end))/U(i,i);          %FOR ANOTHERS
    end
    M(:,exe)=X;                                        %WRITE PRECONDITIONER M
end
---------------------------------------------------------------------------
```

3.2 Diagonal preconditioner

Often a simple diagonal preconditioner is enough to reduce the condition number (improve spectral properties) of the coefficient matrix. Algorithm 3.2 illustrates how to build a M preconditioner with nD diagonals.

Algorithm 3.2: Diagonal preconditioner

```
------------------------- DIAGONAL PRECONDITIONER -----------------------
M=diag(diag(A));                                                 %IF nD IS ONE
if nD>1                                                  %ELSE WE NEED TO BUILD IT
   M=M+triu(A,1)-triu(A,nD)+...
       tril(A,-1)-tril(A,-nD);                        %BUILDING M PRECONDITIONER
end
Minv=M^-1;                                          %BUILDING M^-1 PRECONDITIONER
-------------------------------------------------------------------------
```

Algorithm 3.2 offers M^{-1} mainly for clarity and functionality. Most efficient implementations of the linear solvers have built-in mechanisms to work with matrix M avoinding the explicit computation of the inverse.
The solution of the linear system $\Gamma a = b$, in turn, can be obtained by different kind of linear system solvers, like GMRES, BICG, BICGSTAB, BICGSTABL, MINRES, QMR, TFQMR, LSQR. All of them can be preconditioned. A detailed presentation on non-stationary iterative methods can be found in [13].

4 Framework

To build the MATLAB framework for the Tau Lanczos method a GUI, Graphical User Interface, is being developed. For the moment the library tackles linear differential equations with polynomial coefficients (see Figure 2), applying entire and the step-by-step version of Tau method. Several polynomials bases are offered: Chebyshev, Legendre, Hermite and Laguerre.
Flowchart in Figure 3 shows the main functions based in symbolic computation and numerical methods developed for this work.
So, the framework starts with opening the GUI (figure 2), where all instructions are placed on it. For to get them, like interval of solution $[a, b]$, the polynomial order $[dim]$ and showing step *step*, was used the function (21).

$$newvariable = str2num(get(handles.nameonguide,'String')); \tag{21}$$

After the decision by entire Tau method, the application follows to get the initial conditions vector IC using the same way (function (22)), and then follows to the type of polynomial informed.

$$IC = str2num(get(handles.IC,'String')); \tag{22}$$

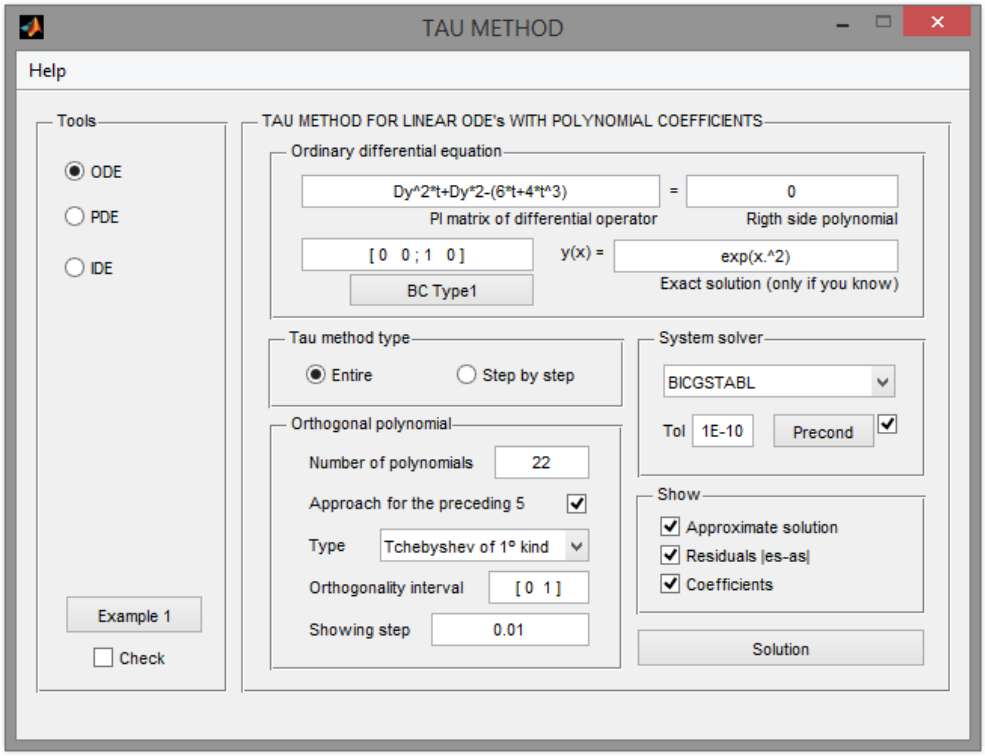


Figure 2: GUI for linear ODEs.

As an example, algorithm (4.1) illustrates how to write the Legendre polynomials showed at equation (9).

Algorithm 4.1: Legendre polynomials

```
---------------------- BUILDING LEGENDRE POLYNOMIALS --------------------
dim=dim+1; P=zeros(dim); P(1,1)=1; P(2,2)=1;      %dim IS POLYNOMIAL ORDER
for n=2:dim-1
    N=n-1;
    P(n+1,1)=-(N/(N+1))*P(n-1,1);
    P(n+1,2:n+1)=((2*N+1)/(N+1))*P(n,1:n)-(N/(N+1))*P(n-1,2:n+1);
end
-----------------------------------------------------------------------
```

After this specification of the basis, the user can change the interval of orthogonality $[a, b]$.

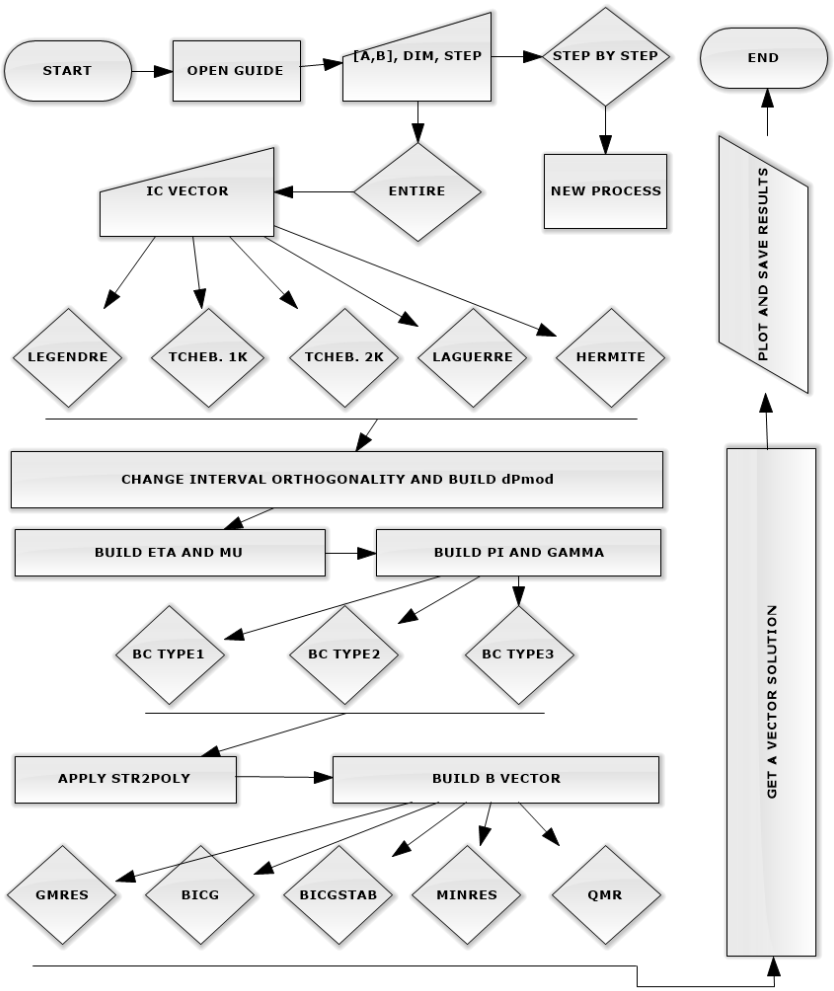


Figure 3: Flowchart to entire Tau method for ODEs.

For this propose, formula (23) is used, showed by [6] to build $Pmod$ (see Algorithm (4.2)).

$$P_n(x) = \sum_{m=0}^{n} a_m x^m, \qquad Pmod_n(x) = P_n\left(\frac{2x-(a+b)}{b-a}\right) = \sum_{m=0}^{n} a_m^* x^m. \tag{23}$$

Algorithm 4.2: Interval of orthogonality

```
---------- CHANGE POLYNOMIALS TO AN ORTHOGONAL INTERVAL IN [ab] ----------
%x\in[a,b]          %FOLOWING MASON J. C. f((2x-(a+b))/(b-a)) = (t1x+t2)^n
```

```
if a==-1 && b==1                    %IT MUST CHANGE ONLY IF a~=-1 AND b~=1
    Pmod=P;
else
    t1=2/(b-a);                                     %FOLOWING MASON J. C.
    t2=(-a-b)/(b-a);                                %FOLOWING MASON J. C.
    Pmod=zeros(size(P));        %MEMORY ALLOCATION FOR MODIFIED POLYNOMIAL
    PASC=abs(pascal(dim,1));
    %---------------------- CODE FOR NEWTON BINOMIAL --------------------%
    for lin=1:dim                       %FOR EACH LINE OF POLINOMIAL MATRIX
        temp=zeros(1,lin);          %MEMORY ALLOCATION FOR TEMPORAY VECTOR
        for n=1:lin-1                    %FOR EACH COEFFICIENT OF THE LINE
            coef=zeros(1,n+1);     %MEMORY ALLOCATION FOR THE COEFFICIENTS
            if P(lin,n+1)~=0        %IF SOME COEFFICIENT IS 0, DON'T APPLY
                for i=0:n                   %NEWTON BINOMIAL FOR n DEGREE
                    coef(i+1)=PASC(n+1,i+1)*t1^(n-i)*t2^(i);
                end
                temp(1:n+1)=temp(1:n+1)+P(lin,n+1)*coef(n+1:-1:1);
            end
        end
        Pmod(lin,1:n+1)=temp;  %WRITING THE NEW TCHEBYCHEV MATRIX IN [a b]
    end
    Pmod(:,1)=Pmod(:,1)+P(:,1);      %ADDING THE FIRST COLUM OF POLINOMIAL
end
-------------------------------------------------------------------------
```

This routine finishes by calculating the derivatives $dPmod$ of the polynomial basis $Pmod$. A matrix block with size $dim \times dim \times k$ is used, where dim is the polynomial order and k is the derivative maximal order of $Pmod$ (see Algorithm (4.3)).

Algorithm 4.3: Building dPmod

```
------- BUILDING THE DERIVATES OF ORTHOGONALS POLYNOMIALS IN [a b] -------
dPmod=zeros(dim,dim,K);         %MEMORY ALlOCATION FOR K DERIVATES OF Pmod
for k=1:lim            %BUILDING DERIVATES OF Pmod = d^nPmod FOR n=1,...,K
    for i=1:dim
        for j=2:i                           %FOR EACH ROW AND EACH COLUMN
            dPmod(i,j-1,k)=dPmod(i,j,k-1)*(j-1);    %BUILDING THE DERIVATE
        end
    end
end
-------------------------------------------------------------------------
```

The next routine builds the η and μ matrices showed in equation (4). In [7] it was proved that η and μ can be written as

$$\eta = \begin{bmatrix} 0 & & & & \\ \eta_{10} & 0 & & & \\ \eta_{20} & \eta_{21} & 0 & & \\ & & \ddots & & \\ \eta_{n0} & \eta_{n1} & \eta_{32} & \dots & 0 \end{bmatrix} \quad \text{e} \quad \mu = \begin{bmatrix} \mu_{00} & \mu_{01} & & & \\ \mu_{10} & \mu_{11} & \mu_{12} & & \\ & \mu_{21} & \mu_{22} & \mu_{23} & \\ & & \ddots & & \\ & & & \mu_{nn-1} & \mu_{nn} \end{bmatrix}, \tag{24}$$

where, for $i = 1 : n$, the $\eta_{i,j}$ e $\mu_{i,j}$ are found by

$$\begin{cases} \eta_{i+1,j} = \alpha_i^{-1}\big[\alpha_{j-1}\eta_{i,j-1} + (\beta_j - \beta_i)\eta_{i,j} + \gamma_{j+1}\eta_{i,j+1} - \gamma_i\eta_{i-1,j}\big], j = 0 : i-1 \\ \eta_{i+1,i} = \alpha_i^{-1}(\alpha_{i-1}\eta_{i,i-1} + 1) \\ \eta_{i,0} = v_1, \quad \eta_{i,1} = v_2 \quad and \quad \eta_{0,j} = v_3 \end{cases} \tag{25}$$

and

$$\begin{cases} \mu_{i,i} = \beta_{i-1} \\ \mu_{i,i+1} = \alpha_{i-1} \\ \mu_{i+1,i} = \gamma_{i-1} \end{cases}, \tag{26}$$

where α, β and γ are the coefficients of the recurrence relation of the orthogonal polynomials used, as showed in (27)

$$\begin{cases} xP_n = \alpha_n P_{n+1} + \beta_n P_n + \gamma_n P_{n-1}, \quad n \geq 0 \\ P_0 = 1, \quad P_{-1} = 0 \end{cases}. \tag{27}$$

Algorithm 4.6: Building Π and Γ

```
-------------------------------------------------------------------------
t=mu; Dy=eta; y=eye(exe);                          % DEFFINING THE MATRICES
PI=eval(get(handles.equation,'String'));              % WRITING PI MATRIX
GAMMA=zeros(size(PI));                % MEMORY ALLOCATION FOR GAMMA MATRIX
GAMMA(:,K+1:end)=PI(:,1:end-K);              % K COLUMS IS NECESSARY TO IC
-------------------------------------------------------------------------
```

The user can select the type of boundary conditions through button "BC Type": in case of "BC Type1", the following conditions are chosen

$$Y(X_0) = Y_0, Y'(X_1) = Y_1, Y''(X_2) = Y_2...; \tag{28}$$

In case of "BC Type2", Dirichlet conditions are selected

$$Y(X_0) = Y_0, Y(X_1) = Y_1, Y(X_2) = Y_2..., \tag{29}$$

and finally, for "BC Type3" specific boundary conditions are defined by the user.

For Type1 and Type2 cases, the matrix

$$IC = [x_0, ..., x_n; y_0, ...y_n] \tag{30}$$

is created, with size ([-,K]=size(IC)), where "K" is equal to the order of the differential equation. For "Type3", an additional row containing, in its elements, the order "d" of each derivative of boundary conditions is created ($r_k = 0$ means $y(x_k) = y_k$ and $r_k = d$ means $y^{(d)}(x_k) = y_k$), then

$$IC = [x_0, ..., x_n; y_0, ...y_n; r_0, ..., r_n]. \tag{31}$$

Thus, for each one of the three types, B part of Γ matrix can be built, respectively by, algorithm (4.5), (4.6) and (4.7).

Algorithm 4.5: Type1 boundary condition

```
----------------- TYPE:(Y(X0)=Y0,Y'(X1)=Y1,Y''(X2)=Y2...) ----------------
for i=1:length(GAMMA)
    GAMMA(i,1)=polyval(Pmod(i,i:-1:1),IC(1,1));              % FOR Y(X0)=Y0
    for ind=2:K                                        % FOR Y^(ind)(Xind)=Yind
        GAMMA(i,ind)=polyval(dPmod(i,i:-1:1,ind-1),IC(1,ind));
    end
end
-------------------------------------------------------------------------
```

Algorithm 4.6: Type2 boundary condition

```
------------------ TYPE:(Y(X0)=Y0,Y(X1)=Y1,Y(X2)=Y2...) ------------------
for i=1:length(GAMMA)
    for ind=1:K
        GAMMA(i,ind)=polyval(Pmod(i,i:-1:1),IC(1,ind));
    end
end
-------------------------------------------------------------------------
```

Algorithm 4.7: Type3 boundary condition

```
------------------------ USER DECIDE ABOUT IT ---------------------------
for i=1:length(GAMMA)
    for ind=1:K
        if IC(3,ind)==0
            GAMMA(i,ind)=polyval(Pmod(i,i:-1:1),IC(1,ind));
```

```
        elseif IC(3,ind)~=0
            GAMMA(i,ind)=polyval(dPmod(i,i:-1:1,IC(3,ind)),IC(1,ind));
        end
    end
end
-------------------------------------------------------------------------
```

To deal with the right-hand side, the library must translate polynomial expressions to arrays; this is achieved with srt2poly function. Thus, b is built following Algorithm (4.8).

Algorithm 4.8: Building b vector

```
------------------------ BUILDING THE B VECTOR -------------------------
if str2double(get(handles.RS,'String'))==0  % IF HOMOGENEOUS ODE JUST JUMP
else                              % WRITE THE POLYNOMIAL IN THE CURRENT BASIS
    rside=get(handles.RS,'String');   % GETTING THE RIGHT SIDE (RS) OF ODE
    if isempty(str2num(rside))==1        % IF THE (RS) ISN'T ONLY A NUMBER
        P=str2poly({rside},'t');                  % WRITE IT LIKE A VECTOR
        p=zeros(max(P{1,1}(:,2))+1,1);         % REARRANGING TO WANTED WAY
        [rm,~]=size(P{1,1});
        for rk=1:rm
            ind=P{1,1}(rk,2);
            p(ind+1)=p(ind+1)+P{1,1}(rk,1);
        end
    else                                    % IF THE (RS) IS ONLY A NUMBER
        p(1)=str2double(rside);
    end
    [gsp,~]=size(p);
    for fi=gsp+1:exe
        p(fi,1)=0;
    end
    BC=Pmod(1:exe,1:exe)^-1;
    A=p'*BC;
    b=zeros(length(GAMMA),1);            % MEMORY ALLOCATION FOR b VECTOR
    b(1:K)=IC(2,1:K);
    b(K+1:exe)=A(1:end-K)';
end
-------------------------------------------------------------------------
```

Finally, the user only needs to select the iterative solver and the preconditioner. For all parameters and choices, a default option is always provided.

4.1 Example 1

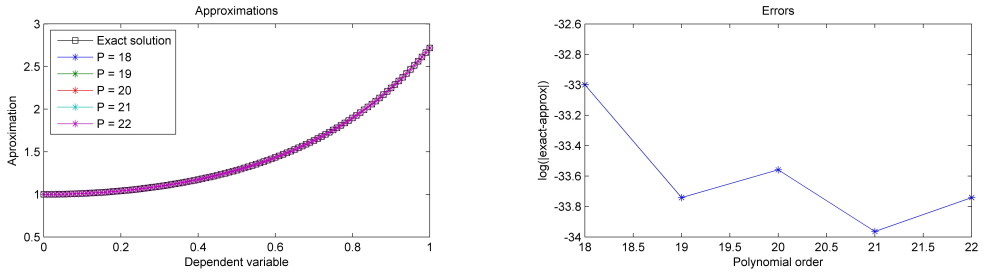
Consider the solution of

$$\begin{cases} ty''(x) + 2y'(x) - (6t + 4t^3)y(x) = 0, \quad x \in [0,1] \\ y(0) = 1, \quad y'(0) = 0 \end{cases} . \tag{32}$$

The user has only to provide

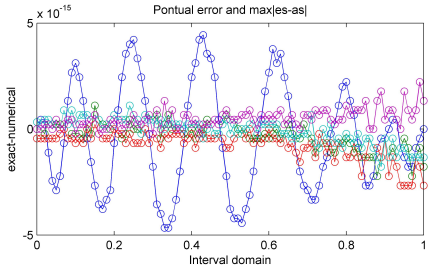
$$Dy^2 * t + Dy * 2 - (6 * t + 4 * t^3) = 0, \quad BC = [0 \quad 0; 1 \quad 0]. \tag{33}$$

The approximation solution will be presented for five consecutive polynomial order, as in Figure 4.



(a) Approximations

(b) Maximal error for each case



(c) Punctual error

Figure 4: Numerical approximatuions using Tau library

4.2 Example 2

In this example a comparison between the proposed implementation of the Tau method and the fourth order Runge-Kutta method is considered, both applied to the solution of

$$\begin{cases} y'(x) = y(x), \quad x \in [0,3] \\ y(0) = 1 \end{cases} . \tag{34}$$

The time spent by the Tau method, using Tchebyshev polynomials of degree 22, was 0.02s with precision $2.8 \cdot 10^{-14}$, while using Runge-Kutta, 8.41s to reach a precision of $1.2 \cdot 10^{-12}$ was required, with a step size of 10^{-6} (see figure 5).

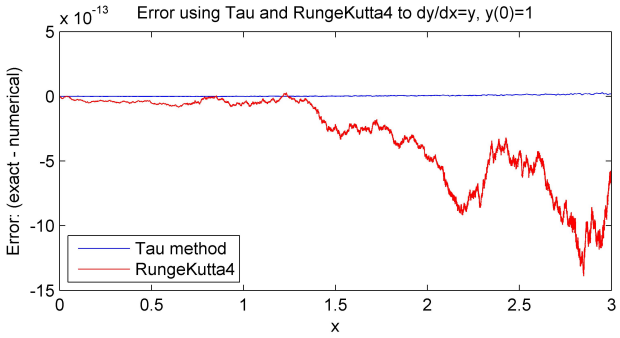


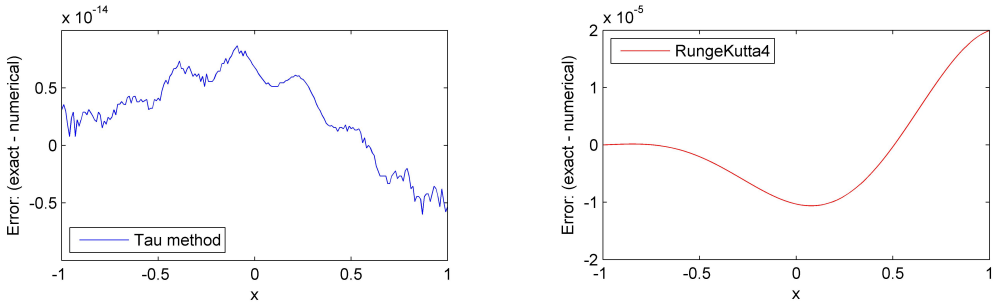
Figure 5: Error using Tau and 4^{th} order Runge-Kutta to $dy/dx = y$, $y(0) = 1$

4.3 Example 3

In this example, we have a comparison solving the problem

$$\begin{cases} d^2y/dx^2 + \pi^2 y = x^2 - 5x^4, & x \in [-1, 1] \\ y(-1) = -1, \quad y'(-1) = \pi \end{cases} . \tag{35}$$

Again Tau method was faster than Runge-Kutta of order 4: Tau with Legendre polynomials of degree 22 and BICGSTAB with ILU preconditioner, took 0.01s to achieve a precision of $7.5 \cdot 10^{-15}$, while Runge-Kutta required 13.99s for a precision of $1.9 \cdot 10^{-5}$ (step-size of 10^{-6} (see figure 6).



(a) Error using Tau

(b) Error using 4^{th} order Runge-Kutta

Figure 6: Error to $d^2y/dx^2 + \pi^2 y = x^2 - 5x^4$, $y(-1) = -1$, $y'(-1) = \pi$

5 Conclusions

In this work a novel library to implement the Tau method is proposed. For the moment, it delivers approximate solutions for linear differential systems of equatuions, with polynomial or polynomial approximations of the right-hand side. The library offers the possibility to solve the problem in any interval, with different type of polynomial bases, different linear solvers, ranging from the most efficient GMRES and BICGSTAB to others less popular like BICGSTABL, MINRES, QMR, TFQMR and LSQR. It offers a set of preconditioners to accelerate the linear solvers, mainly, the more efficient but expensive ILU factorization to the less general but less expensive block diagonal preconditioner.
Numerical results illustrate the effectiveness of the implemented approach: time for computation is fast, even when compared with state-of-the-art methods like the 4^{th} order Runge-Kutta. It is worth to mention that the error produced by the approximate solution driven by the Tau method is more uniform than the one delivered by the traditional Runge-Kutta of order 4.
The framework proposed offers a simple approach to solve systems of differential problems, which can be an attractive feature for the user. Our ultimate goal is to extend the library functionalities to other classes of problems, enabling the use of the method regardless of the type of problem.

REFERENCES

[1] Crisci M.R., Russo E., "An extension of Ortiz's recursive formulation of the tau method to certain linear systems of ordinary differential equations", *Math. Comput.*, 41 (1983) pp. 27-42.

[2] Heat M. T., "Scientific Computing, An introductory Survey", *McGraw Hill*, (1997).

[3] Liu K. M, Pan C. K., "The Automatic Solution to Systems of Ordinary Differential Equations by the Tau Method", *Computers and Mathematics with Applications*, 38 (1999) pp. 197-210.

[4] Lanczos C., "Trigonometric Interpolation of Empirical and Analytical Functions", *Journal Maths Phis*, 17, (1938) pp. 123-199.

[5] Mahmound J. P., Ardabili M. Y. R., Shahorad A., "Numerical solution of the systems of Fredholm integro-differential equations by the tau method", *Applied Mathematics and Computation*, 168 (2005) pp. 465-478.

[6] Mason J. C. and Handscomb D. C., "Chebyshev Polynomials", *Chapman and Hall/CRC*, (2003).

[7] Matos J., Rodrigues M. J., de Matos J. C. and Cruz M., "Avoiding similarity transformations in the operational Tau method", (2015).

[8] Matos J., Rodrigues M. J., Vasconcelos P. B., "New implementation of the Tau method for PDE's", *Journal of Computational and Applied Mathematics*, 164-165 (2004) pp. 555-567.

[9] Navasimayan S., Ortiz E.L., "Best approximation and the numerical solution of partial differential equations with the tau method", Portugal. *Math.*, 41 (1985) pp. 97-119.

[10] Ortiz E.L., "On the numerical solution of nonlinear and functional differential equations with the tau method", R. Ansorge (ed.) W. Trnig (ed.), *Numerical Treatment of Differential Equations in Applications*, Berlin (1978) pp. 127-139.

[11] Ortiz E.L., Dinh A. P. N., "Linear recursive schemes associated with some nonlinear partial differential equations in one dimension and the tau method", SIAM J. *Math. Anal.*, 18 (1987) pp. 452-464.

[12] Ortiz E.L., Samara H., "An operational approach to the tau method for the numerical solution of nonlinear differential equations", *Computing*, 27 (1981) pp. 15-25.

[13] Saad Y., "Iterative methods for sparse linear systems", *SIAM*, (2003).

[14] Trefethen L. N., "Spectral methods in MATLAB", *SIAM*, Oxford University, (2000).

SYMCOMP 2015
Faro, March 26-27, 2015

PAGERANK COMPUTATION USING LUMPING AND EXTRAPOLATION TECHNIQUES

Isabel R. Mendes*, Paulo B. Vasconcelos[†]

*School of Engineering and LEMA
Polytechnic Institute of Porto
Porto, Portugal
Email: irm@isep.ipp.pt

[†] Faculty of Economics and CMUP
University of Porto
Porto, Portugal
Email: pjv@fep.up.pt - Web page: http://www.fep.up.pt/docentes/pjv

Keywords: Web matrix; eigenvector computation; dangling node; lumping; convergence acceleration and Aitken extrapolation

Abstract *Web information retrieval is extremely challenging due to the huge number of web pages. To determine the order of importance in which to display web pages after a query, Google's search engine computes the PageRank vector, the left principal eigenvector of a web matrix that is related to the hyperlink structure of the web, the Google matrix.*
The Power Method is one of the oldest and simplest iterative methods for finding the dominant eigenvalue and eigenvector of a matrix and it was the original method proposed by Brin and Page for finding the PageRank vector. The PageRank computation by the standard power method takes days to converge since matrices involved are large. The number of web pages is increasing rapidly, so, it becomes necessary to find refinements or alternatives to speed up the computation.
Among the most successful approaches for reducing the work associated with the PageRank vector are the extrapolation techniques [4], and the more recent lumping methods [6] and [7]) that proceed with a matrix reordering according to dangling and nondangling nodes.
This work presents a novel approach for the acceleration of the PageRank computation by combining reordered and extrapolation techniques. Two algorithms, called LumpingE methods, considering standard Aitken extrapolation within the original lumping method are proposed. Numerical experiments comparing the new LumpingE methods with standard Power method and original Lumping methods are illustrated. Results show the merits from this new proposal.

1. INTRODUCTION

PageRank is a numerical algorithm to order the relative importance of web documents based on a web graph (within the World Wide Web or other set). The weight of a page, PageRank, is computed recursively and depends on the number and PageRank of all incoming links. The PageRank's thesis is that *a webpage is important if it is pointed to by other important pages*. Comparing the PageRank scores of two pages gives an indication of the relative importance of the two pages.

During the last decade the scientific community was very active in studying and presenting numerical approaches to solve the problem as fast as possible and with the less computational cost possible. A plethora of publications have been produced on this topic. The original paper [1], written by Google founders Larry Page and Sergey Brin, was the seed that led to all this work and the success of Google. However, the PageRank algorithm is much more complex, involving other topics such as spammers' control, author rank, among others. Although other link based ranking algorithms have been developed, and still are, Google's PageRank is, until now, the most studied and successful of them all.

The PageRank problem can be, simply stated by:

$$\pi^T = \pi^T G$$

where $\pi^T = (\pi(1), \ldots, \pi(n))^T$ is the *PageRank vector*, $\| \pi \|_1 = 1$, $\pi(i)$ is the PageRank of page *i*, *n* is the number of pages in Google's index of the Web and *G* is the *Google matrix* of order *n*.

$$G = \alpha S + (1-\alpha)E \tag{1}$$

is a convex combination of two stochastic matrices, *S* that represents the link structure of the web graph and *E* with the purpose to force uniqueness for π^T. Matrix

$$S = H + dw^T \tag{2}$$

is composed by the matrix *H* reflecting the link structure of the web, $h_{ij} = 1/n_i$ if page *i* links to page *j* and 0 otherwise, being n_i the number of outlinks of page *i*, and dw^T to eliminate the zero rows produced by pages with no outlinks, $d_i = 1$ if $n_i = 0$ and 0 otherwise. Vector $w \geq 0$ is known as *the dangling node vector* and it is a probability vector. A dangling node represents a page with no links to other page (such as image files or protected pages); if a link exists, then the node is referred as *nondangling*. Furthermore, uniqueness of π is guaranteed for *G* irreducible and aperiodic, which leads to $G = \alpha S + (1-\alpha)E$, with $E = ev^T$ (rank-1 matrix), where e is the vector of all ones, $v \geq 0$ is a probability vector known as *personalization* or *teleportation vector*. It is usual to assume $w = v = \frac{1}{n}e$. Finally, $\alpha \in [0,1]$ is the *dumping factor*, the fraction of time that the random walk follows a link.

This is an eigenvalue problem, where π^T stands for the left eigenvector of matrix *G*

associated with the dominant eigenvalue λ_1. The PageRank vector exists, is unique and can be found with the simple power method. This numerical method is simple, shows slow convergence but is stable, reliable and converges to the principal eigenvector, depending on α. A dumping factor far less than 1 allows for convergence (linear) of the power method. Usually $\alpha = 0.85$ is the reference value and a higher one closer to 1 makes the computation more difficult. For the former cases, more sophisticated methods, such as Implicit Arnoldi or Krylov-Schur are the only answer to solve those (large) eigenvalue problems [2].
The book [3] offers an excellent overview of Google's PageRank and other search engines.
Due to matrices involved in PageRank computations being large, PageRank computations, using the power method, takes days to converge. Moreover, the number of web pages is increasing rapidly, so, the need to enhance the PageRank computations, by reducing the computation time and/or the number of required iterations to reach a certain precision, is thus mandatory. Some of the most successful approaches are the known extrapolation techniques [4], [5] and the more recent Lumping methods [6], [7] that proceed with a matrix reordering according to dangling and nondangling nodes.

The rest of the paper is structured as follows. In section 2 we briefly describe the Lumping method, Aitken extrapolation and provide some references to extrapolation acceleration techniques.
Section 3 presents the hybrid proposed approach combining reordered techniques with extrapolation. Numerical results illustrating the effectiveness of the proposed approach are given in section 4. Finally section 5 presents a few concluding remarks.

2. ALGORITHM INGREDIENTS

Our new approach relies on combining the lumping methods with extrapolation to achieve less iterations on the iterative process.
While some acceleration techniques are well-known such as the simplest Aitken and quadratic extrapolation [4] to the most sophisticated ones like the $\epsilon-$algorithm [5], Lumping methods are not so popular. In 2.1. and 2.2. is provided a brief presentation about the Lumping methods. In 2.3. we will present the Aitken extrapolation.

2.1. Lumping 1 method

The dangling nodes can be lumped into a single node to obtain a stochastic reduced matrix with the same eigenvalues as the full matrix. The PageRank computation for the nondangling nodes is performed separately. That is the *H* matrix can be partitioned into

$$\begin{bmatrix} H_{11} & H_{12} \\ 0 & 0 \end{bmatrix}$$

where $H_{11} \geq 0$, $k \times k$ represents, respectively, the links among nondangling nodes (ND) and $H_{12} \geq 0$, $k \times (n-k)$, the links from nondangling nodes to dangling nodes. The $(n-k)$ zero rows represent the dangling nodes (D) and the k first rows sums one. Then (2) can be casted

as

$$S = \begin{bmatrix} H_{11} & H_{12} \\ ew_1^T & ew_2^T \end{bmatrix}$$

where $d = [0 \;\; e]^T$, $v = [v_1 \;\; v_2]^T$, $v = [w_1 \;\; w_2]^T$, w_1 $k \times 1$ and w_2 $(n-k)\times 1$. With this partition we can rewrite equation (1)

$$G = \begin{bmatrix} G_{11} & G_{12} \\ eu_1^T & eu_2^T \end{bmatrix}$$

with $u = [u_1 \;\; u_2]^T$, $u_i = \alpha\, w_i + (1-\alpha)v_i$, and $G_{1i} = \alpha\, H_{1i} + (1-\alpha)ev_i^T$, $i = 1, 2$.

Lumping can be viewed as a similarity transformation of the Google matrix. Next we provide a theorem to express the PageRank vector for Lumping 1 method. It comprises two theorems from [6].

Theorem 2.1. (adapted from [6]) Let

$$X = \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}$$

with $L = I_{n-k} - \frac{1}{n-k}\hat{e}e^T$, $\hat{e} = e - e_1 = [0, 1, 1, \ldots, 1]^T$ the first canonical basis vector, and $I_n = [e_1 \cdots e_n]$ the identity matrix of order n. Then,

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}$$

where

$$G^{(1)} = \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}.$$

The matrix $G^{(1)}$ is stochastic of order $k+1$ with the same nonzero eigenvalues as G.

Let $\sigma^T G^{(1)} = \sigma^T$, $\sigma^T \geq 0$, $\|\sigma\| = 1$, with partition $\sigma^T = [\sigma_{1:k}^T \;\; \sigma_{k+1}]$, where σ_{k+1} is a scalar. Then the PageRank vector π^T equals

$$\pi^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}. \blacksquare$$

Proof: The proof follows and details [6].

From

$$X^{-1} = \begin{bmatrix} I_k & 0 \\ 0 & L^{-1} \end{bmatrix}$$

it follows that

$$XGX^{-1} = \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ eu_1^T & eu_2^T \end{bmatrix} X^{-1} = \begin{bmatrix} I_k G_{11} & I_k G_{12} \\ Leu_1^T & Leu_2^T \end{bmatrix} X^{-1} =$$

$$= \begin{bmatrix} G_{11} & G_{12} \\ Leu_1^T & Leu_2^T \end{bmatrix} \begin{bmatrix} I_k & 0 \\ 0 & L^{-1} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12}L^{-1} \\ Leu_1^T & Leu_2^T L^{-1} \end{bmatrix} .$$

Considering $L^{-1} = I_{n-k} + \hat{e}e^T$, $e^T e = n-k$, $\hat{e}\, e^T e = (n-k)\hat{e}$ and simplifying $I_{n-k} = I$,

let $Le = \left(I_{n-k} - \frac{1}{n-k}\hat{e}e^T \right) e = e - \frac{1}{n-k}\hat{e}e^T e = e - \frac{1}{n-k}\hat{e}(n-k) = e - \hat{e} = e_1$

so $Leu_1^T = e_1 u_1^T$

$$Leu_2^T L^{-1} = \left(I_{n-k} - \frac{1}{n-k}\hat{e}e^T \right) eu_2^T L^{-1} = I_{n-k} eu_2^T L^{-1} - \frac{1}{n-k}\hat{e}e^T eu_2^T L^{-1}$$

$$= eu_2^T L^{-1} - \frac{1}{n-k}(n-k)\hat{e}u_2^T L^{-1} = eu_2^T L^{-1} - \hat{e}u_2^T L^{-1} = (e-\hat{e})u_2^T L^{-1} = e_1 u_2^T L^{-1}$$

$$= e_1 u_2^T \left(I_{n-k} + \hat{e}e^T \right) = e_1 u_2^T \left(I + \hat{e}e^T \right)$$

and

$$G_{12}L^{-1} = G_{12}\left(I_{n-k} + \hat{e}e^T \right) = G_{12}\left(I + \hat{e}e^T \right).$$

Then

$$XGX^{-1} = \begin{bmatrix} G_{11} & G_{12}L^{-1} \\ Leu_1^T & Leu_2^T L^{-1} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12}\left(I + \hat{e}e^T \right) \\ e_1 u_1^T & e_1 u_2^T \left(I + \hat{e}e^T \right) \end{bmatrix}$$

has the same eigenvalues as G.

We choose a different partitioning and separate the leading $k+1$ rows and columns, in order to reveal the eigenvalues, and observe that

$$\left(I_{n-k} + \hat{e}e^T \right) e_1 = \left(I + \hat{e}e^T \right) e_1 = e \ , \qquad G_{12}\left(I + \hat{e}e^T \right) e_1 = G_{12}e \, , \qquad u_2^T \left(I + \hat{e}e^T \right) e_1 = u_2^T e$$

and

$$XGX^{-1} = \begin{bmatrix} G_{11} & G_{12}\left(I+\hat{e}e^T\right) \\ e_1u_1^T & e_1u_2^T\left(I+\hat{e}e^T\right) \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12}e & \bullet \\ u_1^T & u_2^Te & \bullet\bullet \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}.$$

So, we obtain the block triangular matrix

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix} \quad \text{where,} \quad G^{(1)} = \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^Te \end{bmatrix}$$

with at least $n-k-1$ zero eigenvalues.

The similarity transformation is thus proved.

Finally, and to express the PageRank vector, consider

$$* = G^{(2)} = \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}\left(I+\hat{e}e^T\right)\begin{bmatrix} e_2 & \cdots & e_{n-k} \end{bmatrix} \quad \text{then} \quad XGX^{-1} = \begin{bmatrix} G^{(1)} & G^{(2)} \\ 0 & 0 \end{bmatrix}.$$

The vector $\begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}$ is an eigenvector for XGX^{-1} associated with the eigenvalue $\lambda = 1$.

Then

$$\begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}XGX^{-1} = \begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}$$

Multiplying by X on the right

$$\begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}XGX^{-1}X = \begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}X \quad \Leftrightarrow \quad \begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}XG = \begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}X$$

So

$$\hat{\pi}^T = \begin{bmatrix} \sigma^T & \sigma^TG^{(2)} \end{bmatrix}X$$

is an eigenvector of G associated with $\lambda = 1$ and a multiple of the stationary distribution π of G. The dominant eigenvalue 1 of G is distinct and $G^{(1)}$ has the same nonzero eigenvalues as G. So, the stationary distribution σ of $G^{(1)}$ is unique.

Using the original partitioning which separates the k leading elements we will express $\hat{\pi}$ in terms of quantities in the matrix G.

Let

$$\hat{\pi}^T = \begin{bmatrix} \sigma_{1:k}^T & \left(\sigma_{k+1} & \sigma^TG^{(2)}\right) \end{bmatrix}\begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}.$$

Multiplying out

$$\hat{\pi}^T = \left[\sigma_{1:k}^T \quad \left(\sigma_{k+1} \quad \sigma^T G^{(2)}\right) L\right]$$

shows that $\hat{\pi}^T$ has the same leading elements as $\sigma^T = \left[\sigma_{1:k}^T \quad \sigma_{k+1}\right]$.

Now, we must examine the trailing $n-k$ components of $\hat{\pi}^T$. To do this we partition the matrix $L = I_{n-k} - \frac{1}{n-k}\hat{e}e^T$ and distinguish the first row and column,

$$L = \begin{bmatrix} 1 & 0 \\ -\frac{1}{n-k}e & I - \frac{1}{n-k}ee^T \end{bmatrix}.$$

Then the eigenvector part associated with the dangling nodes is

$$z^T = \left[\sigma_{k+1} \quad \sigma^T G^{(2)}\right] L = \left[\sigma_{k+1} - \frac{1}{n-k}\sigma^T G^{(2)} e \qquad \sigma^T G^{(2)}\left(I - \frac{1}{n-k}ee^T\right)\right]$$

To remove the terms containing $G^{(2)}$ in z, we simplify

$$\left(I + \hat{e}e^T\right)[e_2 \quad \cdots \quad e_{n-k}]e = \left(I + \hat{e}e^T\right)\hat{e} = (n-k)\hat{e}\,.$$

Hence

$$G^{(2)}e = \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}\left(I + \hat{e}e^T\right)[e_2 \quad \cdots \quad e_{n-k}]e = \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}(n-k)\hat{e} \quad \Leftrightarrow \quad G^{(2)}e = (n-k)\begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}\hat{e} \qquad (3)$$

and

$$\frac{1}{n-k}\sigma^T G^{(2)} e = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}\hat{e} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1 = \sigma_{k+1} - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1\,,$$

where we used $\hat{e} = e - e_1$, and the fact that σ is the stationary distribution of $G^{(1)}$, so

$$\sigma_{k+1} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e\,.$$

Therefore the leading element of z equals

$$z_1 = \sigma_{k+1} - \frac{1}{n-k}\sigma^T G^{(2)} e = \sigma_{k+1} - \left(\sigma_{k+1} - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1\right) = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1\,.$$

For the remaining elements of z, we use (3) to simplify

$$G^{(2)}\left(I-\frac{1}{n-k}ee^T\right)=G^{(2)}-\frac{1}{n-k}G^{(2)}ee^T=G^{(2)}-\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\hat{e}e^T\ .$$

Replacing

$$\left(I+\hat{e}e^T\right)[e_2\quad\cdots\quad e_{n-k}]\ =\ [e_2\quad\cdots\quad e_{n-k}]+\hat{e}e^T$$

in $G^{(2)}$ yields

$$\begin{aligned}z_{2:n-k}^T\ &=\ \sigma^T G^{(2)}\left(I-\frac{1}{n-k}ee^T\right)=\ \sigma^T\left(G^{(2)}-\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\hat{e}e^T\right)=\\&=\ \sigma^T\left(\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\left(I+\hat{e}e^T\right)[e_2\quad\cdots\quad e_{n-k}]-\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\hat{e}e^T\right)=\\&=\ \sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\left(\left(I+\hat{e}e^T\right)[e_2\quad\cdots\quad e_{n-k}]-\ \hat{e}e^T\right)=\\&=\ \sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\left([e_2\quad\cdots\quad e_{n-k}]+\hat{e}e^T\ -\ \hat{e}e^T\right)=\\&=\sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}[e_2\quad\cdots\quad e_{n-k}].\end{aligned}$$

Therefore the eigenvector part associated with the dangling nodes is

$$\begin{aligned}z\ &=\ \left[z_1\quad z_{2:n-k}^T\ \right]=\left[\sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}e_1\quad\sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}[e_2\quad\cdots\quad e_{n-k}]\right]=\\&=\ \sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}[e_1\quad e_2\quad\cdots\quad e_{n-k}]\ =\ \sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}\end{aligned}$$

and

$$\hat{\pi}=\left[\sigma_{1:k}^T\quad\sigma^T\begin{pmatrix}G_{12}\\u_2^T\end{pmatrix}\right].$$

Since π is unique, we conclude that $\hat{\pi}=\pi$ if $\hat{\pi}^Te=1$.

This follows, again, from the fact that σ is the stationary distribution of $G^{(1)}$ and

$$\sigma^T\begin{bmatrix}G_{12}\\u_2^T\end{bmatrix}e\ =\ \sigma_{k+1}.\quad\blacksquare$$

Theorem 2.1 illustrates how the PageRank π^T can be given in terms of σ of the small matrix $G^{(1)}$.

2.2. Lumping 2 method

An alternative more complex to this approach considers a refined division of the ND nodes in strongly nondangling nodes (SND), pages with links to pages that are not dangling nodes, and weakly nondangling nodes (WND), pages that are not dangling but that point to only dangling nodes. This division leads to a matrix H of the form

$$H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} H_{11}^{11} & H_{11}^{12} & H_{12}^{1} \\ 0 & 0 & H_{12}^{2} \\ 0 & 0 & 0 \end{bmatrix}$$

where H_{11}^{11}, $k_1 \times k_1$ represents the links among SND, H_{11}^{12}, $k_1 \times k_2$, the links from SND to WND, H_{12}^{1}, $k_1 \times (n-k)$, the links from SND to D and H_{12}^{2}, $k_2 \times (n-k)$, the links from WND to D.

Figure 1 illustrates the distinction between strongly and weakly nondangling nodes and dangling nodes.

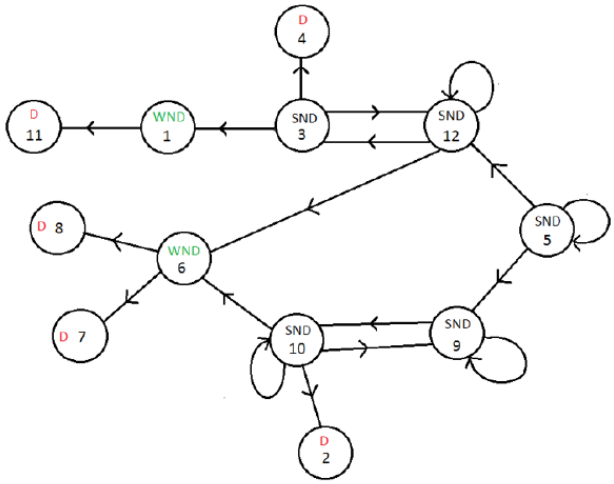


Figure 1. Graph example illustrating the different types of nodes.

With the Lumping 1 method the dangling nodes are lumped into a single node resulting in a stochastic reduced matrix with the same eigenvalues as the full matrix. The PageRank computation for the nondangling nodes are performed separately. Following this idea, in [7] it was shown that weakly nondangling nodes can be also lumped into a single node and the PageRank of the strongly nondangling nodes can be computed separately. The further reduced matrix is also stochastic with the same nonzero eigenvalues as the Google matrix G. Moreover, the full PageRank vector π can be easily recovered from the stationary distribution of the smaller matrix.

Theorem 2.2. [7]

Using the notation above and defining $G^{(2)}$ by

$$G^{(2)} = \begin{bmatrix} G_{11}^{11} & G_{12}^{1} e & G_{11}^{12} e \\ u_1^{(1)T} & u_2^T e & u_1^{(2)T} e \\ (1-\alpha)\, v_1^{(1)T} & \alpha + (1-\alpha) v_2^T e & (1-\alpha)\, v_1^{(2)T} e \end{bmatrix},$$

then $G^{(2)}$ is a stochastic matrix of order $k_1 + 2$ with the same nonzero eigenvalues as the full Google matrix G. Let $\hat{\sigma}^T = \hat{\sigma}^T G^{(2)}$, $\hat{\sigma} \geq 0$, $\| \hat{\sigma} \| = 1$, partitioned by $\hat{\sigma}^T = \begin{bmatrix} \hat{\sigma}_{1:k_1}^T & \hat{\sigma}_{k_1+1} & \hat{\sigma}_{k_1+2} \end{bmatrix}$, where $\hat{\sigma}_{k_1+1}$ and $\hat{\sigma}_{k_1+2}$ are two scalars.

Then the PageRank vector π is given by

$$\pi^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix} \tag{4}$$

where the vector σ is

$$\sigma^T = \begin{bmatrix} \hat{\sigma}_{1:k_1}^T & \hat{\sigma}^T \begin{pmatrix} G_{11}^{12} \\ u_1^{(2)T} \\ (1-\alpha) v_1^{(2)T} \end{pmatrix} & \hat{\sigma}_{k_1+1} \end{bmatrix}. \tag{5}$$ ■

Proof: Following closely [7]

Let $$X = \begin{bmatrix} I_{k_1} & 0 & 0 \\ 0 & I_{k_2} & 0 \\ 0 & 0 & L \end{bmatrix},$$

where $L = I_{n-k} - \frac{1}{n-k} \hat{e} e^T$ and $\hat{e} = e - e_1 = [0, 1, 1, \ldots, 1]^T$.

It follows from Theorem 2.1. that

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix},$$

where

$$G^{(1)} = \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix} = \begin{bmatrix} G_{11}^{11} & G_{11}^{12} & G_{12}^{1}e \\ (1-\alpha)e v_1^{(1)^T} & (1-\alpha)e v_1^{(2)^T} & G_{12}^{2}e \\ u_1^{(1)^T} & u_1^{(2)^T} & u_2^T e \end{bmatrix}$$

$$= \begin{bmatrix} G_{11}^{11} & G_{11}^{12} & G_{12}^{1}e \\ (1-\alpha)e v_1^{(1)^T} & (1-\alpha)e v_1^{(2)^T} & \left(\alpha+(1-\alpha)v_2^T e\right)e \\ u_1^{(1)^T} & u_1^{(2)^T} & u_2^T e \end{bmatrix},$$

where we used $H_{12}^2 e = e$. The matrix $G^{(1)}$ is stochastic of order $k+1$ with the same nonzero eigenvalues as G.

Let $\sigma^T G^{(1)} = \sigma^T$, $\sigma \geq 0$, $\| \sigma \| = 1$, and partition $\sigma = \left[\sigma_{1:k}^T \quad \sigma_{k+1}\right]$, where σ_{k+1} is a scalar.

It follows from Theorem 2.1. that the PageRank vector π satisfies $\pi^T = \left[\sigma_{1:k}^T \quad \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix}\right]$.

Define the permutation matrix Z by $Z = \begin{bmatrix} I_{k_1} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & I_{k_2} & 0 \end{bmatrix}$.

We have

$$\hat{G}^{(1)} = ZG^{(1)}Z^T = \begin{bmatrix} G_{11}^{11} & G_{12}^{1}e & G_{11}^{12} \\ u_1^{(1)^T} & u_2^T e & u_1^{(2)^T} \\ (1-\alpha)e v_1^{(1)^T} & \left(\alpha+(1-\alpha)v_2^T e\right)e & (1-\alpha)e v_1^{(2)^T} \end{bmatrix}.$$

Let $\hat{X} = \begin{bmatrix} I_{k_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \hat{L} \end{bmatrix}$, where $\hat{L} = I_{k_2} - \frac{1}{k_2} \hat{e} e^T$.

From Theorem 2.1., we obtain $\hat{X}\hat{G}^{(1)}\hat{X}^{-1} = \begin{bmatrix} G^{(2)} & * \\ 0 & 0 \end{bmatrix}$,

where $G^{(2)} = \begin{bmatrix} G_{11}^{11} & G_{12}^{1}e & G_{11}^{12}e \\ u_1^{(1)^T} & u_2^T e & u_1^{(2)^T} e \\ (1-\alpha)v_1^{(1)^T} & \alpha+(1-\alpha)v_2^T e & (1-\alpha)v_1^{(2)^T} e \end{bmatrix}$.

The matrix $G^{(2)}$ is stochastic of order k_1+2 with the same nonzero eigenvalues as $\hat{G}^{(1)}$. Therefore, $G^{(2)}$ has the same nonzero eigenvalues as G.

Let $\hat{\sigma}^T G^{(2)} = \hat{\sigma}^T$, $\hat{\sigma} \geq 0$, $\|\hat{\sigma}\| = 1$, and partition $\hat{\sigma}^T = \left[\hat{\sigma}_{1:k_1+1}^T \quad \sigma_{k_1+2} \right]$, where σ_{k_1+2} is a scalar.

It follows from Theorem 2.1 that the stationary distribution vector $\hat{\pi}$ of $\hat{G}^{(1)}$ satisfies

$$\hat{\pi}^T = \begin{bmatrix} \hat{\sigma}_{1:k_1+1}^T & \hat{\sigma}^T \begin{pmatrix} G_{11}^{12} \\ u_1^{(2)^T} \\ (1-\alpha) v_1^{(2)^T} \end{pmatrix} \end{bmatrix}.$$

We have $\hat{\pi}^T = \hat{\pi}^T \hat{G}^{(1)} = \hat{\pi}^T Z G^{(1)} Z^T$, i.e., $\left(Z^T \hat{\pi}\right)^T = \left(Z^T \hat{\pi}\right)^T G^{(1)}$. Therefore, the stationary distribution σ of $G^{(1)}$ satisfies

$$\sigma^T = \hat{\pi}^T Z = \begin{bmatrix} \hat{\sigma}_{1:k_1}^T & \hat{\sigma}^T \begin{pmatrix} G_{11}^{12} \\ u_1^{(2)^T} \\ (1-\alpha) v_1^{(2)^T} \end{pmatrix} & \hat{\sigma}_{k_1+1} \end{bmatrix}.$$

This completes the proof of the theorem. ■

Theorem 2.2. shows that to compute the PageRank vector π, we can compute the stationary distribution $\hat{\sigma}$ of the stochastic matrix $G^{(2)}$ and then recover the PageRank vector π according to (4) and (5).

2.3. Aitken extrapolation method

The power method is one of the simplest iterative methods for finding the dominant eigenvalue and eigenvector of a matrix and it was the original method proposed by Brin and Page for finding the PageRank vector.

As mentioned, the standard power method is known for its slow convergence. So, since 1998, several researchers are working on techniques for accelerate the PageRank computations. Between them there is a group of Stanford researchers that proposed several methods for accelerating the power method.

In [4] this group of researchers proposed an acceleration method that aims to reduce the number of power iterations, the Aitken extrapolation method, which it's based on the well-known Aitken's delta square method for accelerating linearly convergence sequences.

Aitken's process is a well-known sequence transformation to accelerate of the rate of convergence of a slowly converging sequence [8].

Since $1 > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$, it is known that the power method, $\pi^{(k)^T} G$, converges to the principal eigenvector of the Markov matrix G. Assuming that the starting vector lies in the subspace spanned by the eigenvectors of G, the power method can be expanded has

$$\pi^{(l)^T} = \pi^{(l-1)^T} G = \pi^{(0)^T} G^l = \pi^{(0)^T} \left(e\pi^T + \sum_{i=2}^{n} \lambda_i^l x_i y_i^T \right)$$

$$= \pi^T + \alpha_2 \lambda_2^l y_2^T + \sum_{i=3}^{n} \alpha_i \lambda_i^l y_i^T$$

where x_i and y_i, $i = 1, \ldots, n$, are, respectively, the right and left eigenvectors corresponding to λ_i, and $\alpha_i = \pi^{(0)^T} x_i$, $i = 2, \ldots, n$. The size of the subdominant eigenvalue λ_2 governs the number of power iterations.

To improve the convergence, if $|\lambda_2| > |\lambda_3|$, then $\pi^{(l)^T} - \alpha_2 \lambda_2^l y_2^T$ will be a better approximation to π^T, and $\alpha_2 \lambda_2^l y_2^T$ can be estimated by Aitken's delta square process based on three iterates [4]:

$$\alpha_2 \lambda_2^l y_2^T \approx \frac{\left(\Delta \pi^{(l)^T} \right)^2}{\Delta^2 \pi^{(l)^T}}$$

$$\left(\Delta \pi^{(l)^T} \right) = \pi^{(l+1)^T} - \pi^{(l)^T}$$

$$\Delta^2 \pi^{(l)^T} = \pi^{(l+2)^T} - 2\pi^{(l+1)^T} + \pi^{(l)^T}$$

where $(*)^2$ indicates component-wise squaring of elements in the vector $(*)$.

So, by subtracting $\alpha_2 \lambda_2^l y_2^T$ to $\pi^{(l)^T}$ we can accelerate the convergence of the power method.

Next, for simplicity, we will use $\pi^{(l)} = G\pi^{(l-1)}$ instead of $\pi^{(l)^T} = \pi^{(l-1)^T} G$ and consider that matrix G has n distinct eigenvectors u_i: $Gu_i = \lambda_i u_i$.

We will assume that the starting vector $\pi^{(0)}$ lies in the subspace spanned by the eigenvectors of G, $\pi^{(0)} = u_1 + \sum_{i=2}^{n} \alpha_i u_i$ and will use $\alpha_2 \lambda_2^l u_2$ instead of $\alpha_2 \lambda_2^l y_2^T$.

Following Kamvar et al. in [4], we will proof that $\alpha_2 \lambda_2^l u_2$ can be estimated by Aitken's delta square process by using the two subsequent iterates of the power method $\left(\pi^{(l+1)}, \pi^{(l+2)} \right)$:

$\alpha_2 \lambda_2^l u_2 \approx \frac{\left(\Delta \pi^{(l)} \right)^2}{\Delta^2 \pi^{(l)}}$, with $\Delta \pi^{(l)} = \pi^{(l+1)} - \pi^{(l)}$ and $\Delta^2 \pi^{(l)} = \pi^{(l+2)} - 2\pi^{(l+1)} + \pi^{(l)}$.

We will begin by assuming that $\pi^{(l)}$ can be expressed as a linear combination of the first two eigenvectors:

$$\pi^{(l)} = u_1 + \alpha_2 u_2 \tag{6}$$

We will calculate an estimate of the principal eigenvector u_1 in closed form using the successive iterates. This approximation becomes increasingly accurate as l becomes larger.

Appling the power method and knowing that (λ_1, u_1) and (λ_2, u_2) are eigenpairs of G:

$$\pi^{(l+1)} = G\,\pi^{(l)} = G\left[u_1 + \alpha_2 u_2\right] = Gu_1 + \alpha_2 Gu_2 = \lambda_1 u_1 + \alpha_2 \lambda_2 u_2$$

Since the first eigenvalue λ_1 of a Markov matrix is 1,

$$\pi^{(l+1)} = G\,\pi^{(l)} = u_1 + \alpha_2 \lambda_2 u_2 \tag{7}$$

Appling another iteration of the power method,

$$\pi^{(l+2)} = G\,\pi^{(l+1)} = G\left[u_1 + \alpha_2 \lambda_2 u_2\right] = Gu_1 + \alpha_2 \lambda_2 Gu_2 = u_1 + \alpha_2 \lambda_2^2 u_2 \tag{8}$$

Let us define,

$$g_i = \left(\Delta \pi_i^{(l)}\right)^2 = \left(\pi_i^{(l+1)} - \pi_i^{(l)}\right)^2 \tag{9}$$

and

$$h_i = \Delta^2 \pi_i^{(l)} = \pi_i^{(l+2)} - 2\pi_i^{(l+1)} + \pi_i^{(l)} \tag{10}$$

where π_i represents the component i of vector π.

Using (6) and (7) in (9),

$$g_i = \left[(u_1)_i + \alpha_2 \lambda_2 (u_2)_i - (u_1)_i - \alpha_2 (u_2)_i\right]^2 = \alpha_2^2 (\lambda_2 - 1)^2 (u_2)_i^2$$

Using (6), (7) and (8) in (10),

$$h_i = (u_1)_i + \alpha_2 \lambda_2^2 (u_2)_i - 2\left((u_1)_i + \alpha_2 \lambda_2 (u_2)_i\right) + \alpha_2 (u_2)_i = \alpha_2 (\lambda_2 - 1)^2 (u_2)_i$$

Considering f_i as the quotient $\frac{g_i}{h_i}$:

$$f_i = \frac{g_i}{h_i} = \frac{\alpha_2^2 (\lambda_2 - 1)^2 (u_2)_i^2}{\alpha_2 (\lambda_2 - 1)^2 (u_2)_i} = \alpha_2 (u_2)_i$$

Therefore, $f_i = \alpha_2 u_2$

Hence, from equation (6), we have a closed-form solution for u_1:

$$u_1 = \pi^{(l)} - \alpha_2 u_2 = \pi^{(l)} - f$$

3. THE NEW LUMPINGE METHODS

As mentioned, the proposed approach, LumpingE methods, combines Lumping methods with Aitken extrapolation.

3.1. LumpingE 1 method

Theorem 3.1. (LumpingE 1 method) [9]

Let $$X = \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}$$

with $L = I_{n-k} - \frac{1}{n-k}\hat{e}e^T$, $\hat{e} = e - e_1 = [0, 1, 1, \ldots, 1]^T$ the first canonical basis vector, and $I_n = \lfloor e_1 \ \cdots e_n \rfloor$ the identity matrix of order n. Then,

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}$$

where

$$G^{(1)} = \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}.$$

The matrix $G^{(1)}$ is stochastic of order $k+1$ with the same nonzero eigenvalues as G.

Let $\sigma^T G^{(1)} = \sigma^T$, $\sigma^T \geq 0$, $\| \sigma \| = 1$. Then the PageRank vector π^T equals

$$\pi^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}$$

with partition

$$\sigma^T = \begin{cases} \left[\sigma_{1:k}^T - \dfrac{\left(\Delta\sigma_{1:k}^T\right)^2}{\Delta^2 \sigma_{1:k}^T} \quad \sigma_{k+1} - \dfrac{(\Delta\sigma_{k+1})^2}{\Delta^2 \sigma_{k+1}} \right], & \text{for iterations } s, l \times s, \cdots, l \in IN \\ \left[\sigma_{1:k}^T \quad \sigma_{k+1} \right], & \text{for other iterations} \end{cases}$$

where σ_{k+1} is a scalar and *s* defines the step by which the extrapolation is applied. ■

Proof:

Follows the proof of theorem 2.1 and at every s-step apply one Aitken extrapolation.

3.2. LumpingE 2 method

Theorem 3.2. (LumpingE 2 method) [9]

Using the notation above and defining $G^{(2)}$ by

$$G^{(2)} = \begin{bmatrix} G_{11}^{11} & G_{12}^{1}e & G_{11}^{12}e \\ u_1^{(1)T} & u_2^T e & u_1^{(2)T} e \\ (1-\alpha)\, v_1^{(1)T} & \alpha + (1-\alpha) v_2^T e & (1-\alpha)\, v_1^{(2)T} e \end{bmatrix},$$

then $G^{(2)}$ is a stochastic matrix of order $k_1 + 2$ with the same nonzero eigenvalues as the full Google matrix G.

Let $\hat{\sigma}^T = \hat{\sigma}^T G^{(2)}$, $\hat{\sigma} \geq 0$, $\| \hat{\sigma} \| = 1$, partitioned by

$$\hat{\sigma}^T = \begin{cases} \left[\hat{\sigma}_{1:k_1}^T - \dfrac{\left(\Delta \hat{\sigma}_{1:k_1}^T\right)^2}{\Delta^2 \hat{\sigma}_{1:k_1}^T} \quad \hat{\sigma}_{k_1+1} - \dfrac{\left(\Delta \hat{\sigma}_{k_1+1}\right)^2}{\Delta^2 \hat{\sigma}_{k_1+1}} \quad \hat{\sigma}_{k_1+2} - \dfrac{\left(\Delta \hat{\sigma}_{k_1+2}\right)^2}{\Delta^2 \hat{\sigma}_{k_1+2}} \right], \text{ for iterations } s, l \times s, \cdots, l \in IN \\ \left[\hat{\sigma}_{1:k_1}^T \quad \hat{\sigma}_{k_1+1} \quad \hat{\sigma}_{k_1+2} \right], \text{ for other iterations} \end{cases}$$

where $\hat{\sigma}_{k_1+1}$ and $\hat{\sigma}_{k_1+2}$ are two scalars and *s* defines the step by which the extrapolation is applied.

Then the PageRank vector π is given by

$$\pi^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}$$

where

$$\sigma^T = \begin{bmatrix} \hat{\sigma}_{1:k_1}^T & \hat{\sigma}^T \begin{pmatrix} G_{11}^{12} \\ u_1^{(2)T} \\ (1-\alpha) v_1^{(2)T} \end{pmatrix} & \hat{\sigma}_{k_1+1} \end{bmatrix}. \quad \blacksquare$$

Proof:

Follows the proof of theorem 2.2 and at every s-step apply one Aitken extrapolation.

4. NUMERICAL EXPERIMENTS

As an illustrative example, a hyperlink matrix of dimension $10^4 \times 10^4$ with 65% of dangling nodes, 15% and 20%, respectively, weakly and strongly nondangling nodes is considered. The CPU time (in seconds), reported in Table 1, was considered as an average of the time required to compute 10 times the procedures.

The CPU time is greatly reduced with the use of Lumping techniques. A partition on dangling and nondangling nodes gives rise to a reduction of 16× without extrapolation and a reduction of 27× with extrapolation with respect to the time required by the classical power method. The partition of the nondangling nodes in strongly and weakly reduces the computation time in 4× and 8× considering, respectively, without and with extrapolation. Our combined versions, LumpingE, are always better than the original Lumping versions. It should be mentioned that the distinction between nondangling nodes does not provide, at least for the test matrix used, any advantage. The cost to recover the all PageRank vector from the partitioned one, along with the necessary cost to prepare data, is not compensated by the gains in aggregating the nodes in weakly and strongly nondangling.

The number of iterations for the Lumping approaches is smaller than the original power method. The reduction is particularly impressive for the new extrapolated versions. The reduction in the number of iterations with the LumpingE 2 method compared with the original Lumping 2 is significant. This is relevant for real problems, usually of high dimension, since one might be interested in obtaining an approximate solution after a few number of iterations. A computation with just a few iterations can be already enough to provide useful information.

The convergence history is depicted in Figures 2. and 3. For a clear understanding, one should mention that Aitken extrapolation in LumpingE 1 and 2 was taken every $s = 10$ iterations.

method	iterations	Time (s)
power	44	2.45
Lumping 1	37	0.15
Lumping 2	42	0.58
LumpingE 1	23	0.09
LumpingE 2	25	0.31

Table 1. Timings and Number of iterations for Lumping and LumpingE methods

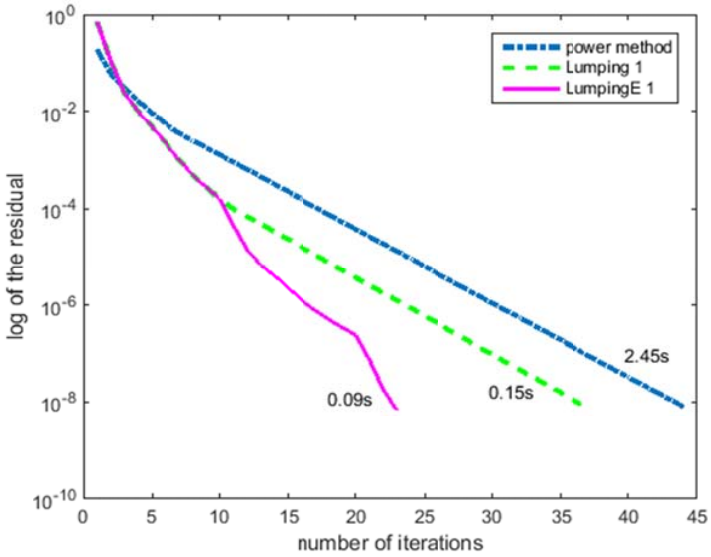


Figure 2. Convergence history for power, Lumping 1 and LumpingE 1 methods.

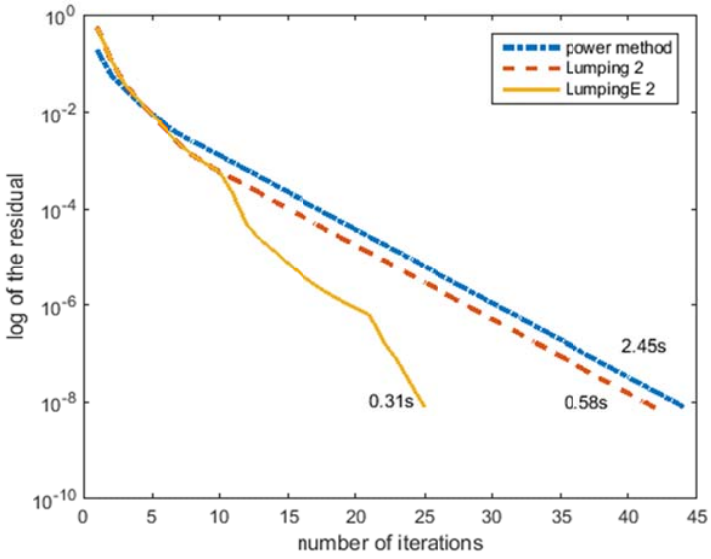


Figure 3. Convergence history for power, Lumping 2 and LumpingE 2 methods.

5. CONCLUSIONS

A family of LumpingE methods, combining partitioning, matrix reduction and extrapolation, is proposed to accelerate PageRank computations. Numerical results illustrating the dynamics of the iterative process, number of iterations and CPU time are provided. The new proposed family allows for a significant reduction both in number of iterations and CPU time required for convergence.

REFERENCES

[1] Brin, S., Page, L., "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, Vol. 30, no. 1, pp. 107–117, 1998.

[2] Golub, G., Greif, C., "An arnoldi-type algorithm for computing page rank," *BIT Numerical Mathematics*, Vol. 46, no. 4, pp. 759–771, 2006.

[3] Langville, A., Meyer, C., *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[4] Kamvar, S., Haveliwala, T., Manning, C., Golub, G., "Extrapolation methods for accelerating pagerank computations," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 261–270.

[5] Brezinski, C., Redivo-Zaglia, M., "The pagerank vector: Properties, computation, approximation, and acceleration," *SIAM Journal on Matrix Analysis and Applications*, Vol. 28, no. 2, pp. 551–575, 2006.

[6] Ipsen, I., Selee, T, "PageRank computation, with special attention to dangling nodes", *SIAM Journal on Matrix Analysis and Applications*, Vol. 29, no. 4, pp. 1281–1296, 2007.

[7] Lin, Y., Shi, X., Wei, Y., "On computing pagerank via lumping the google matrix," *Journal of Computational and Applied Mathematics*, Vol. 224, no. 2, pp. 702–708, 2009.

[8] Brezinski, C., Redivo Zaglia, M., "Generalizations of aitken's process for accelerating the convergence of sequences," *Computational & Applied Mathematics*, Vol. 26, no. 2, pp. 171–189, 2007.

[9] Mendes, I., Vasconcelos, P., "Lumping with acceleration for PageRank computation." *14th International Conference on Computational Science and Its Applications*. IEEE, pp. 221-224, 2014.

SYMCOMP 2015
Faro, March 26-27, 2015

SOLUTION OF NONLINEAR OPTIMAL CONTROL PROBLEMS WITH THE LANCZOS TAU METHOD

A. Gavina[1*] and J.M.A. Matos[2] and P. Vasconcelos[3]

1: Departamento de Matemática
Laboratório de Engenharia Matemática
Instituto Superior de Engenharia do Porto
Porto, Portugal
e-mail: alg@isep.ipp.pt

2: Instituto Superior de Engenharia do Porto and Centro Matemática
University of Porto
Porto, Portugal
e-mail: jma@isep.ipp.pt

3: Faculty of Economics and Centro Matemática
University of Porto
Porto, Portugal
e-mail: pjv@fep.up.pt

Keywords: Operational Tau method, Optimal control

Abstract. *The solution of optimal control problems (OCP) can be obtained following a standard procedure, which consists in applying Pontryagin's Maximum Principle and obtaining the necessary optimality conditions along with the transversality condition resulting into a two-point boundary value problem (BVP).*
The application of numerical methods to solve the BVP is a valid resource. In this work we investigate the application of the Operational Tau method [14] to obtain the approximate solution of optimal control problems, imposing that the initial or boundary conditions are verified exactly and that the approximate solution satisfies the differential equation with a residual minimized in a quadrature sense.
The Operational Tau method allows for the differential problem to be transformed into an algebraic problem, based on a reduced set of matrix operations. The accurate evaluation of those matrices is a sensitive task. In this work we use a recent technique [17] allowing an improvement in the precision of the Tau method, that is crucial in nonlinear problems. In [11] we have shown the application of the operational Tau method to linear OCP with infinite time horizon. In this work we treat nonlinear problems with finite and infinite time horizon and provide.

1 INTRODUCTION

In order to optimize trajectories, optimal control problems are widely used in multi-disciplinary applications: immunology, biological systems, applied mathematics, engineering and socio-economic systems [16]. The objective is to minimize/maximize a certain performance index, determining control and state trajectories for a dynamic system over a period of time. To obtain a solution of these dynamic systems described by linear and nonlinear differential equations, numerical methods are an available resource , in particular the use of spectral methods.
The aim of this work is to investigate the application of the Operational Tau method [14] to solve optimal control problems in finite and infinite horizont time. The Tau method is a spectral method that produces a polynomial approximation of the solution of the differential problem. The detailed description of the Tau method is made in Section 3 and in section 4 we present a brief characterization of optimal control problems with finite and infinite horizon. Section 5 we present numerical examples on how the approximate solution were obtained via the Tau method. Section 6 concludes.

2 PRELIMINAIRES AND NOTATIONS

Let v be the vector space of continuous integrable functions defined on some interval $[a, b]$ and let $\langle P_n, P_m \rangle$ be the inner product defined on v,

$$\langle P_n, P_m \rangle = \int_b^a P_n(x) P_m(x) w(x) dx \tag{1}$$

where P_n and P_m are polynomials of degree n and m, $\langle P_n, P_n \rangle = \|P_n^2\| \neq 0$ and $w(x)$ is a positive weighting function.

Definition 1. *The polynomials P_n and P_m are orthogonal if $\langle P_n, P_m \rangle = 0, \quad \forall n \neq m$*

Definition 2. *$P_0(x), P_1(x), P_2(x), \ldots$ is a sequence of orthogonal polynomials if any two different polynomials in the sequence are orthogonal to each other under the inner product (1).*

Definition 3. *The orthogonal polynomials satisfy a three-term recurrence relation [8]*

$$P_{i+1}(x) = a_i(x - b_i)P_i(x) - c_i P_{i-1}(x), \ i \geq 0$$

with $P_{-1}(x) = 0$ and $P_0(x) = 1$ with matricial representation given by

$$\begin{bmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ P_3(x) \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \ldots \\ \alpha_0 & a_0 & 0 & 0 & 0 & \ldots \\ \alpha_1 & \alpha_2 & a_0 a_1 & 0 & 0 & \ldots \\ \alpha_3 & \alpha_4 & \alpha_5 & a_0 a_1 a_2 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\Phi} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \end{bmatrix} \tag{2}$$

where $\alpha_j,\ j \geq 0$ *are real constants depending on* a_i, b_i *and* $c_i\ i = 0, \ldots, j$.

Definition 4. *Let* D *be a linear differential operator of order* $\nu \geq 1$ *with polynomial coefficients acting on* v *then* D *can be defined on* $]a, b[$ *by*

$$D \equiv \sum_{r=1}^{\nu} \frac{d^r p_r(x)}{dx^r} \tag{3}$$

Definition 5. *Let* $y(x) = \sum_{i\geq 0} a_i x^i$ *be a polynomial in the basis* $\{1, x, x^2, \ldots, x^n, \ldots\}$ *with coefficients* $\boldsymbol{a} = (a_0, a_1, a_2, \ldots, a_n, 0, 0, \ldots)$. *If* $\boldsymbol{x} = [1\ x\ x^2\ \ldots\ x^n\ \ldots]^t$ *is a vector then [14]*

- *the polynomial* $y(x)$ *can be written as* $y(x) = \boldsymbol{ax}$;
- *the effect of the multiplication of a polynomial by* x *is given by*

$$xy(x) = \boldsymbol{a}\mu\boldsymbol{x}$$

- *the effect of the differentiation of a polynomial is*

$$y'(x) = \boldsymbol{a}\boldsymbol{x}' = \boldsymbol{a}\eta\boldsymbol{x}$$

where μ *and* η *are infinite matrices representing, respectively, the multiplication and the differentiation operator given by*

$$\eta = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & \cdots \\ 0 & 2 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \mu = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Lemma 1. *The effect of, respectively,* α *repeated differentiations or* β *shifts on the coefficients of a polynomial* $y_n(x)$ *is equivalent to the post-multiplication of* $\boldsymbol{a}$ *by* η^α *or* μ^β.

Proof. See [14]. □

Definition 6. *Let* $y(x) = \sum_{i\geq 0} a_i P_i(x)$ *be a polynomial written in the basis of orthogonal polynomials* $\{P_0(x), P_1(x), P_2(x), \ldots\}$ *with coefficients* $\boldsymbol{a} = (a_0, a_1, a_2, \ldots, a_n, 0, 0, \ldots)$. *The polynomial* $y(x)$ *can be written as* $y(x) = \boldsymbol{a}\Phi\boldsymbol{x}$, *where* Φ *is the lower triangular matriz defined in (2) [14].*

3 TAU METHOD

In 1938 Lanczos presents in his work [1] the Tau method for the solution of ordinary differential equations. Later it was generalized for partial derivatives equations by Ortiz [2], Matos & al [3], Kong and Wu [4], etc. and for integro-differential equations by Hosseini [5, 6], Pour-Mahmoud & al [7], etc.
The Tau method is a spectral method that produces a polynomial approximation, $y_n(x)$, of the solution, $y(x)$, of a given differential problem $Dy(x) = f(x)$, defined on $]a, b[$. It is based on solving a system of linear algebraic equations, obtained by imposing that the supplementary conditions are verified exactly and the residual is minimized in a quadrature sense, i.e, is orthogonal to the first elements of an orthogonal polynomial basis. It can be applied to initial, mixed or boundary value problems.
Let D be the operator defined in (3) and $y(x)$ be the exact solution of the differential problem

$$\begin{cases} Dy(x) = f(x) & a < x < b \\ g_j(y) = \sigma_j & j = 1 : \nu \end{cases} \tag{4}$$

where $f(x)$ is a polynomial and $g_j(y)$ are ν linear functionals to be satisfied by the required solution $y(x)$, representing initial, boundary or mixed conditions, acting on $C^{\nu}[a, b]$.
The main idea of the Tau method is to approximate $y(x)$ by a polynomial $y_n(x) = \sum_{i=0}^{n} a_i P_i(x)$ where $\{P_i(x),\ i \geq 0\}$ is an orthogonal polynomial basis defined by a lower triangular matrix. The Tau approximant $y_n(x)$ to $y(x)$ is the unique polynomial solution of the perturbed problem

$$\begin{cases} Dy_n(x) = f(x) + \tau(x) & a < x < b \\ g_j(y_n) = \sigma_j & j = 1 : \nu \end{cases} \tag{5}$$

where τ is a polynomial perturbation close to zero in $]a, b[$, in the sense that the first $n + 1 - \nu$ coefficients in the P base are null. So,

$$\tau(x) = \sum_{i \geq n+1-\nu} \tau_i P_i(x)$$

3.1 Operational Tau method

The operational approach of the Tau method for differential equations allows to determinate the approximate solution $y_n(x)$ of the problem (5) without explicitly deal with the perturbation term τ. In fact, as we shall see, this term is a polynomial residual that results from the truncation of an algebraic system of linear equations. This is an operational version since the differential problem is transformed into an algebraic problem, using the matrices μ and η, and will be reduced to a set of matrix operations. The matricial representation of the action of the differential operator D is given by:

$$Dy_n(x) = \mathbf{a}\Pi\mathbf{x}, \quad \Pi = \sum_{i=0}^{v} \eta^i p_i(\mu). \tag{6}$$

Let $B = (B_1, B_2, \ldots, B_\nu)$ be the matrix with column components representing the supplementary conditions of (4) calculated in polynomial basis of the Tau approximant and $\Pi = (\Pi_0, \Pi_1, \Pi_2, ...)$ where Π_i is the i^{th} column of Π. If, in problem (4), $f(x) = \sum_{i=0}^{m} f_i P_i(x)$ is an m degree polynomial, or a polynomial approximation of a given function, then, to determinate the unknown coefficients, $\mathbf{a}$, of y_n the following algebraic infinite set of equations must be satisfied

$$\begin{cases} \mathbf{a}\Pi_0 = f_0 \\ \mathbf{a}\Pi_1 = f_1 \\ \cdots \\ \mathbf{a}\Pi_m = f_m \\ \mathbf{a}\Pi_{m+1} = 0 \\ \vdots \end{cases} \tag{7}$$

and

$$\begin{cases} \mathbf{a}B_1 = \sigma_1 \\ \mathbf{a}B_2 = \sigma_2 \\ \cdots \\ \mathbf{a}B_v = \sigma_\nu \end{cases} . \tag{8}$$

The system (7) - (8) is an overdetermined and infinite, with $n+1$ unknowns. If $n > m+\nu$ then this system can be conveniently truncated into

$$\mathbf{a}G = \mathbf{b}, \tag{9}$$

where $\mathbf{b} = (\sigma_1, \ldots, \sigma_\nu, f_0, \ldots, f_m, 0, \ldots, 0)$ and $G = (B; \Pi_0; \ldots; \Pi_{n-\nu})$ is the augmented matrix that associates the operator D with the supplementary conditions of the problem. The polynomial $y_n = \mathbf{a}P,\ P = (P_0, \ldots, P_n)^T$, defined by (9), is a Tau approximant to the solution of (4)[14].

4 OPTIMAL CONTROL PROBLEMS

We will consider optimal control problems with dynamics described by ordinary differential equations of the form

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_0, t_f] \\ x(t_0) = x_0 \end{cases} \tag{10}$$

where $x(t) \in \mathbb{R}^n,\ u(t) \in \mathbb{R}^m$ and $f : [t_0, t_f] \times \mathbb{R}^{n+m} \to \mathbb{R}^n$ is a continuous function in t, x and u and has continuous partial derivatives. The performance is given by

$$J(\cdot) = \int_{t_0}^{t_f} F(t, x(t), u(t))dt \tag{11}$$

where $F : [t_0, t_f] \times \mathbb{R}^{n+m} \to \mathbb{R}$.

The objective is to find the optimal controller u^* that maximize/minimize the performance J.
Considering the Hamiltonean function

$$H(t, x(t), u(t), \lambda_0, \lambda(t)) = \lambda_0 F(t, x(t), u(t)) + \boldsymbol{\lambda} f(t, x(t), u(t))$$

where F and f are the functions described above, λ_0 is a scalar and $\boldsymbol{\lambda} = [\lambda_1(t), \lambda_2(t), \ldots, \lambda_n(t)]$ is a vector of co-state variables and considering that (x, u^*) is a controlled trajectory defined over the interval $[t_0, t_f]$ then (x, u^*) is optimal, for all admissible controls u, if the the Pontryagin′s maximum principle holds, i.e.,

$$H(t, x, u^*, \boldsymbol{\lambda}) \geq H(t, x, u, \boldsymbol{\lambda})$$

The Pontryagin maximum principle garantees that if (x, u^*) is an optimal pair, solution of the problem (10)-(11), then the first order necessary conditions [16] satisfies the Hamiltonian maximization, i.e.,

$$\begin{array}{ll} i) & \lambda_0 = 1 \\ ii) & H_x = -\dot{\boldsymbol{\lambda}} \\ iii) & H_u = 0 \end{array} \tag{12}$$

with transversality conditions given by $\boldsymbol{\lambda}(t_f) = \mathbf{0}$ for $t_f < \infty$.
In most problems the equation $iii)$ can be manipulated to find a representation of u in terms of t, x and $\boldsymbol{\lambda}$ and substituted into the first equation of the dynamics (10). The equation obtained together with $ii)$ and the the conditions $x(t_0) = x_0$, $\boldsymbol{\lambda}(t_f) = \mathbf{0}$ form the boundary value problem to be solved.

5 NUMERICAL EXAMPLES

In this work we focus on two applications of nonlinear optimal control problems . The first one is defined in finite interval $[t_0, t_f]$ and its dynamics is a variant of Van der Pol oscillator [19]. The second application is an infinite optimal control problem and relates to the Euler equation for the angular velocities of a spacecraft [13]. For both problems the trajectories were obtained using the operational Tau method. In order to have a numerical comparational, we also use the Matlab bvp4c solver. In this solver he range of integration is divided into several subintervals and for each subinterval bvp4c computes a cubic polynomial function that collocates at the ends and at the midpoint, i.e., Bvp4c can be viewed as collocation with C^1 piecewise cubic polynomial that satisfies the boundary conditions.

5.1 Optimal control model for the Rayleigh equation

Considering the unconstrained Rayleigh [12] optimal control problem

$$\min \int_0^T (u^2(t) + x_1^2(t))dt \tag{13}$$

subject to

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -x_1(t) + x_2(t)(1.4 - px_2^2(t))x_2(t) + 4u(t) \\ x_1(0) = -5 \\ x_2(0) = -5 \end{cases} \tag{14}$$

The trajectories of the problem (13)-(14) will be obtained after applying the Hamiltonean, calculating the necessary conditions and the transversality conditions at the final time $T = 4.5$ with parameter $p = 0.14$, i.e., solving the BVP

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 + 1.4x_2 - px_2^3 - 8\lambda_2 \\ \dot{\lambda}_1 = \lambda_2 - 2x_1 \\ \dot{\lambda}_2 = 3px_2^2\lambda_2 - \lambda_1 - 1.4\lambda_2 \\ x_1(0) = -5 \\ x_2(0) = -5 \\ \lambda_1(4.5) = 0 \\ \lambda_1(4.5) = 0 \end{cases} \tag{15}$$

To implement the operational Tau method, first we need to linearize the second and the fourth differential equations in the problem (15). Using the Taylor polynomial approximation, the expressions x_2^3 and $x_2^2\lambda_2$ will be replaced , respectively, by $3x_{20}^2x_2 - 2x_{20}^3$ and by $2x_{20}\lambda_{20}x_2 + x_{20}^2\lambda_2 - 2x_{20}^2\lambda_{20}$ into the system (15), where $x_{2,0}$ and $\lambda_{2,0}$ are initial approximations to x_2 and λ_2. Thus the problem becomes

$$\begin{cases} \dot{x}_1 - x_2 = 0 \\ \dot{x}_2 + x_1 - 1.4x_2 + 8\lambda_2 + 3px_{20}^2x_2 = 2p2x_{20}^3 \\ \dot{\lambda}_1 - \lambda_2 + 2x_1 = 0 \\ \dot{\lambda}_2 + \lambda_1 + 1.4\lambda_2 - 6px_{20}\lambda_{20}x_2 - 3px_{20}^2\lambda_2 = -6x_{20}^2\lambda_{20} \\ x_1(0) = -5 \\ x_2(0) = -5 \\ \lambda_1(4.5) = 0 \\ \lambda_2(4.5) = 0 \end{cases} \tag{16}$$

The matricial representation of (16), is given by

$$\Pi = \begin{bmatrix} \eta & I & 2I & 0 \\ -I & \eta - 1.4I + 3p\left(x_{20}(\mu)\right)^2 & 0 & -6px_{20}(\mu)\lambda_{20}(\mu) \\ 0 & 0 & \eta & I \\ 0 & 8I & -I & \eta + 1.4I - 3p\left(x_{20}(\mu)\right)^2 \end{bmatrix} \tag{17}$$

with

$$\mathbf{f} = \begin{bmatrix} 0 \\ 2px_{20}^3 \\ 0 \\ -6x_{20}^2\lambda_{20} \end{bmatrix} \tag{18}$$

where $x_{20}(\mu)$ and $\lambda_{20}(\mu)$ are matrices evaluated as linear combinations of powers of matrix μ with the coefficients of polynomial $x_{2,0}$ and $\lambda_{2,0}$. That is, for instancies, if $x_{2,0}(x) = \sum_{i=0}^{n} a_i x^i$ then $x_{2,0}(\mu)$ is also a polynomial in terms of powers of μ, $x_{2,0}(\mu) = \sum_{i=0}^{n} a_i \mu^i$.
The Tau approximants will be determinated solving the system (9) where $\mathbf{b} = [-5, -5, 0, 0, \mathbf{f}]$ and $G = [B; \Pi_0; \ldots; \Pi_{n-\nu}]$, B is the matrix that represents the boundary conditions of (16) calculated in polynomial basis of the Tau approximant.
With the Operational Tau method, we have implemented an iterative procedure, replacing, from the second iteration on, the polynomials $x_{2,0}$ and $\lambda_{2,0}$ by their homonyms x_2 and λ_2 obtained in the previous iteration. This operation actually means that, in each iteration we need to update the matrix G and the vector b components with this new polynomials expressions.

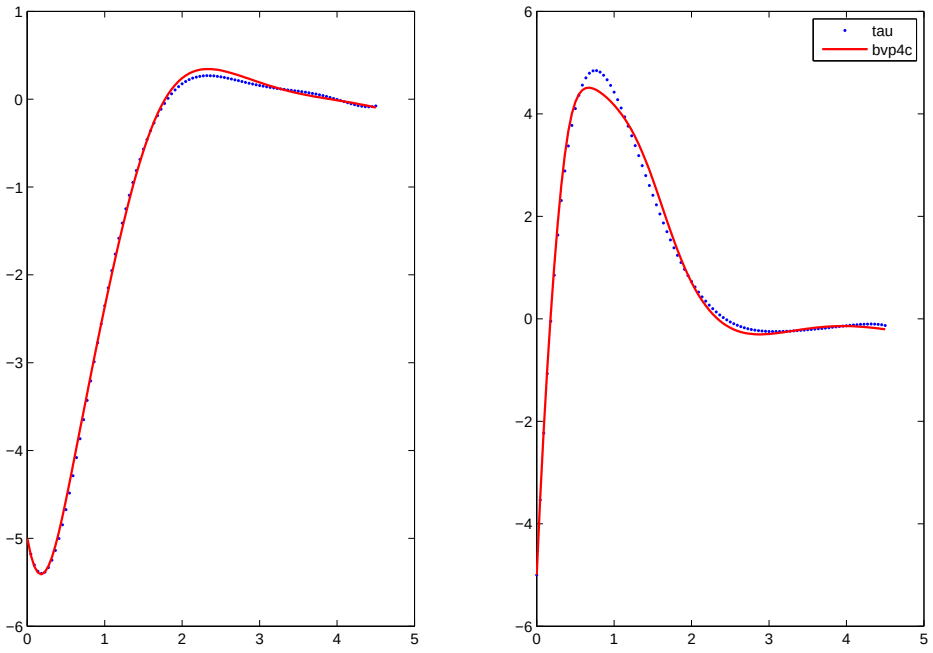


Figure 1: Curves of the state trajectories for the Rayleigh problem.

Our numerical experiments were obtained with the Legendre polynomials $P_n(x)$ with $n = 6$. In Figure 1, we can see, the trajectories curves for x_1 and x_2 obtained wiyh the operational

Tau method and with the Matlab solver bvp4c. It can be seen that both methods produced approximations with similar numerical behavior. In addition, we can refer that in terms of the machine computation time, the Tau method spented about 30% of the time spented by bvp4c, to produce the approximants pictured in that figure. Our rotine needs 20 iterations, in each one the main cost is the recalculation of an $4(n+1)$ square matrix G and the resolution of a linear system associated with that matrix.We have also observed that the Tau method seems to be less sensitive to changes in the parameter p. For example, for $p = 1.14$ we still have solution for the Rayleigh problem while in bvp4c we need to use more than 2500 mesh points to meet the tolerance.

5.2 Optimal manouvres of a rigid asymetric spacecraft

Let

$$J = \int_0^\infty \frac{x_1^2 + x_2^2 + x_3^2 + u_1^2 + u_2^2 + u_3^2}{2} dt \tag{19}$$

be the infinite cost functional to minimize, subject to

$$\begin{cases} \dot{x}_1 = A_1 x_2 x_3 + \frac{1}{I_1} u_1 \\ \dot{x}_2 = A_2 x_1 x_3 + \frac{1}{I_2} u_2 \\ \dot{x}_3 = A_3 x_1 x_2 + \frac{1}{I_3} u_3 \end{cases} \tag{20}$$

with initial conditions

$$x_1(0) = 0.01, \quad x_2(0) = 0.005, \quad x_3(0) = 0.001 \tag{21}$$

The parameters $I_1 = 86.24, I_2 = 85.07, I_3 = 113.59$ are the same as in [18] and $A_1 = -\frac{I_3 - I_2}{I_1}$, $A_2 = -\frac{I_1 - I_3}{I_2}$, $A_3 = -\frac{I_2 - I_1}{I_3}$.
As in the previous problem, after applying the Hamiltonean, calculate the necessary and the transversality conditions, given by $\lambda_i(\infty) = 0,\ i = 1, 2, 3$, we get a nonlinear BVP.
Again, in order to remove the nonlinear component, products in the form xy will be replaced by $xy \approx y_0 x + x_0 y - x_0 y_0$. Then, the matricial representation of the resulting BVP, according to (6) is the matrix

$$\Pi = \begin{bmatrix} \eta & -A_2 x_{30}(\mu) & -A_3 x_{20}(\mu) & I & A_3\lambda_{30}(\mu) & A_2\lambda_{20}(\mu) \\ -A_1 x_{30}(\mu) & \eta & -A_3 x_{10}(\mu) & A_3\lambda_{30}(\mu) & I & A_1\lambda_{10}(\mu) \\ -A_1 x_{20}(\mu) & -A_2 x_{10}(\mu) & \eta & A_2\lambda_{20}(\mu) & A_1\lambda_{10}(\mu) & I \\ \frac{1}{(I_1)^2} I & 0 & 0 & \eta & A_1 x_{30}(\mu) & A_1 x_{20}(\mu) \\ 0 & \frac{1}{(I_2)^2} I & 0 & A_2 x_{30}(\mu) & \eta & A_2 x_{10}(\mu) \\ 0 & 0 & \frac{1}{(I_3)^2} I & A_3 x_{20}(\mu) & A_3 x_{10}(\mu) & \eta \end{bmatrix}$$

with

$$\mathbf{f} = \begin{bmatrix} -A_1 x_{20} x_{30} \\ -A_2 x_{10} x_{30} \\ -A_3 x_{10} x_{20} \\ A_2 x_{30}\lambda_{20} + A_3 x_{20}\lambda_{30} \\ A_1 x_{30}\lambda_{10} + A_3 x_{10}\lambda_{30} \\ A_1 x_{20}\lambda_{10} + A_2 x_{10}\lambda_{20} \end{bmatrix}$$

where $x_{i0},\ \lambda_{i0}$ represents the initial polynomial approximations for x_i, λ_i and $x_{i0}(\mu),\ \lambda_{i0}(\mu)$ are matrices evaluated as linear combinations of powers of matrix μ with the coefficients of polynomials $x_{i0},\ \lambda_{i0}$ for $i = 1, 2, 3$. In this problem, $\mathbf{b} = [10^{-2}, 5 \times 10^{-3}, 10^{-3}, 0, 0, 0, \mathbf{f}\,]$, $G = [B; \Pi_0; \ldots; \Pi_{n-\nu}]$, B is the matrix that represents the boundary conditions of (16) calculated in polynomial basis of the Tau approximant. Once, again we solve the system given by (9) to obtain the Tau approximants.
We got our numerical experiments using Chebyshev polynomials $T_n(x)$ with $n = 9$. To obtain the approximate solution were needed only 10 iterations each one representing a linear matrix $2(n+1)$ dimension. In Figure 2, we represent together the polynomial aproximation of $x_i,\ i = 1, 2, 3$, obtained using the operational Tau method, and the solution given by the bvp4c solver in Matlab for $t = 1000$. Comparing both representations of the approximate solutions the resultas are numerically close.

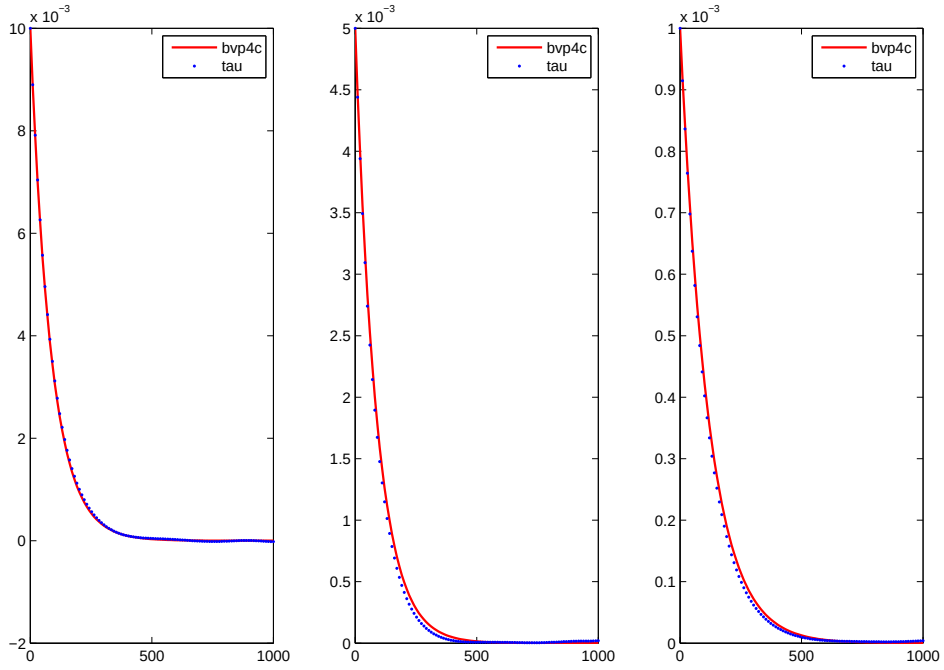


Figure 2: Curves of the state trajectories of a rigid asymetric spacecraft.

6 CONCLUSION

In this work we present the operational Tau method as a viable numerical approach to solve finite and infinite nonlinear optimal control problems. Numerical results illustrate that optimal control problems can be solved using the operational Tau method, benefiting from a polynomial approach. Even with low degree polynomials, the Tau method provides approximate solutions

comparable with robust and efficient solvers such as bvp4c from Matlab. Since we need to change the basis of the polynomials involved in the operational Tau method, the matrices, in nonlinear problems, become ill conditioned as we increase de degrees of these polynomials. The drawback is that, with increasing n polynomials degrees this method become unstable. For future work we intend to introduce some numerical stability in these matrices.

REFERENCES

[1] Lanczos, C., *Interpolation of empirical and analytical functions*, J. Math. Phys., 17, 123-199, (1938).

[2] Ortiz, E. L., *The Tau method*, SIAM J. Numer. Ana., 6, 480-492 (1969).

[3] Matos, J.; Rodrigues, M. J. and Vasconcelos, P. B., *New implementation of the Tau method for PDE's*, Journal of computational and applied mathematics, Elsevier, 164, 555-567 (2004).

[4] Kong, W. and Wu, X., *Chebyshev Tau matrix method for Poisson-type equations in irregular domain*, Journal of Computational and Applied Mathematics, Elsevier, 228, 158-167 (2009).

[5] Hosseini, S. M. and Shahmorad, S., *Numerical solution of a class of integro-differential equations by the Tau method with an error estimation*, Applied mathematics and computation, Elsevier, 136, 559-570 (2003).

[6] Hosseini, S. M. and Shahmorad, S., *Tau numerical solution of Fredholm integro-differential equations with arbitrary polynomial bases*, Applied Mathematical Modelling, Elsevier, 27, 145-154 (2003).

[7] Pour-Mahmoud, J.; Rahimi-Ardabili, M. and Shahmorad, S., *Numerical solution of the system of Fredholm integro-differential equations by the Tau method*, Applied Mathematics and Computation, Elsevier, 168, 465-478 (2005)

[8] Gautschi, W., *Orthogonal Polynomials: computation and approximation*, Oxford university press, (2004)

[9] Adeniyi, R.; Onumanyi, P. and Taiwo, O. *A computational error estimate of the tau method for nonlinear ordinary differential equations*, J. Nig. Mathssoc, 9, 21-32 (1990)

[10] Matos, J.M.A, Rodrigues, M.J., Matos, J.C.M, Cruz, M. *Avoid Similarity Transformations in the Operational Tau method*, *submitted* , 2014

[11] Gavina, A., Matos, J., Vasconcelos, P.B, *Tau Method for Linear Quadratic Regulator Problems*, *Journal of Applied Nonlinear Dynamics* Vol. **3(2)** , pp. 139-146, 2014

[12] Maurer, H. and Oberle, H. J. *Second order sufficient conditions for optimal control problems with free final time: The Riccati approach* SIAM journal on control and optimization, SIAM, 41, 380-403 (2002)

[13] Junkins, J. L. and Turner, J. D., *Optimal spacecraft rotational maneuver*, Elsevier, (1986)

[14] Ortiz, E. L. And Samara, H., *An operational approach to the Tau method for the numerical solution of non-linear differential equations*, Computing, vol. 27, 15-25, (1981).

[15] Fahroo, F. and Ross, I.M., *Pseudospectral Methods for Infinite-Horizon Nonlinear Optimal Control Problems*, Journal of Guidance, Control, and Dynamics, Vol. 31, No. 4, pp. 927-936 (2008).

[16] Athans, M. and Falb, P.L., *Optimal Control: An introduction to the theory and its applications*, Dover Publications, Inc, (2007).

[17] Rodrigues, Maria João and Matos, José, *A tau method for nonlinear dynamical systems*, Numerical Algorithms, pp. 1-18 (2012).

[18] Jajarmi, A., Pariz, N., Effati, S., and Kamyad, A. V., *Solving infinite horizon nonlinear optimal control problems using an extended modal series method*, Journal of Zhejiang University SCIENCE C, Vol. 12, No. 8, pp. 667-677 (2011)

[19] Guckenheimer, J., *Dynamics of the van der Pol equation Circuits and Systems*, IEEE Transactions on, IEEE, 27, 983-989 (1980)

SYMCOMP 2015
Faro, March 26-27, 2015

NUMERICAL SOLUTION OF NEW NEOCLASSICAL SYNTHESIS' MODELS

João C. Rocha[1*], Paulo B. Vasconcelos[2] & Pedro Gil[3]

1: Economics Master
Faculty of Economics
Porto, Portugal
e-mail: jpppcr@gmail.com

2: Faculty of Economics and CMUP
University of Porto
Porto, Portugal
e-mail: pjv@fep.up.pt

3: Faculty of Economics and CEF.UP
University of Porto
Porto, Portugal
e-mail: pgil@fep.up.pt

Keywords: Projection Methods, Perturbation Methods, New Neoclassical Synthesis models

Abstract.
New Neoclassical Synthesis models incorporate the prominent role of expectations from the New Classical school, the intertemporal optimising dynamic framework of Real Business Cycles theory, and distortions arising from New Keynesian real and nominal rigidities. Based on microeconomic foundations, and therefore less subject to Lucas' critique, these models are increasingly used as forecasting tools in economics and for policy analysis. Their analytic solution is, however, only possible for certain strict and simpler cases, and as such it is common to resort to numerical approximations.
This work focuses on numerical solution methods for this family of economic models, namely by applying projection methods instead of the most common perturbation approach based on Taylor's expansion. Numerical results on a Dynamic Stochastic General Equilibrium model with staggered prices illustrating this approach is provided.

1 Introduction

New Neoclassical Synthesis (NNS) models, as described by Goodfriend and King [6], Woodford [16], or Clarida et al. [4], emerged in the 1990s as a result of the combination of elements from the Real Business Cycles (RBC) school and New Keynesian school.
From the RBC school, NNS models inherited its dynamic and stochastic framework, which on its turn was based on the standard neoclassical growth model. Within that framework, RBC models featured economic cycles, which arose both from microeconomic decisions of intertemporal profit or utility maximisation from representative agents, and from stochastic behaviour of real variables such as productivity. This intertemporal maximisation depended not only on the expected result of agents' own actions, but also on the expected behaviour of other agents, namely fiscal or monetary authorities modelled in the form of policy rules.
On top of this mathematical framework, some authors began incorporating Keynesian and New Keynesian features, including real rigidities such as imperfect competition, or nominal rigidities such as staggered or sticky price and wage setting. These additions were seen as necessary to improve the way these models reflected real data, and to restore a degree of effectiveness to monetary policy, seen by the RBC school as having little or no effect.
In the context of the New Classical school, Lucas [11] argued against models estimated from historical statistical data as tools for forecasting the effect of economic policies. To support this criticism, Lucas cited the lack of consideration, in the referred practice, for the role of the change of agents' expectations when faced with a given policy, in estimating the outcome of that same policy. In this context, RBC models' microeconomic foundations were seen as making them less subject to Luca's critique. By incorporating the expected actions of authorities into the way how representative agents acted, any possible change in the former's behaviour would already be reflected in the latter's current decisions.
This perceived robustness was inherited by NNS models, which had incorporated the referred Keynesian rigidities as deriving, too, from microeconomic behaviour of households and firms. This, together with the added realism from the referred rigidities, turned them into popular tools of policy analysis and economic forecasting.
Dynamic Stochastic General Equilibrium (DSGE) models, which include the NNS, however, do not always have analytic solutions, except for the case of a particular set of models and restricted to smaller dimensions, always in the presence of certain function specifications. Numerical methods are thus the key answer to solve this type of problems.
The solution of DSGE models is usually defined either by a set of Euler equations, resulting from the application of the Lagrange multipliers method, or a policy function, resulting from the use of dynamic programming. Numerical solution methods can be categorised by the kind of solution definition they apply to, that is, the type of function they attempt to approximate: Euler equations or policy function. A second categorisation of methods can be made based on whether the approximation is local (only valid near a point) or

global (valid in the whole domain).
One of the most widely used approximations is the family of perturbation methods, which rely on Taylor series expansions of different orders of the problem's Euler equations in the neighbourhood of a point, usually its steady-state. Often a single Taylor expansion is used, as in the case of linearisation or log-linearisation – two particular applications of this family of methods – which may exclude important features of the original problems. For this reason, second and higher order have been proposed and are applied. These techniques are fast but their locality precludes the analysis of larger shocks that imply bigger deviations from the point of approximation.
Discrete State Space methods, on the other hand, are global methods that can be applied to policy function problems. They involve approximating the referred function through a recursive iterative process, called policy function iteration, after imposing a fixed grid of possible values for its variables. Although global and always converging to the true function provided that enough iterations are considered, this method can become computationally intensive with an increase of the complexity of the problem, or when finer grids are employed.
This work follows a third approach, consisting projection methods, a global method with similarities to regression that applies to functional problems, such as those defined by either policy function or Euler equations [8]. This method starts with the choice of a generic approximation function, such as Chebychev polynomials or Artificial Neural Networks, which is meant to replace the unknown of the functional problem. Then, a measure of distance between the approximated and the true function, or a residual function, is established. Finally, a method of parameterising the generic function in order to minimise the referred residual is chosen, such as the Least squares method, Chebychev, Galerkin or other collocation method. Being a global method, the projection method does not preclude the analysis of larger shocks. Additionally, depending on the approximation function used, the full nonlinearities of the problem can be preserved. However, the larger the model or the number of approximation function parameters to be computed, the higher the computational cost is.
In this work we extend a model based on Lim and McNelis [10], similar to the standard NNS or New Keyensian model, by considering staggered adjustment of prices. The use of projection method is explained. Finally, a set of simulations and impulse-response shocks are used to showcase the model, extension, method and application.

2 Numerical solution methods for New Neoclassical Synthesis models

The solution to the underlying optimisation problems of NNS, and Dynamic General Equilibrium models in general, is usually obtained via dynamic programming [13] or via application of the Kuhn-Tucker theorem or Lagrange multipliers method [3]. Both approaches turn the optimisation problem into a functional one. In the case of dynamic programming, its solution is the policy function, which returns the optimising choice of control variables, for each given value of the state variables. Similarly, the Lagrange

multipliers approach results in one or more Euler equations, forming a system of difference equations (differential equations, for continuous time) which describe the optimal path taken by the control variables that solves the optimisation problem.
Regardless of the solution approach, analytic solutions for these models only exist in certain restrictive conditions, such as certain utility or production function specifications, described in Canova [2] or Heer and Mauß ner [7], for example. Given the impossibility or, in certain cases, the difficulty of obtaining such analytic solutions, it is common to resort to approximation techniques instead.
Heer and Mauß ner [7] categorise approximation techniques using two types of features. The first is related to the function it tries to approximate: policy function or Euler equation. The second is concerned with the kind of approximation, which can be local or global. Local techniques try to obtain a function that only approximates the true one around a given point, usually its steady-state, contrasting with global ones, which aim to obtain valid approximations for the whole domain.
Linear approximation is a common local method, which involves turning the original problem's first order conditions or constraints into approximate linear substitutes. Different linearisation techniques exist, including Log-linear and Linear quadratic (LQ) approximation. Log-linearisation applies to both the Euler equation and budget constraints, which are approximated by taking logarithms and first-order Taylor series expansions around the problem's steady-state, subsequently solving the resulting linearised system. Uhlig [15] is an example of this approach. LQ approximation, on the other hand, consists of quadratic approximations for the problem's objective function and linear approximations for its constraints, once again around its steady-state.
These linear approximation methods are included in the larger group of perturbation methods, which also resort to Taylor expansions of different orders around the problem's steady-state. They rest on the assumption that the modelled economy never diverges too far away from its steady-state, so that higher order members of the Taylor series can be ignored without severe accuracy loss. These methods are less computationally intensive than others, and in most cases the assumption is acceptable, and the approximation good enough. As noted by Kim and Kim [9], however, although a wide range of the literature involving DSGE models relies on linear approximations for their solution, their use may result in high approximation errors and distort the results of their analysis. For this reason, several second and higher-order perturbation methods and algorithms have been advanced [14, 12].
Another kind of techniques, Discrete State Space methods, also called Discretisation, are global methods that can be applied to policy function problems. According to Canova [2], they generally involve forcing the states and exogenous variables to take values from a fixed grid of possible values. This, on the other hand, facilitates obtaining the policy function through a recursive iterative process called policy function iteration. This approach always converges to the true policy function. However, with an increase of the number of variables or shocks, the problem may become too highly computationally intensive, in what is known

as the curse of dimensionality. Furthermore, this approach involves a trade-off between speed and accuracy: the finer the grid, the better the approximation, but the higher the number of computations involved.
A third alternative, the one applied in this work, are Projection methods. Described in Judd [8], these are global methods that apply to problems with solution defined either by policy function or Euler equations. They involve choosing an approximation function for each of the decision rules, which in turn is tuned within an iterative process controlled via a chosen residual function. This way, the functional problem is turned into one of minimising the residual function. Variations of the method are possible given the different existing choices of approximating function (Chebychev polynomials or neural networks, for example), residual function or minimisation procedure (Least squares, Galerkin method or other collocation methods, for example). This approach, like other global methods, does not rely on the smallness of shocks for a good approximation. Additionally, the validity of the approximation it not restricted to the neighbourhood of the point around which it is made. However, the better approximation comes with a greater computational cost, especially when a large number of parameters has to be calculated, such as in the case of higher order polynomials or more complex neural networks.

3 Model exposition and numerical solution

As previously mentioned, the particular model to be studied in the present work is based on a flexible prices baseline model described by Lim and McNelis [10]. It consists of a dynamic in discrete time, stochastic general equilibrium model. In this work this baseline model is modified with the addition of sticky, or staggered, prices, and the corresponding numerical solution developed in detail.
The model is built upon three main sectors in an open economy: households, firms and monetary authorities; and two smaller ones, namely fiscal authorities and rest of the world.
The household sector decides, at each time period, between consumption and leisure, subject to a budget constraint. This budget constraint includes, among other items, the firm's profits, given that the former are fully owned by households. The price dynamics of the model stem from production and pricing decisions of the domestic (consumption and intermediate) goods producing firms, and the monetary authority acts on inflation by setting the interest rate (via a simple Taylor rule). On its turn, the fiscal authority is presented as an exogenously defined amount of government spending and lump-sum taxes. Finally, capital or producer goods, which fully depreciate at the end of each time period, are imported by households from the rest of the world, being subsequently lent to domestic firms.
There are several ways by which sticky prices are usually included in NNS models. The one chosen here is a Calvo rule [1], which in practice separates, for each time period, intermediate goods firms into two groups: one allowed to change prices, and another that is not. Accordingly, firms that can change prices will do so by setting the price that

maximises its present and discounted future profits, taking into account both the demand they face and the probability of being able to set prices again in the future. The ability to set prices, on its turn, arises from the assumption of imperfect competition in the intermediate goods' market, caused by imperfect substitutability between differentiated products.

3.1 Rest of the world

The rest of the world comes into the model both as a producer of capital goods, as an importer and as a lender of one period bonds. Capital goods are imported by households and subsequently rented to firms. Exports to the rest of the world, on its turn, are introduced via a constant exogenous variable $X = \overline{X}$. The external balance relationship is given by the equation [1]:

$$P_t X_t - S_t P_t^F I_t = \left(1 + R_{t-1}^* + \Phi_{t-1}\right) S_t F_{t-1} - S_t F_t \tag{1}$$

with the variables representing:

- P_t, the domestic price level;
- S_t, the exchange rate (expressed in indirect quotation);
- I_t, the capital imports, and P_t^F, the corresponding price;
- R_t^*, the external interest rate;
- F_t, the stock of bonds issued by households in the rest of the world, or external debt;

Finally, the interest paid on external debt bonds includes Φ_t, a risk premium, which increases as the external financial position, or external debt, increases above its steady-state $\overline{F}$ (according to sensitivity parameter $\varphi > 0$), and is given by

$$\Phi_t = sign\left(F_t\right) \times \varphi \left[e^{\left(|F_t| - \overline{F}\right)} - 1\right] \tag{2}$$

[1] The left side of (1) being the trade balance, and the right side the change in the economy's external debt

3.2 Households

The household sector's choice between consumption and leisure translates, in practice, to an optimisation problem as follows.
Firstly, the representative household values its present and future consumption according to the function

$$V = E_0 \sum_{t=0}^{\infty} \beta^t U_t \left(C_t, L_t \right) \tag{3}$$

where its preferences are represented by U_t, the utility function, which is of the type

$$U_t(C_t, L_t) = \frac{C_t^{1-\eta}}{1-\eta} - \frac{L_t^{1+\varpi}}{1+\varpi} \tag{4}$$

Value function equation (3) constitutes the expected present value, given the discount factor β, for the representative household, of both current and all future utility. This utility, as seen on equation (4), depends positively on the amount of consumption C_t, and negatively on the amount of labour L_t of the corresponding period. The utility function is of Constant Relative Risk Aversion type. The measure of the risk aversion curve, η, is the relative risk aversion coefficient (or elasticity of marginal utility of consumption), and ϖ the elasticity of marginal disutility of labour. C_t is an index of consumption goods, given on its turn by (5), a Dixit-Stiglitz, or Armington, aggregator [5].

$$C_t = \left[\int_0^1 \left(C_{j,t} \right)^{(\zeta-1)/\zeta} dj \right]^{\zeta/(\zeta-1)} \tag{5}$$

Here, an infinite number of goods, indexed by j, become perfect substitutes when ζ, the constant elasticity of substitution coefficient, approaches infinity, and perfect complements when it approaches zero.
The choice that maximises the utility function above is, on other hand, restricted by the budget constraint

$$\begin{aligned} W_t L_t + \Pi_t + P_t^K K_t + \left(1 + R_{t-1}\right) B_{t-1} + S_t F_t \\ = P_t C_t + P_t^F I_t + B_t + \left(1 + R_{t-1}^* + \Phi_{t-1}\right) S_t F_{t-1} + T_t \end{aligned} \tag{6}$$

with the variables representing:

- W_t, the wages;

- Π_t, the firms' profits;
- R_t, the domestic interest rate;
- B_t, the stock of domestic government bonds bought by households;
- T_t, the amount of lump-sum taxes.
- I_t, the capital imports, which equals K_t since the later fully depreciates at the end of each period; and P_t^F, the price of the said imports, which equals P_t^K;
- K_t, the stock of capital goods, which fully depreciates at each period, and as such equals I_t, the capital imports;
- P_t^K, the capital goods' price, charged by households to firms;

Setting up the Lagrangian for the maximisation of equation (3) with respect to (6), solving the first order conditions and rearranging, results in the Euler equations of the problem, describing the path that solves the intertemporal maximisation problem:

$$\begin{aligned} \frac{C_t^{-\eta}}{P_t} &= \frac{C_{t+1}^{-\eta}}{P_{t+1}}\beta\left(1+R_t\right) && (7)\\ W_t &= L_t^{\varpi} P_t C_t^{\eta} && (8)\\ P_t^F &= P_t^K && (9)\\ \left(1+R_t\right) S_t &= \left(1+R_t^* + \Phi_t' F_t + \Phi_t\right) S_{t+1} && (10) \end{aligned}$$

3.3 Firms

The production sector is made up of two types of firms: intermediate goods firms, and final goods firms. The former hire labour and capital goods from households, paying wages and the capital rent price in return, combining both into intermediate goods. The later buy the referred intermediate goods, which are combined into a final good subsequently sold to households. Intermediate goods firms behave in a monopolistic competition setting, having a degree of market power given the imperfect substitutability between their products.

Final goods firms

As previously described, the final goods firms take intermediate goods, $Y_{j,t}$ (with the different firms being indexed by $j \in [0;1]$), as production inputs, transforming them into a single final good, Y_t, according to the aggregating function

$$Y_t = \left[\int_0^1 (Y_{j,t})^{(\zeta-1)/\zeta}\, dj\right]^{\zeta/(\zeta-1)} \tag{11}$$

Taking both final good P_t and intermediate goods price $P_{j,t}$ as given, it maximises profits:

$$\max_{Y_{j,t}} \quad P_t Y_t - \int_0^1 P_{j,t} Y_{j,t} dj \tag{12}$$

Setting up and solving the first order condition, while taking (11) into account, leads to (13), which can be seen as the demand faced by intermediate goods firms as the final goods firm produces Y_t:

$$Y_{j,t} \quad = \left(\frac{P_{j,t}}{P_t}\right)^{-\zeta} Y_t \tag{13}$$

Intermediate goods firms

Each intermediate goods firm, on its turn, employs, in each period t, capital goods ($K_{j,t}$) and labour ($L_{j,t}$) as factors of production, producing $Y_{j,t}$ according to the Constant Elasticity of Substitution production function

$$Y_{j,t} = Z_t \left[(1-\alpha)\,(L_{j,t})^{\kappa} + \alpha (K_{j,t})^{\kappa}\right]^{1/\kappa} \tag{14}$$

The transformation $\frac{1}{1-\kappa}$, $0 < \kappa < 1$, stands for the elasticity of substitution between the two factors of production, and $0 < \alpha < 1$ for the usage share of the same factors. Z_t, on the other hand, is a random productivity shock, which follows an autoregressive process around its steady-state value $\overline{Z}$, subject to a random disturbance ϵ_t^Z as follows:

$$\ln(Z_t) = \rho \ln(Z_{t-1}) + (1-\rho) \ln\left(\overline{Z}\right) + \epsilon_t^Z,\ \epsilon \sim N\left(0, \sigma_Z^2\right) \tag{15}$$

Profit maximisation, in this case, implies (constrained) minimisation of total costs:

$$\begin{aligned} \min_{L_{j,t}, K_{j,t}} \quad & TC_{j,t} = W_{j,t} L_{j,t} + P_{j,t}^K K_{j,t} \\ s.t. \quad & (14) \end{aligned}$$

Setting up the Lagrangian, solving and rearranging the first order conditions results into the optimal factor combination equation:

$$K_{j,t} = \left(\frac{W_t}{(1-\alpha)} \cdot \frac{\alpha}{P_t^K}\right)^{\frac{1}{1-\kappa}} L_{j,t} \tag{16}$$

Combining the above equation with production function (14), and solving for L and K, returns the firm's conditional demands for labour and capital, respectively:

$$\begin{aligned} L_{j,t} &= \left(\frac{Y_{j,t}}{Z_t}\right)\left[(1-\alpha)+\alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\frac{\kappa}{(1-\kappa)}}\right]^{-\frac{1}{\kappa}} \\ K_{j,t} &= \left(\frac{Y_{j,t}}{Z_t}\right)\left[\alpha+(1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\frac{\kappa}{(1-\kappa)}}\right]^{-\frac{1}{\kappa}} \end{aligned} \tag{17}$$

Amongst intermediate goods firms, the ones that can change prices do so by maximising the present value of their profits. Being θ the probability of a given intermediate goods firm changing its price on a given period, and $P_{j,t}^A$ its chosen price, the profit maximisation problem becomes:

$$\max_{P_{j,t}^A} \quad V = E_t \sum_{t=0}^{\infty} \theta^t \beta^t \left(P_{j,t}^A Y_{j,t} - \left(W_{j,t} L_{j,t} + P_{j,t}^K K_{j,t}\right)\right) \tag{18}$$

Plugging into the corresponding Lagrangian the optimal factor conditional demands (17) and intermediate goods' demand (13), and solving the first order condition the following profit maximisation condition results:

$$P_{j,t}^A = \frac{\zeta}{\zeta-1} \cdot \frac{E_t \sum_{t=0}^{\infty} \theta^t \beta^t A_t (P_t)^{\zeta} Y_t}{E_t \sum_{t=0}^{\infty} \theta^t \beta^t (P_t)^{\zeta} Y_t} \tag{19}$$

where $A_{j,t}$ represents the marginal cost of intermediate goods firms.[2] Imperfect competition in the intermediate goods' market translates into a markup set by the corresponding firms above this cost, as seen on the left multiplier of the profit maximisation condition above.

Finally, having derived the price set by firms that are allowed to do so, one needs to combine it with the price of the remaining firms to obtain the aggregate price index, which is done using a Dixit-Stiglitz aggregator as follows:

$$P_t = \left[\theta \left(P_{t-1}\right)^{1-\zeta} + (1-\theta)\left(P_t^A\right)^{1-\zeta}\right]^{1/(1-\zeta)} \tag{20}$$

[2] $A_{j,t} = \frac{\partial TC_t}{\partial Y_{j,t}} = \frac{1}{Z_t}\left[W_t\left[(1-\alpha)+\alpha\left(\frac{\alpha W_t}{(1-\alpha)P_t^K}\right)^{\frac{\kappa}{(1-\kappa)}}\right]^{-\frac{1}{\kappa}} + P_t^K\left[\alpha+(1-\alpha)\left(\frac{(1-\alpha)P_t^K}{\alpha W_t}\right)^{\frac{\kappa}{(1-\kappa)}}\right]^{-\frac{1}{\kappa}}\right]$

3.4 Monetary and fiscal authorities

The monetary authority adjusts the interest rate (R_t) partially every period, according to a Taylor rule that takes into account both inflation (π_t) and external interest rate (R_t^*), with adjustment parameters $\phi_1 > 1$ and $0 < \phi_2 < 1$:

$$R_t = \phi_2 R_{t-1} + (1 - \phi_2)\left(R_t^* + \phi_1 \pi_t\right) \tag{21}$$

Inflation, on its turn, is defined as

$$\pi_t = \left[\left(\frac{P_t}{P_{t-1}}\right)^4 - 1\right] \tag{22}$$

As for the fiscal authority, its role in the model is simplified to a single constant exogenous variable for public spending, $G = \overline{G}$.

3.5 Closing the model

The remaining equation required to complete the model is the aggregate demand, or demand for final, or consumption, goods:

$$Y_t = C_t + G_t + X_t \tag{23}$$

The above expression, in addition to equations (1), (14), (16), (20) and (21), and Euler equations (7), (8), (9) and (10), form the full system that describes the model, to be solved for C_t, S_t, R_t, W_t, L_t, P_t^K, K_t, P_t, F_t and Y_t.

3.6 Numerical solution

The chosen solution method for the simulation of the model consists of the Projection method, as described by Judd [8] and placed into context earlier in this work. It involves:

- Choosing an approximation function specification for each of the functions to be approximated;
- Choosing a measure of the approximation error;
- Choosing an optimisation algorithm to iterate until the error is reduced to a previously defined tolerance.

Approximation function. In this case, the functions to be approximated include Euler difference equations (7), (10), and aggregate price index function (20), which describe the optimal path of C_t, S_t and P_t.
Both the numerator and denominator of the profit maximisation condition (19) can be rewritten as a recursive relationship by using two auxiliary variables, $A_t^{p1}$and A_t^{p2}, defined as

$$\begin{aligned} A_t^{p1} &= A_t(P_t)^{\zeta}Y_t + \theta\beta A_{t+1}^{p1} \\ A_t^{p2} &= (P_t)^{\zeta}Y_t + \theta\beta A_{t+1}^{p2} \end{aligned}$$

This, in turn, allows (19) to be rewritten as

$$P_t^A = \frac{\zeta}{1-\zeta} \cdot \frac{A_t^{p1}}{A_t^{p2}} \tag{24}$$

The approximation function specification consists of an artificial neural network, namely a multilayer perceptron. Each variable is approximated by a single neuron, denoted by n, with three input nodes z – one for each of the state variables Z_t, F_t and R_t, defined as

$$n_i = f\left(\sum_{k=1}^{j} w_k z_k\right)$$

where i is the index of approximated functions, k the index of the neuron's inputs, and with activation function $f(x) = \dfrac{1}{1+e^{-x}}$. Finally, the approximation is parameterised by weights w.

Approximation error. The error measures are based on the approximated Euler equations. Rearranging the referred equations, and letting $\hat{C}_t$, $\hat{S}_t$, $\hat{A}_t^{\ p1}$ and $\hat{A}_t^{\ p2}$ denote the approximated values of C_t, S_t, A_t^{p1} and A_t^{p2} respectively, the Euler errors are given by

$$\epsilon_t^c = \frac{\hat{C}_t^{-\eta}}{P_t}\left[\frac{1}{1+R_t}\right] - \beta\left[\frac{\hat{C}_{t+1}^{-\eta}}{P_{t+1}}\right] \tag{25}$$

$$\epsilon_t^S = (1+R_t)\,\hat{S}_t - (1+R_t^* + \Phi_t' F_t + \Phi_t)\,\hat{S}_{t+1} \tag{26}$$

$$\epsilon_A = \frac{\hat{A}_t^{p1}}{\hat{A}_t^{p2}} - \frac{A_t\,(P_t)^{\zeta}\,Y_t + \theta\beta\hat{A}_{t+1}^{p1}}{\hat{A}_t^{p2} - (P_t)^{\zeta}\,Y_t + \theta\beta\hat{A}_{t+1}^{p2}} \tag{27}$$

measuring the distance between the approximated and the true values – those that verify the optimal path as described by the Euler equations.

Approximation algorithm. The computational implementation of the model comprises three separate parts. The first one consists of an algorithm, encapsulated in a MATLAB function, described below, which simulates the approximated model and returns the simulation's Euler errors, for any given set of parameters and neural network weights. The second consists of a MATLAB script that iterates on the referred function, based on an initial guess of the approximation weights, until the sum of squared Euler errors are minimised. Finally, a third script simulates the model as-is, by calling the model's simulation function, and where impulse-response shocks, for example, can be defined.

Algorithm 1 Model simulation function

for all simulation periods **do**
  Draw values for random processes (Z_t);
  $\hat{C}_t, \hat{S}_t, \hat{A_t^{p1}}, \hat{A_t^{p2}} \leftarrow$ Output of neural network;
  $Y_t, L_t, K_t, P_t^K, P_t, W_t, R_t, F_t \leftarrow$ Solve rest of system numerically;
  Compute Euler errors ϵ_t^S, ϵ_t^C, ϵ_t^A;
end for

4 Simulations and results

Having described the model, the projection method and its application, we now aim to show it in practice, simulating an impulse-response shock to productivity. The results in the staggered prices model are compared to a version of the baseline model with flexible prices.

In the case of flexible prices, the intermediate firms profit maximisation optimisation problem, having into account intermediate goods' demand (13), becomes instead:

$$\max_{Y_{j,t}} \Pi_{j,t} = P_{j,t} \left(\frac{P_{j,t}}{P_t} \right)^{-\zeta} Y_t - \left(L_{j,t} W_{j,t} + P_t^K K_{j,t} \right) \tag{28}$$

Solving its first order condition results into the profit maximisation condition:

$$P_{j,t} = \frac{\zeta}{\zeta - 1} \cdot A_t \tag{29}$$

where once again final prices include a markup over marginal cost, sourced in the imperfect substitutability between intermediate goods. This equation replaces (19) for the flexible prices model.

Finally, for the model's parameters needed for the simulation, this work relies on the already calibrated values of Lim and McNelis [10], reproduced here in Table 1.

η	Relative risk aversion coefficient	1.5
ϖ	Elasticity of marginal disutility of labour	0.25
β	Discount factor	1/1.01
α	Production factors usage share	0.15
κ	Elasticity of substitution between production factors parameter	0.1
φ	Risk premium sensitivity	0.1
ρ	Productivity shock adjustment factor	0.9
ϕ_1	Taylor rule interest rate adjustment factor	0.9
ϕ_2	Taylor rule inflation sensitivity	1.5
ζ	Elasticity of substitution between differentiated consumption / final goods	6
θ	Staggered prices adjustment factor	0.85

Table 1: Model parameters

4.1 Productivity shock

The productivity shock, or impulse-response, is implemented firstly by starting the simulation with Z in its stationary value, while setting productivity's disturbance parameter, ϵ^Z, equal to zero for all periods except for the moment of the shock, where it is given a positive value. Finally, the model is simulated, with the results shown in Figures 1 and 2.

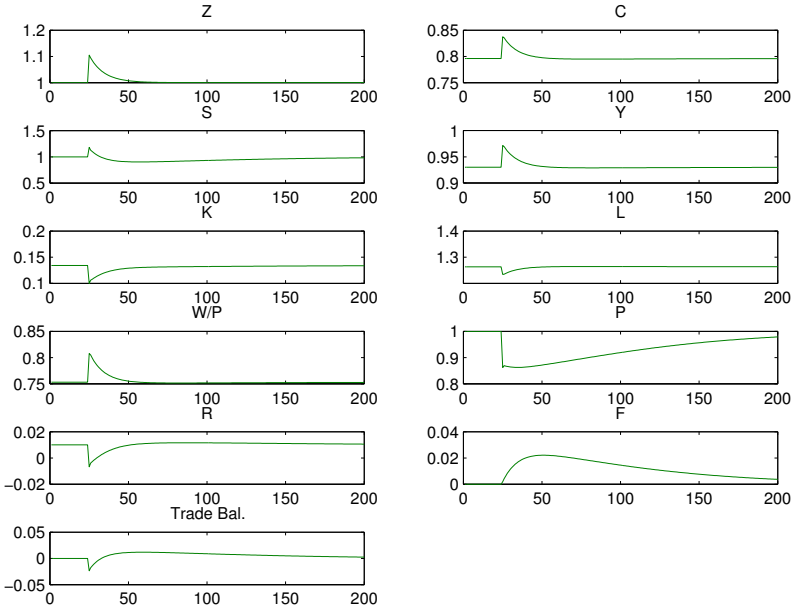


Figure 1: Productivity shock ($\epsilon^Z = 0.1$) with flexible prices

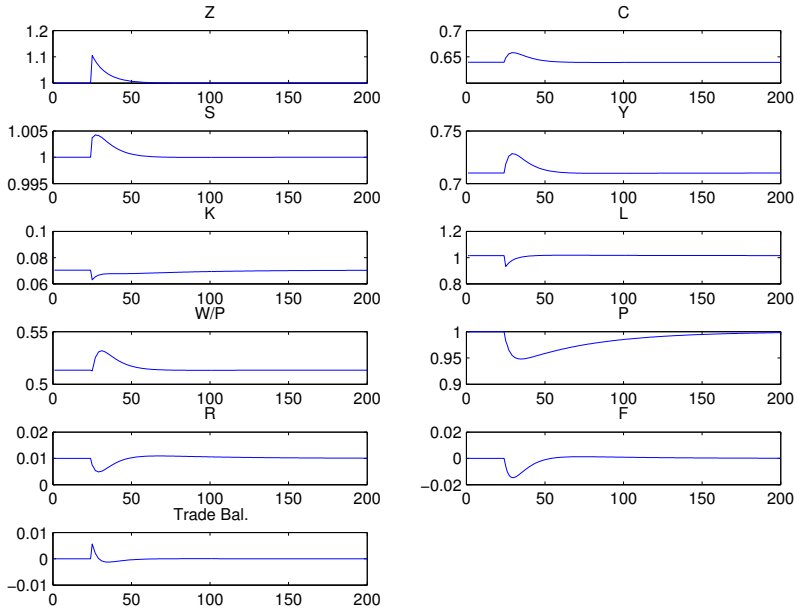


Figure 2: Productivity shock ($\epsilon^Z = 0.1$) with staggered prices

Although the magnitudes of the effects aren't directly comparable due to different steady-state effects of markups in both versions, one can note the direction of the effects being the same for both versions of the model, with the exception of the trade balance and external debt. Additionally, the adjustment process also tends to differ, being smoother in the staggered prices case.

The increase in productivity can be seen as having a corresponding increase in production, real wages and consumption, while both production factors' usage decrease and the price level falls. However, while with flexible prices the trade balance deteriorates, with the corresponding increase in external indebtedness, with staggered prices the opposite effect is found.

5 Conclusion

In the present work a DSGE model of the New Neoclassical Synthesis was derived, extending a baseline version with the addition of staggered prices. For its solution a projection method was employed, using an Artificial Neural Network as the approximation function, trained for minimising the sum of squared Euler errors, the measure of the approximation fitness. Finally, a set of impulse-response shocks were simulated on both the baseline and extended versions of the model, after approximation.

The correct solution of this model with the addition of imperfect competition and staggered prices is an important intermediate step for solving larger DSGE models, and for

answering economic questions, given the prevalence of the referred extensions in most of the NNS literature.

In this respect the projection method, although not widely used – mainly in favour of perturbation methods of different orders –, is shown to provide a good approximate solution, both as measured by the Euler errors and impulse-response results. Projection methods have the advantage of providing a global solution, in addition to fully retaining the nonlinearities of the approximated problem solution (depending on the applied approximation function).

As for future work, in the Economics front, a large literature on additional frictions and shocks exists, all trying to improve the fit to real data or the analysis of specific problems, such as the inclusion of money, consumption habits persistence, financial frictions, heterogeneous agents, and improved fiscal sections, for example, which could be explored with this method.

Concerning the application of the method itself, several speed improvements could be introduced, for example, by removing the nested loops through vectorization of the MATLAB code. Finally, although operations within each model simulation are interrelated and not easily parallelised, iterations of the Monte Carlo method for drawing wealth distributions and variable correlations, a common practice in the literature, are fully separate from each other and can be done concurrently, exploring parallel processing. A GPU implementation shows potential and seems an attractive scientific computing problem to tackle.

REFERENCES

[1] Guillermo A. Calvo. Staggered prices in a utility-maximizing framework. *Journal of Monetary Economics*, 12(3):383–398, 1983.

[2] Fabio Canova. *Methods for Applied Macroeconomic Research*. Princeton University Press, 2007. ISBN 9781400841028.

[3] Gregory C. Chow. *Dynamic economics: optimization by the Lagrange method*. Oxford University Press, New York, 1997.

[4] Richard Clarida, Jordi Gali, and Mark Gertler. The Science of Monetary Policy: A New Keynesian Perspective. *Journal of Economic Literature, American Economic Association*, 37(4):1661–1707, 1999.

[5] A. K. Dixit and Joseph E. Stiglitz. Monopolistic competition and optimum product diversity. *The American Economic Review*, 67(3):297–308, June 1977.

[6] Marvin Goodfriend and Robert G. King. The New Neoclassical Synthesis and the Role of Monetary Policy. In *NBER Chapters*, pages 231–296. National Bureau of Economic Research, Inc, 1997.

[7] Burkhard Heer and Alfred Mauß ner. *Dynamic General Equilibrium Modeling*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-03148-9. doi: 10.1007/978-3-540-85685-6.

[8] Kenneth L. Judd. *Numerical Methods in Economics*. The MIT Press, 1998. ISBN 0262100711.

[9] Jinill Kim and Sunghyun Henry Kim. Spurious welfare reversals in international business cycle models. *Journal of International Economics*, 60(2):471–500, August 2003. ISSN 00221996. doi: 10.1016/S0022-1996(02)00047-8.

[10] G. C. Lim and Paul D. McNelis. *Computational Macroeconomics for the Open Economy*. The MIT Press, 2008. ISBN 0262123061.

[11] Robert E. Lucas. Econometric policy evaluation: A critique. *Carnegie-Rochester Conference Series on Public Policy*, 1:19–46, 1976. ISSN 0167-2231. doi: http://dx.doi.org/10.1016/S0167-2231(76)80003-6.

[12] Stephanie Schmitt-Grohé and Martín Uribe. Solving dynamic general equilibrium models using a second-order approximation to the policy function. *Journal of Economic Dynamics and Control*, 28(4):755–775, January 2004. ISSN 01651889. doi: 10.1016/S0165-1889(03)00043-5.

[13] Nancy L. Stokey and Edward C. Prescott. *Recursive Methods in Economic Dynamics*. Harvard University Press, 1989. ISBN 0674750969.

[14] Eric Swanson, Gary Anderson, and Andrew Levin. Higher-Order Perturbation Solutions to Dynamic, Discrete-Time Rational Expectations Models: Methods and an Application to Optimal Monetary Policy. Computing in Economics and Finance 2005 146, Society for Computational Economics, 2005.

[15] Harald F. Uhlig. *A toolkit for analyzing nonlinear dynamic stochastic models easily*. Institute for Empirical Macroeconomics, Federal Reserve Bank of Minneapolis, 1995.

[16] Michael Woodford. *Interest and prices: Foundations of a theory of monetary policy*. Princeton University Press, 2003. ISBN 0691010498; 9780691010496.

SYMCOMP 2015
Faro, March 26-27, 2015

OPTIMIZATION OF DESALINATION HEAT EXCHANGER'S GEOMETRIES USING BAT-INSPIRED TECHNIQUES

Costa, D.M.S.[1]*, Trindade, J.M.F.[1,2,3] and Loja, M.A.R.[1,3]

1: GI-MOSM, Grupo de Investigação em Modelação e Optimização de Sistemas Multifuncionais
ISEL, IPL - Instituto Superior de Engenharia de Lisboa
Av. Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal

2: ENIDH, Escola Superior Náutica Infante D. Henrique
Av. Eng.º Bonneville Franco, Paço de Arcos, 2780-122 Oeiras, Portugal

3: LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal

e-mails: {a38293@alunos.isel.pt, jorgetrindade@enautica.pt, amelialoja@dem.isel.ipl.pt}

Keywords: Desalination, Heat transfer coefficient, Falling film evaporators, Bat-inspired optimization, Symbolic computation

Abstract

In many regions of the world there are persisting situations of lack of potable water with the well-known consequences in population's daily life, where a poverty logic loop seems to be definitely installed. In some of those regions, where seawater and solar energy resources are abundant, the use of this kind of energy may be an effective alternative to consider for seawater desalination.

Falling film evaporators allow high heat transfer rates and are an important key to achieve greater efficiencies throughout several industries. Desalination systems, especially when driven by renewable energies, tend to be highly dependent of appropriate efficiencies. In order to study this matter, a meta-heuristic optimization of a desalination model of this kind is done, focusing on some design variables of the evaporator's geometry.

Because of the context of the present work, the achievement of an optimal solution, from the global heat transfer perspective is a particularly important objective. To enable this optimization study, different implementations of bat-inspired optimization technique will be considered, aiming at the minimization and maximization of the falling film heat transfer coefficient, within a desalination model, based on a horizontal falling film evaporator.

A parametric study concerning the heat exchangers' selected design variables will be carried out and discussed, along with an influence analysis of other optimization parameters.

1. INTRODUCTION

Millions of people have no access to secure fresh water. Seawater desalination is a reasonable alternative for coastal arid regions, thus desalination systems have been used to supply potable water for coastal communities and urban centres. Drinking water can be produced from seawater using membrane or thermal processes. A desalination plant performs a separation process in which the incoming saline water is separated into two outgoing streams of brine and product water. Desalination uses a large amount of energy to remove a portion of pure water from a salt water source. The energy optimization in the desalination process is therefore crucial, in order to achieve good efficiencies.

Many studies of water desalination costs appear regularly in water desalination and renewable energy related publications. The choice of desalination method affects significantly the water desalination cost. Thermal methods are used mainly in medium and large size systems, while membrane methods, mainly reverse osmosis, are used by medium and low capacity systems [1]. Renewable energy options for an environmental friendly option, when decreasing global reserves of fossil fuels threatens the long-term sustainability of global economy, are discussed in [2]. The use of renewable energies for desalination is nowadays a reasonable option for the emerging energy and water problems. In some regions throughout the world, characterized by very high temperatures during summer, water scarcity can be quite evident, as noted by [3]. In their work they study the possibility of supplying saturated steam to a thermal desalination system. This is done by using the absorption cycle system of an air conditioning application, where its driving energy is collected from solar radiation. Since air conditioning is extensively used to cool down buildings during summer, this can be an efficient way of desalination, reducing the electricity consumption. Falling film evaporators are present in many industry processes as a key component for performance and efficiency. For desalination applications, falling film evaporators allow high heat transfer rates with reduced maintenance costs, mitigating corrosion and scaling [4]. The liquid is evaporated at the outside of the tubes. It flows from one tube to the other in the form of droplets, jets or as a continuous sheet. Due to the impinging effect when water flows from one tube to another, the heat transfer is higher when compared to vertical falling film evaporators.

It is thus, crucial to make an effective research investment to find not only innovative and sustainable ways of desalting water, by means of renewable energies, but also to optimize these processes. Optimizing fresh water production is therefore a key issue, being possible to understand this objective as a dual goal problem, as it is possible to think it in the perspective of the production maximization and/or in terms of the production cost minimization. To this purpose it was considered a Bat-inspired optimization technique, recently developed and proposed by [5], which is based on the echolocation behaviour of bats in searching within a design space. Another algorithm inspired on the bat behaviour was proposed by [6] being designated as Doppler Effect Bat Algorithm as it combines bat-algorithm (BA) with the Doppler effect. In their work the authors proposed a new frequency equation, as well as the velocities and locations updating where inertia weights are considered. The performance of

this approach is evaluated with benchmark functions.
A brief survey on meta-heuristics algorithms based on grouping of animals by its social behaviour was carried out by [7], for the travelling salesman problem. Their authors propose additionally a classification for meta-heuristics algorithms, not according to the swarm intelligence but subdivided into swarm, schools, flocks and herds algorithms. Bat optimization is considered within this last group, according to their authors. More recently, another study on the performance of BA optimization technique was carried out by [8], where standard benchmark functions were also used to achieve that comparative evaluation. These authors also hybridized BA with differential evolution to improve the local search step, and concluded on a significant improvement of the original bat technique.

The present work focus on a falling film evaporator model, submitted to a set of experiments, which results were subsequently analysed [4]. In a first stage of the present paper, a validation example is carried out, after which a set of case studies is considered using bat-inspired optimization techniques. In this last stage it is intended to understand and characterize the influence of each heat transfer variable on the heat transfer coefficient correlation used, as well as the influence of the optimization parameters.

2. FALLING FILM EVAPORATION

Due to the advantages of using falling film evaporators on desalination systems, it is important to understand the influence of the evaporator's design variables on the heat transfer efficiency. In order to find an interesting desalination model of this kind to support this work, several references were analysed. The work carried out in [4] has experimental data related to a falling film desalination model and an empirical correlation based on that, which is dependent of enough design variables. As a consequence of that, it is possible to compute an objective function (OF) with adequate complexity, to be optimized by a BA, enabling a validation of the correlation and the optimization algorithm.

Basically, falling film evaporation consists on spraying fine seawater droplets along a tube bundle, which is located inside an evaporator chamber – in this specific case, the tube bundle is horizontal. Outside those tubes, a liquid film is formed and the liquid falls continuously from tube to tube due to the gravitational acceleration. On the other hand, chilled water flows inside the tube bundle and its temperature is maintained above the seawater's saturation temperature, to ensure constant evaporation – as a result of this temperature gradient, the heat flux from the chilled water to the external liquid saline film causes its evaporation [4].

In [9], Han & Fletcher had a significant contribute to the study of falling film evaporation behaviour on horizontal tubes, proposing the following empirical correlation, to describe the falling film heat transfer coefficient (FFHTC):

$$h_{Han\ \&\ Fletcher} = 0.0028 \cdot \left[\frac{{\mu_{s,l}}^2}{g \cdot {\rho_{s,l}}^2 \cdot {k_{s,l}}^3}\right]^{-0.333} \cdot (Re_{\Gamma})^{0.5} \cdot (Pr_{s,l})^{0.85} \quad (1)$$

Conducting their experiments with pure water only at saturation temperatures of 322 K and above, the authors concluded that with the increase in saturation temperature the average

convective heat transfer coefficient increases too.

However, with the work carried out in [4], Shahzad et al. figured out that the Han & Fletcher correlation (1) is unable to predict the additional heat transfer enhancement mechanism that may occur when the vapour specific volume increases rapidly at low saturation temperatures, i.e., below 300-298 K. As a result of that fast expansion, micro-bubbles within the thin film layer emerge, causing a significant flow and thermal gradient agitation – such phenomena enhances the FFHTC [4]. By means of a falling film evaporator and experimental measurements, the authors were able to propose a new correlation, incorporating the additional effect from the micro-bubble agitation, including parameters such as salinity, saturation temperature and vapour specific volume:

$$h_{Shahzad\ et\ al.} = \left[0.277 \cdot \left[\frac{{\mu_{s,l}}^2}{g \cdot {\rho_{s,l}}^2 \cdot {k_{s,l}}^3}\right]^{-0.333} \cdot (Re_\Gamma)^{-2.11} \cdot (Pr_{s,l})^{4.55} \cdot \left[2 \cdot \exp\left(\frac{sal_{s,l}}{sal_{ref}}\right) - 1\right]^{-0.41} \cdot \left(\frac{T_{s,sat}}{T_{sat,ref}}\right)^{14.70}\right] + \left[0.885 \cdot \frac{q}{deltaT_{ch,l-s,l}} \cdot \left(\frac{v_{s,g}}{v_{ref}}\right)^{-0.34}\right] \quad (2)$$

Since Shahzad et al. correlation predicts bubble agitation, which occurs at low saturation temperatures, it is valid for the following evaporator absolute pressure range [4]:

$$P \in [0.93\ ;\ 3.60]\ kPa\ abs \quad (3)$$

Which approximately corresponds to the following evaporator saturation temperature range, assuming pure water [4]:

$$T_{s,sat} \in [279\ ;\ 300]\ K \quad (4)$$

3. BAT-INSPIRED OPTIMIZATION

Swarm intelligence theory considers the modelling of populations of interacting agents, where the local interactions between all the members may give an overall intelligent global behaviour. Examples of swarm intelligence based algorithms are the ant colony and the particle swarm optimization, the artificial bee colony, the cuckoo search and the firefly algorithm, the human seeker optimization and the BA.

BA mimetic the behaviour of microbats which can find their prey and discriminate different types of insects, in complete darkness. This is done through the use of a type of sonar, called, echolocation, which enable them to detect prey and to avoid eventual obstacles. The algorithm is then supported upon the following main three principles: the echolocation capacity, used to sense distances and to differentiate between prey and barriers; the random flying capacity, characterized by velocity, frequency and loudness and the ability to vary frequency and loudness, depending on the proximity of the object.

To this purpose, in BA each bat in a population of *N* individuals is initialized with a starting position, rate of pulse, loudness and a frequency, which evolutes during the number of iterations/generations necessary to meet the stopping criteria.

Each new generation of bats is updated through the movement of the existing ones, according to the relations:

$$Q_i^k = Q_{\min} + \left(Q_{\max} - Q_{\min}\right)\beta \tag{5}$$

$$v_i^{k+1} = v_i^k + \left(x_i^k - Best\right)Q_i^k \tag{6}$$

$$x_i^{k+1} = x_i^k + v_i^k \tag{7}$$

where $\beta \in [0,1]$, Q_i^k is the pulse frequency of the i-th bat at an instant k. The lateral bounds of the pulse frequency are $Q_{\min}$ and $Q_{\max}$. The position and velocity of the i-th bat at the instant k are denoted by x_i^k and v_i^k respectively. *Best* stands for the best solution until the current instant.

The local search that starts depending on the proximity of the pulse rate of each bat r_i^k, to a randomly generated rate, will modify the current global best solution as follows:

$$x^k = Best + \varepsilon A_i^k \left(2\beta - 1\right) \tag{8}$$

which differ from the original bat algorithm, as in this later the loudness was an average quantity, considering the contribution of all the bats in the population. In the present case an improved local search proposed in [8] was adopted. The parameters involved on the updating of the k-th bat position is the position of the global best solution, a scaling factor ε and the loudness of each bat A_i^k at that generic instant *k*. Again β is a random value in $[0,1]$.

Pulse rate and loudness parameters closely mimetic the behaviour of bats, as the first increases when the bat finds a prey and at the same the loudness decreases. These relations are established according to:

$$A_i^{k+1} = \alpha A_i^k \tag{9}$$

$$r_i^k = r_i^0 \left(1 - e^{(-\gamma\varepsilon)}\right) \tag{10}$$

with α and γ being constants that enable an progressive variation of loudness and pulse rate. These constants have a similar effect to the cooling schedule parameter present in simulated

annealing method [10]. As in other optimization techniques, the current iteration best is registered along with the corresponding design variables, maintaining a log that allows tracing the evolution of the whole process.

In the present work, a hybridized version of BA is also implemented where the local search is performed according to differential evolution (DE) technique [8],[11]. At each generation, new bat positions are achieved by differential mutation wherein it is possible to adopt either one or two scaled differences of trial solutions. In both cases the mutation is based on the global best solution so far, so DE schemes used can be in short designated by DE/Best/1 and DE/Best/2. Considering the recombination/crossover operator, an arithmetic recombination is used, wherein an adaptive or not, perturbation factor can be selected. This is also used in other evolutionary approaches for other particular parameters, such as in the case of the particle swarm optimization [12], [13].

4. DEFINITION AND OPTIMIZATION OF THE DESALINATION MODEL

4.1. FIRST VALIDATION STUDY

In order to correctly compute the OF, which is the FFHTC predicted by Shahzad et al. correlation (2), a validation is needed, based on the same experimental conditions described in [4]. As a consequence of the fluids' properties dependency on certain variables, such as temperature, interpolation procedures must be programmed. Such conditions are summarized in Table 1, as well as the evaluative input variables for the interpolation procedures.

Vapour phase related properties are evaluated at mean vapour film temperature, being calculated as:

$$T_{\Gamma} = \frac{\left[\frac{Ti_{ch,l} + To_{ch,l}}{2} + T_{s,sat}\right]}{2} \tag{11}$$

Concerning to the film Reynolds number and to the input heat flux, they are given according to equations (12) and (13), respectively, where in this last case it is used the concept of log mean temperature difference.

$$Re_{\Gamma} = \frac{4 \cdot \Gamma}{\mu_{s,l}} \tag{12}$$

$$q = \frac{(\dot{m}_{ch,l} \cdot Cp_{ch,l} \cdot {deltaT_{ch,l}}^2) \cdot \ln\left(\frac{To_{ch,l} - T_{s,sat}}{Ti_{ch,l} - T_{s,sat}}\right)}{\left(To_{ch,l} - Ti_{ch,l}\right) \cdot Ai} \tag{13}$$

All fluid properties evaluated as pure fresh water, instead of seawater.

Table 1 – Variables' conditions for the OF validation, based on [4].

Parameter	Lower Bound	Upper Bound	Evaluated at
g [m/s^2]	9.81		–
v_{ref} [m^3/kg]	52.65		–
sal_{ref} [ppm]	30,000		–
T_{ref} [K]	322		–
P [kPa] (3)	0.93	3.60	–
$T_{s,sat}$ [K] (4)	279	300	P [kPa] (3)
$Ti_{ch,l}$ [K]	324	345	–
$To_{ch,l}$ [K]	289	310	–
$\mu_{s,l}$ [kg/(m.s)]	0.00148	0.00086	$T_{s,sat}$ [K] (4)
$\rho_{s,l}$ [kg/m^3]	999.87	996.63	$T_{s,sat}$ [K] (4)
$\rho_{ch,l}$ [kg/m^3]	994.66	985.58	$deltaT_{ch,l-s,l}$ [K] + $T_{s,sat}$ [K] (4)
$k_{s,l}$ [W/(m.K)]	0.573	0.610	$T_{s,sat}$ [K] (4)
$Pr_{s,l}$	10.90	5.87	$T_{s,sat}$ [K] (4)
$v_{s,g}$ [m^3/kg]	143.91	40.41	T_Γ [K] (11) and P [kPa] (3)
Re_Γ (12)	49.44	85.32	–
Γ [kg/(m.s)]	0.01833	0.01827	–
T_Γ [K] (11)	292.75	313.75	–
$deltaT_{ch,l-s,l}$ [K]	27.5	27.5	–
$deltaT_{ch,l}$ [K]	35	35	–
Ai [m^2]	0.08715		–
At [m^2]	0.00017		–
D [m]	0.0146		–
L [m]	1.9		–
$sal_{s,l}$ [ppm]	15,000	90,000	–
$Cp_{ch,l}$ [J/(kg.K)]	4178.00	4182.74	$deltaT_{ch,l-s,l}$ [K] + $T_{s,sat}$ [K] (4)
$u_{ch,l}$ [m/s]	0.09955		–
$\dot{m}_{ch,l}$ [kg/s]	0.01658	0.01643	–
q [W/m^2] (13)	41,837	41,502	–

Figure 1 depicts the results of the present OF as well as the experimental results obtained in [4], for a 15,000 ppm salinity, as well as the Han & Fletcher correlation (1). Comparing the results, it is possible to conclude that the OF is acceptable. In addition, it is verified that all different curves converge to the Han & Fletcher empirical correlation, for saturation temperatures above 300 K, as it should be.

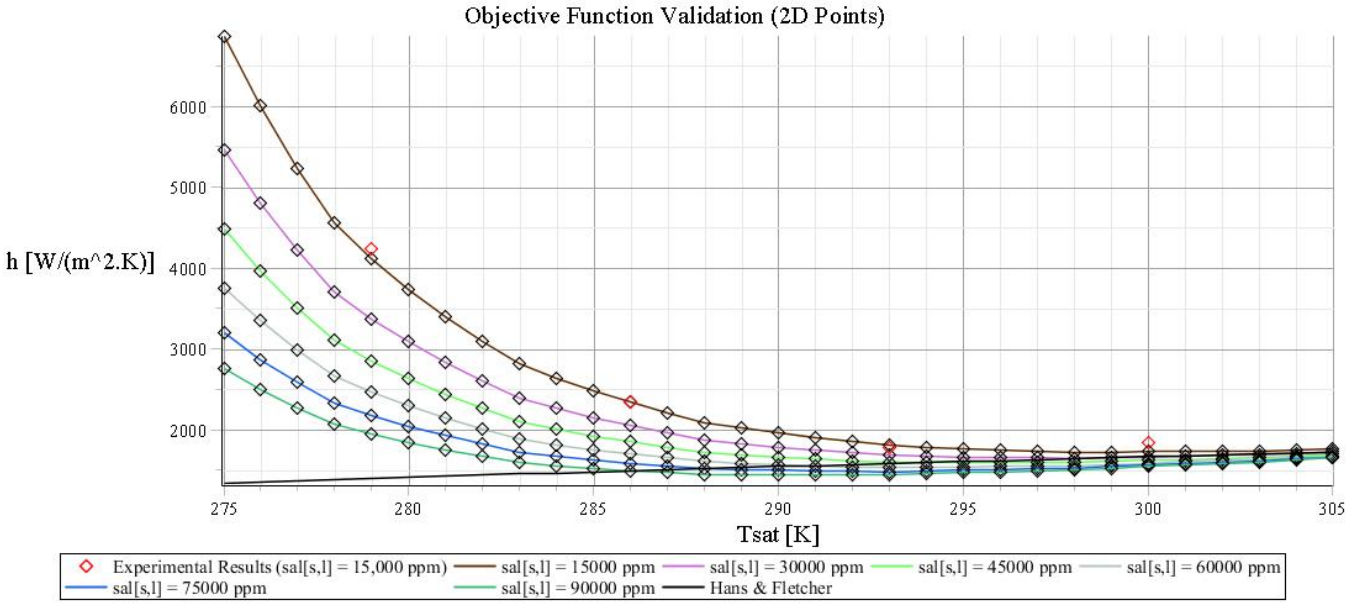


Figure 1 – FFHTC results, using the computed OF (2D Points).

Based on the results presented in Figure 1 a 3D surface was modelled, as one can observe in Figure 2.

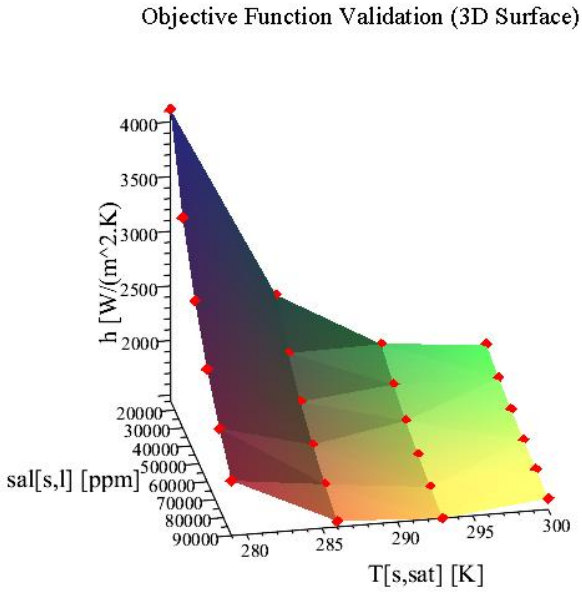


Figure 2 – FFHTC results, using the computed OF (3D Surface).

It can be concluded that when both saturation temperature $T_{s,sat}$ and salinity $sal_{s,l}$ decrease, the FFHTC increases (see Figure 1 and Figure 2).

In Figure 3, it is possible to analyse the tube diameter influence on the FFHTC, assuming a constant velocity of 0.09955 m/s of the chilled water inside the tube bundle and a feed water salinity of 15,000 ppm. In addition, the inlet and outlet temperatures were assumed constant, only for validation purposes. As a result of the direct proportionality between the input heat flux (from chilled water to the external saline film solution) and the tube diameter, it is verified that when tube diameter increases, the FFHTC increases too (see equation (14)) – this happens due to the direct proportionality between the chilled water mass flow rate and tube diameter, when the flow velocity is constant.

$$q = \frac{(\rho_{ch,l} \cdot u_{ch,l} \cdot D \cdot Cp_{ch,l} \cdot deltaT_{ch,l}{}^2) \cdot \ln\left(\frac{To_{ch,l} - T_{s,sat}}{Ti_{ch,l} - T_{s,sat}}\right)}{4 \cdot L \cdot \left(To_{ch,l} - Ti_{ch,l}\right)} \tag{14}$$

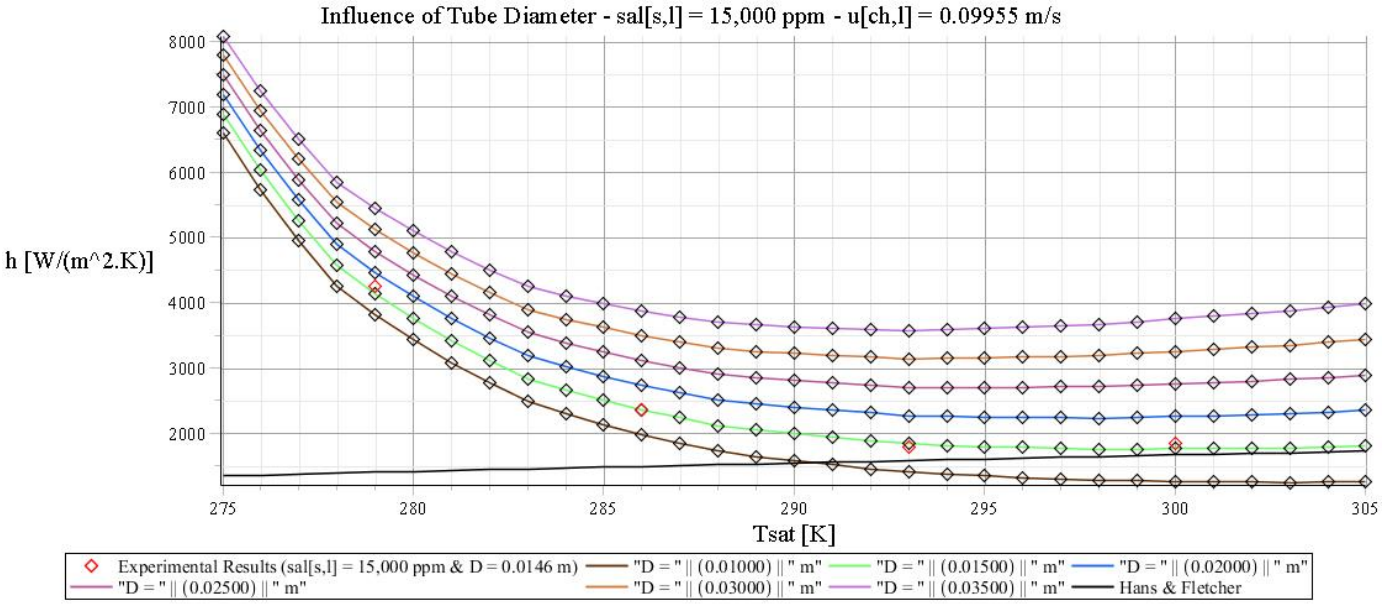


Figure 3 – Influence of tube diameter on FFHTC.

In Figure 4, it is possible to analyse the tube length influence on the FFHTC, assuming a feed water salinity of 15,000 ppm. Differently from the influence of tube diameter on FFHTC, it is verified from equation (14) that the input heat flux is inversely proportional to the tube length. As it can be observed through the curves' trend presented in Figure 4, when the tube length decreases, the FFHTC increases.

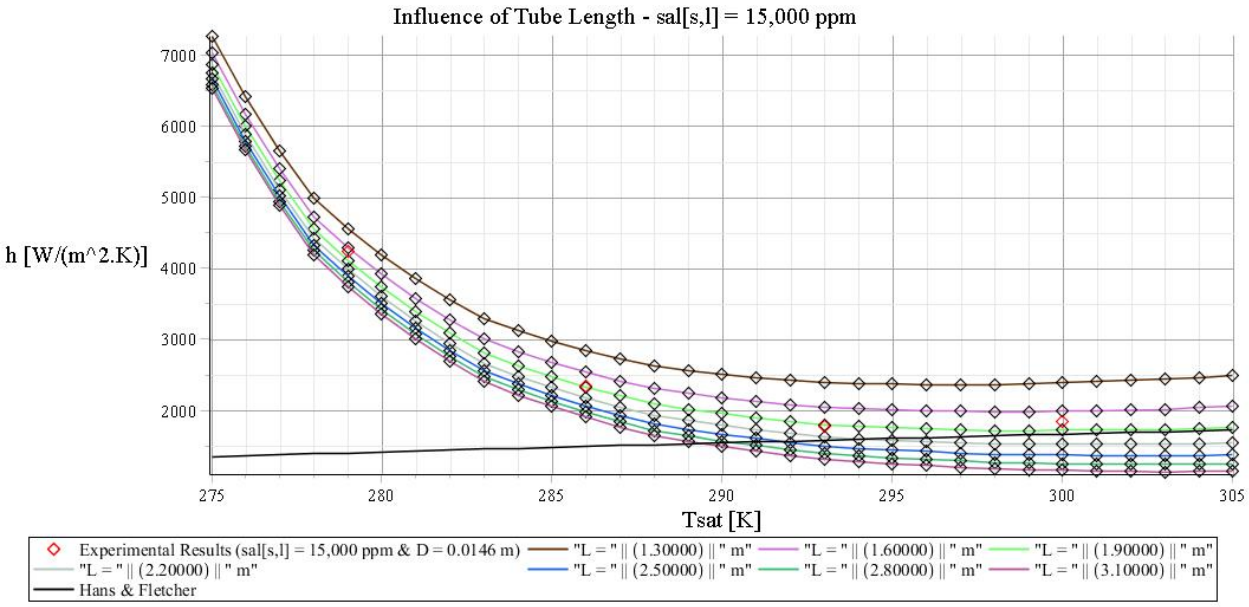


Figure 4 – Influence of tube length on FFHTC.

4.2. SECOND VALIDATION STUDY

After the OF validation, it is convenient to run the optimization algorithm and compare the results, in order to understand its validity.

For validation and comparison purposes, only two design variables are defined – the evaporator saturation temperature ($T_{s,sat}$ [K]) and the feed water salinity ($sal_{s,l}$ [ppm]). In Table 2 their bounds are presented. The remaining conditions are exactly the same, comparing with Table 1.

Table 2 – Design variables' bounds for the BA validation.

	Design Variable	Lower Bound	Upper Bound
1	$T_{s,sat}$ [K]	279	300
2	$sal_{s,l}$ [ppm]	15,000	90,000

As a result of the BA characteristics, several input variables must be predefined – Table 3 summarizes those parameters, for this particular validation example.

Table 3 – BA input parameters for optimization algorithm validation purpose.

Parameter	Value
Optimization Objective	MAXIMIZATION
Hybrid	TRUE
Adaptive	TRUE
$Fper_i$	3
$Fper_f$	0.5
n	10
Ngen	10
A0	1
r0	0.5
α	0.95
γ	0.98
Q_{min}	0
Q_{max}	1
ε	0.001
d	2
Number of Runs	1

The worst and optimum values were found, respectively, in the fifth and seventh generation, corresponding to the variables summarized in Table 4.

Table 4 – Parametric results for the BA validation.

Gen	$T_{s,sat}$ [K]	$sal_{s,l}$ [ppm]	Best [W/(m^2.K)]	Worst [W/(m^2.K)]
5	300	90,000	-	1545
7	279	15,000	4116	-

Comparing the worst and best values from Table 4 with the curves presented in Figure 1, it is concluded that the BA easily converges for the optimum conditions, in order to maximize the FFHTC – which is achieved when $T_{s,sat} = 279\ K$ and $sal_{s,l} = 15{,}000\ ppm$ ($T_{s,sat}$ and $sal_{s,l}$ lower bounds), corresponding to $h \cong 4116\ W/(m^2.K)$. As we expected, for higher saturation temperature and salinity values, the OF decreases – in Table 4, $Gen = 5$ is an example of that, where $h \cong 1545\ W/(m^2.K)$ is the worst result.

In order to understand the mean value and standard deviation behaviour along generations, Figure 5 presents both variables.

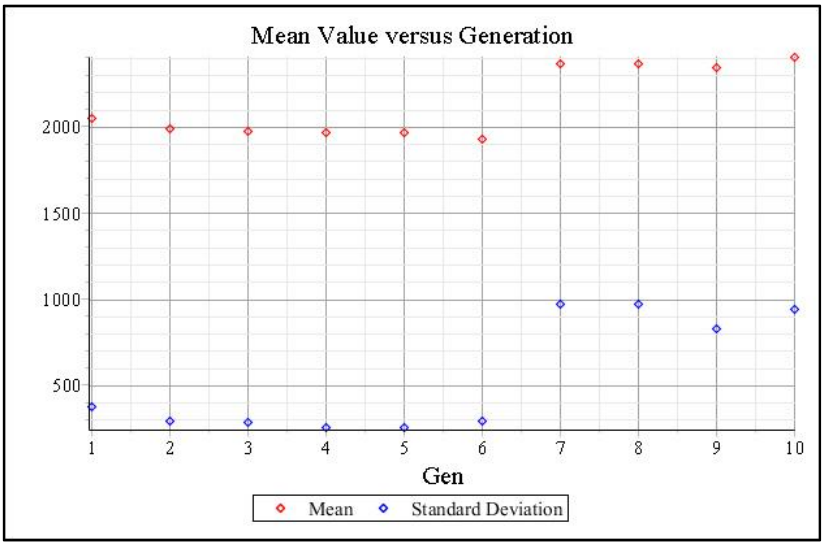


Figure 5 – Mean value and standard deviation versus generation, for the BA validation.

For a qualitative illustration of the obtained data for this particular validation, Figure 6 displays the maximum evolution along generations.

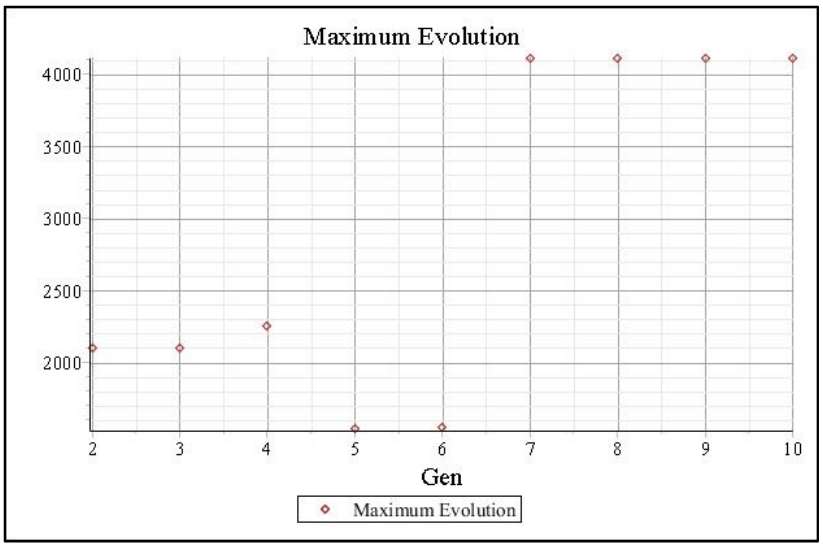


Figure 6 – Maximum evolution versus generation, for the BA validation.

4.3. OPTIMIZATION STUDIES

The objective of the present subsection is to study the influence of several design variables on the OF maximum and minimum. As a consequence of that, it is possible to analyse the heat transfer behaviour of a falling film evaporator at low saturation temperatures, with a design and optimization perspective. On the other hand, it is important to understand the effects of BA parameters on the optimization results and its convergence to optimum conditions, because the algorithm's efficiency depends on them. From the OF validation it is possible to understand that in the maximization case, for the specified conditions, the optimum is always verified when both design variables assume their bounds' ends – for the specific situation submitted to validation, the optimum appears when both saturation temperature and salinity assume their lowest bound. Simultaneously, when salinity starts to increase, the curves' behaviour shows that in the minimization case, minimum only appears for saturation temperatures in between its lower and upper bound. This type of analysis is crucial for this kind of problem, especially if the input heat flux is driven by solar energy. The outlet temperature of chilled water ($To_{ch,l}$) is highly dependent on the evaporation rate along the tube. For the optimization purpose, it is convenient to predict that temperature, for different types of situations, because its dependence on the design variables is evident. In order to do that, two empirical correlations have to be used, depending on flow conditions along the tube.

Firstly, the chilled water Reynolds number is calculated by the equation (15), in order to understand the flow regime. Since the outlet temperature is not known in the first place, an iterative process is needed to approximate its "true" value progressively. In the first iteration all the fluid properties are evaluated at $Ti_{ch,l}$.

$$Re_{ch,l} = \frac{\rho_{ch,l} \cdot u_{ch,l} \cdot D}{\mu_{ch,l}} \tag{15}$$

Secondly, an empirical correlation has to be chosen, depending on the flow regime and fluid properties. In order to simplify the present analysis, only two correlations are used – one for laminar and other for turbulent flow. Equation (16) is the traditional expression for heat transfer prediction in fully developed turbulent flows, inside tubes (forced convection), by Dittus and Boelter [14]:

$$Nu_{ch,l\ Dittus\ \&\ Boelter} = 0.023 \cdot {Re_{ch,l}}^{0.8} \cdot {Pr_{ch,l}}^{0.3} \tag{16}$$

Correlation (16) can be applied if the following restrictions are verified:

1. Smooth tubes;
2. Fully developed turbulent flow;
3. Approximately $0.6 < Pr_{ch,l} < 100$;
4. Approximately $2500 < Re_{ch,l} < 1.25 \cdot 10^5$.

On the other hand, for fully developed laminar flows, Hausen [14] recommends the following correlation (17):

$$Nu_{ch,l\ Hausen} = 3.66 + \frac{0.0668 \cdot (D/L) \cdot Re_{ch,l} \cdot Pr_{ch,l}}{1 + 0.04 \cdot \left[(D/L) \cdot Re_{ch,l} \cdot Pr_{ch,l}\right]^{2/3}} \quad (17)$$

Correlation (17) can be applied if the following restrictions are verified:

1. Smooth tubes;
2. Fully developed laminar flow;
3. Approximately $Re_{ch,l} < 2500$.

Thirdly, the convective heat transfer coefficient inside the tube can be calculated by:

$$h_{ch,l} = \frac{Nu_{ch,l} \cdot k_{ch,l}}{D}\ [W/(m^2.K)] \quad (18)$$

Fourthly, the energy balance is expressed by:

$$q_{ch,l} = h_{ch,l} \cdot \pi \cdot D \cdot L \cdot \left[T_{s,sat} - \left(\frac{Ti_{ch,l} + To_{ch,l}}{2}\right)\right] = \dot{m}_{ch,l} \cdot Cp_{ch,l} \cdot \left(To_{ch,l} - Ti_{ch,l}\right)\ [W] \quad (19)$$

Solving equation (19) for $To_{ch,l}$, the outlet temperature of chilled water is determined. After that, the iterative process starts, re-evaluating all the fluid properties at the mean temperature described by equation (20), until the value of $To_{ch,l}$ converges.

$$T_{mean} = \frac{To_{ch,l} + Ti_{ch,l}}{2} \quad (20)$$

In order to start the optimization runs, several design variables are predefined, as long as their lower and upper bounds (see Table 5).

Table 5 – Design variables' bounds for the desalination model optimization.

	Design Variable	**Lower Bound**	**Upper Bound**
1	$T_{s,sat}$ [K]	279	300
2	$sal_{s,l}$ [ppm]	15,000	90,000
3	$Ti_{ch,l}$ [K]	350	380
4	D [m]	0.020	0.030
5	L [m]	1.5	2

For this optimization section, the chilled water velocity ($u_{ch,l}$) was set to 0.5 m/s.

4.3.1. FIRST OPTIMIZATION STUDY (1S) – MAXIMIZATION

Table 6 summarizes the BA input parameters used in the first optimization study (maximization).

Table 6 – BA input parameters for the desalination model optimization – maximization (1S).

Parameter	Value
Optimization Objective	MAXIMIZATION
Hybrid	FALSE
Adaptive	FALSE
$Fper_i$	10
$Fper_f$	5
n	10
Ngen	10
A0	0.01
r0	0.01
α	0.95
γ	0.98
Q_{min}	0
Q_{max}	1
ε	0.001
d	5
Number of Runs	1

The optimum and worst values were found, respectively, in the second and sixth generation, corresponding to the variables summarized in Table 7.

Table 7 – Parametric results for the desalination model optimization – maximization (1S).

Gen	$T_{s,sat}$ [K]	$sal_{s,l}$ [ppm]	$Ti_{ch,l}$ [K]	D [m]	L [m]	Best [W/(m^2.K)]	Worst [W/(m^2.K)]
2	279.00	15,000	350.00	0.0200	1.5000	4040	-
6	300.00	90,000	380.00	0.0300	2.0000	-	1588

However, the best value presented in Table 7 does not correspond to a maximum optimum (for those conditions), which means that the input parameters in Table 6 are not the best combination in this case. In order to understand the mean value and standard deviation behaviour along generations, Figure 7 presents both variables. For a qualitative illustration of the obtained data for this maximization case study, Figure 8 displays the maximum evolution along generations.

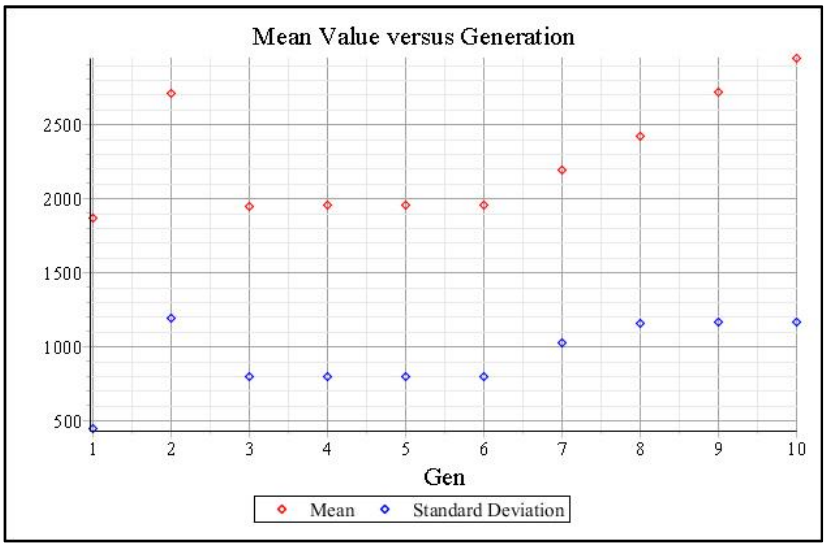


Figure 7 – Mean value and standard deviation versus generation, for the desalination model maximization (1S).

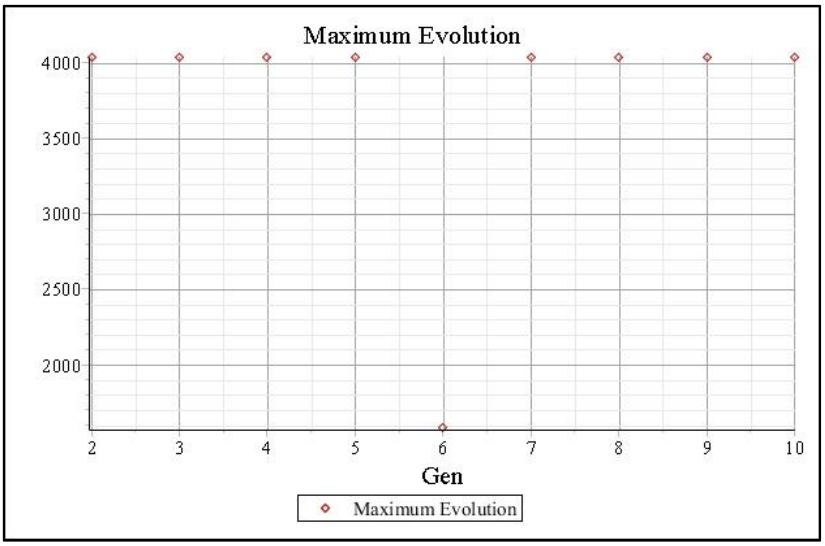


Figure 8 – Maximum evolution versus generation, for the desalination model maximization (1S).

In order to find the global maximum, new input parameters are defined in Table *8*, for the second maximization attempt.

Table 8 – BA input parameters for the desalination model optimization – second attempt of maximization (1S).

Parameter	Value
Optimization Objective	MAXIMIZATION
Hybrid	TRUE
Adaptive	TRUE
Fperi	2
Fperf	0.5
n	20
Ngen	20
A0	0.7
r0	0.5
α	0.98
γ	0.97
Qmin	0
Qmax	2
ε	0.001
d	5
Number of Runs	9

For this second attempt, the global maximum was found in the seventh run (see Figure 9). In addition, the worst and optimum values were found, respectively, in the first and sixteenth generation, corresponding to the variables summarized in Table 9.

Table 9 – Parametric results for the desalination model optimization – second attempt of maximization (1S).

Gen	$T_{s,sat}$ [K]	$sal_{s,l}$ [ppm]	$Ti_{ch,l}$ [K]	D [m]	L [m]	Best [W/(m^2.K)]	Worst [W/(m^2.K)]
1	279.92	38,645	362.69	0.0270	1.7517	-	2669
16	279	15,000	380.00	0.0200	2.0000	4688	-

The optimum value evolution along runs is presented in Figure 9. In order to understand the mean value and standard deviation behaviour along generations, Figure 10 presents both variables.

It is important to note that the optimum is achieved when tube diameter and length correspond, respectively, to its lower and upper bounds. At the first sight, this can be contradictory comparing with equation (14). However, for this optimization case we are predicting the outlet chilled water temperature, instead of setting it to a constant value, which has high influence on the results.

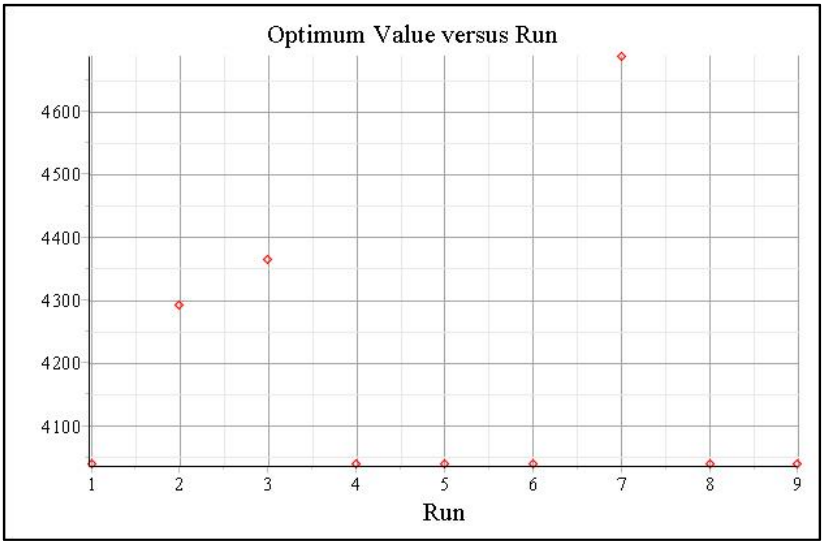


Figure 9 – Optimum value versus run, for the desalination model maximization (second attempt) (1S).

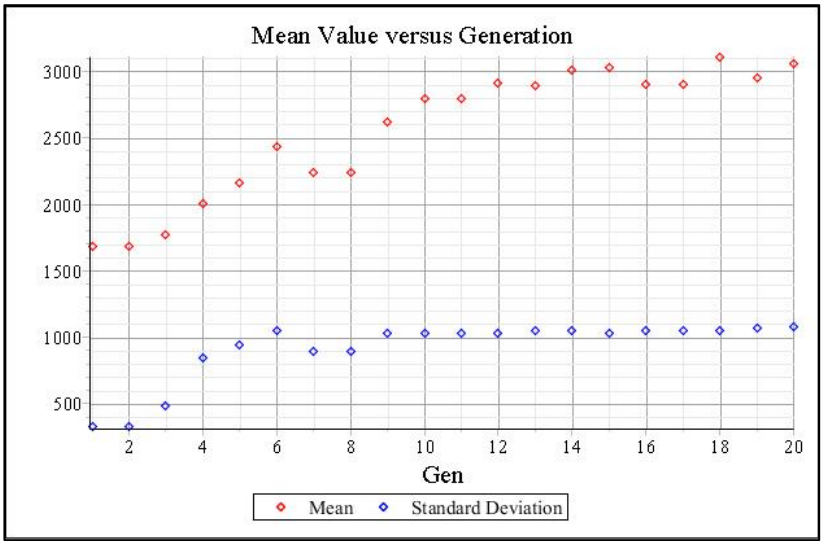


Figure 10 – Mean value and standard deviation versus generation, for the desalination model maximization (second attempt) (1S).

4.3.2. SECOND OPTIMIZATION STUDY (2S) – MINIMIZATION

For the minimization case, the next results correspond to the BA input parameters presented in Table 6.

The optimum and worst values were found, respectively, in the first and fifth generation, corresponding to the variables summarized in Table 10.

Table 10 – Parametric results for the desalination model optimization – minimization (2S).

Gen	$T_{s,sat}$ [K]	$sal_{s,l}$ [ppm]	$Ti_{ch,l}$ [K]	D [m]	L [m]	Best [W/(m².K)]	Worst [W/(m².K)]
1	295.61	59,465	361.64	0.0244	1.7989	1566	-
5	300.00	90,000	380.00	0.0300	2.0000	-	1588

In Figure 11 are presented both variables mean value and standard deviation evolution.

Analysing Table 10, BA finds as the best minimum optimum, the value of $h \cong 1566\ W/(m^2.K)$. However, this result does not corresponds to a global optimum. The target global minimum is verified for a saturation temperature of 293 K, where $h \cong 895\ W/(m^2.K)$.

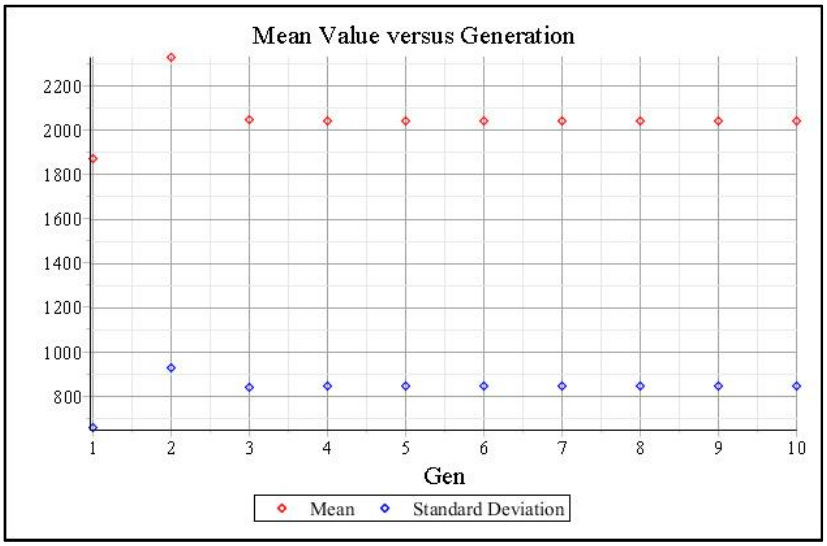


Figure 11 – Mean value and standard deviation versus generation, for the desalination model minimization (2S).

Once again to allow for a qualitative illustration of the obtained data for this particular case study, Figure 12 displays the minimum evolution along generations.

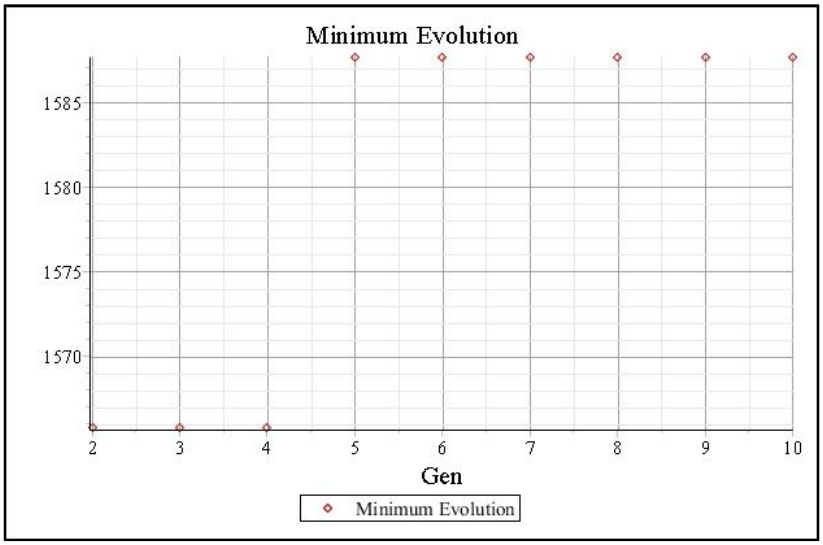


Figure 12 – Minimum evolution versus generation, for the desalination model minimization (2S).

Since the BA did not find the optimum minimum with the input parameters referred in Table 6, we need to change these. Table 11 summarizes the BA input parameters used in this new attempt.

Table 11 – BA input parameters for the desalination model optimization – second attempt of minimization (2S).

Parameter	**Value**
Optimization Objective	MINIMIZATION
Hybrid	TRUE
Adaptive	TRUE
Fperi	10
Fperf	5
n	10
Ngen	10
A0	0.01
r0	0.01
α	0.95
γ	0.98
Qmin	0
Qmax	1
ε	0.001
d	5
Number of Runs	20

The hybrid and adaptive modes have been both activated. In addition, the number of runs is set to 20, in order to increase the number of iterations. The worst and optimum values were found, respectively, in the first and third generation, corresponding to the variables summarized in Table 12, only for the best run. In the present case study, the best run was the second one (see Figure 13).

Table 12 – Parametric results for the desalination model optimization – second attempt of minimization (2S).

Gen	$T_{s,sat}$ [K]	$sal_{s,l}$ [ppm]	$Ti_{ch,l}$ [K]	D [m]	L [m]	Best [W/(m^2.K)]	Worst [W/(m^2.K)]
1	284.60	81,478	356.88	0.0275	1.5234	-	1257
3	300.00	90,000	350.00	0.0300	1.5000	933	-

The BA finds that the lower optimum is verified when the salinity increases to 90,000 ppm, as we expected. However, the saturation temperature is still on its end bounds (300 K). The most important conclusions retrieved from this second attempt of minimization are the following:

- The use of Hybrid = TRUE and Adaptive = TRUE, related to DE local search enables a more efficient optimum convergence;
- The BA finds that the minimum optimum appears when:
  - $sal_{s,l}$ is equal to its respective upper bound (90,000 ppm);
  - $Ti_{ch,l}$ is equal to its respective lower bound (350 K);
  - D is equal to its respective upper bound (0.030 m);
  - L is equal to its respective lower bound (1.5 m).

The optimum value evolution along runs is presented in Figure 13.

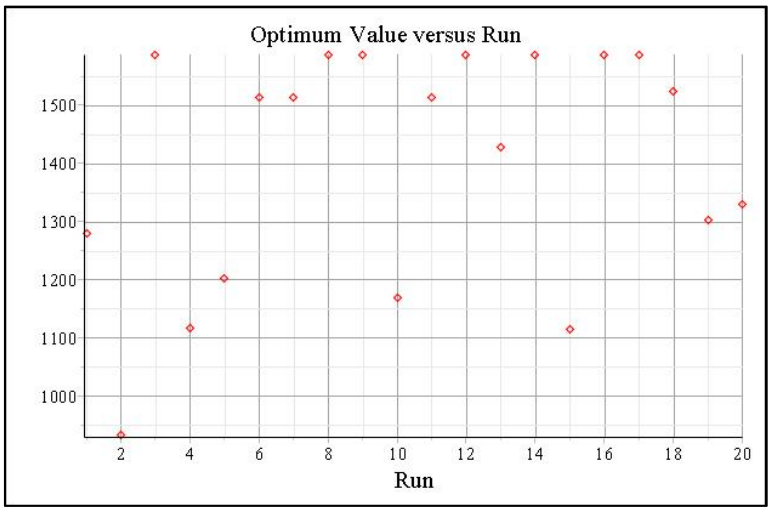


Figure 13 – Optimum value versus run, for the desalination model minimization (second attempt) (2S).

The minimum optimum was not found yet, because with only 10 generations (Ngen), the BA did not have enough iterations to properly update the saturation temperature $T_{s,sat}$.

Based on the previous conclusions, Table 13 gather several results for the same conditions analysed in the previous studies (to enable a valid comparison), but with different BA input parameters, to analyse their influence on the results.

For every optimization run presented in Table 13, the objective is minimization, hybrid and adaptive modes are activated, $Q_{min} = 0$, $\varepsilon = 0.001$ and $N_{gen} = 50$.

Table 13 – BA input parameters (left side) and their respective parametric results (right side) for the desalination model optimization (BA inputs' influence) – minimization.

BA Input Parameters								**Heat Transfer Parameters**					
$Fper_i$	**$Fper_f$**	**n**	**A0**	**r0**	**α**	**γ**	**Q_{max}**	**$T_{s,sat}$ [K]**	**$sal_{s,l}$ [ppm]**	**$Ti_{ch,l}$ [K]**	**D [m]**	**L [m]**	**OF Optimum Evolution [W/(m².K)]**
1	0.5	20	1.5	0.25	0.95	0.98	1	300.00	90,000	352.88	0.0300	2.0000	1233
1.5	0.5	10	1	0.5	0.94	0.99	1	283.48	90,000	359.44	0.0300	1.5000	1174
1	0.5	10	1	0.5	0.95	0.98	1	290.51	63,468	357.93	0.0279	1.5000	1158
1	0.5	20	2	0.25	0.95	0.98	1	292.10	84,441	359.97	0.0267	1.5000	1133
1.5	0.25	10	1.5	1.25	0.90	0.96	1	295.96	63,587	350.00	0.0300	1.8366	1121
1	0.5	10	3	0.1	0.92	0.96	1	295.12	79,243	350.79	0.0300	1.8714	1105
1	0.5	10	1.5	0.25	0.95	0.98	2	285.43	89,302	354.47	0.0300	1.5665	1090
1	0.5	40	1.5	0.25	0.95	0.98	1	289.53	78,804	355.11	0.0300	1.6794	1087
1	0.5	10	2	0.25	0.95	0.98	1	287.53	83,418	351.62	0.0274	1.5000	1084
1	0.5	20	3	0.1	0.92	0.96	1	289.64	90,000	366.21	0.0300	1.5000	1056
1	0.5	10	2.5	0.05	0.90	0.96	1	297.09	59,454	350.00	0.0300	1.5000	969
1	0.5	10	1	0.5	0.94	0.99	1	293.33	76,288	350.00	0.0292	1.5000	954
1	0.5	10	1.5	0.25	0.95	0.98	1	297.86	90,000	350.00	0.0300	1.5000	912
1	0.5	10	1.5	0.25	0.95	0.98	1	295.62	90,000	350.00	0.0300	1.5000	903

5. CONCLUSIONS

In the present paper, several optimization studies of a falling film desalination model were performed, using a BAT-inspired algorithm. Several validations were carried out, as well as different case studies, in order to understand and characterize the influence of each heat transfer variable on the heat transfer coefficient correlation used, as well as the influence of the optimization input parameters.

For saturation temperatures above 322 K the Han & Fletcher empirical correlation should be used to predict the FFHTC – the increase of saturation temperature corresponds to the increase of the FFHTC. On the other hand, for saturation temperature less than 300-298 K, Shahzad et al. correlation incorporates the additional effect from the micro-bubble agitation (which enhances the FFHTC), including parameters such as salinity, saturation temperature and vapour specific volume.

The computed OF revealed an appropriate approximation in comparison with the experimental data retrieved from [4]. In addition, all different curves converge to the Han & Fletcher correlation, for saturation temperatures above 300 K, as it should be. In validation conditions, assuming $Ti_{ch,l}$ and $To_{ch,l}$ as constant temperatures, it can be concluded that when both saturation temperature $T_{s,sat}$ and salinity $sal_{s,l}$ decrease, the FFHTC increases.

The results retrieved from the BA optimization, in validation conditions, confirm the conclusions of the first validation study, with the exact same variables and OF values.

Predicting the outlet chilled water temperature $To_{ch,l}$, the input heat flux is not directly and inversely proportional with, respectively, the tube diameter and length, as was verified in the validation conditions (with both $Ti_{ch,l}$ and $To_{ch,l}$ constant) – or at least, this statement cannot be made without running the algorithm, because the evaporation rate is highly dependent on the inlet and outlet temperatures.

From the optimization section it is possible to state that the maximum optimum corresponds to the following variables:

- $T_{s,sat}$ is equal to its respective lower bound (279 K);
- $sal_{s,l}$ is equal to its respective lower bound (15,000 ppm);
- $Ti_{ch,l}$ is equal to its respective upper bound (380 K);
- D is equal to its respective lower bound (0.020 m);
- L is equal to its respective upper bound (2 m).

On the other hand, the minimum optimum corresponds to the following variables:

- $T_{s,sat}$ is in between its lower and upper bounds (approximately 293 K);
- $sal_{s,l}$ is equal to its respective upper bound (90,000 ppm);
- $Ti_{ch,l}$ is equal to its respective lower bound (350 K);
- D is equal to its respective upper bound (0.030 m);
- L is equal to its respective lower bound (1.5 m).

In addition, is it verified that the input parameters of BA are determinant to achieve the optimization goals. For the minimization case study the global minimum was not found by the algorithm, but it is concluded that the most important parameters that influence the results are the loudness (A0), pulse rate (r0) and their respective constants for progressive variation, which are, respectively, α and γ. In order to achieve better results on this particular case, further studies need to be done to better characterize this heat transfer behaviour.

ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support given by FCT/MEC through Project PTDC/ATP-AQI/5355/2012 and Project LAETA - UID/EMS/50022/2013

NOMENCLATURE

Related to heat transfer

T_{ref}	Reference saturation temperature [K] ($T_{ref} = 322\ K$)
$\dot{m}$	Mass flow rate [kg/s]
sal_{ref}	Reference seawater salinity [ppm] ($sal_{ref} = 30{,}000\ ppm$)
v_{ref}	Reference specific volume at 295 K [m^3/kg] ($v_{ref} = 52.65\ m^3/kg$)
FFHTC	Falling film heat transfer coefficient
h	Heat transfer coefficient [W/(m^2.K)]
Γ	Liquid film mass flow rate per meter of tube length [(kg/s)/m of tube length]
Ai	Tube inner area [m^2]
At	Tube transversal area [m^2]
Cp	Specific heat [J/(kg.K)]
D	Tube inner diameter [m]
L	Tube length [m]
Nu	Nusselt number
P	Evaporator absolute pressure [kPa abs]
Pr	Prandtl number
Re	Reynolds number
T	Saturation temperature [K]
Ti	Inlet temperature [K]
To	Outlet temperature [K]
$deltaT$	Temperature difference [K]
g	Gravitational acceleration [m/s^2] ($g = 9.81\ m/s^2$)
k	Thermal conductivity [W/(m.K)]
q	Input heat flux [W/m^2]
sal	Salinity [ppm]
u	Velocity [m/s]
v	Specific volume [m^3/kg]
μ	Dynamic viscosity [kg/(m.s)]
ρ	Density [kg/m^3]

Subscripts

Γ	Liquid film (outside the tube)
ch	Chilled water (flowing inside the tube)
g	Vapor phase

l	Liquid phase
ref	Reference
s	Feed water (saline/seawater)
sat	Saturation

Related to optimization

A0	Loudness
BA	Bat-algorithm
d	Number of design variables
Fper	Perturbation factor in DE
n	Population's dimension
Ngen	Number of generations
OF	Objective function
Q_{max}	Pulse frequency maximum bound
Q_{min}	Pulse frequency minimum bound
r0	Pulse rate
α	Constant for progressive variation of loudness
γ	Constant for progressive variation of pulse rate
ε	Scaling factor

Subscripts

f	final state
i	initial state

REFERENCES

[1] I.C. Karagiannis, P.G. Soldatos. Water desalination cost literature: review and assessment. Desalination 223 (2008) 448–456.

[2] M.A. Eltawil, Z. Zhengming, L. Yuan. A review of renewable energy technologies integrated with desalination systems, Mohamed. Renewable and Sustainable Energy Reviews, 13 (2009), 2245–2262.

[3] Hassan K. Abdulrahim, Mohamed A. Darwish. Thermal desalination and air conditioning using absorption cycle. Desalination and Water Treatment, (2014) 1-20.

[4] M.Wakil Shahzad, A. Myat, W.G. Chun, K.C. Ng. Bubble-assisted film evaporation correlation for saline water at sub-atmospheric pressures in horizontal-tube evaporator. Applied Thermal Engineering, 50 (2013) 670-676.

[5] Xin-She Yang and Amir H. Gandomi, Bat Algorithm: A Novel Approach for Global Engineering Optimization, Engineering Computations, 29(5), (2012) 464-483.

[6] G. Liu, H. Huang, S. Wang, Z. Chen. An Improved Bat Algorithm with Doppler Effect for Stochastic Optimization. International Journal of Digital Content Technology and its Applications 6(21) (2012) 326-336.

[7] J.A. Ruiz-Vanoye, O. Díaz-Parra, F. Cocón, A. Soto, M.A.B. Arias, G. Verduzco-Reyes, R. Alberto-Lira. Meta-Heuristics Algorithms based on the Grouping of Animals by Social Behavior for the Traveling Salesman Problem. International Journal of Combinatorial Optimization Problems and Informatics, 3(3), (2012) 104-123.

[8] I. Fister Jr., D. Fister, X-S Yang. A Hybrid Bat Algorithm. Elektrotehniski Vestnik 80(1-2), (2013) 1-7.

[9] J.C. Han, L.S. Fletcher. Falling film evaporation and boiling in circumferential and axial grooves on horizontal tubes. Industrial and Engineering Chemistry Process Design and Development, 24 (1985) 570-575.

[10] M.A. Ramos Loja, C.M. Mota Soares, C.A. Mota Soares. Modelling and Design of Adaptive Structures Using B-Spline Method. Composite Structures, 57, (2002) 245-251.

[11] M.A.R. Loja, C. M. Mota Soares, J. I. Barbosa. Optimization of Magneto-Electro-Elastic Composite Structures Using Differential Evolution. Composite Structures, 107, (2014), 276-287.

[12] Salerno J. Using the Particle Swarm Optimization Technique to Train a Recurrent Neural Model, IEEE International Conference on Tools with Artificial Intelligence (1997) 45-49.

[13] M.A.R. Loja. On the Use of Particle Swarm Optimization to Maximize Bending Stiffness of Functionally Graded Structures. Journal of Symbolic Computation, 61-62, (2014) 12-30.

[14] Holman, J. P. Heat Transfer. 10th. s.l.: McGraw-Hill Series in Mechanical Engineering, 2009.

SYMCOMP 2015
Faro, March 26-27, 2015

AN APPROACH TO ROBUST PADÉ APPROXIMATION OF ORTHOGONAL POLYNOMIAL EXPANSIONS

J.C. Matos[1*], J. Matos[2] and M.J. Rodrigues[3]

1: Departamento de Matemática
Laboratório de Engenharia Matemática
Instituto Superior de Engenharia do Porto
Porto, Portugal
e-mail: jem@isep.ipp.pt

2: Instituto Superior de Engenharia do Porto
Centro Matemática da Universidade do Porto
Laboratório de Engenharia Matemática
Porto, Portugal
e-mail: jma@isep.ipp.pt

3: Faculdade of Ciências da Universidade do Porto
Centro Matemática da Universidade do Porto
Universidade of Porto
Porto, Portugal
e-mail: mjsrodri@fc.up.pt

Keywords: Padé approximation, Froissart doublets, Spectral methods.

Abstract. *We present an approach to robust Padé approximation of orthogonal polynomial expansions. This approach is based on the robust Padé approximation of power series, introduced by P. Gonnet et al, which allows to overcome some numerical problems in the computation of classical Padé approximants. These problems are due mainly to the use of spectral coefficients with noise and to the resolution of ill-conditioned systems of linear equations. As an application, we give an example of robust Padé approximants Legendre expansions, computed from spectral solutions of stiff differential equations.*

1 INTRODUCTION

Let f be a function represented by a formal power series

$$f(z) \sim \sum_{k=0}^{\infty} c_k z^k. \tag{1}$$

A Padé approximant (PA) of f is a rational function whose numerator and denominator are chosen so that its power series expansion agrees with f as far as possible. The PA are mentioned, for first time, by George Andersen (1736-1740) and Leonhard Euler (1707-1783). Charles Hermite (1822-1901) and Carl Lindemann (1852-1939) proved the transcendence of the numbers e and π using a generalization of Padé approximants. This kind of rational approximation received the name Padé approximation in honour of Henri Padé (1863-1953) who was the first mathematician to make a systematic study of these approximants in his PhD thesis.
Nowadays, PA are found in many numerical algorithms: equation solving, integration, integro-differential equations, approximation of special function and $z-$transform. The PA are also related with non linear extrapolation methods, e.g. the $\epsilon-$algorithm and the Shanks transformation, and they are a powerful technique to investigate non-linear problems.
There are several generalizations of Padé approximants: multi-point Padé approximants, algebraic and differential Hermite-Padé approximants, Baker-Gammel approximants, Padé-Borel approximants and Padé approximants from orthogonal polynomial expansions (being the last ones the main subject studied in this work). These rational approximations have a wide areas of applications: pure mathematics, numerical analysis, theoretical physics, chemistry, mechanics.
Throughout this paper, we work with PA from power series and PA from orthogonal polynomial expansions. To prevent eventual ambiguities, we will use the term *Taylor-Padé approximant* (TPA) to refer Padé approximants from power series and *Fourier-Padé approximants* (FPA) when we refer to a Padé approximant from orthogonal polynomial expansion. If we use a prescribed orthogonal polynomial system, e.g. Chebyshev or Legendre polynomials, we will use the terms *Chebyshev-Padé approximants* (CPA) and *Legendre-Padé approximants* (LPA), respectively.
The definition of a FPA is similar to the definition of a TPA. In fact, let $\{\varphi_k\}_{k\geq 0}$ be a system of orthogonal polynomials, with respect to an weight function w. Given a function f represented by a formal orthogonal polynomial expansion

$$f(z) \sim \sum_{k=0}^{\infty} c_k \varphi_k(z). \tag{2}$$

A FPA of f is a rational function whose numerator and denominator are chosen so that is expansion agrees with f as far as possible.
In practice, the algorithms of calculation of these rational approximations are fragile. The reasons for that are essentially due to the following facts:

a) The Padé table can be *non-normal*,

b) The existence of *Froissart doublets*, originated by rounding errors on a computer and errors on the series coefficients.

In order to overcome this drawback P. Gonnet *et al* (2013) proposed a new rational approximant, called robust Padé approximants and gave an algorithm to compute them [3]. B. Beckermann and Ana C. Matos, have studied the algebraic properties of these approximants and they gave a proof for their forward stability (or robustness).
The main purpose of his paper is to present a robust FPA, in the sense that this PA have not Froissart doublets.
Next section is dedicated to analyse the location of Froissart doublets and in the section 3 we describe the robust TPA algorithm given in the P. Gonnet *et al* paper [3]. The section 4 is dedicated to present a procedure to find robust FPA, in the sense that, this FPA have not Froissart doublets. In section 5 we present an example as application, and in the last section we will make some conclusions.

2 FROISSART DOUBLETS AND THEIR LOCATION

M. Froissart [2] had performed numerical experiences and studied the location of poles and zeros of TPA. He noted, that diagonal (i.e. type (p,p) TPA) of perturbed power series have pairs of poles with a close-by zeros. Later these pairs of zeros/poles have received the name of Froissart doublets. Here, we expand the Froissart observations to the location of Froissart doublets originated by noisy data on the Fourier coefficients or by the use of floating point arithmetic. First of all we remark that the formal definition of a Froissart doublet given by H. Stahl [9] is an "asymptotic definition". Thus, this definition is a useless for our purpose. We define a Froissart doublet of a PA (TPA or a FPA) in the following way.
Given a non negative real number tol we say that a pair (η, ξ) is a Froissart doublet of a (p, q) type PA if: η is a zero, ξ is a pole and $|\eta - \xi| < \text{tol}$.
Consider the power series $f(z) = \sum_{k=0}^{\infty} c_k z^k$ and the perturbed power series, $f_\epsilon(z) = \sum_{k=0}^{\infty}(c_k + \epsilon r_k)z^k$ and $f_\omega(z) = \sum_{k=0}^{\infty}(c_k + \omega 2^{-k} r_k)z^k$, where ϵ, ω are two small positive numbers and r_k, $k = 0, 1, \ldots$ are random complex numbers i.i.d. uniformly distributed on the disk $|z| < 1$. Then the diagonal PA of the perturbed series f_ϵ (f_ω) have Froissart doublets located nearby the circumference $|z| = 1$ ($|z| = 2$). These property may be explained with the help of the following result about random power series [7]

Theorem 1 *Let $(\Omega, \mathcal{A}, \mathcal{P})$ a probability space and X_n, $n = 0, 1, \cdots$, are independent complex random variables. Then given $\omega \in \Omega$:*

1. *The convergence ratio $r(w)$ of the power series $\sum_{n=0}^{\infty} X_n(\omega) z^n$ with probability one is given by*
$$r(w) = \left(\lim_{n \to \infty} \sup |X_n(\omega)|^{1/2} \right)^{-1}.$$
2. *If X_n are symmetric and $0 < r(\omega) < \infty$ then, the circumference $|z| = r(\omega)$ is, with probability one, a natural boundary of the function $F(z; \omega) = \sum_{n=0}^{\infty} X_n(\omega) z^n$.*

Thus, we have, for the non deterministic part of f_ϵ and f_ω, the following

Corollary 1 *The random series* $\sum_{k=0}^{\infty}(\epsilon r_k)z^k$ *and* $\sum_{k=0}^{\infty}(\omega 2^{-k} r_k)z^k$ *have respectively convergence ratios, with probability one,* $r_\epsilon = 1$ *and* $r_\omega = 2$*, being the circumference,* $|z| = 1$ *a natural boundary of the series* $\sum_{k=0}^{\infty}(\epsilon r_k)z^k$ *and* $|z| = 2$ *natural boundary of the series* $\sum_{k=0}^{\infty}(\omega 2^{-k} r_k)z^k$*.*

Consequently, as the singularities of functions are represented by poles of its Padé approximants, the location of Froissart doublets obtained in numerical experiences are justified (the Froissart doublets lie nearby the natural boundary of the noise series).
If we consider random Chebyshev series or random Legendre series $\sum_{k=0}^{\infty}(\epsilon r_k)\varphi_k(z)$ and $\sum_{k=0}^{\infty}(\omega 2^{-k} r_k)\varphi_k(z)$, where $\varphi_k(z)$ represent the Chebyshev or the Legendre orthogonal polynomials on the interval $[-1, 1]$, with ϵ, ω and r_k, $k = 0, 1, \ldots$, in the same condition as in Taylor's case. Numerical experiences, performed in [6], suggest that the diagonal CPA or the LPA have Froissart doublets nearby the image of the Joukowski transform $J(z) = (z+z^{-1})/2$ of the natural boundaries of the correspondent power series. To be more precise The diagonal CPA or LPA computed with series $\sum_{k=0}^{\infty}(\epsilon r_k)\varphi_k(z)$ have Froissart doublets nearby the real interval $[-1, 1]$ and the diagonal CPA or LPA computed with series $\sum_{k=0}^{\infty}(\omega 2^{-k} r_k)\varphi_k(z)$ have Froissart doublets nearby the ellipse $\left|z + \sqrt{z^2 - 1}\right| = 2$. We remark that in all experiences performed the Froissart doublets, originated exclusively from the use of floating point arithmetic, are located on the circumference $|z| = 1$ (in TPA case) or on the interval $[-1, 1]$ (in CPA or LPA). This is really a serious drawback of FPA relatively to the TPA, since the poles of Froissart doublets are real (or are "almost" real) and they are nearby the orthogonal interval. Thus we must be very careful when we use FPA. In fact it is essential to eliminate the Froissart doublets, at least those who are in the neighbourhood of the interval of orthogonality. In order to illustrate the observations above we give the following example.

Example 1 *Consider the rational function* $f(z) = (4z^2 + 4z - 11)/4z^3 - 12z^2 - 3$*. We perturbed the function with the two kind of noises (described above) with* $\epsilon = \omega = 10^{-5}$ *and we have computed the type* $(20, 20)$ *TPA with the function* f *perturbed with the two kind of noises. The location of poles and zeros of* $\pi_{20,20}$ *is shown in the pictures in left column of Figure 1. In the right column we present the location of poles and zeros of the type* $(20, 20)$ *CPA with* f *also perturbed with both kind of noises. In all pictures we can see* 3 *isolated poles indicated with black dots (that represent the 3 poles,* $z = 3$*,* $z = \pm i/2$ *of the non perturbed function* f*),and two zeros isolated that indicated with magenta circles (that represent the 2 zeros* $z = -(\sqrt{3}+1/2)$ *and* $z = \sqrt{3}-1/2$ *of* f*). The red circumferences in the pictures of left column represent the natural boundaries of the noises and in the right column the red lines represent the Joukowski transform of the natural boundaries. We can observe that there are, in all pictures,* 17 *Froissart doublets. They are located nearby the natural boundaries of the noises (left column) and nearby their Joukowski transform (right column). We note that, for the sake of clarity, the remain isolated zero is not included in all pictures.*

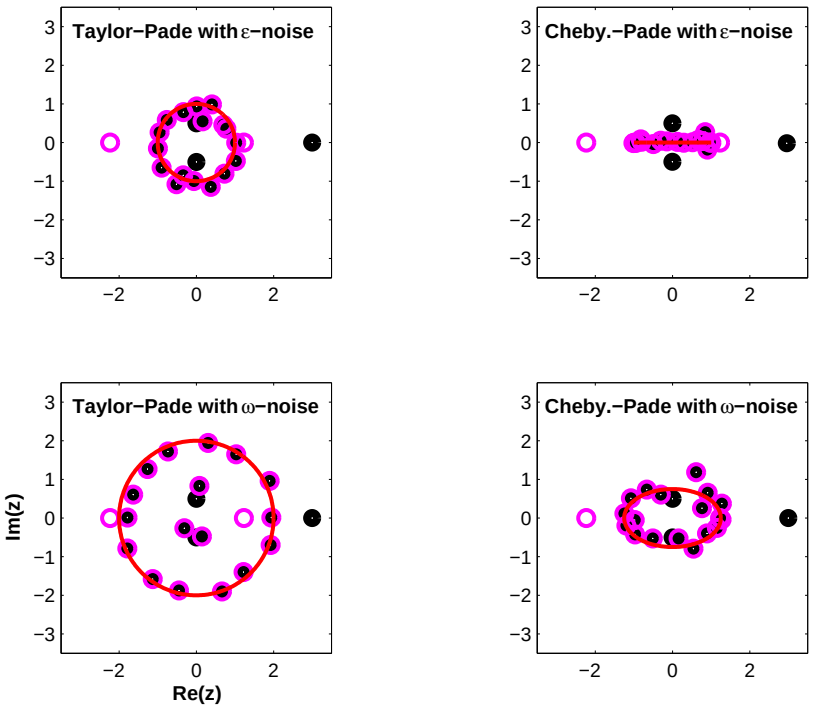


Figure 1: Location of poles (black dots) and zeros (magenta circles) of type $(20, 20)$ TPA and of type $(20, 20)$ CPA of the function f perturbed with two kind of noises.

3 ROBUST TPA ALGHORITHM

Let $\mathcal{P}_n$ denote the set of polynomials of degree up to n. Given two non negative integers p and q, the type (p, q) TPA of the function f, represented by the power series (1) is the rational function $\pi_{p,q} = N_{p,q}/D_{p,q}$, $N_{p,q} \in \mathcal{P}_p$ and $D_{p,q} \in \mathcal{P}_q$, that satisfies the relation

$$\pi_{p,q}(z) - f(z) = \mathcal{O}\left(z^{maximum}\right). \tag{3}$$

The equation (3) is non-linear, but if we multiply by $D_{p,q}$, we set the linear condition

$$f(z)D_{p,q}(z) - N_{p,q}(z) = \mathcal{O}\left(z^{maximum}\right). \tag{4}$$

Obviously, we require $D \not\equiv 0$ otherwise the condition becomes meaningful. The condition (5) has $p + q + 2$ unknowns (the coefficients of the polynomials $D_{p,q}$ and $N_{p,q}$), Thus it can always be satisfied through degree $p + q$ or higher,

$$f(z)D_{p,q}(z) - N_{p,q}(z) = \mathcal{O}\left(z^{p+q+1}\right). \tag{5}$$

Setting $N_{p,q} = a_0 + a_1 z + \ldots + a_p z^p$ and $D_{p,q} = b_0 + b_1 z + \ldots + b_p z^p$, we can write the condition (5) in matrix form depending of the values of p and q.
For example, if $p \geq q$ then condition (5) takes the Toeplitz form

$$\left[\begin{array}{c} a_0 \\ a_1 \\ \vdots \\ a_p \\ \hline a_{p+1} \\ \vdots \\ a_{p+q} \end{array}\right] = \left[\begin{array}{cccc} c_0 & & & \\ c_1 & c_0 & & \\ \vdots & \vdots & \ddots & \\ c_q & c_{q+1} & \cdots & c_0 \\ \vdots & \vdots & & \vdots \\ c_p & c_{p-1} & \cdots & c_{p-q} \\ \hline c_{p+1} & c_p & \cdots & c_{p-q+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p+q} & c_{p+q-1} & \cdots & c_p \end{array}\right] \left[\begin{array}{c} b_0 \\ b_1 \\ \vdots \\ b_q \end{array}\right] \tag{6}$$

with the convention that $a_k = 0$, $k \geq m+1$. The other case ($p < q$) is similar and is not relevant to the exposition, so we do not include it here. Thus the coefficients of the denominator are given by a non trivial solution of the linear system formed by q equations (represented by the block matrices below the horizontal line) and $q+1$ unknowns b_k, $k = 0, 1, \ldots, q$. To be more precise, the vector $\mathbf{b} = [b_0 \; b_1 \; \cdots \; b_q]^T$ is a non trivial solution of the matrix equation

$$\mathbf{0} = \mathbf{C}\mathbf{b}, \tag{7}$$

or that the vector $\mathbf{b}$ is a null vector of $\mathbf{C}$, where $\mathbf{C}$ is the $q \times (q+1)$ Toeplitz matrix

$$\mathbf{C} = \left[\begin{array}{cccc} c_{p+1} & c_p & \cdots & c_{p-q+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p+q} & c_{p+q-1} & \cdots & c_p \end{array}\right]. \tag{8}$$

The algorithm used in [3] is based in the singular value decomposition of $\mathbf{C}$, and, (7) takes the form

$$\mathbf{0} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\mathbf{b}, \tag{9}$$

where the $q \times (q+1)$ matrix $\mathbf{\Sigma}$ is diagonal with real entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_q \geq 0$. We have two cases, depending of the value of the last singular value σ_q.

- If $\sigma_q > 0$, then $\text{rank}\,(\mathbf{C}) = q$ and the last column of $\mathbf{V}$ provides a unique non trivial solution, up to a scale factor, of (7). If the matrix $\tilde{\mathbf{C}}$, that results from elimination of the first column of $\mathbf{C}$, is singular then from (6) we have $b_0 = a_0 = 0$. Thus, $N_{p,q}$ and $D_{p,q}$ have a common factor z^λ, $\lambda \geq 1$ and the degree of the numerator (denominator) of the type (p, q) TPA are less then p (q).

- If $\sigma_\delta = \ldots = \sigma_q = 0$ and $\sigma_{\delta-1} \neq 0$, then $\text{rank}\,(\mathbf{C}) = \delta < q$ and the SVD of $\mathbf{C}$ has a non zero null vector that is zero in its first $q - \delta$ positions. Thus, we have $b_0 = \ldots = b_{q-\delta-1} = 0$ and, from (6), $a_0 = \ldots = a_{q-\delta-1} = 0$.

The above observations are the key idea that motivated the algorithms proposed in [3].

Algorithm: Robust TPA for noisy data or floating point arithmetic.

Input: $p, q \geq 0$ ($p \geq q$), Taylor coefficients $c_0, \ldots, c_{p+q}$ of a function f, tolerance tol ≥ 0.
Output: $N_{p,q}(z) = a_0 + \ldots a_p z^p$, $D_{p,q}(z) = b_0 + \ldots b_q z^q$.

1. Define $\tau = \text{tol} \cdot \|\mathbf{c}\|$. If $|c_0|, \ldots, |c_q| \leq \tau$, set $N_{p,q} = 0$ and $D_{p,q} = 1$ and STOP.
2. If $q = 0$, set $N_{p,q}(z) = a_0 + \ldots a_p z^p$ and $D_{p,q} = 1$ and STOP.
3. Compute the SVD of the matrix $\mathbf{C}$. Let δ be the number of singular values of $\mathbf{C}$ that are greater than τ.
4. If $\delta < p$, set $q := \delta$ and $p := q - (p - \delta)$ and return to step 3.
5. Get $D_{p,q}$ from the null right singular vector $\mathbf{b}$ and $N_{p,q}$ of the upper part of (6).
6. If $|b_0|, \ldots, |b_{\lambda-1}| \leq \text{tol}$ for some $\lambda \geq 1$, zero the first λ coefficients of $N_{p,q}$ and $D_{p,q}$ and cancel the factor z^λ.
7. If $|b_{p+1-\lambda}|, \ldots, |b_p| \leq \text{tol}$ for some $\lambda \geq 1$, remove the last λ coefficients of $D_{p,q}$.
8. If $|a_{q+1-\lambda}|, \ldots, |a_q| \leq \tau$ for some $\lambda \geq 1$, remove the last λ coefficients of $N_{p,q}$.

4 A ROBUST FPA PROCEDURE

In this section we describe a direct approach to compute FPA without Froissart doublets. Unlike the approach described in the previous section, our procedure is a "direct approach", in the sense that we need to find the zeros and the poles of FPA in order to eliminate the Froissart doublets from FPA. Next we will describe an algorithm to compute linear proposed by A.C. Matos [5]

4.1 An algorithm to compute FPA

Consider a function f represented by the orthogonal polynomial expansion (2). Given two non negative integers p and q, a type (p, q) FPA of f is a rational function

$$\Phi_{p,q}(z) = \frac{N_{p,q}(z)}{D_{p,q}(z)} = \frac{\sum_{k=0}^{p} a_k \varphi_k(z)}{\sum_{k=0}^{q} b_k \varphi_k(z)}$$

that satisfies the condition

$$f(z) D_{p,q}(z) - N_{p,q}(z) = \mathcal{O}\left(\varphi_{\text{maximum}}(z)\right). \tag{10}$$

Like in previous section the condition (11) can always be satisfied through order $p + q$ or higher

$$f(z) D_{p,q}(z) - N_{p,q}(z) = \sum_{k \geq p+q+1} e_k \varphi_k(z). \tag{11}$$

Assuming that the orthogonal expansion of the functions $\varphi_k f$, $k = 0, 1, \ldots$ take the form

$$\varphi_k(z)f(z) = \sum_{j=0}^{\infty} h_{j,k}\varphi_j(z), \qquad k = 0, 1, \ldots$$

$$\sum_{i=0}^{q} h_{j,i}b_i = a_j, \quad j = 0, \ldots, p \tag{12}$$

$$\sum_{i=0}^{q} h_{j,i}b_i = 0, \quad j = p+1, \ldots, p+q. \tag{13}$$

The denominator coefficients are given by a non trivial solution of the homogeneous linear system (13) with $q+1$ equations and q unknowns. Defining the matrix

$$\mathbf{H_{p,q}} = \begin{bmatrix} h_{p+1,0} & \cdots & h_{p+1,q-1} \\ \vdots & & \vdots \\ h_{p+q,0} & \cdots & h_{p+q,q-1} \end{bmatrix}, \tag{14}$$

we have [5]

Proposition 1 *If* $\det(\mathbf{H_{p,q}}) \neq 0$*, then exist one and only one FPA such that* $b_q = 1$ *and* $e_{p+q+1} \neq 0$.

Assuming that $\mathbf{H_{p,q}}$ is regular and the normalization condition $b_q = 1$, the type FPA $\Phi_{p,q}$ is determined by equations (13)- (13) that take the matricial form

$$\mathbf{H_{p,q}} \cdot \mathbf{b_{p,q}} = -\mathbf{h}_{p,q} \tag{15}$$

$$\mathbf{a_{p,q}} = \mathbf{G_{p,q}} \cdot \mathbf{b_{p,q}} + \mathbf{g_{p,q}} \tag{16}$$

where,

$$\mathbf{a_{p,q}} = \begin{bmatrix} a_0 \ldots a_p \end{bmatrix}^T, \quad \mathbf{b_{p,q}} = \begin{bmatrix} b_0 \ldots b_{q-1} \end{bmatrix}^T,$$
$$\mathbf{g_{p,q}} = \begin{bmatrix} h_{0,q} \ldots h_{p,q} \end{bmatrix}^T, \quad \mathbf{h_{p,q}} = \begin{bmatrix} h_{p+1,q} \ldots h_{p+q,q} \end{bmatrix}^T,$$

and

$$\mathbf{G_{p,q}} = \begin{bmatrix} h_{0,0} & \cdots & h_{0,q-1} \\ \vdots & & \vdots \\ h_{p,0} & \cdots & h_{p,q-1} \end{bmatrix}.$$

The coefficients $h_{j,i}$ can be computed by recurrence using the following result [5]. Lets $x\varphi_i(x) = \alpha_i\varphi_{i+1}(x) + \beta_i\varphi_i(x) + \gamma_i\varphi_{i-1}(x)$, $i \geq 0$, be the recursion relation associated with the family of orthogonal polynomials $\{\varphi_k\}_{k\geq 0}$ and μ_i the w-norm of φ_i, $i \geq 0$ ($\mu_i = \left[\int \varphi_i^2(z)w(z)\mathrm{d}z\right]^{1/2}$).

Proposition 2 *The coefficients $h_{j,i}$ satisfy the recursion recurrence*

$$h_{i,j+1} = \frac{1}{\alpha_j}\left(\frac{\mu_{i+1}}{\mu_i}\alpha_i h_{i+1,j} + (\beta_i - \beta_j)h_{i,j} + \frac{\mu_{i-1}}{\mu_i}\gamma_i h_{i-1,j} - \gamma_j h_{i,j-1}\right),\ i,j \geq 1 \qquad (17)$$

with initial values $h_{i,0} = \varphi_0 c_i$, $i \geq 0$. And, the coefficients $h_{j,i}$ satisfy the relation $h_{j,i} = \mu_i/\mu_j h_{i,j}$.

We remark that to build a (p,q) type TPA we need to use the first $p+q+1$ coefficients. On other hand to compute a FPA of same type we must know the first $p+2q+1$ coefficients. This results because the multiplication law in powers of z, $z^n \cdot z^m = z^{n+m}$, it is much simpler than the multiplication law of orthogonal polynomials which takes the form $\varphi_n(z)\cdot\varphi_m(z) = \sum_{\ell=|m-n|}^{m+n} A_{m,n,\ell}\varphi_\ell(z)$.

4.2 Robust FPA procedure

In many practical problems, we only have access to a finite set of coefficients (Polynomials) instead we have access to a series. In this case, we will only be able to compute a finite number of FPA (or TPA). For example if we have access to the first $n+1$ coefficients then we are limited to compute the type (p,q) FPA that satisfy the inequality $p+2q \leq n$. Thus, it is important to distinguish the accessible FPA that have not Froissart doublets from the ones that have Froissart doublets. More, if a type (p,q) FPA, $\Phi_{p,q}$, have $\nu_{p,q}$ Froissart doublets thus the numerator and the denominator of $\phi_{p,q}$ share $\nu_{p,q}$ factors that "almost" cancel and $\phi_{p,q}$ is "almost" a type $(p-\nu_{p,q}, q-\nu_{p,q})$ rational function. A useful tool to distinguish these FPA is to build a table (that we named *Froissart table*) where, the (p,q)th entry is the number of Froissart doublets of $\Phi_{p,q}$. Note that in this table depends on the number tol that we set previously.
The next example clarifies the utility of the Froissart table and shows the relationship between the presence of noise and the presence of Froissart doublets.

Example 2 *Consider the exponential function $f(z) = e^z$ with Chebyshev expansion [4] $f(z) \sim 2\sum_{k\geq 0}' I_k(1)T_k(z)$ where the dash means that the first term is halved and $I_k(z)$ is the modified Bessel function of the first kind. Consider also the perturbed exponential function $f_\epsilon(z) \sim 2\sum_{k\geq 0}' (I_k(1)+\epsilon r_k)T_k(z)$, where r_k, $k \geq 0$, are random real numbers i.i.d. uniformly distributed on the interval $[-1,1]$ and ϵ is the strong of the noise.*
On Figure 2, we show the Froissart tables of f (left picture) and f_ϵ (right picture) with a tolerance tol $= 10^{-6}$ for CPA $\Phi_{p,q}$, $1 \leq p,q \leq 15$ and strong noise $\epsilon = 10^{-8}$ (in f_ϵ). We have marked the positive entries with a red rectangle to better distinguish the CPA that have Froissart doublets (we will say that these CPA are in the "red region") from the CPA that do not have Froissart doublets (we will say that these CPA are in the "white region"). This example is paradigmatic, it shows the fragility of FPA in presence of noisy data. In fact, the red region on the Froissart table of the unperturbed exponential function consists only of two entries (only $\Phi_{13,15}$ and $\Phi_{15,14}$ have one Froissart doublet each, cause

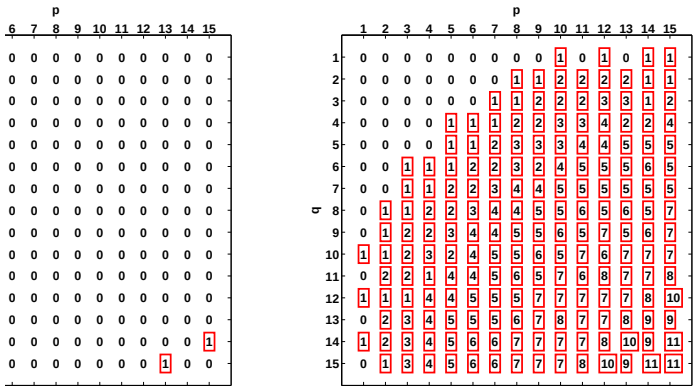


Figure 2: Froissart table of exponential function e^z with tol $= 10^{-6}$ without noise (left) and with noise, $\epsilon = 10^{-8}$ (right).

by rounding errors). By other hand, the introduction of a small noise has changed the pattern of the Froissart table (represented on right picture). Now, almost all entries $\nu_{p,q}$ *such that* $p + q > 9$*, are in the red region.*
If we restrict ourselves to a particular sequence of FPA of f_ϵ*, e.g. the diagonal sequence* $\Phi_{p,p}$ *(*$p = 1, 2, \ldots, 15$*), we have only four CPA in white region,* $\Phi_{p,p}$*,* $1 \leq p \leq 4$*. Thus, assuming that the quality of the approximation increases when p increase,* $\Phi_{4,4}$ *is the best diagonal FPA that have not Froissart doublets. If we cancel the factors related with the Froissart doublets. By inspecting the diagonal entries in the red region* $\nu_{p,p}$*,* $5 \leq 15$*, we can see that the degree of the numerators and denominators of* $\Phi_{p,p}$*,* $p \in 5, 6, 7, 8, 9, 15$*, is* $p - \nu_{p,p} = 4$ *while the FPA* $\Phi_{p,p}$*,* $10 \leq p \leq 14$ *have degree* $p - \nu_{p,p} = 5$*.*

This example suggests 3 different approaches to get a good robust FPA of a given function/series. Given a sequence of FPA Φ_{p_i,q_i}, where $\{p_i\}_{i\geq 0}$ and $\{q_i\}_{i\geq 0}$ are two non decrease sequences of non negative integers.

1. We just choose the last FPA Φ_{p_i,q_i} such (p_i, q_i) is in white region of Froissart table.

2. We choose a FPA Φ_{p_i,q_i} such that $p_i + q_i - 2\nu_{p_i,q_i}$ is maxim, and we compute the rational function $\tilde{\Phi}_{p_i,q_i}$, that results from cancelling the factors of numerator an denominator of Φ_{p_i,q_i} related with the Froissart doublets.

5 FILTERING LEGENDRE-TAU METHOD WITH A ROBUST LPA

Consider the non linear differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} - \alpha y^3 = 0, \tag{18}$$

with the condition $y(-1) = (1 + 2\alpha + \alpha^2)^{-1/2}$ and $\alpha \in]-1, 1[$. This ODE has solution

$$y(x) = \frac{1}{\sqrt{\alpha^2 + 1 - 2\alpha x}},$$

which has Legendre expansion

$$y(x) = \sum_{k=0}^{\infty} c_k P_k(x) = \sum_{k=0}^{\infty} \alpha^k P_k(x), \quad -1 < x < 1$$

where $P_0, P_1, \ldots$, are the Legendre polynomials.
Setting, in (18), $\alpha = 9/10$ the solution y has the closest singularity $\zeta = 1.05(5)$ nearby the interval of orthogonality $[-1, 1]$ and the Tau method has slow convergence rate. We pretend to filter a Legendre-Tau solution [8] of the non linear EDO using LPA. The Legendre-Tau solution, of order n, of (18) takes the form

$$y_n(x) = \sum_{k=0}^{n} c_k^{(n)} P_k(x),$$

and we define, the absolute coefficients errors $\Delta c_k^{(n)} = \left|(9/10)^k - c_k^{(n)}\right|$, $k = 0, 1, \ldots, n$, and the absolute Legendre-Tau error $\Delta y_{50}(x) = |y(x) - y_{50}(x)|$.
We present in Figure 5 the errors of the Legendre-Tau solution of order 50, $\Delta y_{50}(x)$ (top image) and the absolute coefficients errors $\Delta c_k^{(50)}$, $k = 0, 1, \ldots, 50$ (bottom image). We had choose the L-T solution of order 50 because it the best L-T approximation of y that we can compute, using a double precision software. By other hand, as we had observed in section 2, it is convenient to avoid random errors on the coefficients, which imply the existence of Froissart Doublets on the PA. We can observe that these errors are essentially due to the Tau projection.
In order to find a "good" robust LPA we show the Froissart table of LPA of y_{50}, with tolerance tol $= 10^{-6}$, on Figure LegendreFroissartTable. For simplicity, we will restrict to analyse the diagonal sequence, for others LPA sequences the procedure method is similar. We note that this table include all diagonal LPA of y_{50} since the LPA $\Phi_{p,q}$ may satisfy the relation $p + 2q \leq 50$. Inspecting the table we observe that:

1. $\Phi_{3,3}$ is the last diagonal LPA that is in white region,

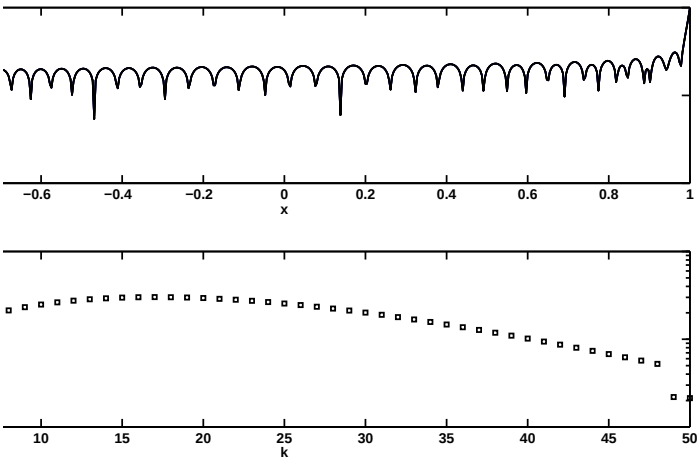


Figure 3: Absolute Legendre-Tau error $\Delta y_{50}(x)$ (top) and absolute Tau coefficients errors $\Delta c_k^{(50)}$, $k = 0, 1, \ldots, 50$ (bottom).

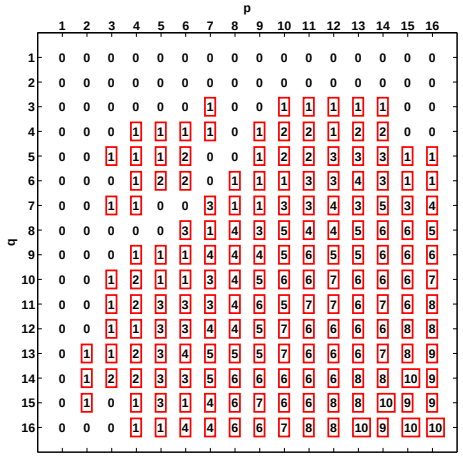


Figure 4: Froissart Table of $y_{50}(x)$ with tolerance tol $= 10^{-4}$ and $1 \leq p, q \leq 16$.

2. $\Phi_{13,13}$ is the the unique diagonal LPA $\Phi_{p,p}$ such has maximum numerical degree $p - \nu_{p,p} = 7$.

Thus, $\Phi_{3,3}$ is a robust LPA of y_{50} and we may to compute others robust approximants, like $\tilde{\Phi}_{13,13}$, using $\Phi_{13,13}$.

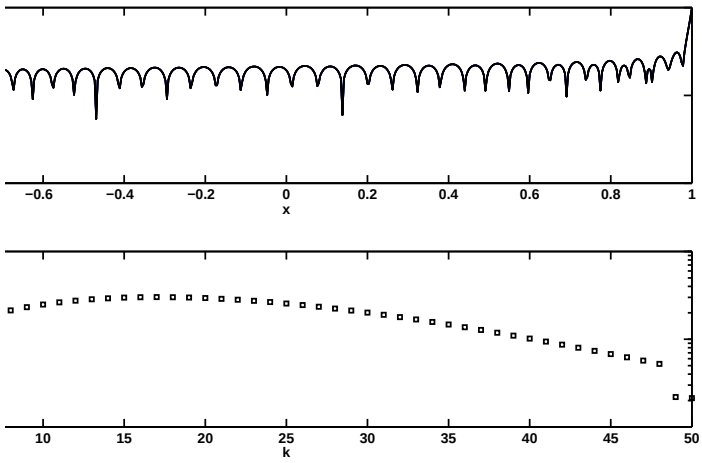


Figure 5: Absolute Legendre-Tau error $\Delta y_{50}(x)$ (top) and absolute Tau coefficients errors $\Delta c_k^{(50)}$, $k = 0, 1, \ldots, 50$ (bottom).

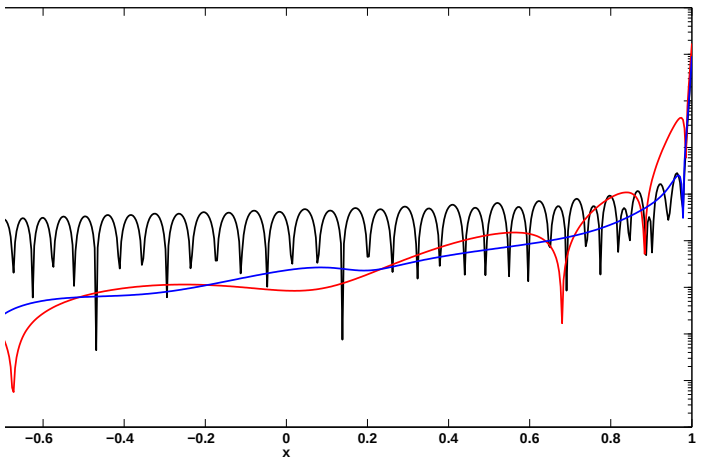


Figure 6: Absolute errors: $\Delta y_{50}(x)$ (black), $\Delta \Phi_{3,3}(x)$ (red) and $\Delta \tilde{\Phi}_{13,13}(x)$ (blue).

On Figure 6 we illustrate the filtering results. The Legendre-Tau error $\Delta y_{50}(x)$ (black line), The error of (3,3) type LPA $\Delta \Phi_{3,3}(x)$ (red line) and the error of the approximant $\Delta \tilde{\Phi}_{13,13}(x)$ i.e, the type $(13, 13)$ FPA with the factors related with the Froissart doublets cancelled (blue line). We can observe that the approximant $\tilde{\Phi}_{13,13}$ only improve the

approximation given by $\Phi_{3,3}$ for values nearby $x = 1$.

6 CONCLUSIONS

Numerical experiences suggest that the last type (p, q) FPA, of a given sequence, in the white region of the Froissart table, gives a good approximation. This approximation can frequently be improved, using robust approximants computed with FPA in the red region, but the improvement is not significant.
This procedure is a direct method, in the sense that we need to compute the zeros and the poles of FPA, in contrast with the algorithm to compute robust TPA, described in section 3, proposed by Gonnet et al. Thus it have some advantages, e.g. they have not Froissart doublets while the Gonnet's algorithms could produce robust TPA with Froissart doublets [1]. By other hand our procedure depends on the precision that we compute the zeros and poles of FPA, and this is not easy, e.g. in case of polynomials with high degrees. So it would be interesting to find a non direct algorithm to compute robust FPA.

REFERENCES

[1] Beckermann, B. and Matos, A.C. "Algebraic properties of robust Padé approximants", *Journal of Approximation Theory* Vol. **190**, pp. 91-115, 2015.

[2] Froissart M. "Approximation de Padé: application à la physique des particules élémentaires". *in RCP, Programme*, No. 25, Vol. **9**, pp.1-13, CNRS, Strasbourg, 1969.

[3] Gonnet, et al. "Robust Padé Approximation via SVD", SIAM Review, Vol. **55**, pp. 101-117, 2013.

[4] Mason, J.C. and Handscomb, D.C. "Chebyshev Polynomials", Chapman and Hall, 2002.

[5] Matos, A.C., "Recursive computation of Padé-Legendre approximants and some acceleration properties", *Numerische Mathematik*, pp. 535-560, 2001. Vol. **89**,

[6] Matos, J.C., Matos, J. and Rodrigues, M.J., "On the localization of Zeros and Poles of Chebyshev-Padé approximants from perturbed functions", *In Computational Science and its applications ICCSA*, Vol. **8584** of lecture notes in computer science, pp. 481-492 Springer International Publishing, 2014.

[7] Kahane, J.P., *Some Random series of functions*, Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1993.

[8] Ortiz, E.L. and Samara, H. "An operational approach to the tau method for the numerical solution of non-linear differential equations", *Computing*, Vol. **27**, pp. 15-25, 1981.

[9] Stahl, H. "Spurious poles in Padé approximation", *Journal of Approximation Theory* Vol. **86**, pp. 139-204, 1997.

SYMCOMP 2015
Faro, March 26-27, 2015

LEARNING TO SOLVE LINEAR SECOND-ORDER ORDINARY DIFFERENTIAL EQUATIONS WITH CONSTANT COEFFICIENTS BY USING INTERACTIVE SOFTWARE

Celestino Coelho[1] and Rui Marreiros[2]

1: Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade do Algarve
Campus de Gambelas 8005-139 Faro, Portugal
e-mail: ccoelho@ualg.pt

2: Center for Functional Analysis, Linear Structures and Applications (CEAFEL)
Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade do Algarve
Campus de Gambelas 8005-139 Faro, Portugal
e-mail: rmarrei@ualg.pt

Keywords: Linear Ordinary Differential Equations, Constant Coefficients, Method of Undetermined Coefficients, Symbolic Computation, Wolfram Mathematica, Computable Document Format

Abstract. *From the early days of the calculus, differential equations have been an area of great theoretical research and practical applications in several branches of science. The scope of this work is to present new software that is able to show all the steps in the process of obtaining the solution of a linear second-order ordinary differential equation by using the method of the undetermined coefficients.*

1 INTRODUCTION

In the present work we present interactive software that can be used to aid in the process of teaching linear second-order ordinary differential equations (ODEs) with constant coefficients. This work can be seen as a natural complement to our previous work [3]. In [3] we also studied the second-order linear differential equations with constant coefficients, but used a different method to obtain the particular solution. In what relates to the implementation of the computational tool, in [3] we started with the homogeneous case and then extended it to the construction of a particular solution by using the variation of parameters method (the "general method") in the nonhomogeneous case. In the present work we consider the undetermined coefficients method to construct a particular solution for the same type of equations. As we will see below, this method can only be applied to a special form that the right-hand side (rhs) of the differential equation can assume.
This work is organized as follows. We begin shortly recalling the basic concepts of the theory of the second-order linear ODE; for references we address the reader to our cited work [3], and, for instance, [9], [7] and [8]. Then follows the main section where we describe the implementation and the use of the software.

2 BASIC CONCEPTS

This section is used to recall the basic concepts of the theory of the second-order linear differential equations.

Definition 2.1 *A second-order linear ordinary differential equation in the dependent variable y and the independent variable x is an equation that can be written in the form*

$$y'' + ay' + by = f, \tag{1}$$

where a, b and f are continuous real functions on a real interval I, i.e., $x \in I \subset \mathbb{R}$.
If f is not identically zero on I, then the equation (1) is said to be nonhomogeneous.
If f is identically zero on I, then we obtain the so called corresponding homogeneous equation to (1), being its form naturally given by

$$y'' + ay' + by = 0. \tag{2}$$

The following result takes place.

Theorem 2.1 *Let y_1 and y_2 be two linearly independent solutions of the homogeneous equation (2) on an interval I. Then the general solution of (2) is given by*

$$y = C_1 y_1 + C_2 y_2,$$

where $C_1, C_2 \in \mathbb{R}$.

The general solution of the nonhomogeneous equation (1) is given in the next theorem.

Theorem 2.2 *Let y be the general solution of the nonhomogeneous equation* (1)*, y_h be the general solution of the homogeneous equation* (2) *and y_p a particular solution of the nonhomogeneous equation* (1)*. Then*

$$y = y_h + y_p.$$

First, let us consider the homogeneous equation with constant coefficients, i.e.,

$$y'' + ay' + by = 0, \tag{3}$$

where $a, b \in \mathbb{R}$. Associated to the equation (3) we define the characteristic equation,

$$r^2 + ar + b = 0. \tag{4}$$

From the fundamental theorem of Algebra we know that the roots of (4) have to fall into one of the following cases: two real distinct roots; one real root with multiplicity two; or two complex conjugate roots. Let r_1, r_2 be the roots of (4) and $C_1, C_2 \in \mathbb{R}$, according to each case, the general solution of the equation (3) is written in one of the following forms.

Case 1. Distinct real roots, $r_1 \neq r_2$; then

$$y(x) = C_1 e^{r_1 x} + C_2 e^{r_2 x}.$$

Case 2. Repeated real root, $r_1 = r_2$; then

$$y(x) = e^{r_1} \left(C_1 + C_2 x\right).$$

Case 3. Complex conjugate roots, $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $\alpha, \beta \in \mathbb{R}$, $\beta \neq 0$; then

$$y(x) = e^{\alpha x} \left(C_1 \cos(\beta x) + C_2 \sin(\beta x)\right).$$

Next, we consider the nonhomogeneous equation with constant coefficients, i.e.,

$$y'' + ay' + by = f, \tag{5}$$

where $f(x) \neq 0$ and $a, b \in \mathbb{R}$. By Theorem 2.2 we know that the general solution of this equation is given by $y = y_h + y_p$, where is y_h is the general solution of the associated homogeneous equation and y_p is a particular solution of the given nonhomogeneous equation. The process to obtain y_h was explained for equation (3); therefore we only need to know how to obtain one particular solution for (5). There are two methods to perform this task, the undetermined coefficients method and the variation of parameters method.

The undetermined coefficients method: This method can only be applied to a special form that the function on the rhs of the equation (5) can assume; the general form of f allowing the use of this method is

$$f(x) = e^{\alpha x} \left(P_m(x) \cos(\beta x) + Q_n(x) \sin(\beta x)\right),$$

where P_m and Q_n are polynomials of degree m and n, respectively.
In this case a particular solution $y_p(x)$ of the equation (5) is sought in the form

$$y_p(x) = x^s e^{\alpha x} \left(A_k(x) \cos(\beta x) + B_k(x) \sin(\beta x) \right),$$

where $k = \max\{m, n\}$, A_k and B_k are polynomials of degree k of the general form with undetermined coefficients, and s is the multiplicity of the root $r = \alpha \pm i\beta$ of the characteristic equation of the corresponding homogenous differential equation to (5) (if $\alpha \pm i\beta$ is not a root of the characteristic equation, then $s = 0$). Let us briefly explain the use of this method. Depending on the function f we seek a particular solution $y_p(x)$ of the equation (5) in a similar form with undetermined coefficients. Then we substitute the expression for $y_p(x)$ in the given equation. Equating coefficients of the similar terms of the first and the second member of (5), we obtain a linear system of equations, which is solved to define the undetermined coefficients. Solving this system we get the particular solution sought $y_p(x)$.

The variation of parameters method: For sake of self-contained present work, we briefly recall this "general method" (see our previous work [3]) that we have to use in the general case. Consider the nonhomogeneous linear differential equation

$$y'' + ay' + by = f,$$

where a, b and f are continuous functions on some interval I. Let y_1 and y_2 be two linearly independent solutions of the corresponding homogeneous equation

$$y'' + ay' + by = 0,$$

on the interval I. Then a particular solution of the nonhomogeneous equation is given by

$$y_p = uy_1 + vy_2,$$

provided that u and v are functions that satisfy the following conditions

$$\begin{cases} u'y_1 + v'y_2 &= 0 \\ u'y_1' + v'y_2' &= f \end{cases}.$$

The superposition principe: It is convenient to use the following result when the rhs $f(x)$ of the nonhomogeneous equation is given by a sum of several functions. If $y_{pk}(x)$ is a solution of the equation

$$y'' + ay' + by = f_k(x), \quad k = 1, ..., n,$$

then the function

$$y_p(x) = \sum_{k=1}^{n} y_{pk}(x)$$

is a solution of the equation

$$y'' + ay' + by = \sum_{k=1}^{n} f_k(x).$$

3 SOFTWARE IMPLEMENTATION AND USE

Our goal is to create an interactive software that can be used by our students when learning the main subject of this work, the linear second-order ODEs with constant coefficients. To achieve this goal we use the Wolfram's software, more precisely we implement the code to create the Computable Document Format (CDF) file with the program *Mathematica*®. To accomplish this task we follow the references [11], [10], [12] and [14].
The CDF file type is a type of file that, as the name itself indicates, has the ability of doing calculus within the document. It is worth mentioning that this type of file requires a previous installation of the CDF Player program that can be downloaded from Wolfram's webpage[1]. This intrinsic characteristic of the file gives the programmer the chance to create files for an incredible wide set of applications that can be used to teach or to present any subject that involves any type of calculus. In this section we will describe the outlook of our software and explain the main commands behind its implementation.
In what concerns to the outlook, we can see it as two main areas, one of input and another of output results. In figure 1 the regions that correspond to input conditions are signalized as regions 1 and 2. Region 1 is not exactly an input region but it is directly related with the input region marked as region 2. Figure 2 shows the changes that occur when the user chooses one of the types presented to define the function on the rhs of the ODE. When the function on the rhs has the form of a product between a polynomial and an exponential, the user is compelled to choose the first option. This way it will be necessary to define the coefficients of the polynomial as well as the exponent constant coefficient. Since this was created to be a didactical tool we limited the degree of the polynomial to a maximum of two. So, in this case the user has to give the program the coefficients of the ODE, the coefficients of the polynomial, and the exponent coefficient, that is, the user has to define six input values. The other option to define the function on the rhs includes trigonometric functions. This is actually the general case of the function that we can have on the rhs of the ODE when using the undetermined coefficient method to solve the type of ODEs covered in this work. Attending to figure 2 we notice a substancial increment in the number of input fields in this case. In fact, we have to define the coefficients of another polynomial, that will have, again, a maximum degree of two, and the constant coefficient in the argument of the trigonometric functions, which sums up to ten input values. To deal with the trigonometric functions the user must be a little bit more careful, because sometimes the function on the rhs of the ODE depends only on a cosine function, case that will be accomplished by setting the coefficient β equal to 0, but in other occasions the function defined on the rhs depends only on a sine function. In this situation the manipulation is slightly different, because we can not define a coefficient β that eliminates the first part of the function, the part that results from the product of the polynomial, the exponential and the cosine function. So, in this case, the user must set all the coefficients of the polynomial equal to zero. Other situation that is important to

[1] Wolfram's webpage (at the bottom of the webpage look for Products/CDF Player).

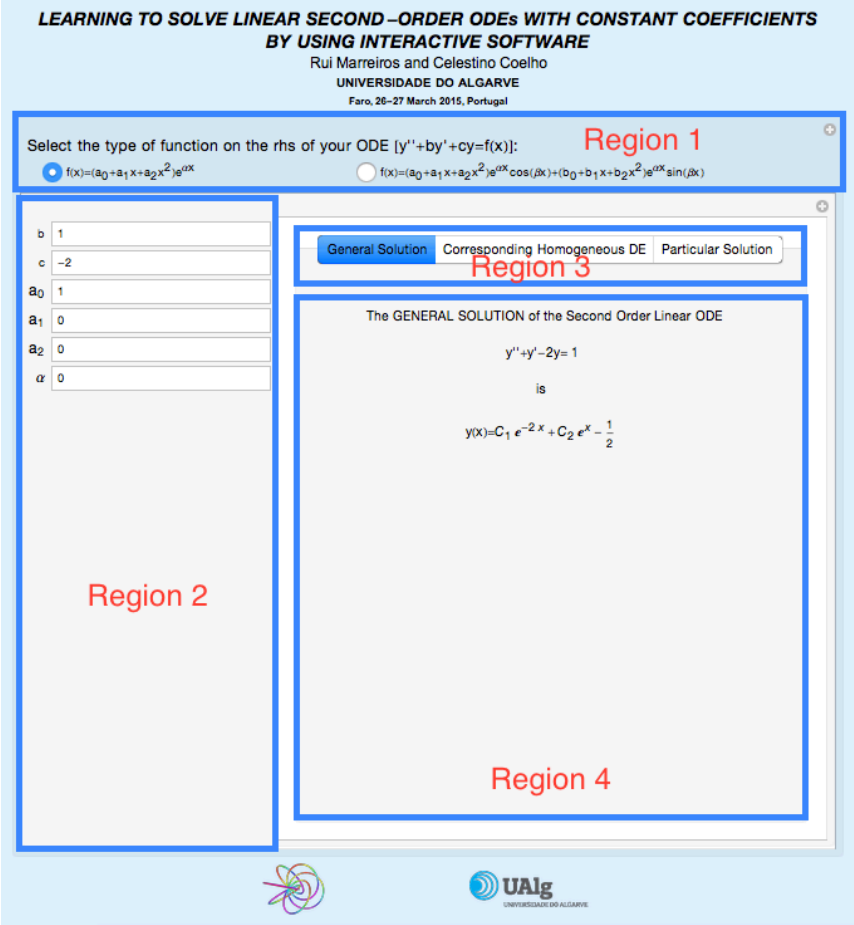
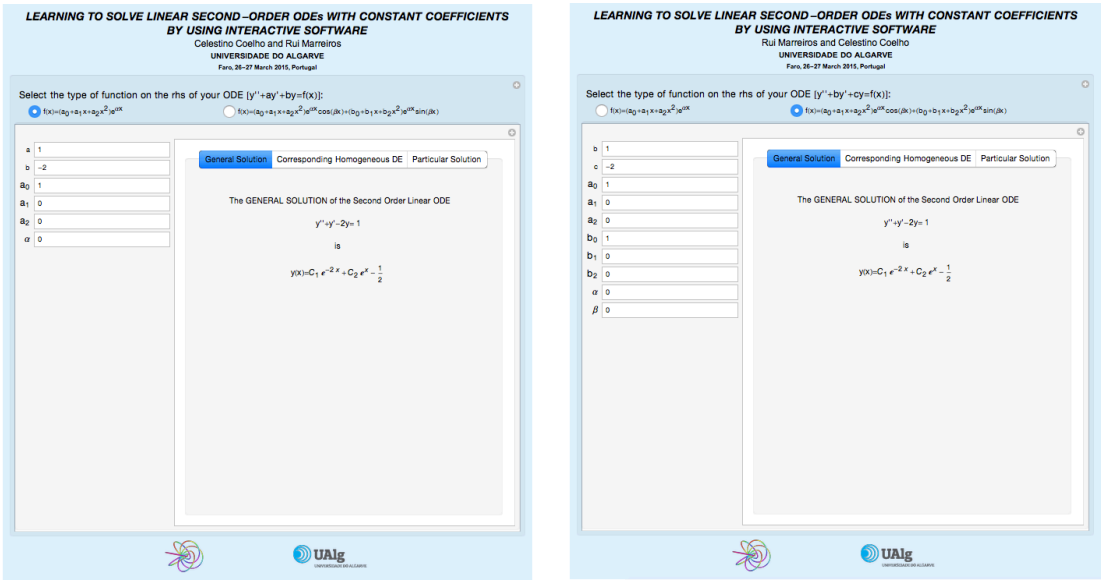


Figure 1: Outlook of the software - regions of input and output.

emphasize is the one that happens when the function on the rhs depends simultaneously on sine and cosine functions. In this case additional caution must be taken, because if the constant coefficients in the arguments of the trigonometric functions are not the same the problem can not be solved by a solely application of the program. To clarify this situation, when the constant coefficients in the arguments of the trigonometric functions are exactly the same the user can define the function on the rhs in one application of the program, in any other occasion it will be necessary to apply the principle of superposition described in the first section. In fact, this principle has to be applied in all the situations that occur when the user is confronted with the impossibility of defining the rhs function in the way that the program exhibits.

Turning to the output area we divide it in two regions, one where the user defines the part of the solution he wants to see or analyze, if the idea is to see the process of obtaining the solution. In figure 1 this is marked as region 3. By clicking on one of three options

available, the user will see the general solution, the process of obtaining the solution to the homogeneous associated ODE, or the process of obtaining the particular solution to the ODE defined in the input area. The results are shown in the region marked as region 4 in figure 1.



(a) Case 1: $f(x) = p_n(x)e^{\alpha x}$ **(b)** Case 2: $f(x) = p_n(x)e^{\alpha x}\cos(\beta x) + q_m(x)e^{\alpha x}\sin(\beta x)$

Figure 2: Outlook of the software - choices available for f in the ODE: $y'' + ay' + by = f$.

In what follows we will describe in a very briefly way the most important commands used to create the software presented here. As the list of commands used to implement the software is vast, it will be impossible point and explain all of them. The central command used to construct this type of files is the `Manipulate` command (see figure 3), used to enable interactive manipulation. Another important command used is the `InputField` command, presented in figure 4, which is used to represent an editable input field that, in our case, contains the values the user wants to attribute to the constant coefficients of the ODE, the polynomial and also to the trigonometric constant arguments.

```
Manipulate[
 TabView[{
   {"op1", "General Solution" → Column[
```

Figure 3: Command `Manipulate`.

Analyzing figure 3 we notice the existence of another command, also with greater importance, used to manipulate the output. The command `TabView` will give the chance to choose, by clicking, the type of output.

```
{{alfa, 0, alfaL}, -10, 10}, Alignment → Center, ControlPlacement → Left, ControlType → InputField
], (*end manipulate*)
```

Figure 4: Command `InputField`.

We also used commands that are intrinsic to *Mathematica*® and enable the user to perform symbolic calculus, such as `DSolve` (see figure 5).

```
"y(x)=" FullSimplify[y[x] /. DSolve[ y''[x] + a y'[x] + b y[x] == 0, y[x], x,
    GeneratedParameters → (Subscript[C, #] &)][[1]]],
```

Figure 5: Command `DSolve`.

4 EXAMPLES

The examples we decide to show here were chosen from the references [2], [13], and [4]. Our choice was performed in a way that we can be able to present the most important situations that the user can bump into when using the software. In what follows, we state the problem we want to solve and give the main direction guidelines to obtain its solution. This way the user can see how the software works.
To start, consider the case where the function on the rhs of the ODE is defined by a polynomial only.

Example 4.1 *Find the general solution of the ODE* $y'' - 4y' + 3y = 1 + x + 3x^2$.

In this first example, the homogeneous associated ODE has the following solution

$$y_h(x) = C_1 e^x + C_2 e^{3x},$$

where $C_k,\ k = 1, 2$ are arbitrary real constants. Since $r = 0$ is not a solution to the characteristic equation, $r^2 - 4r + 3 = 0$, we will be looking for a particular solution in the form of a second degree polynomial, that is,

$$y_p(x) = A_0 + A_1 x + A_2 x^2.$$

Obtaining y_p' and y_p'' and inserting their expressions in the original equation we get a linear system of three equations and three unknowns, which is easily solved. The result of this operation will lead us to the conclusion that

$$y_p(x) = \frac{11}{3} + 3x + x^2.$$

Therefore, using the theory explained in the first section of this work, we get the general solution of the ODE, which is given by

$$y(x) = y_h(x) + y_p(x) = C_1 e^x + C_2 e^{3x} + \frac{11}{3} + 3x + x^2,\ C_1, C_2 \in \mathbb{R}.$$

Figure 6 shows the general solution of the equation defined in this example, as well as the process of obtaining it.

(a) General solution **(b)** Associated homogeneous ODE **(c)** Particular solution

Figure 6: Application of the software to solve example 4.1.

Example 4.2 *Find the general solution of the ODE* $y'' + 2y' = 4x$.

Here, the homogeneous associated ODE has the following solution

$$y_h(x) = C_1 + C_2 e^{-2x},$$

where $C_k,\ k = 1, 2$ are arbitrary real constants. In this case $r = 0$ is a solution to the characteristic equation, $r^2 + 2r = 0$. Consequently, we need to look for a particular solution in the following form,

$$y_p(x) = x\,(A_0 + A_1 x) = A_0 x + A_1 x^2.$$

The reason to multiply the general first degree polynomial by x is due to the fact that the general solution to the associated homogeneous ODE already has the part that is constant in its solution (part that corresponds to C_1). Now, proceeding in the same way as in example 4.1, we conclude that,

$$y_p(x) = -x + x^2.$$

This implies that,

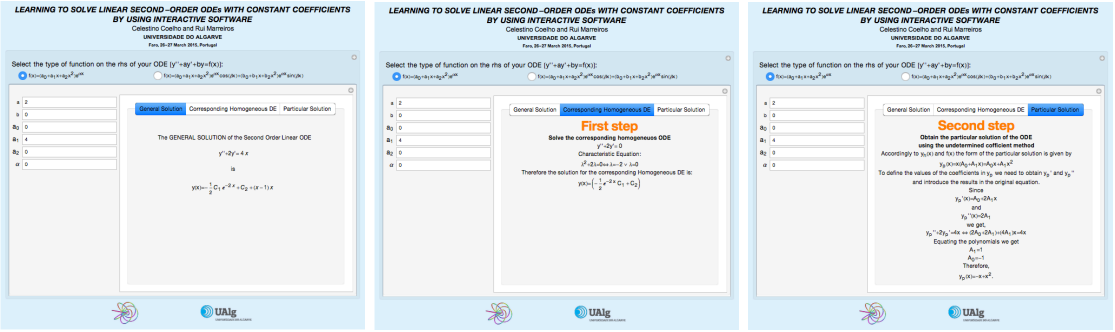
$$y(x) = y_h(x) + y_p(x) = C_1 + C_2 e^{-2x} - x + x^2,\ C_1, C_2 \in \mathbb{R}.$$

The results obtained when solving this example with our software are presented in figure 7.

Example 4.3 *Find the general solution of the ODE* $y'' + 5y' + 6y = (10 - 10x)e^{-2x}$.

The homogeneous associated ODE has the solution

$$y_h(x) = C_1 e^{-3x} + C_2 e^{-2x},$$



(a) General solution **(b)** Associated homogeneous ODE **(c)** Particular solution

Figure 7: Application of the software to solve example 4.2

where C_k, $k = 1, 2$ are arbitrary real constants. Since $r = -2$ is a solution to the characteristic equation, $r^2 + 5r + 6 = 0$, we need to be looking for a particular solution in the form,

$$y_p(x) = x\left(A_0 + A_1 x\right) e^{-2x} = \left(A_0 x + A_1 x^2\right) e^{-2x}.$$

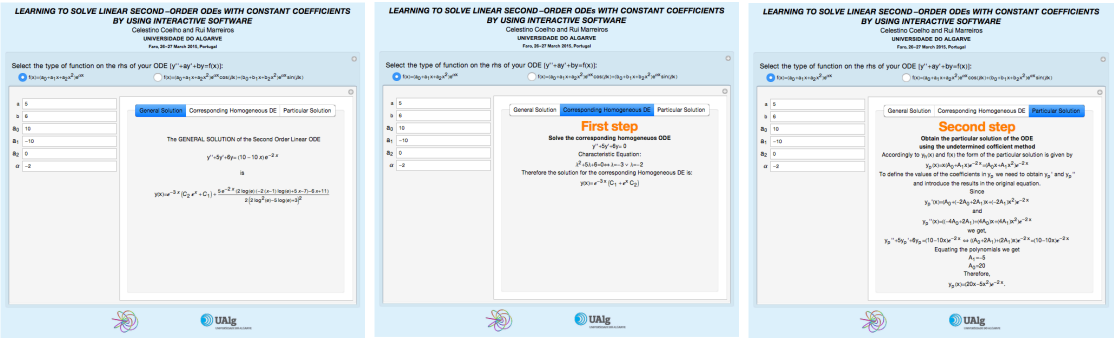
The situation presented in this case and the one described in example 4.2 are analogous. So proceeding in the same way as the one presented in example 4.1 we get

$$y_p(x) = \left(20x - 5x^2\right) e^{-2x}.$$

Therefore, the general solution is given by

$$y(x) = y_h(x) + y_p(x) = C_1 e^{-3x} + C_2 e^{-2x} + \left(20x - 5x^2\right) e^{-2x},\ C_1, C_2 \in \mathbb{R}.$$

Figure 8 shows the results obtained when applying our software.



(a) General solution **(b)** Associated homogeneous ODE **(c)** Particular solution

Figure 8: Application of the software to solve example 4.3.

Example 4.4 *Find the general solution of the ODE* $y'' - 2y' + y = 6xe^x$.

Solving the homogeneous associated ODE we get

$$y_h(x) = C_1 e^x + C_2 x e^x,$$

where $C_k,\ k = 1, 2$ are arbitrary real constants. The characteristic equation, $r^2 + 2r = 0$, has a unique solution with multiplicity two, $r = 1$. This implies that we need to look for a particular solution that has the following form,

$$y_p(x) = x^2 \left(A_0 + A_1 x\right) e^x = \left(A_0 x^2 + A_1 x^3\right) e^x.$$

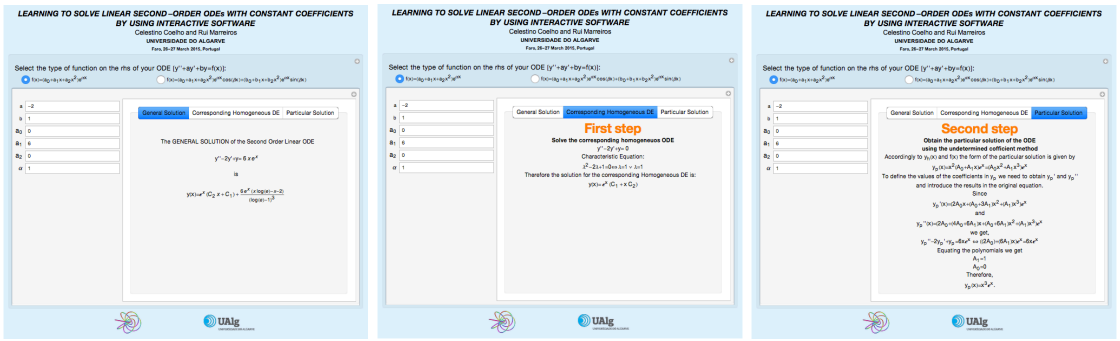
In this situation we need to multiply by x^2 because the associated homogeneous ODE has a solution that already is a linear combination of the functions in the set $\{e^x, xe^x\}$. This means that we need to look for a solution that is a linear combination of $x^2 e^x$ and $x^3 e^x$. The application of the undetermined coefficient method to this case yields,

$$y_p(x) = x^3 e^x.$$

Which leads us to conclude that

$$y(x) = y_h(x) + y_p(x) = C_1 e^x + C_2 x e^x + x^3 e^x,\ C_1, C_2 \in \mathbb{R}.$$

In figure 9 we present the results obtained by our software when it is applied to example 4.4.



(a) General solution **(b)** Associated homogeneous ODE **(c)** Particular solution

Figure 9: Application of the software to solve example 4.4.

Example 4.5 *Find the general solution of the ODE* $y'' - 3y' + 2y = 4\sin(x)$.

To end this section we present an example were the function on the rhs of the ODE is defined in terms of a trigonometric function. In what concerns to the homogeneous associated ODE we follow the same procedure to obtain its solution, which give us

$$y_h(x) = C_1 e^x + C_2 e^{2x},$$

where $C_k,\ k = 1, 2$ are arbitrary real constants. In this situation we will be looking for a particular solution in the form,

$$y_p(x) = A_0 \cos(x) + B_0 \sin(x).$$

Applying the method described in all of the previously presented examples we obtain

$$(A_0 - 3B_0)\cos(x) + (3A_0 + B_0)\sin(x) = 4\sin(x),$$

which leads to the following system of linear equations,

$$\begin{cases} A_0 - 3B_0 &= 0 \\ 3A_0 + B_0 &= 4 \end{cases} \Leftrightarrow \begin{cases} A_0 &= 6/5 \\ B_0 &= 2/5 \end{cases},$$

and, consequently,

$$y_p(x) = \frac{6}{5}\cos(x) + \frac{2}{5}\sin(x).$$

Therefore, the general solution of the ODE is given by

$$y(x) = y_h(x) + y_p(x) = C_1 e^x + C_2 e^{2x} + \frac{6}{5}\cos(x) + \frac{2}{5}\sin(x),\ C_1, C_2 \in \mathbb{R}.$$

In figure 10 we picture the results obtained with our software.

(a) General solution **(b)** Associated homogeneous ODE **(c)** Particular solution

Figure 10: Application of the software to solve example 4.5.

5 CONCLUSION

In [3] we presented software that could be used to solve this type of ODEs, but, due to the way that we choose to create it, the user can only apply it freely when the function on the rhs of the equation is constant. In any other situation the user has to have a license for *Mathematica*® or CDF Player. As a consequence of this fact, we can assure that one of

most important conclusions of this work is that the software presented here can be used without any license or any payment. This feature allows us to share it with our students, so they can use it at home, or anywhere else, when studying linear second-order ODEs with constant coefficients. This may be the most important conclusion of this work, but we may draw other minor considerations about it, such as the importance it represents as a complement in studying this subject and the simplicity of its use.
We should point out that in the last couple of years there has been a little revolution in the publication and distribution of this type of applications, as we can see in Wolfram's webpage, but in our humble opinion the one we present in this is more complete and give a more concise approach about the process of how we should obtain the particular solution. The major part of the applications available only give the solution, giving no clue of how to get it.

REFERENCES

[1] Birkhoff, G., and Rota, G.-C.. *Ordinary Differential Equations*, 4th edition, John Wiley & Sons, 1989.

[2] Braun, M., *Differential Equations and Their Applications, An Introduction to Applied Mathematics*, 3rd edition, Springer-Verlag, New York, 1983.

[3] Coelho, C. and Marreiros, R.. "Solving second-order linear ordinary differential equations interactively", In A. Loja, J.I. Barbosa, and J.A. Rodrigues (Eds), Proceedings of the 1 st ECCOMAS Conference on Algebraic and Symbolic Computation - SYMCOMP 2013, Lisbon, Portugal, pp. 243-258, September 9-10, 2013.

[4] Demidovich, B.. *Problems in Mathematical Analysis*, 2nd edition, Mir Publishers, Moscow, 1978.

[5] Edwards, H., and Penney, D.E.. *Elementary Differential Equations*, 6th edition, Pearson Education, Inc., 2008.

[6] Gill, Dennis Z.. *A First Course in Differential Equations with Modeling Applications*, 9th edition. Brooks/Cole, 2009.

[7] Krasnov, M.L., Kiselev, A.I., and Makarenko, G.I.. *A Book of Problems in Ordinary Differential Equations*, Mir, 1981.

[8] Marreiros, Rui. *Apontamentos de equações diferenciais ordinárias*, Universidade do Algarve, 2012.

[9] Ross, S.. *Differential Equations*, John Wiley & Sons, 1984.

[10] Ruskeepää, H.. *Mathematica® Navigator. Mathematics, Statistics, and Graphics*, 3rd edition, Academic Press, 2009.

[11] Torrence, B.F. and Torrence, E.A., *The Student's Introduction to Mathematica®, A Handbook for Precalculus, Calculus, and Linear Algebra*, 2nd edition, Cambridge University Press, Cambridge, 2009.

[12] Trott, M., *The Mathematica Guidebook for Symbolics*, Springer Science+Business Media, Inc., New York, 2006.

[13] Vrabie, I.I., *Differential Equations, An Introduction to Basic Concepts, Results and Applications*, World Scientific Publishing Co. Pete. Ltd., Singapore, 2004.

[14] Wolfram, S.. *The Mathematica® Book*, 5th edition, Wolfram Media, 2003.

SYMCOMP 2015
Faro, March 26-27, 2015

INTERACTIVE LEARNING OF MODELING WITH ORDINARY DIFFERENTIAL EQUATIONS

Celestino Coelho[1], Rui Marreiros[2], and Ana C. Conceição[2]

1: Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade do Algarve
Campus de Gambelas 8005-139 Faro, Portugal
e-mail: ccoelho@ualg.pt

2: Center for Functional Analysis, Linear Structures and Applications (CEAFEL)
Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade do Algarve
Campus de Gambelas 8005-139 Faro, Portugal
e-mail: {rmarrei, aconcei}@ualg.pt

Keywords: Ordinary Differential Equations, Modeling, Applications, Computable Document Format.

Abstract. *Differential equations constitute a large and very important branch of mathematics, in particular those related to science and engineering applications. To help students to assimilate the knowledge related to this topic we present a new educational software that allows interaction with the user and can be freely used, both in the classroom, when teaching ordinary differential equations, and as a self-learning tool.*

1 INTRODUCTION

An ordinary differential equation (ODE) of order n assumes the following form

$$F\left(x, y(x), y'(x), ..., y^{(n)}(x)\right) = 0, \tag{1}$$

where F is a given function, x is the independent variable and $y = y(x)$ is the sought solution of the ODE.
The general solution of the ODE (1) is the set of all of its solutions, which is defined by

$$y = \varphi\left(x, C_1, C_2, ..., C_n\right)$$

containing n arbitrary constants $C_1, C_2, ..., C_n$. Any solution obtained from the general solution for particular values of the arbitrary constants $C_1, C_2, ..., C_n$ is called a particular solution of the ODE (1).
The problem of finding a particular solution $y = y\left(x\right)$ of the ODE (1) satisfying the n initial conditions

$$y\left(x_0\right) = y_0, y'\left(x_0\right) = y_0', ..., y^{(n-1)}\left(x_0\right) = y_0^{(n-1)},$$

is called the Cauchy problem, or the initial value problem, for the ODE (1). For example, for the first order equation

$$F\left(x, y(x), y'(x)\right) = 0$$

the initial condition is of the form $y\left(x_0\right) = y_0$, and for the second order equation

$$F\left(x, y(x), y'(x), y''(x)\right) = 0$$

the initial conditions are of the form $y\left(x_0\right) = y_0$, $y'\left(x_0\right) = y_0'$; where x_0, y_0, y_0' are given real numbers.
The process of solving a physical phenomenon problem starts with the construction of the mathematical model that translates the problem to be solved. This is usually made in two steps: first we establish the physical quantities representing the unknown function, whose knowledge should give us an exact and complete idea of the phenomenon evolution; secondly, we find the physical laws that govern the phenomenon being considered.

2 SOFTWARE

The main goal of this work is to present new software that can be used to teach the subject of ODEs, more precisely some of their applications in modeling physical phenomenon. This software is designed to have distribution free, so it can be used without buying any license during the classes or by our students anytime they need to study the topics presented here. In this section we will describe some of the commands and ideas behind its implementation and explain the main features of the software.

2.1 Implementation

The software we present in this work was constructed using the computer algebra system program *Mathematica*®, which is a Wolfram's registered brand. Using this program we build a file that is able to do calculus within the document itself. This type of files is called Computable Document Format (CDF) file and can only be programmed using *Mathematica*®. The guidelines to create this type of documents can be found, for example, in the references [10], [9], [11] and [12].
To run a CDF file the user must have a previous installation of CDF Player program, which can be downloaded from Wolfram's webpage[1]. The CDF Player program does not obligates the user to buy any kind of license from Wolfram, at least for the software we present in this work. However, the user should be aware that there are cases when it will be required a license.
The list of the commands use to build the software is vast, so we will turn only to the most important ones. These are the ones that are intrinsic to the CDF files and those predefined in *Mathematica*® available to perform symbolic calculus.
To allow interactivity between the file and the user we use the command `Manipulate` (see figure 1). All the features we want to allow the user to interact must be defined within this command.

```
Manipulate[
 TabView[{
   {"op1", "Graphical Solution" → Grid[{{{
```

Figure 1: Command `Manipulate`.

Other important command that was used is the `InputField` command. In our case all the data inputed by the user in these fields must be constant, otherwise it will generate an error. Unless the user has a valid license for CDF Player or *Mathematica*® (see figure 2).

```
{{q0, 0.05, Qinit}, 1, 10},
{{i0, 0, Iinit}, -1, 1},
Alignment → Center, ControlPlacement → Top, ControlType → InputField,
```

Figure 2: Command `InputField`.

To create the tabs corresponding to the graphical solution and the analytical solution, we use the `TabView` command (see figure 3). This command is defined to create as many tabs as we want to separate information. To switch from one tab to another the user just have to click on the title to see the pretended output.

[1] Wolfram's webpage

```
Manipulate[
 TabView[{
   {"op1", "Graphical Solution" → Grid[{{
```

Figure 3: Command `TabView`.

In what concerns to the commands that allow us to perform symbolic calculus within the document, we suppose the command `DSolve` is the most important. Using this command we can directly solve the initial value problem defined by the user, and use the output result to plot its solution.

2.2 How to use it

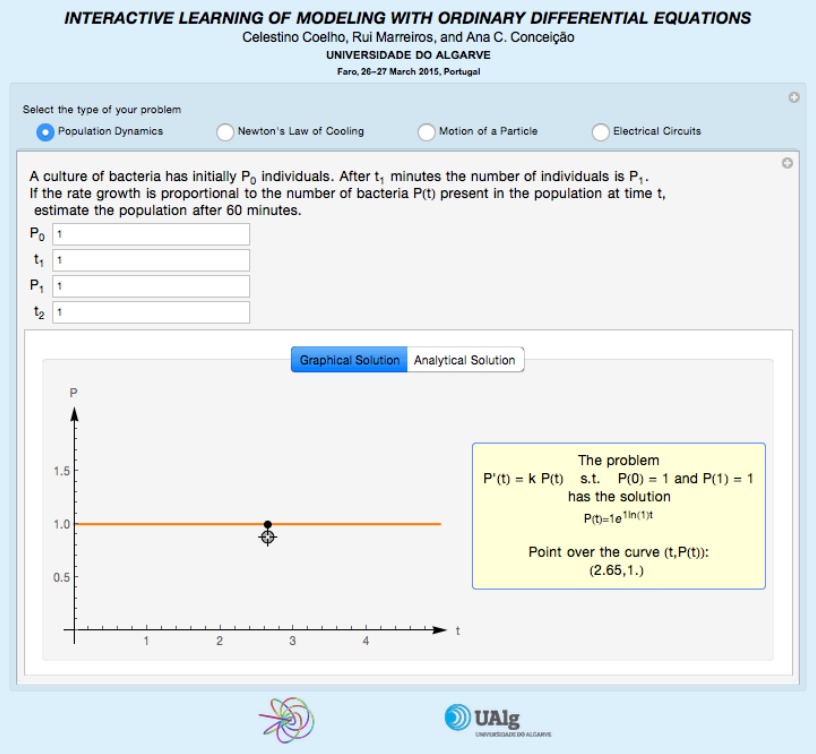


Figure 4: Initial interface.

Regarding to the outlook of the software, we can divide it into two main regions, one of input and another that corresponds to the output results.
Figure 4 shows that the input region is composed by two types of input selection. On top we see four radio buttons that correspond to the selection of the problem we want to solve. The set of problems available to solve correspond to the examples that will be

presented and explained in the next section. After setting the physical phenomena we want to model we will see the statement of a standard problem and the associated input fields. Note that the set of input fields is directly related to the problem to be solved, that is, it will change from problem to problem. To jump from one input field to another the user just have to press `TAB` key on the keyboard, or use the mouse.
The output region is located below the statement of the problem and the set of input fields. Here we see an option on top that gives the user the possibility to see the graphical output with the mathematical model of the problem and the corresponding solution, or the analytical solution, that is the mathematical methodology to solve the problem selected. The graphical window is programmed to give the user the possibility to move the cursor over the plot. This enables the user to get additional information about the problem solved without having to define new parameters in the input fields. Nevertheless, the use of this trick is, in many situations, difficult to use, because sometimes it will be extremely difficult to stabilize the pointer of the mouse over the point we want to colect information. These cases require an extra care when answering the questions, because, due to the nature of the question, in some cases the answer has to be overestimated and in other underestimated.

3 APPLICATIONS

In this section we present the examples we decide to tackle with our software. Due to the countless number of examples that we can found in almost any branch of science, we decide that would be a good ideia to focus on examples that are more directed to the areas of engineering, making a choice of examples where the mathematical model uses different types of ODEs. With this in mind, the four models presented here are based on four different types of ODEs, more precisely, two of them are created with first-order ODEs, one is separable variables and the other is linear, as the other two are based on second-order ODEs, where one is solved by direct integration on both sides of the ODE and the other has to be solved using the method of undetermined coefficients. The examples presented here were collected from the references [1], [2], and [6].

3.1 Population dynamics

One of the earliest models presented to model the dynamic change of populations was introduced by Thomas Malthus, what explains the name of the model, malthusian model. To derive the mathematical model we consider the quantity $P(t)$ to be the number of the individuals present in the population at time t evolving in an environment in which resources are unlimited. In such case, a simple mathematical model of growth/decay of population assumes that the rate of the population is proportional to the population itself. This proportionality can simply be expressed by the differential equation

$$\frac{\mathrm{d}P}{\mathrm{d}t} = kP, \tag{2}$$

where the proportionality constant k is the difference between the birth rate and the death rate of the population being studied. Dimensional analysis reveals that this constant must be expressed in the physical dimension time^{-1}.
The ODE that defines the model (2) is a linear and separable equation. Therefore, to obtain the solution we can simply consider

$$\frac{1}{P}\mathrm{d}P = k\mathrm{d}t \tag{3}$$

Integrating both sides of (3) we get

$$\ln|P| = kt + C \tag{4}$$

which is equivalent to write

$$P(t) = C_1 e^{kt}, \tag{5}$$

for an arbitrary constant C_1. The equation (5) gives us the general solution of the ODE (2). To define a particular solution we need to specify an initial condition $P(t_0) = P_0$. This way we get

$$P(t_0) = P_0 \Leftrightarrow C_1 e^{kt_0} = P_0 \Leftrightarrow C_1 = P_0 e^{-kt_0}. \tag{6}$$

The conjugation of (6) with (5) gives us the solution to the Cauchy problem associated with the malthusian model

$$P(t) = P_0 e^{k(t-t_0)}. \tag{7}$$

The analysis of the equation (7) shows us three limit situations that deserve to be studied. The limit $t \to \infty$ depends directly on the constant of proportionality k: when k is positive, the value of P tends to infinity, that means that the population grows exponentially because there are more births than deaths; when k is negative, the value of P tends to zero, meaning that the population will die out because there are more deaths than births; when k is equal to zero, the value of P will remain unaltered with time, which represents a static population where births and death rates are exactly the same.
In many real problems we do not know the value of the constant of proportionality, but we are able to count the number of the individuals at some initial time t_0, $P(t_0) = P_0 > 0$, and at some other ulterior time t_1, $P(t_1) = P_1 > 0$. Using this information we can derive the value of the constant of proportionality and, therefore, conclude if the population will be static, grow exponentially or eventually extinct. This is accomplish by using

$$P(t_0) = P_0 \Leftrightarrow C_1 e^{kt_0} = P_0$$

and

$$P(t_1) = P_1 \Leftrightarrow C_1 e^{kt_1} = P_1.$$

Relating the last two equations we get,

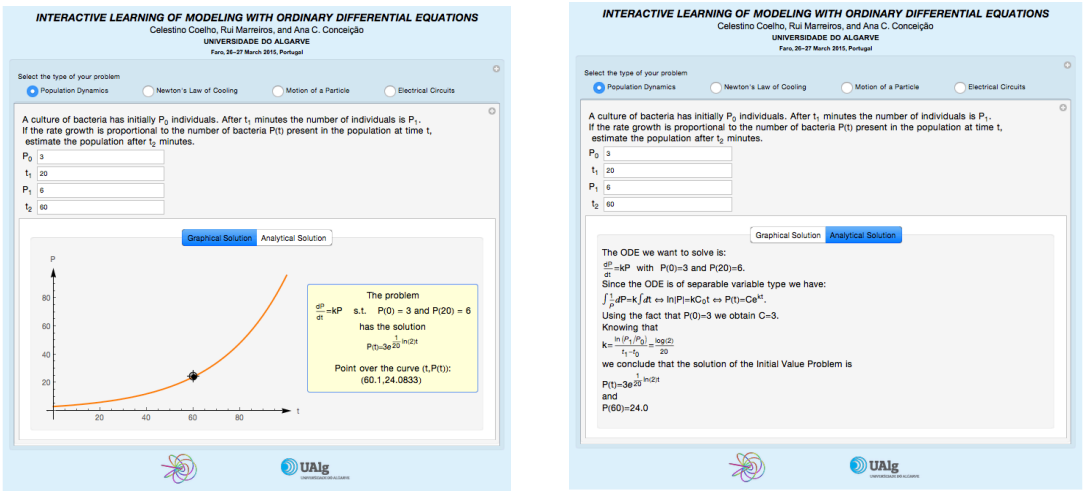
$$\frac{C_1 e^{kt_1}}{C_1 e^{kt_0}} = \frac{P_1}{P_0} \Leftrightarrow e^{k(t_1-t_0)} = \frac{P_1}{P_0} \Leftrightarrow k(t_1 - t_0) = \ln\left(\frac{P_1}{P_0}\right) \Leftrightarrow k = \frac{\ln(P_1/P_0)}{t_1 - t_0}.$$

This will be the case of the example we decided to present in this work.

Example 3.1 (*e-coli* population growth) *A culture of bacteria (e-coli) has initially $P_0 = 3$ individuals. After 20 minutes the population is twice the original value. Assuming a growth rate proportional to the number of bacteria $P(t)$ present in the population at time t:*

a) find the number of bacteria $P(t)$ present in the population after 60 minutes;

b) what will be the time necessary to guarantee a population 5 times bigger than the initial population?



(a) Graphical output **(b)** Analytical output

Figure 5: Tool application to solve example 1.

3.2 Newton's law of cooling

The Newton's law of cooling states that the rate of heat loss from the surface of an object is proportional to the difference in temperature between the object's surface and its environment. If the object is a very good conductor of heat, what is often true in real applications, then the internal temperature of the body is effectively the one that is verified at the surface. Let $T(t)$ and $S(t)$ be the temperatures of the object and its environment at time t, respectively. These are the quantities that are to be considered in the modeling of the phenomenon. The physical law to be applied is the Newton's law of cooling which states that

$$\frac{\mathrm{d}T}{\mathrm{d}t} = -k\,(T - S) \tag{8}$$

where the proportionality in the statement of the law has been changed to an equality by the introduction of a constant k. Performing a simple dimensional analysis we verify

that the physical dimension of the constant k is time^{-1}. In this ODE t is the independent variable and T represents the unknown variable of the problem. So, the equation (8) can be written in the following form

$$\frac{\mathrm{d}T}{\mathrm{d}t} + kT = kS \tag{9}$$

which is a linear ODE. To obtain the general solution we can use the integrating factor e^{kt}. Multiplying both sides of the equation (9) by e^{kt} we get

$$\frac{\mathrm{d}T}{\mathrm{d}t}e^{kt} + ke^{kt}T = ke^{kt}S \Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}t}\left(Te^{kt}\right) = ke^{kt}S. \tag{10}$$

The integration of both sides of the equation (10) yields,

$$T(t) = ke^{-kt}\int e^{kt}S(t)\,\mathrm{d}t. \tag{11}$$

In order to obtain a particular solution of (11), consider the simple situation in which the temperature of the environment remains constant, that is, $S(t) = S_0$. In this case, the equation (11) integrates to

$$T(t) = S_0\left(1 + C_1e^{-kt}\right) \tag{12}$$

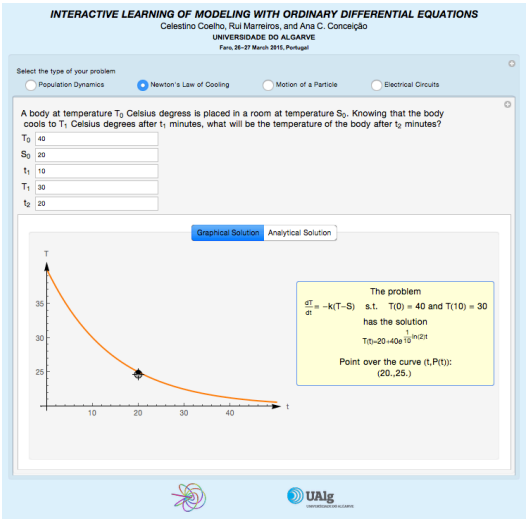
for some integration constant C_1. To define C_1 and completely define the particular solution we need to stipulate an initial condition. Under the condition $T(t_0) = T_0$ we get

$$T(t) = S_0 + (T_0 - S_0)\,e^{-k(t-t_0)}. \tag{13}$$

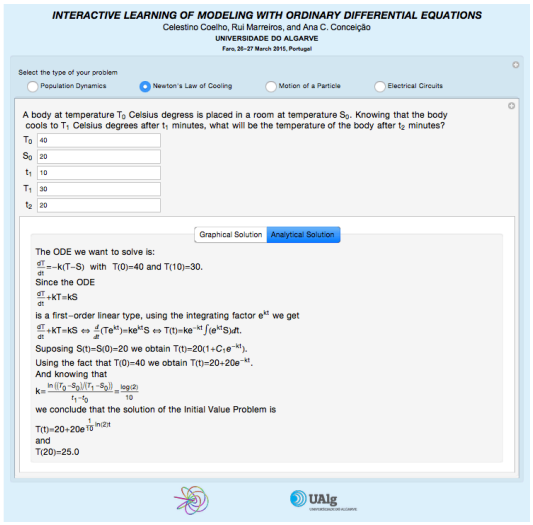
Knowing the values of S_0, T_0 and k we will be able to calculate the temperature of the object for all times t. However, it is often common to use the particular solution (13) to obtain a parameter of the model by providing additional information, as it was the case explained in the first example presented. For example, if k, which is a material parameter specific to a certain object, is an unknown value, assuming the T_0 and S_0 are known, we can derive the value of this parameter by adding a further measurement of the temperature at some time $t_1 > t_0$. Denoting this measurement by T_1 and using (13) we get

$$k = \frac{1}{t_1 - t_0}\ln\left(\frac{T_0 - S_0}{T_1 - S_0}\right). \tag{14}$$

Example 3.2 *A body at temperature* 40 *Celsius degrees is placed in a room at temperature* 20 *Celsius degrees. Knowing that the body cools to* 30 *Celsius degrees after* 10 *minutes, what will be the temperature of the body after* 20 *minutes?*



(a) Graphical output



(b) Analytical output

Figure 6: Tool application to solve example 2.

3.3 Motion of a particle

To exemplify this case let us study the free fall of a body. In this case the free fall is modeled as a material point that moves vertically towards the earth surface. Since the problem of knowing the motion is equivalent to the one of knowing the distance of the particle from the impact point at every instant of time t, we consider the unknown function as a function that measures the displacement along the vertical. Therefore y is a real function that depends on only one independent variable, the time, denoted by t.
In what concerns to the physical laws that are used to model the phenomenon we have to use the law of mechanics that governs the free fall of a body, which is expressed by the Newton's second law of motion,

$$ma = F, \tag{15}$$

where a represents the acceleration and F is the resultant of the forces acting upon the body. Due to the nature of the phenomenon being modeled there is only one force to be considered, the force of gravity, denoted by G, defined by

$$G = -mg, \tag{16}$$

where $g = 9.81\,\mathrm{m}\cdot\mathrm{s}^{-2}$ is the gravity acceleration. The minus signal present in the equation (16) means that this force acts downwards, unlike y, which is upwards directed.
The velocity of the body will be expressed as the derivative of the displacement with respect to time,

$$\dot{y} \equiv \frac{\mathrm{d}y}{\mathrm{d}t},$$

and, since the acceleration is the derivative of the velocity with respect to t, this quantity will be defined as the second order derivative of the displacement with respect to time,

$$\ddot{y} \equiv \frac{\mathrm{d}^2 y}{\mathrm{d}t^2}. \tag{17}$$

The notation used in the last two equations is the standard notation used in mechanics to express the derivatives with respect to time.
Using (17) and (16) in (15) we get

$$\ddot{y} = -g, \tag{18}$$

which represents the mathematical model for the free fall body. This is a second-order ODE where g is a constant, which means that we can simply integrate the equation on both sides,

$$\dot{y} = -gt + C_1,$$

where C_1 is an arbitrary constant to be defined under the initial conditions of the problem. Analyzing the result we see that, at this stage, we have a first-order ODE. To obtain the general solution of the original ODE we need to perform another integration on both sides of the equation, which is possible because the right-hand side (rhs) of the equation depends only on time,

$$y(t) = -g\frac{t^2}{2} + C_1 t + C_2, \tag{19}$$

where C_2 is another arbitrary constant.
According to the theory presented in the introduction, the equation (19) represents the general solution of the ODE presented in (18). Due to the order of the ODE we notice the existence of two arbitrary constants in (19). As a consequence of this we need to have supplementary conditions that enable the specification of the value of these constants. In this case, the nature of the problem suggests that the conditions must be related to the position of the body at the beginning of the motion (the initial position) and its velocity at the same moment (initial velocity). If the motion starts at the moment $t = 0$, then these conditions must be stated in the following form

$$\begin{cases} y(0) &= y_0, \\ \dot{y}(0) &= v_0, \end{cases} \tag{20}$$

where y_0 and v_0 are two known values of the problem. The conditions presented in (20) are called Cauchy conditions. These are the conditions that are used to define the constants C_1 and C_2 obtained in the general solution. To accomplish this task we conjugate the conditions with the general solution, which immediately yields

$$\begin{cases} y(0) &= y_0, \\ \dot{y}(0) &= v_0, \end{cases} \Leftrightarrow \begin{cases} C_2 &= y_0, \\ C_1 &= v_0. \end{cases} \tag{21}$$

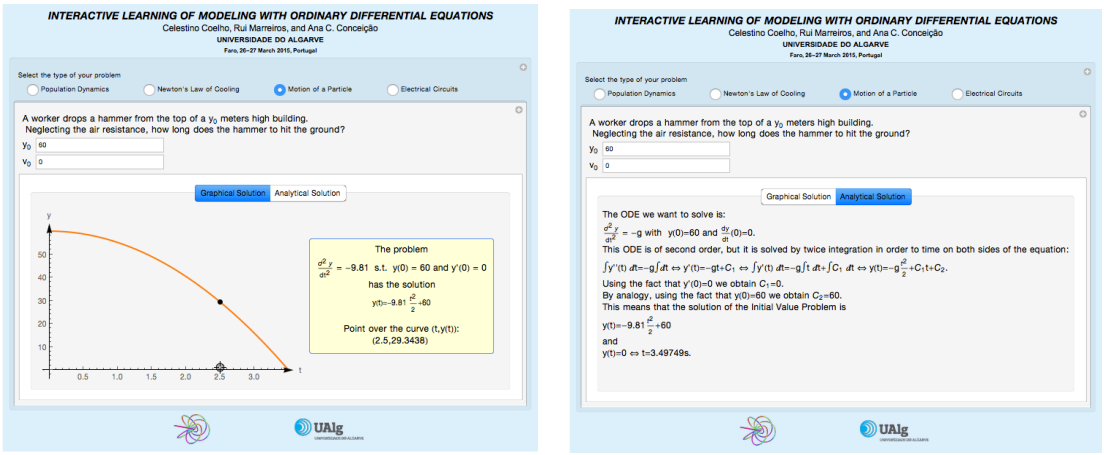
This means that the Cauchy problem

$$\begin{cases} \ddot{y} &= -g \\ y(0) &= y_0, \\ \dot{y}(0) &= v_0 \end{cases} \tag{22}$$

has the unique solution

$$y(t) = -g\frac{t^2}{2} + v_0 t + y_0.$$

Example 3.3 *A worker drops a hammer from the top of a* 60 m *high building. Neglecting the air resistance, how long does the hammer takes to hit the ground?*



(a) Graphical output **(b)** Analytical output

Figure 7: Tool application to solve example 3.

3.4 Electrical circuits

The implementation of a simple electrical circuit is nothing more than an arrangement of various electrical elements such as, for example, a source of electromotive force, often a battery, resistors, inductors and capacitors. The basic quantity of the mathematical model will be the electrical charge, usually denoted by Q. The electrical current flowing in a circuit along the time is just the rate of the electrical charge that flows with respect to time, that means the rate of flow of charge. Defining by I the electrical current we will get

$$I = \frac{\mathrm{d}Q}{\mathrm{d}t}. \tag{23}$$

Other quantity that is used in the mathematical model is the electrical voltage, U, that represents the potential drops/differences in the circuit elements.
The behaviour of the electrical circuits is based on Kirchhoff's Laws. There are two laws, the first of which simply states that the current is conserved at any junction, that is, any point at which wires connect. The method of loop currents introduces dependent variables in such a way that Kirchhoff's First Law is automatically satisfied. It is therefore the second of Kirchhoff's Laws that provides the differential equations. To apply this law we assume that the applied electromotive force in any closed circuit balances all the voltage drops in the circuit being considered, or, in other words, the sum of all voltage drops around a closed circuit is zero. To exemplify consider the single-loop LRC-series circuit presented in figure 8, which contains an inductor, a resistor and a capacitor, and corresponds to example 3.4. The letters L, R, and C are known as inductance, resistance, and capacitance, respectively, and, in general, are constants, which will the case covered here. Using the equation (23) and adding the voltage associated to the inductor,

$$L\frac{\mathrm{d}I}{\mathrm{d}t} = L\frac{\mathrm{d}^2Q}{\mathrm{d}t^2},$$

with the one associated to the resistor,

$$IR = R\frac{\mathrm{d}Q}{\mathrm{d}t}$$

and the one associated to the capacitor,

$$\frac{1}{C}Q,$$

and equating the sum to the impressed voltage, we get a second-order differential equation

$$L\frac{\mathrm{d}^2Q}{\mathrm{d}t^2} + R\frac{\mathrm{d}Q}{\mathrm{d}t} + \frac{1}{C}Q = E(t),$$

where $E(t)$ represents the impressed voltage at time t. For additional insight about the subject covered by this example see, for instance, [7].

Example 3.4 *An electric circuit consists of an inductance of* 0.05 *henrys, a resistance of* 20 *ohms, a condenser of capacitance* $100 \cdot 10^{-6}$ *farads, and EMF of* $E(t) = 100$ *volts. Find the charge* Q *and the current* I *at time* t*, given the initial conditions* $Q = 0$ *coulombs,* $I = 0$ *amperes when* $t = 0$*.*

In this example we are considering $E(t) = C,\ C \in \mathbb{R}$, but, in general, E is defined by a sine or a cosine, so, using the undetermined coefficient method, we will be looking for a particular solution with the following general form:

$$Q_p(t) = A_0 \cos(\omega t) + B_0 \sin(\omega t).$$

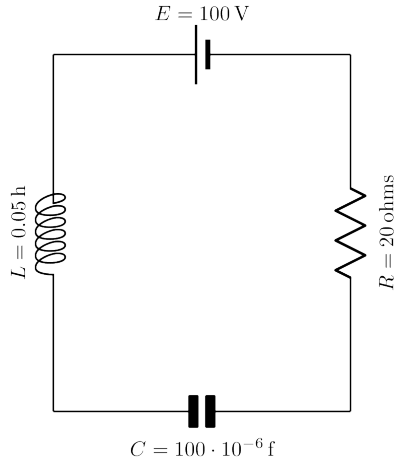


Figure 8: Electric circuit considered in example 3.4.

Solving for A_0 and B_0 we reach to the conclusion that

$$A_0 = \frac{L^2 A\left(\omega^2 - \omega_0\right)}{R^2 + L^2\omega^3 + L^2\omega_0\left(\omega_0 - \omega\left(1+\omega\right)\right)}$$

and

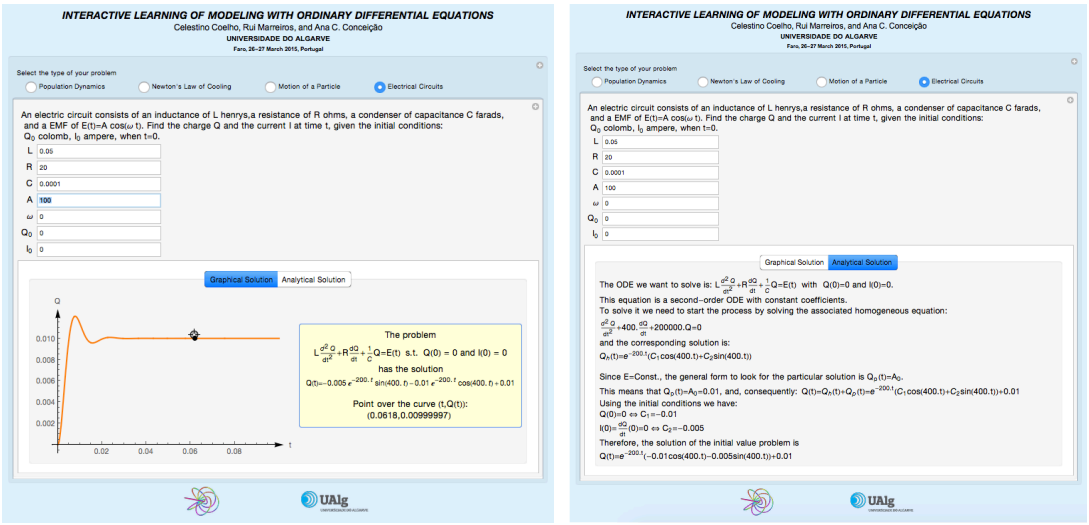
$$B_0 = \frac{LRA}{R^2 + L^2\omega^3 + L^2\omega_0\left(\omega_0 - \omega\left(1+\omega\right)\right)},$$

where, $1/(LC)$ was replaced by ω_0^2 and considering the rhs of the equation in its general form for the problem of electrical circuits, $f(x) = A\cos(\omega t)$. Other way to solve these cases is by using the variation of parameter method presented in [3].

4 CONCLUSIONS

Over the last couple of years the software available to help students in their understanding of the subjects taught in higher education degrees has growth exponentially, specially in the subjects related to mathematics, see, for example, [4], [5], and [8]. In our opinion this kind of tools provide a new kind of help to students when learning these subjects. In many situations, we see tools that output only the solution of the problem, other give the analytical solution but not any insight about the graphical solution, and, in most part of the cases, they are not free. The license free use is, perhaps, the most important feature of the software presented in this work. The way used to build the associated CDF file guarantees this characteristic of the program. It means that after the installation of the program CDF Player the user can test as many examples as he wants, without the need of any payment.

Other important conclusion about this work is the simplicity, in what concerns to the use, of the tool. Any user can read the statement of the problem and easily input the data



(a) Graphical output **(b)** Analytical output

Figure 9: Tool application to solve example 4.

related to the problem to be solved, getting, instantaneously, a graphical insight of the solution, and, if needed, the analytical steps performed to derive it.

REFERENCES

[1] Ayres Jr., Frank, *Schaum's Outline of Theory and Problems of Differential Equations*, McGraw-Hill Book Company, 1952.

[2] Braun, M., *Differential Equations and Their Applications, An Introduction to Applied Mathematics*, 3rd edition, Springer-Verlag, New York, 1983.

[3] Coelho, C. and Marreiros, R.. "Solving second-order linear ordinary differential equations interactively", *SYMCOMP2013 Proceedings of the 1* st *International Conference on Algebraic and Symbolic Computation (ECCOMAS Thematic Conference)*, Ed. A. Loja, J.I. Barbosa, and J.A. Rodrigues, Lisboa, Portugal, pp. 243-258, September 9-10, 2013.

[4] Conceição, A.C., Pereira, J.C., Silva, C.M., Simão, C.R. "Software Educacional em Pr- Clculo e Clculo Diferencial: o Conceito F-Tool", *Bol. Soc. Port. Mat.*, Spec. Iss., pp. 57-60, 2013.

[5] Conceição, A.C., Pereira, J.C., Silva, C.M., Simão, C.R. "Mathematica in the Class-Room: New Tools for Exploring Precalculus and Differential Calculus", *CSEI2012 Proceedings of the 1st National Conference on Symbolic Computation in Education and Research*, Lisboa, Portugal, P03, 2012.

[6] Demidovich, B.. *Problems in Mathematical Analysis*, 2nd edition, Mir Publishers, Moscow, 1978.

[7] Humbly, A.R., *Electrical Engineering, Principles and Applications*, 5th edition, Prentice Hall, New Jersey, 2011.

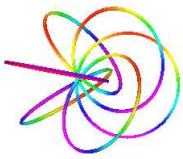
[8] Pereira, J.C., Conceição, A.C. "F-Tool 2.0: Exploring the Logistic Function in the Classroom", *SYMCOMP2013 Proceedings of the 1st International Conference on Algebraic and Symbolic Computation (ECCOMAS Thematic Conference)*, Ed. A. Loja, J. I. Barbosa, and J. A. Rodrigues, Lisboa, Portugal, pp. 149-158, September 9-10, 2013.

[9] Ruskeepää, H.. *Mathematica® Navigator. Mathematics, Statistics, and Graphics*, 3rd edition, Academic Press, 2009.

[10] Torrence, B.F. and Torrence, E.A., *The Student's Introduction to Mathematica®, A Handbook for Precalculus, Calculus, and Linear Algebra*, 2nd edition, Cambridge University Press, Cambridge, 2009.

[11] Trott, M., *The Mathematica Guidebook for Symbolics*, Springer Science+Business Media, Inc., New York, 2006.

[12] Wolfram, S.. *The Mathematica® Book*, 5th edition, Wolfram Media, 2003.



SYMCOMP 2015
Faro, March 26-27, 2015

6TH-ORDER FINITE VOLUME APPROXIMATION FOR THE STEADY-STATE BURGER AND EULER EQUATIONS: THE MOOD APPROACH

Gaspar J. Machado[1*], Stéphane Clain[1,2], Raphael Loubère[2], and Steven Diot[3]

1: Centre of Mathematics
School of Science
University of Minho
Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: {gjm,clain}@math.uminho.pt

2: Institut de Mathématiques de Toulouse
Université Paul Sabatier
31062 Toulouse, France
e-mail: raphael.loubere@math.univ-toulouse.fr

3: Fluid Dynamics and Solid Mechanics (T-3)
Los Alamos National Laboratory
Los Alamos, NM 87545, USA
e-mail: diot@lanl.gov

Keywords: Finite volume, MOOD, sixth-order approximation, Burgers' equation, Euler's equations

Abstract. *We propose an innovative method based on the MOOD technology (Multi-dimensional Optimal Order Detection) to provide a 6th-order finite volume approximation for the one-dimensional steady-state Burger and Euler equations. The main ingredient consists in using an 'a posteriori' limiting strategy to eliminate non physical oscillations deriving from the Gibbs phenomenon while keeping a high accuracy for the smooth part. A short overview of the MOOD method will be presented and numerical tests with regular or discontinuous solutions will assess the method capacity to produce excellent approximations. In the latter situation, the numerical results enable to detect the zone where it is necessary to reduce the degree of the polynomial reconstructions to preserve the scheme robustness.*

1 INTRODUCTION

Computation of accurate approximations for steady-state hyperbolic systems such as the Euler's equations is a constant challenge due to the wide panel of applications in aeronautic, aerospace, and environmental problems. In contrast with non stationary situations, the steady-state case brings more fundamental difficulties such as non uniqueness or oscillations around discontinuities keeping the iterative procedure from converging. Another important issue concerns the accuracy of the approximations where the numerical diffusion may prevent the solver from converging to the correct solution.
Most of the finite volume commercial softwares (FLUENT for instance) use a MUSCL strategy to provide a second-order of accuracy of the steady-state approximations [2, 3, 5, 4]. Other strategies based on WENO/CENO technique propose fourth- or fifth-order methods [11, 12, 13] to capture the correct steady-state solutions but they almost use structured grids since the technique is very time consuming in an unstructured mesh context [1, 16, 15].
Very recently, a new limiter paradigm has been proposed to control the oscillations in the vicinity of the shocks and has been successfully applied to the non stationary Euler's equations [6, 7, 8]. The Multidimensional Optimal Order Detection (MOOD) is an *a posteriori* limiter which enables very high-order approximations. It prevents from the Gibbs phenomenon and is very low time consuming with respect to the WENO/CENO methods. The question we tackle in this paper is the capacity of the MOOD method to deal with steady-state solutions and to provide the correct one.
We first consider the Burgers' equation since it is the standard non linear scalar case that one has to first solve correctly. We introduce the discretisation and the polynomial reconstruction and then we present the MOOD method for the scalar case. Numerical simulations have been carried out to prove that the technology can be successfully applied to a stationary problem. We then deal with the steady-state Euler's equations and adapt the limiting procedure. New numerical simulations are carried out to show that the MOOD strategy enables to capture the solution with a very high accuracy.
The rest of the paper is organized as follows. In section 2 we present the models, section 3 deals with the generic finite volume schemes, and in section 4 we introduce the MOOD technology for the steady-state case which is very different from the non stationary context. Sections 5 and 6 are dedicated to the numerical tests and in the last section we present the conclusions of the work.

2 Models

We will consider in this work the one-dimensional steady-state Burgers' equations (scalar case) and the one-dimensional steady-state Euler's equations (vectorial case).

2.1 Burgers' equations

We seek the velocity function $u = u(x)$, solution of the 1D steady-state inviscid Burgers' equation

$$\frac{\mathrm{d}}{\mathrm{d}x}F(u) = f, \text{ in } \Omega = (0,1) \tag{1a}$$

with Dirichlet boundary conditions

$$u = u_{\mathrm{lf}}, \text{ on } x = 0, \tag{1b}$$
$$u = u_{\mathrm{rg}}, \text{ on } x = 1 \tag{1c}$$

where the flux is given by

$$F(u) = \frac{u^2}{2}$$

and $f = f(x)$ represents the source term.
Burgers' equation is one of the simplest nonlinear equation which is often used as a prototype problem for which shocks can develop.

2.2 Euler's equations

We seek the density $\rho = \rho(x)$, the velocity $u = u(x)$ and the pressure $p = p(x)$ solutions of the 1D steady-state Euler's equations

$$\frac{\mathrm{d}}{\mathrm{d}x}F(U) = f, \text{ in } \Omega = (0,1) \tag{2a}$$

with Dirichlet boundary conditions

$$U = U_{\mathrm{lf}}, \text{ on } x = 0, \tag{2b}$$
$$U = U_{\mathrm{rg}}, \text{ on } x = 1, \tag{2c}$$

where the conservative variable U is given by

$$U = (\rho, \rho u, E)^T$$

and the flux is given by

$$F(U) = (\rho u, \rho u^2 + p, u(E+p)^T.$$

The source term is represented by $f = (f_1, f_2, f_3)^T = (f_1(x), f_2(x), f_3(x))^T$ and the total energy per unit volume is given by

$$E = \frac{1}{2}\rho u^2 + e, \tag{3}$$

where e is the specific internal energy. For an ideal gas, this system is closed by the equation of state

$$e = \frac{p}{\rho(\gamma - 1)} \tag{4}$$

with γ the ratio of specific heats ($\gamma = \frac{7}{5}$ in our studies).

3 FINITE VOLUME SCHEMES

In this section we describe the numerical finite volume schemes for Burgers' equation, being their immediate extensions to Euler's equations, starting by presenting the notations and the polynomial reconstruction machinery.

3.1 Polynomial reconstruction

Let $\mathcal{T}_h$ be a mesh of the interval $\overline{\Omega} = [0,1]$ constituted of cells $K_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, $i = 1,\dots,I$, with centroid x_i, where $x_{\frac{1}{2}} = 0$, $x_{I+\frac{1}{2}} = 1$, and $x_{i+\frac{1}{2}} = x_{i-\frac{1}{2}} + h_i$, $i = 1,\dots,I$ stand for the interfaces.
To achieve high-order numerical approximations, we introduce local polynomial reconstructions of the underlying solutions. At the first stage, we define the stencils associated to the cells. For any cell K_i, $i = 1,\dots,I$, and any degree $\mathsf{d_i}$ of the polynomial reconstruction, we shall denote by S_i the stencil composed of the $\mathsf{d_i}+1$ closest neighbour cells (excluding cell K_i). The second stage consists in defining the polynomial reconstructions based on the data of the associated stencil. To this end, and for any given scalar function ϕ defined on Ω, let $\Phi = (\phi_i)_{i=1,\dots,I}$ where $\phi_i \in \mathbb{R}$ is an approximation of the mean value of ϕ over cell K_i. So, the polynomial reconstruction of degree $\mathsf{d_i}$ associated to cell K_i is defined as

$$\boldsymbol{\phi}_i(x;\mathsf{d_i}) = \phi_i + \sum_{\alpha=1}^{\mathsf{d_i}} R_{i,\alpha}\left[(x-x_i)^\alpha - M_{i,\alpha}\right],$$

where we set $M_{i,\alpha} = \frac{1}{h_i}\int_{K_i}(x-x_i)^\alpha \,\mathrm{d}x$ to provide a conservative property, that is, $\frac{1}{h_i}\int_{K_i}\boldsymbol{\phi}_i(x)\,\mathrm{d}x = \phi_i$, and the vector $R_i = (R_{i,\alpha})_{\alpha=1,\dots,\mathsf{d_i}}$ gathers the polynomial coefficients. For a given stencil S_i, we consider the quadratic functional

$$\widehat{E}_i(R_i) = \sum_{j\in S_i}\left[\frac{1}{h_j}\int_{K_j}\boldsymbol{\phi}_i(x;\mathsf{d_i})\,\mathrm{d}x - \phi_j\right]^2.$$

We denote by $\widehat{R}_i$ the unique vector which minimizes the quadratic functional and set $\widehat{\boldsymbol{\phi}}_i$ the associated polynomial that corresponds to the best approximation in the least squares sense of the data of the stencil.

3.2 First-order finite volume scheme

Using the classical finite volume methodology, equation (1a) is integrated over cell K_i, $i = 1,\dots,I$, resulting in

$$\frac{1}{h_i}\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right) - \bar{f}_i = 0, \tag{5}$$

with $F_{i+\frac{1}{2}} = F(u(x_{i+\frac{1}{2}}))$ and $\bar{f}_i = \frac{1}{h_i}\int_{K_i} f(\xi)\,\mathrm{d}\xi$. The exact mean source term $\bar{f}_i$ is approximated by f_i through Gaussian quadrature approximation and the physical flux

at the interface $F_{i+\frac{1}{2}}$, denoted by $\mathcal{F}_{i+\frac{1}{2}}$, is approximated by the Rusanov numerical flux, namely:

- left boundary interface ($i = 0$)

$$\mathcal{F}_{\frac{1}{2}}(\Phi) = \frac{1}{2}\left(\frac{\phi_{\text{lf}}^2}{2} + \frac{\phi_1^2}{2}\right) - \frac{\max(|\phi_{\text{lf}}|, |\phi_1|)}{2}(\phi_1 - \phi_{\text{lf}});$$

- inner interfaces ($i = 1, \ldots, I-1$)

$$\mathcal{F}_{i+\frac{1}{2}}(\Phi) = \frac{1}{2}\left(\frac{\phi_i^2}{2} + \frac{\phi_{i+1}^2}{2}\right) - \frac{\max(|\phi_{i+1}|, |\phi_i|)}{2}(\phi_{i+1} - \phi_i);$$

- right boundary interface ($i = I$) — similar to the left boundary interface.

Let us now define the residual at cell K_i by

$$\mathcal{G}_i(\Phi) = \frac{1}{h_i}\left(\mathcal{F}_{i+\frac{1}{2}} - \mathcal{F}_{i-\frac{1}{2}}\right) - f_i, \tag{6}$$

and introduce the nonlinear operator $\mathcal{G}(\Phi) = (\mathcal{G}_1(\Phi), \ldots, \mathcal{G}_I(\Phi))^T$. The numerical solution is then given by vector $\Phi^\dagger = (\phi_i^\dagger)_{i=1,\ldots,I}$ which is the solution of the nonlinear problem $\mathcal{G}(\Phi) = 0_I$. This solution is known to be first-order accurate.

3.3 Generic high-order finite volume scheme

To construct a generic high-order solver of (1a)-(1c) one has to substitute the left and right states in by states evaluated through high-order polynomial reconstructions. Let us assume that a cell polynomial degree map is given $\mathcal{M} = (\mathsf{d_1}, \ldots, \mathsf{d_I})^T$ with its associated stencil map $S = (S_1, \ldots, S_I)^T$. Using these maps, we define the state values:

- first cell ($i = 1$)

$$\widehat{\phi}_1^- = \widehat{\boldsymbol{\phi}}_1(x_{\frac{1}{2}}; \mathsf{d_1}) \quad \text{and} \quad \widehat{\phi}_1^+ = \widehat{\boldsymbol{\phi}}_1(x_{\frac{3}{2}}; \min(\mathsf{d_1}, \mathsf{d_2}));$$

- inner cells ($i = 2, \ldots, I-1$)

$$\widehat{\phi}_i^- = \widehat{\boldsymbol{\phi}}_i(x_{i-\frac{1}{2}}; \min(\mathsf{d_{i-1}}, \mathsf{d_i})) \quad \text{and} \quad \widehat{\phi}_i^+ = \widehat{\boldsymbol{\phi}}_i(x_{i+\frac{1}{2}}; \min(\mathsf{d_i}, \mathsf{d_{i+1}}));$$

- last cell ($i = I$) — similar to the first cell.

Note that the minimal polynomial degree $\min(\mathsf{d_i}, \mathsf{d_{i+1}})$ at interface $x_{i+\frac{1}{2}}$ is mandatory to ensure that the cell is updated with the first-order scheme when $\mathsf{d_i} = 0$ or $\mathsf{d_{i+1}} = 0$. The numerical fluxes write:

- left boundary interface ($i = 0$)

$$\mathcal{F}_{\frac{1}{2}}(\Phi) = \frac{1}{2}\left(\frac{(\phi_{\text{lf}})^2}{2} + \frac{\left(\widehat{\boldsymbol{\phi}}_1(0)\right)^2}{2}\right) - \frac{\max\left(|\phi_{\text{lf}}|, \left|\widehat{\boldsymbol{\phi}}_1(0)\right|\right)}{2}\left(\widehat{\boldsymbol{\phi}}_1(0) - \phi_{\text{lf}}\right);$$

- inner interfaces ($i = 1, \ldots, I-1$)

$$\mathcal{F}_{i+\frac{1}{2}}(\Phi) = \frac{1}{2}\left(\frac{\left(\widehat{\phi}_i^+\right)^2}{2} + \frac{\left(\widehat{\phi}_{i+1}^-\right)^2}{2}\right) - \frac{\max\left(\left|\widehat{\phi}_i^+\right|, \left|\widehat{\phi}_{i+1}^-\right|\right)}{2}\left(\widehat{\phi}_{i+1}^- - \widehat{\phi}_i^+\right);$$

- right boundary interface ($i = I$) — similar to the left.

The residual associated to cell K_i turns to be

$$\mathcal{G}_i(\Phi) = \frac{1}{h_i}\left(\mathcal{F}_{i+\frac{1}{2}} - \mathcal{F}_{i-\frac{1}{2}}\right) - f_i, \tag{7}$$

and we get again the nonlinear operator $\mathcal{G}(\Phi) = (\mathcal{G}_1(\Phi), \ldots, \mathcal{G}_I(\Phi))^T$. The numerical solution is then given by vector $\Phi^\dagger = (\phi_i^\dagger)_{i=1,\ldots,I}$ which is the solution of the nonlinear problem $\mathcal{G}(\Phi) = 0_I$. This final solution does explicitly depends on the map $\mathcal{M}$ and the stencil S. As a consequence the system can be written in a more systematic way into

$$\mathcal{G}(\Phi; \mathcal{M}, S) = 0_I. \tag{8}$$

If the solution is regular enough the high-order scheme must retrieve it with high accuracy and an optimal rate of convergence. In presence of local discontinuities, Gibbs phenomenon will be triggered and oscillations are expected and high-order polynomial reconstructions can not be used any more leading the cell polynomial degree d to drop to 0 in the vicinity of the discontinuity. In the following, we construct a scheme capable of dealing with this situation.
For the numerical implementation treating one scalar equation like Burgers' equation or one system like Euler's equations does not modify the strategy: we solve a set of nonlinear coupled equations, only the nonlinear coupling between them may change also the flux and the dependencies between the variables.

4 MOOD SCHEMES

Recently a new limiting technology named the Multidimensional Optimal Order Detection (MOOD) has been proposed and tested in the Euler system context [6, 7, 8]. We propose a short overview of the method and detail the adaptation to the steady-state context.

4.1 Philosophy of Detection and Decrementing

Let us assume that a candidate solution $\Phi^{\dagger,1}$ has been computed with the high-order scheme with a maximal map $\mathcal{M}^1 = (\mathsf{d}^{\mathsf{max}}, \dots, \mathsf{d}^{\mathsf{max}})^T$ and its associated stencil S^1.
The detection criteria used in this work is mimicked from the original Detection criteria in [6, 7, 8]: a first filter detects all extrema (called ED) and a second filter (called u2) applied on detected extrema splits them into regular or non-regular ones. For non-regular extrema, say in cell K_i, one decrements the cell polynomial degree $\mathsf{d_i}$ modifying *de facto* the map into $\mathcal{M}^2$ and the associated stencil S^2. This new map generates several new polynomial reconstructions and consequently new state variable evaluations. Next the solution is recomputed taking into account these modifications. This new candidate solution $\Phi^{\dagger,2}$ is then checked using the detection criteria. New decrementing may occur leading to a new map $\mathcal{M}^3$ with stencil S^3 and a new candidate solution $\Phi^{\dagger,3}$. The iterative procedure stops when the map does not change any more.

4.2 MOOD method

The classical MOOD method would apply the following steps:

0. Initialize. Set $k = 1$. Choose the initial map $\mathcal{M}^1 = (\mathsf{d}^{\mathsf{max}}, \dots, \mathsf{d}^{\mathsf{max}})^T$, its associated stencil $S^1 = (S_1^1, \dots, S_I^1)^T$, and an initial guess Φ^0.
1. Compute $\Phi^{\dagger,k}$ solution of $\mathcal{G}(\Phi; \mathcal{M}^k, S^k) = 0_I$.
2. Detect and Decrement
   - Detect and flag non-regular extrema in set $\mathcal{E}$ using ED+u2 detector.
   - Compute the new cell polynomial degree map $\mathcal{M}^{k+1}$ with $\mathsf{d_i}^{k+1} = \varphi(\mathcal{M}^k)$ for $i \in \mathcal{E}$ and $\mathsf{d_i}^{k+1} = \mathsf{d_i}^k$ for the other cells.
   - Compute the new stencil map S^{k+1} on the basis of $\mathcal{M}^{k+1}$.
3. Exit test. If $\Phi^{\dagger,k+1}$ is close to $\Phi^{\dagger,k}$ or if $\mathcal{M}^{k+1} = \mathcal{M}^k$ then exit the loop else set $k = k + 1$ and go back to 1.

In our simulations the decrementing function φ implements a three step sequence: $\mathsf{d}^{\mathsf{max}} \to 2 \to 0$.

5 NUMERICAL RESULTS FOR BURGERS' EQUATION

In this section we present the numerical tests that have been carried out for the Burgers' equation. The methodology is made on three steps

First, we test the numerical method for a regular solution. Using a sequence of refined meshes one computes the relative errors and infere the order of convergence of the numerical scheme. When $\mathbb{P}_\mathsf{d}$ polynomial reconstructions are employed, that is, $\mathcal{M} = (\mathsf{d}, \dots, \mathsf{d})^T$, one expects a $\mathsf{d} + 1$ order of convergence.

Second, irregular solution is simulated with the first order numerical scheme to validate the shock capturing behavior. Also schemes above first order are tested to enlighten the Gibbs phenomenon, that is to say parasitical numerical oscillations that are observed when discontinuous profiles are computed.

Third the MOOD scheme is tested with cell polynomial decrementing as presented in section 4.2 for regular and irregular solutions. One expects that the MOOD scheme provides both the shock capturing behavior around discontinuity and an optimal order of convergence on smooth solutions. The accuracy of the scheme must increase when polynomial reconstruction employs high degree.

5.1 Regular solution

In order to check the implementation of the method and assess the convergence rates, we manufacture an analytical solution for the given problem setting

$$u(x) = \sin(3\pi x)\exp(x) + 2.$$

Then, the source term is given by

$$f(x) = (\exp(x)\sin(3\pi x) + 2)(\exp(x)\sin(3\pi x) + 3\pi\exp(x)\cos(3\pi x))$$

and the Dirichlet boundary conditions derive from the exact solution, namely on the left boundary point prescribe $u_{\text{lf}} = 2$ and on the right boundary point $u_{\text{rg}} = 2$.
Since we deal with a regular solution, we use the L^∞ norm to compute the error between the solution and the approximation, which is given by

$$E_\infty(I) = \max_{i=1}^{I} |u_i - \bar{u}_i|,$$

and the convergence order between two meshes characterized by I_1 and I_2 cells is given by

$$O_\infty(I_1, I_2) = \frac{|\log(E_\infty(I_1)/E_\infty(I_2))|}{|\log(I_1/I_2)|},$$

where $\bar{u}_i$ is the exact mean value of u over cell K_i. The initial guess is the constant vector $(2, \ldots, 2)^T$.
Table 1, which reports the errors and convergence rates for this smooth solution, shows that we achieve in all situations the optimal order. Moreover, no problematic cells have been detected and the MOOD algorithm preserves the accuracy.
We display in Fig. 1 the exact solution and the approximate one (top panels) as well as the error curves between the approximations (bottom panels) using respectively the $\mathbb{P}_0$, $\mathbb{P}_1$, and $\mathbb{P}_5$ reconstructions. As expected the solution is well-captured with a better accuracy when the polynomial degree of the reconstruction increases.

Table 1: Convergence table for the regular solution of the Burgers' equation.

	I	E_∞	$\mathcal{O}_\infty$	# bad cells
$\mathbb{P}_0$	40	3.2E−01	—	0
	80	1.6E−01	1.0	0
	160	8.2E−02	1.0	0
	320	4.1E−02	1.0	0
$\mathbb{P}_1$	40	2.9E−02	—	0
	80	5.4E−03	2.4	0
	160	1.1E−03	2.3	0
	320	2.5E−04	2.2	0
$\mathbb{P}_5$	40	2.9E−05	—	0
	80	3.8E−07	6.2	0
	160	8.5E−09	5.5	0
	320	1.6E−10	5.7	0

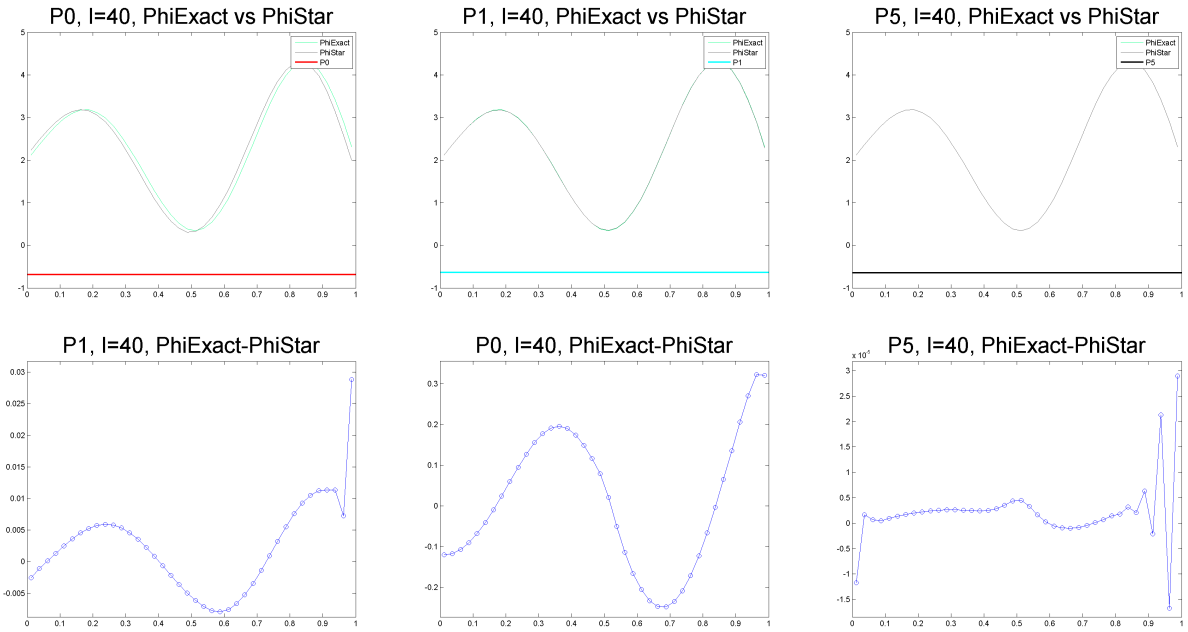


Figure 1: Sinus/Exponential regular test case for a 40 cell mesh and several unlimited polynomial reconstructions. Top panels: exact solution (cyan) and approximate solutions (black) — Bottom panels: error. Left panel: $\mathbb{P}_0$ reconstruction — Middle panel: $\mathbb{P}_1$ reconstruction — Right panel: $\mathbb{P}_5$ reconstruction.

5.2 Solution with a shock

Following the test proposed in [9], we check the new algorithm with a discontinuous solution. As instance considering the source term $f(x) = -\pi\cos(\pi x)u(x)$ and the boundary conditions $u_{\text{lf}} = 1$ and $u_{\text{rg}} = -0.1$, we can derive the exact solution

$$u(x) = \begin{cases} 1 - \sin(\pi x) & \text{if } 0 \leq x \leq x_{\text{s}}, \\ -0.1 - \sin(\pi x) & \text{if } x_{\text{s}} \leq x \leq 1, \end{cases}$$

where x_{s} is the location of the shock. Two solutions are possible $x_{\text{s}} = 0.1486$ and $x_{\text{s}} = 0.8514$ but only the one with the first shock is stable for small perturbation.
The initial guess is chosen to be relatively close to the exact solution and given by the vector $U^0 = (u_i^0)_{i=1,\dots,I}$,

$$u_i^0 = \begin{cases} 1 & \text{if } 0 \leq x_i \leq \frac{1}{4}, \\ -0.1 & \text{if } \frac{1}{4} \leq x_i \leq 1. \end{cases}$$

First-order $\mathbb{P}_0$ and unlimited second-order $\mathbb{P}_k$ schemes We first consider the $\mathbb{P}_0$, $\mathbb{P}_1$, and $\mathbb{P}_5$ polynomial reconstructions with $I = 120$ where the MOOD loop is not triggered meaning that the schemes are "unlimited" (*cf.* Fig. 2). As expected the $\mathbb{P}_0$ scheme solution is rather diffused while the $\mathbb{P}_1$ and $\mathbb{P}_5$ scheme solutions seem sharper and more accurate but with numerical oscillations as expected.

High-order MOOD schemes We now trigger the MOOD loop to eliminate the non-physical oscillations. We start the MOOD iteration with maximal degree $\mathsf{d}^{\mathsf{max}} = 1$ and 5. In Fig. 3 we display the results obtained with the MOOD schemes. We observe that the MOOD method provides a stable and sharp solution without oscillation. The cell polynomial degree map shows that only cells around the shock wave have be decremented whereas the other cells are treated with the most accurate scheme. Table 2 prints the errors computed on the smooth part of the domain and show that we get a second-order of accuracy for the regular part of the curve even with higher polynomial reconstructions. The existence of the shock keeps the numerical solution to be better than a second-order one.

6 NUMERICAL RESULTS FOR EULER SYSTEM OF EQUATIONS

We now turn to the vectorial case using the MOOD methodology for the Euler's equations.

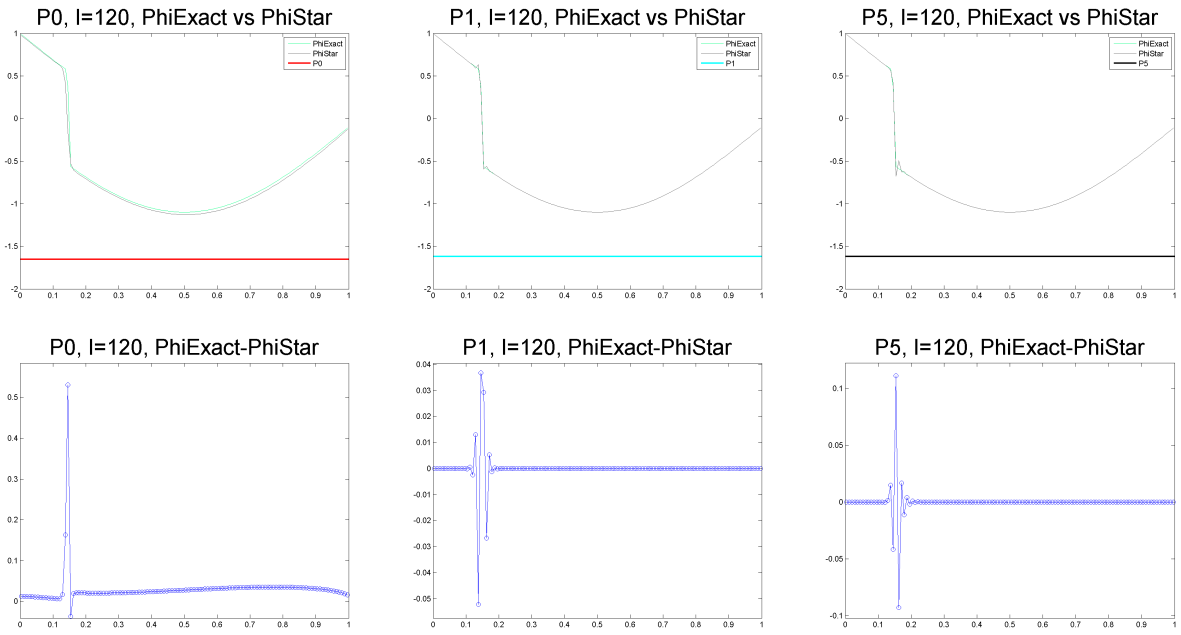


Figure 2: Discontinuous sinus test case for a mesh with 120 cells and several unlimited polynomial reconstructions. Top panels: exact solution (cyan) and approximate solutions (black) — Bottom panels: errors. Left panel: $\mathbb{P}_0$ reconstruction — Middle panel: $\mathbb{P}_1$ reconstruction — Right panel: $\mathbb{P}_5$ reconstruction.

Table 2: Convergence table for the discontinuous sinus test case with the and iterative MOOD loop $\mathsf{d}^{\mathsf{max}} \to 2 \to 0$. Errors are computed on the smooth part of the domain.

I	$E_\infty([0.3;1])$	$\mathcal{O}_\infty([0.3;1])$
80	9.8E−05	—
100	6.3E−05	2.0
120	4.3E−05	2.0
140	3.2E−05	2.0

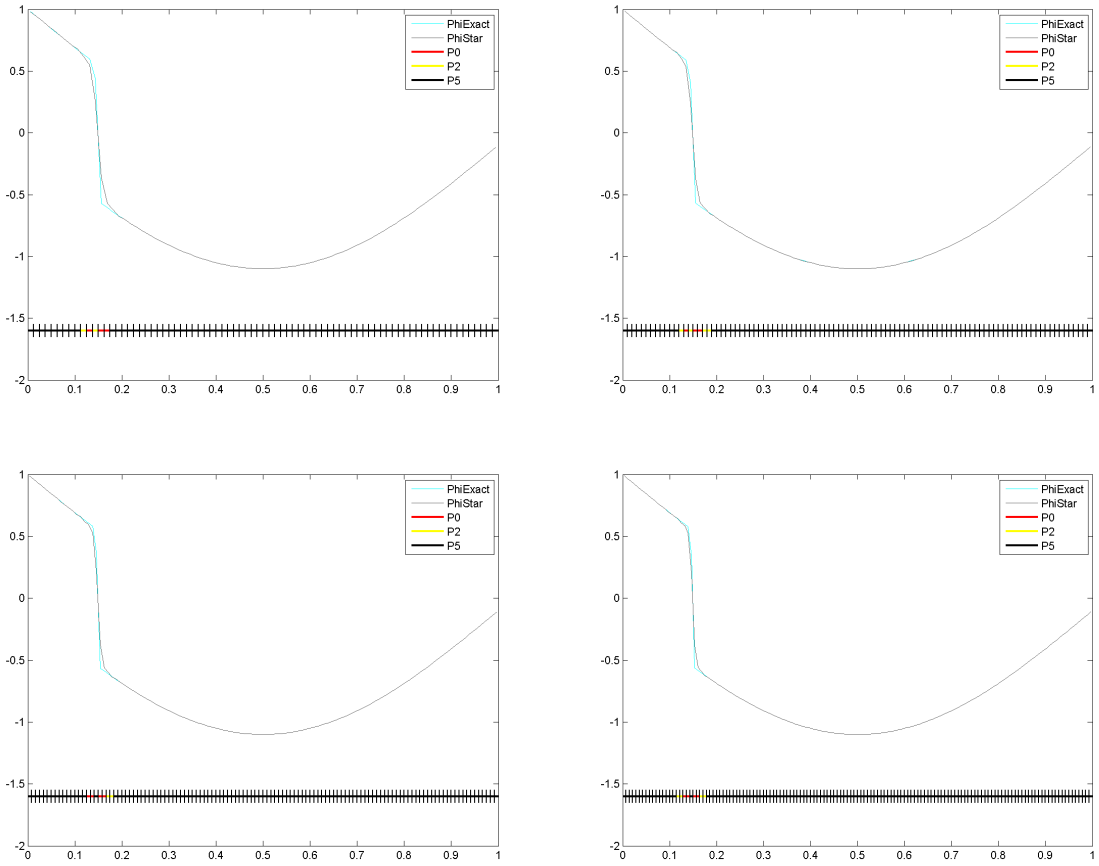


Figure 3: Discontinuous sinus test case for a mesh with 120 cells and iterative MOOD loop $\mathsf{d}^{\max} \to 2 \to 0$. Exact solution and approximate solution (narrow colored line) along with the mesh (bottom) colored with the cell polynomial degree at convergence.

6.1 Regular solution

The first test corresponds to a manufactured smooth solution given by

$$\begin{aligned}\rho(x) &= \exp(x) + \exp(x)\sin(3\pi x) + 2,\\ u(x) &= \sin(2\pi x) + \exp(x),\\ p(x) &= \exp(x).\end{aligned}$$

Algebraic computations provide the source term

$$\begin{aligned}f_1(x) &= (\exp(x) + 2\pi\cos(2\pi x))(\exp(x) + \exp(x)\sin(3\pi x) + 2) +\\ &\quad (\exp(x) + \sin(2\pi x))(\exp(x) + \exp(x)\sin(3\pi x) + 3\pi\exp(x)\cos(3\pi x)),\\ f_2(x) &= \exp(x) + (\exp(x) + \sin(2\pi x))^2(\exp(x) + \exp(x)\sin(3\pi x) + 3\pi\exp(x)\cos(3\pi x)) +\\ &\quad 2(\exp(x) + \sin(2\pi x))(\exp(x) + 2\pi\cos(2\pi x))(\exp(x) + \exp(x)\sin(3\pi x) + 2),\\ f_3(x) &= (\exp(x) + \sin(2\pi x))(((\exp(x) + \sin(2\pi x))^2(\exp(x) + \exp(x)\sin(3\pi x) +\\ &\quad 3\pi\exp(x)\cos(3\pi x)))/2 + \exp(x)/(\gamma - 1) + (\exp(x) + \sin(2\pi x))(\exp(x) +\\ &\quad 2\pi\cos(2\pi x))(\exp(x) + \exp(x)\sin(3\pi x) + 2)) + (\exp(x) + 2\pi\cos(2\pi x))(((\exp(x) +\\ &\quad \sin(2\pi x))^2(\exp(x) + \exp(x)\sin(3\pi x) + 2))/2 + \exp(x)/(\gamma - 1)) + \exp(x)(\exp(x) +\\ &\quad \sin(2\pi x)) + \exp(x)(\exp(x) + 2\pi\cos(2\pi x)),\end{aligned}$$

while the Dirichlet boundary conditions derive from the exact solution, namely on the left boundary point, we prescribe

$$U_{\text{lf}} = \left(3, 3, \frac{1}{\gamma - 1} + \frac{3}{2}\right)^T$$

and on the right boundary point

$$U_{\text{rg}} = \left(\mathrm{e} + 2, \mathrm{e}(\mathrm{e} + 2), \frac{\mathrm{e}^2(\mathrm{e} + 2)}{2} + \frac{\mathrm{e}}{\gamma - 1}\right)^T.$$

Table 3 gives the errors and convergence rates while Fig. 4 displays the exact and the numerical density for the $\mathbb{P}_0$, $\mathbb{P}_1$, and $\mathbb{P}_5$ reconstructions with a mesh of 40 cells. We report an excellent optimal rate and we underline that the $\mathbb{P}_0$ solution does not provide the correct approximation. Indeed, the low level approximation converges only with very fine meshes while the higher approximations immediately provide the good solution.

6.2 Solution with a shock

We now consider the case with a discontinuity where we test a steady shock. To do so, we impose a supersonic condition on the left boundary and a subsonic condition on the

Table 3: Convergence table for the regular solution of the Euler's equation.

	I	$E_\infty(\rho)$	$\mathcal{O}_\infty(\rho)$	$E_\infty(\rho u)$	$\mathcal{O}_\infty(\rho u)$	$E_\infty(E)$	$\mathcal{O}_\infty(E)$
$\mathbb{P}_0$	40	7.9E−01	—	2.7E−01	—	1.4E+00	—
	80	5.1E−01	0.6	1.7E−01	0.7	9.9E−01	0.5
	160	2.3E−01	1.2	1.0E−01	0.7	4.0E−01	1.3
	320	1.0E−01	1.2	5.5E−02	0.9	1.6E−01	1.3
$\mathbb{P}_1$	40	3.4E−02	—	1.7E−02	—	4.0E−02	—
	80	9.2E−03	1.9	3.7E−03	2.2	8.9E−03	2.2
	160	2.3E−03	2.0	9.0E−04	2.0	2.2E−03	2.0
	320	5.7E−04	2.0	2.2E−04	2.0	5.4E−04	2.0
$\mathbb{P}_5$	40	6.5E−04	—	3.7E−04	—	1.3E−03	—
	80	1.0E−05	6.0	2.0E−06	7.5	1.6E−05	6.4
	160	1.3E−07	6.3	1.0E−07	4.3	2.0E−07	6.3
	320	1.9E−09	6.1	2.2E−09	5.6	3.0E−09	6.0

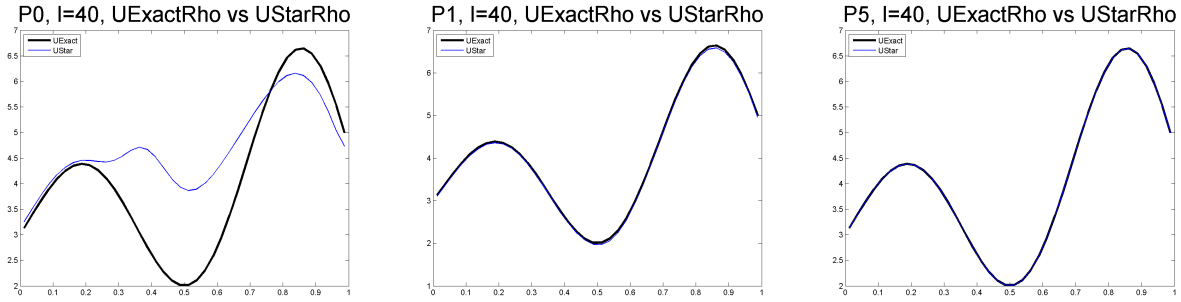


Figure 4: Exact (black) versus numerical (blue) density for $\mathbb{P}_0$ (left), $\mathbb{P}_1$ (centre), and $\mathbb{P}_5$(right) reconstructions.

right boundary. The data is the following:

$$
\begin{aligned}
&f_1(x) = 0,\\
&f_2(x) = 1,\\
&f_3(x) = 0,\\
&U_{\text{lf}} = (2.4142, 1.0000, 0.7058)^T,\\
&U_{\text{rg}} = (3.9598, 1.0000, 1.0967)^T.
\end{aligned}
$$

Figure 5 displays the numerical solutions for the density again with $\mathbb{P}_0$, $\mathbb{P}_1$, and $\mathbb{P}_5$ reconstructions with a mesh of 200 cells and no limiting process. Oscillations are clearly observed and localized around the discontinuity. The last part of the figure shows the numerical solution using the MOOD algorithm where we initialized the cell polynomial degree map with degree 5. We report that the oscillations vanish and the cell polynomial degree map is essentially equal to 5 except in the vicinity of the shock. The MOOD strategy succeeds in eliminating the oscillations while preserving the high degree of reconstruction in the smooth part of the function.

7 CONCLUSION

In this document, we show that the MOOD methodology is well-adapted to the steady-state configuration and we design a new strategy to get very high accuracy for smooth solution and robustness for rough solution. The method has been experimented in the one-dimensional framework both for the scalar case and the vectorial case and future works will be dedicated to the multidimensional case.

ACKNOWLEDGEMENTS

This research was financed by FEDER Funds through Programa Operational Fatores de Competitividade — COMPETE and by Portuguese Funds FCT — Fundação para a Ciência e a Tecnologia, within the Projects PEst-C/MAT/UI0013/2014, PTDC/MAT/-121185/2010, and FCT-ANR/MAT-NAN/0122/2012.

REFERENCES

[1] Abgrall, R., "On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation", *J. Comput. Phys.* 114, pp. 45-58, 1994.

[2] Berthon, C. "Robustness of MUSCL Schemes for 2D unstructured meshes", *J. Comput. Phys.* 218, pp. 495-509, 2006.

[3] Buffard, T., Clain, S., "Monoslope and Multislope MUSCL Methods for unstructured meshes", *J. Comput. Phys.* 229, pp. 3745-3776, 2010.

[4] Clain, S., "Finite Volume Maximum Principle for Hyperbolic scalar problems", *SIAM J. Numer. Anal.* 51(1), pp. 467-490, 2013.

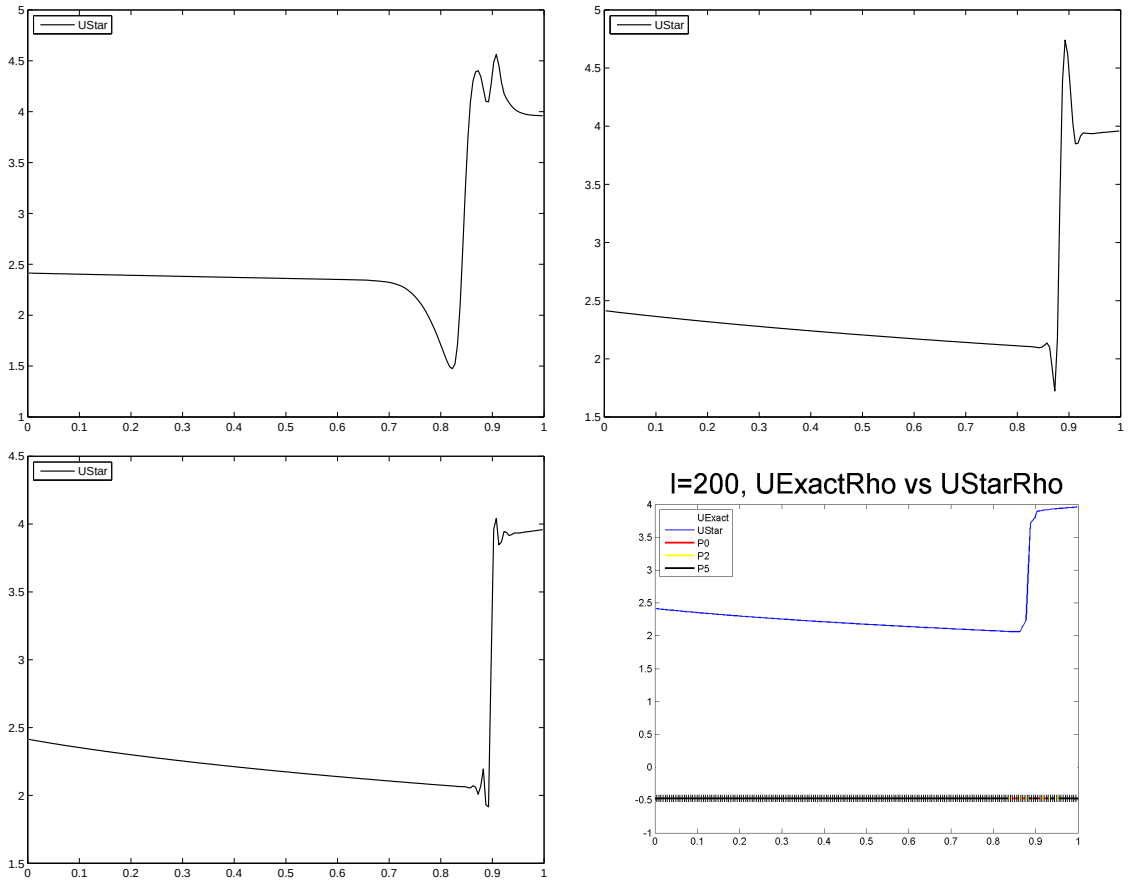


Figure 5: Numerical approximation for the density variable with $\mathbb{P}_0$ (top, left), $\mathbb{P}_1$ (top, right), and $\mathbb{P}_5$ (bottom, left) reconstructions and the numerical solution after the application of the MOOD algorithm (bottom, right).

[5] Clain, S., Clauzon, V., "L^∞ stability of the MUSCL methods", *Numer. Math.* 116, pp. 31-64, 2010.

[6] Clain, S., Diot, S., Loubère, R., "A high-order finite volume method for systems of conservation laws — Multi-dimensional Optimal Order Detection (MOOD)", *J. Comput. Phys.* 230, pp. 4028-4050, 2011.

[7] Diot, S., Clain, S., Loubère, R., "Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials", *Comput. Fluids* 64, pp. 43-63, 2012.

[8] Diot, S., Loubère, R., Clain, S., "The MOOD method in the three-dimensional case: Very-High-Order Finite Volume Method for Hyperbolic Systems", *Int. J. Numer. Meth. Fl.* 73(4), pp. 362-392, 2013.

[9] Chen, W., Chou, C.-S., Kao,C.-Y., "Fast Sweeping Methods for Steady State Problems for Hyperbolic Conservation Laws", *Journal of Computational Physics* 234, pp. 452-471, 2012.

[10] Hu, C., Shu, C. W., "Weighted essentially non-oscillatory schemes on triangular meshes", *J. Comput. Phys.* 150, pp. 97-127, 1999.

[11] Jiang,G.-S., Shu,C.-W., "Efficient implementation of weighted ENO schemes", *J. Comput. Phys.* 126, pp. 202-228, 1996.

[12] Liu, X. D., Osher, S., Chan, T., "Weighted essentially non-oscillatory schemes", *J. Comput. Phys.* 115, pp. 200-212, 1994.

[13] Ollivier-Gooch, C., Nejat, A., Michalak, K., "Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations", *AIAA Journal* 47, pp. 2105-2120, 2009.

[14] Toro, E. F., "Riemann Solvers and Numerical Methods for Fluid Dynamics, 3^{rd} revision", Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2009.

[15] Wolf, W. R., Azevedo, J. L. F., "High-order ENO and WENO schemes for unstructured grids", *International Journal for Numerical Methods in Fluids* 55, pp. 917-943, 2007.

[16] Zhang,Y.-T. and Shu, C.-W., "Third-order WENO scheme on three dimensional tetrahedral meshes", *Com. Comput. Phys.* 5 pp. 836-848, 2009.

SYMCOMP 2015
Faro, March 26-27, 2015

6TH-ORDER FINITE VOLUME APPROXIMATIONS FOR THE STOKES EQUATIONS WITH A CURVED BOUNDARY

Ricardo Costa[1*], Stéphane Clain[1,2] Gaspar J. Machado[1]

1: Centre of Mathematics
School of Science
University of Minho
Campus de Azurm, 4800-058 Guimares, Portugal
e-mail: pg24046@alunos.uminho.pt,{clain,gjm}@math.uminho.pt

2: Institut de Mathématiques de Toulouse
Université Paul Sabatier
31062 Toulouse, France

Keywords: finite volume, sixth-order approximation, Stokes equation, curved boundary

Abstract. *A new solver for the Stokes equations based on the finite volume method is proposed using very accurate polynomial reconstruction to provide a 6th-order scheme. We face two main difficulties: the gradient-divergence duality where the divergence free condition will impose the pressure gradient, and on the other hand, we assume that the domain has a regular curved boundary. The last point implies that a simple approximation of the boundary using piecewise segment lines dramatically reduces the scheme accuracy to at most a second-order one. We propose a new and simple technology which enables to restore the full scheme accuracy based on a specific polynomial reconstruction only using the Gauss points of the curved boundary and does not require any geometrical transformation.*

1 INTRODUCTION

The Stokes or Navier-Stokes equations represent critical issues in modelling and simulations since they encompass within a lot of applications and accurate numerical simulations turns to be a big challenge to provide approximations. Since the finite element method is still the major technique to tackle the discretisation question, finite volume method has received considerable attention due to its intrinsics qualities: built-in conservative and versatility. We refer to the pioneer book of Patankar [16] and the textbook of Ferziger and Peric [7] for an overview of the finite volume for the Navier-Stokes equations.
Very high-order schemes for incompressible fluid flow have been developed using the finite difference framework with the Padé methodology (the so-called compact scheme) [11] on staggered structured grids (see [12][9] and references herein) providing fourth-order or sixth-order approximations [3]. Finite element [10][8] and discontinuous Galerkin methods [13][6][14] also received important contributions to achieve very high-order approximation both in time and space. Sixth-order finite volume approximation is the current state-of-the-art on structured using compact schemes but the unstructured mesh case is still confidential and remains a important issue.
When dealing with very-high order scheme, a crucial point is the evaluation of boundary conditions when the domain is curved. Finite elements or Discontinuous Garlekin methods used isoparametric elements which turns the implementation very complex while the finite volume approach is simpler. Two techniques have been proposed in the convection diffusion context. The first one directly used the Gauss points on the curved boundary to achieve polynomial reconstructions [15] whereas the second one used the Gauss points on the segment for the polynomial reconstruction but adjust a free parameter to reproduce the Dirichlet condition at the Gauss points of the curve [4].
We present a finite volume scheme to provide a sixth-order approximation of the solution of the Stokes problem involving curved boundary. We use a staggered discretization with a primal unstructured mesh for the pressure and the associated diamond mesh for the velocity to avoid the Rhie-Chow interpolation [17][20]. The coupled velocity-pressure approach is employed to avoid the pressure correction intermediate step to provide the divergence-free velocity [9]. Moreover, we do not treat the steady-state as the asymptotic limit of an artificial time marching problem but we directly solve the linear system associated to the saddle point problem. The main difficulty is to achieve an efficient approximation of the solution taking into account the divergence-free velocity constraint to determine the pressure. The method is based, on the one hand, in different kinds of polynomial reconstructions to compute the viscous flux, the pressure gradient, the velocity divergence up to a sixth-order of accuracy and, on the other hand, in a matrix-free formulation using the residual method as in [5][4] solve with the algebraic solver GMRES [19][18]. We detail in the paper the specific treatment of Dirichlet conditions with curved boundary. Indeed, a straightforward approximation using the polygonal domain will lead to a strong degradation of the order (at most second-order) and correction of the

traditional reconstruction will be implemented to recover the optimal order. We design a new local reconstruction involving a free parameter to be fixed such that the Dirichlet conditions on the curved boundary are satisfied. Numerical tests are carried out to prove the efficiency of the method.
The paper is organized as follow. Section 2 presents the generic finite volume scheme for very high order approximation in the context of the Stokes equations. In the third section we tackle the question of the polynomial reconstructions while section four is dedicated to the specific case of the curved boundary. In section 5, we present the numerical results and end the paper with a short conclusion.

2 FINITE VOLUME SCHEME FOR THE STOKES EQUATIONS

Let Ω be an open bounded domain of $\mathbb{R}^2$ with boundary $\partial\Omega$ and $x = (x_1, x_2)$. We seek functions $U = (U_1, U_2) \equiv (U_1(x), U_2(x))$, the velocity field, and $P \equiv P(x)$, the pressure, solutions of the steady-state flow of an incompressible Newtonian fluid governed by the Stokes equations

$$\nabla \cdot (-\mu \nabla U + P I_2) = f, \qquad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot U = 0, \qquad \text{in } \Omega, \tag{2}$$

where the dynamic viscosity $\mu \equiv \mu(x)$ and the source term $f = (f_1, f_2) \equiv (f_1(x), f_2(x))$ are given regular functions. The tensor ∇U is defined as $[\nabla U]_{\alpha\beta} = \frac{\partial U_\alpha}{\partial x_\beta}$, $\alpha, \beta = 1, 2$, and I_2 stands for the identity matrix in $\mathbb{R}^{2\times 2}$. The system (1-2) is completed with the Dirichlet boundary condition

$$U = U_{\mathrm{D}}, \qquad \text{on } \partial\Omega, \tag{3}$$

where $U_{\mathrm{D}} = (U_{1,\mathrm{D}}, U_{2,\mathrm{D}}) \equiv (U_{1,\mathrm{D}}(x), U_{2,\mathrm{D}}(x))$ is a given regular function on $\partial\Omega$ which satisfies the compatibility condition

$$\int_{\partial\Omega} U_{\mathrm{D}} \cdot n \, \mathrm{d}s = 0,$$

with $n = (n_1, n_2)$ the outward unit normal vector on $\partial\Omega$. Moreover, uniqueness for the pressure is guaranteed by the additional constraint $\int_\Omega P \, \mathrm{d}x = 0$.

2.1 Primal and diamond meshes

The primal mesh of Ω, that we denote by $\mathcal{M}$, is a partition of Ω into I non-overlapping convex polygonal cells c_i, $i \in \mathcal{C}_\mathcal{M} = \{1, \ldots, I\}$, and adopt the notations we detail hereafter (see Fig. 1, left):

- for any cell c_i, $i \in \mathcal{C}_\mathcal{M}$, we denote by ∂c_i its boundary and by $|c_i|$ its area; the reference cell point is denoted by m_i which can be any point in c_i (in the present work we shall consider the centroid);

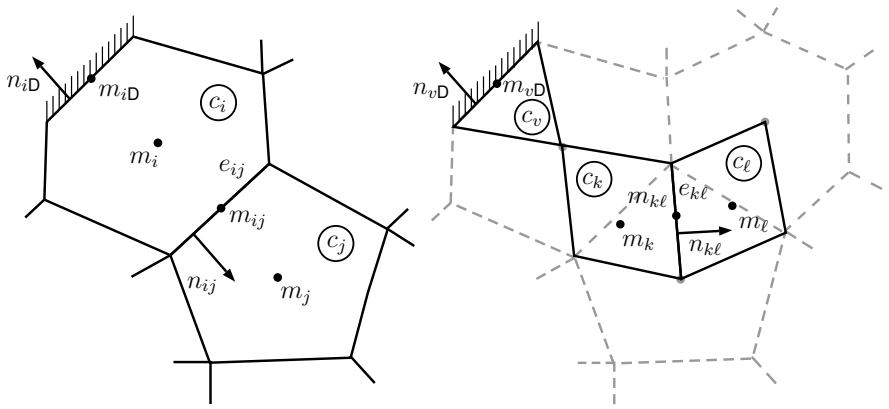


Figure 1: Notation for the primal mesh (left) and for the diamond mesh (right).

- two cells c_i and c_j share a common edge e_{ij} whose length is denoted by $|e_{ij}|$ and $n_{ij} = (n_{1,ij}, n_{2,ij})$ is the unit normal vector to e_{ij} outward to c_i, *i.e.* $n_{ij} = -n_{ji}$; the reference edge point is m_{ij} which can be any point in e_{ij} (in the present work we consider the midpoint); if an edge of c_i belongs to the boundary, the index j is tagged by the letter D;

- for any cell c_i, $i \in \mathcal{C}_\mathcal{M}$, we associate the index set $\nu(i) \subset \{1, \cdots, I\} \cup \{\mathrm{D}\}$ such that $j \in \nu(i)$ if e_{ij} is a common edge of cells c_i and c_j or with the boundary if $j = \mathrm{D}$.

The diamond mesh of Ω, that we denote by $\mathcal{D}$, derives from the primal mesh $\mathcal{M}$ and is constituted of K non-overlapping diamond-shape cell (which degenerate to triangles in the boundary) c_k, $k \in \mathcal{C}_\mathcal{D} = \{I+1, \ldots, I+K\}$. Indeed, for each inner primal edge e_{ij} corresponds a unique cell of the diamond mesh defined by the reference points m_i and m_j and the vertices of the edge (the dual cell associated to a boundary edge $e_{i\mathrm{D}}$ is defined by the reference point m_i and the vertices of the edge).
The notation for the diamond mesh follows the notation introduced for the primal mesh where we substitute the index $i \in \mathcal{C}_\mathcal{M}$ by $k \in \mathcal{C}_\mathcal{D}$ and the index $j \in \nu(i)$ by $\ell \in \nu(k)$ (see Fig. 1, right). In particular m_k is any point in c_k (in the present work we shall consider the centroid) and $m_{k\ell}$ is any point in $e_{k\ell}$ (in the present work we consider the midpoint).
To define the association between diamond cells and primal edges, we introduce the correspondence operator $\Pi_\mathcal{D}$ such that for given arguments (i,j), $i \in \mathcal{C}_\mathcal{M}$, $j \in \nu(i)$, we associate the corresponding diamond cell index $k = \Pi_\mathcal{D}(i,j) \in \mathcal{C}_\mathcal{D}$. In the same way, for each diamond edge, we introduce the correspondence operator $\Pi_\mathcal{M}$ such that for given arguments (k,ℓ), $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, we associate the corresponding primal cell index $i = \Pi_\mathcal{M}(k,\ell) \in \mathcal{C}_\mathcal{M}$.
The numerical integrations on the edges are performed with Gaussian quadrature where for the primal edges e_{ij}, $i \in \mathcal{C}_\mathcal{M}$, $j \in \nu(i)$, we denote by $q_{ij,r}$, $r = 1, \ldots, R$, their Gauss points and for the diamond edges $e_{k\ell}$, $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, we denote by $q_{k\ell,r}$, $r = 1, \ldots, R$,

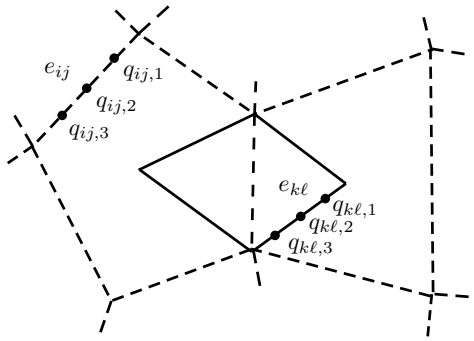


Figure 2: Gauss points on the edges of the primal cells (dashes lines) and on the edges of the diamond cells (solid lines).

their Gauss points, both sets with weights ζ_r, $r = 1, \ldots, R$ (see Fig. 2).

2.2 Generic finite volume scheme

To provide the generic very high-order finite volume scheme, we first integrate equation (1) over each diamond cell c_k, $k \in \mathcal{C}_\mathcal{D}$, and then apply the divergence theorem, yielding

$$\int_{\partial c_k} (-\mu \nabla U + P I_2)\, n \, \mathrm{d}s = \int_{c_k} f \, \mathrm{d}x,$$

which can be rewritten in the scalar form as

$$\int_{\partial c_k} (-\mu \nabla U_\beta \cdot n + P n_\beta) \, \mathrm{d}s = \int_{c_k} f_\beta \, \mathrm{d}x, \ \beta = 1, 2.$$

Considering the Gaussian quadrature with R points, *i.e.* of order $2R$, for the line integrals, we get the residual expression

$$\sum_{\ell \in \nu(k)} \frac{|e_{k\ell}|}{|c_k|} \left[\sum_{r=1}^{R} \zeta_r \left(\mathbb{F}^U_{\beta,k\ell,r} + \mathbb{F}^P_{\beta,k\ell,r} \right) \right] - f_{\beta,k} = \mathcal{O}\left(h_k^{2R} \right), \ \beta = 1, 2, \tag{4}$$

with the physical fluxes given by

$$\mathbb{F}^U_{\beta,k\ell,r} = -\mu(q_{k\ell,r}) \nabla U_\beta(q_{k\ell,r}) \cdot n_{k\ell}, \quad \mathbb{F}^P_{\beta,k\ell,r} = P(q_{k\ell,r}) n_{\beta,k\ell},$$

$h_k = \max_{\ell \in \nu(k)} |e_{k\ell}|$, and $f_{\beta,k}$ an approximation of order $2R$ of the mean value of f_β over cell c_k (if cell c_k is not triangular, we split it into sub-triangles which share the cell centroid as a common vertex and apply the quadrature rule on each sub-triangle as in [**?**]).
We now integrate equation (2) over each primal cell c_i and apply again the divergence theorem, yielding

$$\int_{\partial c_i} U \cdot n \, \mathrm{d}s = 0.$$

Considering again Gaussian quadrature with R points for the line integrals, we get the residual expression

$$\sum_{j\in\nu(i)} \frac{|e_{ij}|}{|c_i|} \sum_{r=1}^{R} \zeta_r \, \mathbb{F}^{\nabla}_{ij,r} = O(h_i^{2R}), \tag{5}$$

with the physical flux given by

$$\mathbb{F}^{\nabla}_{ij,r} = U(q_{ij,r}) \cdot n_{ij}$$

and $h_i = \max_{j\in\nu(i)} |e_{ij}|$.

3 POLYNOMIAL RECONSTRUCTIONS

The polynomial reconstruction is a powerful tool to provide an accurate local representation of the underlying solution and was initially introduced in [1, 2] for hyperbolic problems. In [5] a new methodology was proposed in the context of convection-diffusion problems in order to achieve very accurate approximations of the gradient fluxes and to take into account the boundary conditions. The authors introduced different types of polynomial reconstructions namely the conservative reconstruction in cells and on Dirichlet boundary edges and the non-conservative reconstruction on inner edges, in order to compute approximations of the convective and the diffusive fluxes. We now adapt this technology for the specific Stokes problem where the main difficulty is to handle the two meshes.

3.1 Stencil and data

A stencil is a collection of cells situated in the vicinity of a reference geometrical entity, namely an edge or a cell where the number of elements of the stencil shall depend on the degree d of the polynomial function we intend to construct. For each diamond edge $e_{k\ell}$, $k \in \mathcal{C}_{\mathcal{D}}$, $\ell \in \nu(k)$, we associate the stencil $S_{k\ell}$ consisting of the indices of neighbor diamond cells. Analogously, we associate the stencil S_k for each diamond cell c_k, $k \in \mathcal{C}_{\mathcal{D}}$, and the stencil S_i for each primal cell c_i, $i \in \mathcal{C}_{\mathcal{M}}$, consisting of the indices of neighbor dual and primal cells, respectively. Remark. *A polynomial reconstruction of degree d requires $n_d = (d+1)(d+2)/2$ coefficients. So, in practice, a stencil consists of the N_d closest cells to each geometrical entity (edge or cell) in the respective mesh, with $N_d \geq n_d$ (we consider $N_d \approx 1.5 n_d$ for the sake of robustness).*

Now, we want to compute the polynomial reconstructions based on the data of the associated stencil. To this end, we assume that vectors $\mathbb{U}_1 = (U_{1,k})_{k=I+1,\dots,I+K}$, $\mathbb{U}_2 = (U_{2,k})_{k=I+1,\dots,I+K}$, and $\mathbb{P} = (P_i)_{i=1,\dots,I}$ gather the approximations of the mean values of U_1 and U_2 over the diamond cells and P over the primal cells, *i.e.*

$$U_{1,k} \approx \frac{1}{|c_k|}\int_{c_k} U_1 \, \mathrm{d}x, \quad U_{2,k} \approx \frac{1}{|c_k|}\int_{c_k} U_2 \, \mathrm{d}x, \quad P_i \approx \frac{1}{|c_i|}\int_{c_i} P \, \mathrm{d}x.$$

3.2 Conservative reconstruction for primal cells

For each primal cell c_i, $i \in \mathcal{C}_\mathcal{M}$, the local polynomial approximation of the underlying solution P based on vector $\mathbb{P}$ of degree d is defined as

$$\boldsymbol{P}_i(x) = P_i + \sum_{1\leq|\alpha|\leq d} \mathcal{R}_i^\alpha \left[(x-m_i)^\alpha - M_i^\alpha\right],$$

where $\alpha = (\alpha_1, \alpha_2)$ with $|\alpha| = \alpha_1 + \alpha_2$ and the convention $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2}$, vector $\mathcal{R}_i = (\mathcal{R}_i^\alpha)_{1\leq|\alpha|\leq d}$ gathers the polynomial coefficients, and $M_i^\alpha = \frac{1}{|c_i|}\int_{c_i}(x-m_i)^\alpha \,\mathrm{d}x$ in order to guarantee the conservation property

$$\frac{1}{|c_i|}\int_{c_i} \boldsymbol{P}_i(x)\,\mathrm{d}x = P_i.$$

For a given stencil S_i, we consider the quadratic functional

$$E_i(\mathcal{R}_i) = \sum_{q\in S_i}\left[\frac{1}{|c_q|}\int_{c_q}\boldsymbol{P}_i(x)\,\mathrm{d}x - P_q\right]^2. \tag{6}$$

We denote by $\widehat{\mathcal{R}}_i$ the unique vector which minimizes the quadratic functional (6) and we set $\widehat{\boldsymbol{P}}_i(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

3.3 Conservative reconstruction for diamond cells

For each diamond cell c_k, $k \in \mathcal{C}_\mathcal{D}$, the local polynomial approximation of the underlying functions U_1 and U_2 based on vectors $\mathbb{U}_1$ and $\mathbb{U}_2$ of degree d are defined as

$$\boldsymbol{U}_{\beta,k}(x) = U_{\beta,k} + \sum_{1\leq|\alpha|\leq d} \mathcal{R}_{\beta,k}^\alpha \left[(x-m_k)^\alpha - M_k^\alpha\right], \quad \beta = 1,2,$$

where vector $\mathcal{R}_{\beta,k} = (\mathcal{R}_{\beta,k}^\alpha)_{1\leq|\alpha|\leq d}$ gathers the polynomial coefficients and $M_k^\alpha = \frac{1}{|c_k|}\int_{c_k}(x-m_k)^\alpha \,\mathrm{d}x$ in order to guarantee the conservation property

$$\frac{1}{|c_k|}\int_{c_k} \boldsymbol{U}_{\beta,k}(x)\,\mathrm{d}x = U_{\beta,k}.$$

For a given stencil S_k, we consider the quadratic functional

$$E_{\beta,k}(\mathcal{R}_{\beta,k}) = \sum_{q\in S_k}\left[\frac{1}{|c_q|}\int_{c_q}\boldsymbol{U}_{\beta,k}(x)\,\mathrm{d}x - U_{\beta,q}\right]^2. \tag{7}$$

We denote by $\widehat{\mathcal{R}}_{\beta,k}$ the unique vector which minimizes the quadratic functional (7) and we set $\widehat{\boldsymbol{U}}_{\beta,k}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

3.4 Non-conservative reconstruction for inner diamond edges

For each inner diamond edge $e_{k\ell}$, $k \in \mathcal{C}_{\mathcal{D}}$, $\ell \in \nu(k)$, the local polynomial approximations of degree d of the underlying functions U_1 and U_2 are defined as

$$\boldsymbol{U}_{\beta,k\ell}(x) = \sum_{0\leq|\alpha|\leq d} \mathcal{R}^{\alpha}_{\beta,k\ell}(x - m_{k\ell})^{\alpha}, \quad \beta = 1, 2,$$

where vector $\mathcal{R}_{\beta,k\ell} = (\mathcal{R}^{\alpha}_{\beta,k\ell})_{0\leq|\alpha|\leq d}$ gathers the polynomial coefficients (notice that in this case $|\alpha|$ starts with 0 since no conservation property is required). For a given stencil $S_{k\ell}$ with $\#S_{k\ell}$ elements and vector $\omega_{\beta,k\ell} = (\omega_{\beta,k\ell,q})_{q=1,\ldots,\#S_{k\ell}}$ of the positive weights of the reconstruction, we consider the quadratic functional

$$E_{\beta,k\ell}(\mathcal{R}_{\beta,k\ell}) = \sum_{q\in S_{k\ell}} \omega_{\beta,k\ell,q} \left[\frac{1}{|c_q|}\int_{c_q} \boldsymbol{U}_{\beta,k\ell}(x)\,\mathrm{d}x - U_{\beta,q}\right]^2. \tag{8}$$

We denote by $\widetilde{\mathcal{R}}_{\beta,k\ell}$ the unique vector which minimizes the quadratic functional (8) and we set $\widetilde{\boldsymbol{U}}_{\beta,k\ell}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

3.5 Conservative reconstruction for diamond boundary edges

We treat the boundary diamond edges in a particular way in order to take into account the Dirichlet boundary conditions prescribed for the velocity. For each boundary diamond edge $e_{k\mathrm{D}}$, $k \in \mathcal{C}_{\mathcal{D}}$, the local polynomial approximations of degree d of the underlying functions U_1 and U_2 are defined as

$$\boldsymbol{U}_{\beta,k\mathrm{D}}(x) = U_{\beta,k\mathrm{D}} + \sum_{1\leq|\alpha|\leq d} \mathcal{R}^{\alpha}_{\beta,k\mathrm{D}}\left[(x - m_{k\mathrm{D}})^{\alpha} - M^{\alpha}_{k\mathrm{D}}\right], \quad \beta = 1, 2,$$

where vector $R_{\beta,k\mathrm{D}} = (R^{\alpha}_{\beta,k\mathrm{D}})_{1\leq|\alpha|\leq d}$ gathers the polynomial coefficients, $U_{\beta,k\mathrm{D}}$ is an approximation of the mean value $U_{\beta,\mathrm{D}}$ of order $2R$ over the diamond boundary edge $e_{k\mathrm{D}}$, and $M^{\alpha}_{k\mathrm{D}} = \frac{1}{|e_{k\mathrm{D}}|}\int_{e_{k\mathrm{D}}}(x - m_{k\mathrm{D}})^{\alpha}\,\mathrm{d}x$ in order to guarantee the conservation property

$$\frac{1}{|e_{k\mathrm{D}}|}\int_{e_{k\mathrm{D}}} \boldsymbol{U}_{\beta,k\mathrm{D}}(x)\,\mathrm{d}s = U_{\beta,k\mathrm{D}}.$$

For a given stencil $S_{k\mathrm{D}}$ with $\#S_{k\mathrm{D}}$ elements and vector $\omega_{\beta,k\mathrm{D}} = (\omega_{\beta,k\mathrm{D},q})_{q=1,\ldots,\#S_{k\mathrm{D}}}$ of the positive weights of the reconstruction, we consider the quadratic functional

$$E_{\beta,k\mathrm{D}}(\mathcal{R}_{\beta,k\mathrm{D}}) = \sum_{q\in S_{k\mathrm{D}}} \omega_{\beta,k\mathrm{D},q} \left[\frac{1}{|c_q|}\int_{c_q} \boldsymbol{U}_{\beta,k\mathrm{D}}(x)\,\mathrm{d}x - U_{\beta,q}\right]^2. \tag{9}$$

We denote by $\widehat{\mathcal{R}}_{\beta,k\mathrm{D}}$ the unique vector which minimizes the quadratic functional (9) and we set $\widehat{\boldsymbol{U}}_{\beta,k\mathrm{D}}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

Remark *The motivation for introducing the weights in the case of a non-conservative polynomial reconstruction and in the case of a conservative polynomial reconstruction for Dirichlet boundary edges, is presented in [5] as well as the importance to set larger values for the adjacent cells.*

3.6 High-order finite volume scheme

This subsection is dedicated to design high-order numerical flux approximations based on the polynomial reconstructions presented in the previous subsections to provide the global residual operator.

3.6.1 Numerical fluxes

For a given polynomial degree d and the associated stencils which guarantee the d-consistency property (see [5]), four numerical fluxes situations arise:

- for an inner diamond edge $e_{k\ell}$, the fluxes at the quadrature point $q_{k\ell,r}$ write

$$\mathcal{F}^U_{\beta,k\ell,r} = -\mu(q_{k\ell,r})\nabla\widetilde{\boldsymbol{U}}_{\beta,k\ell}(q_{k\ell,r})\cdot n_{k\ell} \quad \text{and} \quad \mathcal{F}^P_{\beta,k\ell,r} = \widehat{\boldsymbol{P}}_i(q_{k\ell,r})n_{\beta,k\ell}, \quad \beta = 1,2,$$

  with the correspondence $i = \Pi_{\mathcal{M}}(k,\ell)$;

- for a boundary diamond edge $e_{k\mathrm{D}}$, the fluxes at the quadrature point $q_{k\mathrm{D},r}$ write

$$\mathcal{F}^U_{\beta,k\mathrm{D},r} = -\mu(q_{k\mathrm{D},r})\nabla\widehat{\boldsymbol{U}}_{\beta,k\mathrm{D}}(q_{k\mathrm{D},r})\cdot n_{k\mathrm{D}} \quad \text{and} \quad \mathcal{F}^P_{\beta,k\mathrm{D},r} = \widehat{\boldsymbol{P}}_i(q_{k\mathrm{D},r})n_{\beta,k\mathrm{D}}, \quad \beta = 1,2,$$

  with the correspondence $i = \Pi_{\mathcal{M}}(k,\mathrm{D})$;

- for an inner primal edge e_{ij}, the flux at the quadrature point $q_{ij,r}$ writes

$$\mathcal{F}^{\nabla}_{ij,r} = \widehat{\boldsymbol{U}}_{1,k}(q_{ij,r})n_{1,ij} + \widehat{\boldsymbol{U}}_{2,k}(q_{ij,r})n_{2,ij},$$

  with the correspondence $k = \Pi_{\mathcal{D}}(i,j)$;

- for a boundary primal edge $e_{i\mathrm{D}}$, the flux at the quadrature point $q_{i\mathrm{D},r}$ writes

$$\mathcal{F}^{\nabla}_{i\mathrm{D},r} = \widehat{\boldsymbol{U}}_{1,k\mathrm{D}}(q_{i\mathrm{D},r})n_{1,i\mathrm{D}} + \widehat{\boldsymbol{U}}_{2,k\mathrm{D}}(q_{i\mathrm{D},r})n_{2,i\mathrm{D}},$$

  with the correspondence $k = \Pi_{\mathcal{D}}(i,\mathrm{D})$.

3.6.2 Residual operators

For any vector $\Phi = (\mathbb{U}_1, \mathbb{U}_2, \mathbb{P})$ in $\mathbb{R}^{2K+I}$, we define the residual operators for each diamond cell c_k, $k \in \mathcal{C}_D$, as

$$\mathcal{G}^{\beta}_k(\Phi) = \sum_{\ell\in\nu(k)} \frac{|e_{k\ell}|}{|c_k|}\left[\sum_{r=1}^{R}\zeta_r\left(\mathcal{F}^U_{\beta,k\ell,r} + \mathcal{F}^P_{\beta,k\ell,r}\right)\right] - f_{\beta,k}, \quad \beta = 1,2,$$

and for each primal cell c_i, $i \in \mathcal{C}_\mathcal{P}$, as

$$\mathcal{G}_i^\nabla(\Phi) = \sum_{j\in\nu(i)} \frac{|e_{ij}|}{|c_i|} \sum_{r=1}^{R} \zeta_r \mathcal{F}_{ij,r}^\nabla,$$

which correspond to the finite volume scheme (4-5) cast in residual form. Gathering all the components of the residuals in vectors $\mathcal{G}^\beta(\Phi) = \big(\mathcal{G}_k^\beta(\Phi)\big)_{k=I+1,\dots,I+K}$ and $\mathcal{G}^\nabla(\Phi) = \big(\mathcal{G}_i^\nabla(\Phi)\big)_{i=1,\dots,I}$, we introduce the global affine operator from $\mathbb{R}^{2K+I}$ into $\mathbb{R}^{2K+I}$, given by

$$\mathcal{H}(\Phi) = \big(\mathcal{G}^1(\Phi), \mathcal{G}^2(\Phi), \mathcal{G}^\nabla(\Phi)\big)^T,$$

such that vector $\Phi^\star = (\mathbb{U}_1^\star, \mathbb{U}_2^\star, \mathbb{P}^\star)^T \in \mathbb{R}^{2K+I}$, solution of the problem $\mathcal{H}(\Phi) = 0$, provides a constant piecewise approximation of U_1, U_2, and P.

4 DIRICHLET CONDITION ON CURVED BOUNDARY

When dealing with curved boundaries, the substitution of domain Ω with the polygonal domain deriving from the mesh dramatically reduces the method order due to a coarse approximation of the boundary by line segments. A specific treatment of the polynomial reconstruction associated to the edge on the boundary is required.

4.1 Second-order approach

Let e_{kD} be an edge of the dual mesh situated on the boundary Γ_D and set the mean value $\bar{U}_{\beta,kD} = \dfrac{1}{|e_{kD}|}\displaystyle\int_{e_{kD}} U_{\beta,D}(s)\, ds$ with $\beta = 1, 2$ the two components of the velocity. In [5] the following conservative polynomial function of degree d has been considered

$$\widehat{U}_{\beta,kD}(x;d) = U_{\beta,kD} + \sum_{1\le|\alpha|\le d} \mathfrak{R}_{\beta,kD}^{d,\alpha}\Big\{(x - m_{kD})^\alpha - M_{kD}^\alpha\Big\}, \tag{10}$$

taking $U_{\beta,kD} = \bar{U}_{\beta,kD}$, m_{kD} the midpoint of edge e_{kD} and $M_{kD}^\alpha = \dfrac{1}{|e_{kD}|}\displaystyle\int_{e_{kD}} (x - m_{kD})^\alpha\, ds$ such that the conservative property $\dfrac{1}{|e_{kD}|}\displaystyle\int_{e_{kD}} \widehat{U}_{\beta,kD}(x;d)\, ds = U_{\beta,kD}$ holds. To fix the coefficients, we introduce the functional

$$E_{\beta,kD}(\mathfrak{R}_{\beta,kD}^d; d) = \sum_{\ell\in S(e_{kD},d)} \omega_{\beta,kD,\ell}\Big[\frac{1}{|c_\ell|}\int_{c_\ell} \widehat{U}_{\beta,kD}(x;d)\, dx - U_{\beta,\ell}\Big]^2, \tag{11}$$

where $\omega_{\beta,kD,\ell}$ are positive weights and vector $\widehat{\mathfrak{R}}_{\beta,kD}^d$ stands for the unique vector minimizing the functional which provides the best approximation. Such an approach gives, at most, a second-order approximation since we substitute the mean value on a curved arc by the mean value on edge e_{kD} leading to a loose of accuracy. The keypoint is to evaluate the polynomial reconstruction with a better $U_{\beta,kD}$ choice, different to the candidate $\bar{U}_{\beta,kD}$, in order to provide better approximations of $\widehat{U}_{\beta,kD}(x;d)$ at the Gauss points $m_{kD,r}$, $r = 1, \cdots, R_2$.

4.2 High-order approximation

A local parametrization is introduced and a new quadrature formula is used to perform accurate numerical integration on the curved arc. Let e be a generic boundary edge, we denote by v_1 e v_2 the vertexes and denote $e = v_1v_2$ the segment with lenght $|v_1v_2|$ while $\widehat{v_1v_2}$ represents the boundary arc between v_1 and v_2 with length $|\widehat{v_1v_2}|$. We introduce the edge parametrization $q(t) = (1-t)v_1 + tv_2$, $t \in [0,1]$ which satisfies $|q'(t)| = |v_1v_2|$ while $p(t)$ is a parametrization of the arc such that

$$p(0) = v_1,\ p(1) = v_2,\ |p'(t)| = |\widehat{v_1v_2}| \text{ is constant.}$$

Let us denote by q_1, ..., q_{R_2} the Gauss points on edge e associated to parameters t_1, ..., t_{R_2}, then $p_1 = p(t_1)$, ..., $p_{R_2} = p(t_{R_2})$ are the corresponding Gauss points on the boundary arc. Indeed, using the quadrature rule for the numerical integration over the arc, one has

$$\int_{\widehat{v_1v_2}} U_{\beta,D}(p)dp = \int_0^1 U_{\beta,D}(p(t))|p'(t)|dt \approx \sum_{r=1}^{R_2} \xi_r U_{\beta,D}(p_r)|p'(t_r)| = |\widehat{v_1v_2}| \sum_{r=1}^{R_2} \xi_r U_{\beta,D}(p_r).$$

Notice that the following property then holds for $r = 1, \cdots, R_2$

$$\frac{|\widehat{v_1p_r}|}{|v_1q_r|} = \frac{|\widehat{v_2p_r}|}{|v_2q_r|} = \frac{|\widehat{v_1v_2}|}{|v_1v_2|}.$$

Let us now consider $\widehat{e}_{kD}$ the boundary arc associated to the edge e_{kD} and assume that the mean value approximations U_i, are given on the stencil. We consider the following linear operator

$$U_{\beta,kD} \to \widehat{U}_{\beta,kD}(x; d, U_{\beta,kD}) \in \mathbb{P}_d$$

where $U_{\beta,kD}$ represents an approximation of the mean value on e_{kD} **and not** on $\widehat{e}_{kD}$ and is seen as a free parameter. The main difficulty is that the Dirichlet condition is defined on $\widehat{e}_{kD}$ **and not** on e_{kD}. To overcome the problem we introduce the functional

$$H(U_{\beta,kD}) = \sum_{r=1}^{R_2} \left(\widehat{U}_{\beta,kD}(p_{kD,r}; d, U_{\beta,kB}) - U_{\beta,D}(p_{kD,r})\right)^2$$

which corresponds to the error at the boundary arc Gauss points between the polynomial approximation and the real Dirichlet condition. Since we are dealing with a quadratic functional, existence and uniqueness of the minimum $U^\star_{\beta,kD}$ is guaranteed and $\widehat{U}_{\beta,kD}$ will be the polynomial reconstruction where we take $U_{\beta,kD} = U^\star_{\beta,kD}$.
To compute $U^\star_{\beta,kD}$ in practice, we propose the following simple algorithm. We consider the sequence $(U^n_{\beta,kD})^n$ initialized with $U^0_{\beta,kD} = \bar{U}_{\beta,kD}$ and given by the following:

1. with $U^n_{\beta,kB}$ in hand, compute the associated polynomial function $\widehat{U}_{\beta,kD}(x; d, U^n_{\beta,kB})$,

2. evaluate the errors $\delta_r^n = U_{\beta,D}(p_{kD,r}) - \widehat{U}_{\beta,kD}^n(p_{kD,r}; d, U_{\beta,kB}^n)$,

3. update the mean value on e_{kD} with $U_{\beta,kD}^{n+1} = U_{\beta,kD}^n + \sum_{r=1}^{R_2} \xi_r \delta_r^n$,

4. stop if $|U_{\beta,kD}^{n+1} - U_{\beta,kD}^n| < \epsilon_B \bar{U}_{\beta,nD}$ where the tolerance ϵ_B has been prescribed and set $U_{\beta,kD}^\star = U_{\beta,kD}^{n+1}$, else goto step (1).

Numerical experiments shows that we quickly converge with two or three steps using $\epsilon_B = 10^{-12}$. Furthermore, when $\bar{U}_{\beta,kD} = 0$, we use the absolute error criterion $|U_{\beta,kD}^{n+1} - U_{\beta,kD}^n| < \epsilon_B$ in place of the relative error criterion. Remark *Notice that the method only requires the arc length $|\widehat{e}_{kD}|$ and the Gauss point $p_{kD,r}$ on the boundary arc. No geometrical tranformation is performed which provides a very simple method, easy to implement.*

5 NUMERICAL RESULTS

To perform the numerical tests, we consider a fluid with viscosity $\mu = 1$ flowing in a circlar domain $\Omega = \{x : \tau^2 < 1\}$ where $\tau^2 = x_1^2 + x_2^2$. In order to check the implementation of the method and assess the convergence rates, we manufacture an analytical solution for the given problem setting

$$U_1(x) = -ay\exp(\tau^2)(1-\tau), \quad U_2(x) = ax\exp(\tau^2)(1-\tau), \quad P(x) = \cos(\pi\tau^2),$$

where $a = 1/\left((1/(2\sqrt{2}))\exp(1/2)\right)$ in order to normalize the velocity. Then, the source terms are computed such that equations (1) and (2) hold. The homogeneous Dirichlet boundary condition $U_{\mathrm{D}}(x) = (0,0)$ prescribed on $\partial\Omega$, derives from de exact solution. Vectors $\mathbb{U}_\beta^\star = (U_{\beta,k}^\star)_{k\in\mathcal{C}_\mathcal{D}}$, $\beta = 1,2$, and $\mathbb{P}^\star = (P_i^\star)_{i\in\mathcal{C}_\mathcal{M}}$ gather the approximate mean values while vectors $\overline{\mathbb{U}}_\beta = (\overline{U}_{\beta,k})_{k\in\mathcal{C}_\mathcal{D}}$, $\beta = 1,2$, and $\overline{\mathbb{P}} = (\overline{P}_i)_{i\in\mathcal{C}_\mathcal{M}}$ gather the exact mean values of the solution given by

$$\overline{U}_{\beta,k} = \frac{1}{|c_k|}\int_{c_k} U_\beta \,\mathrm{d}x, \beta = 1,2, \quad \text{and} \quad \overline{P}_i = \frac{1}{|c_i|}\int_{c_i} P \,\mathrm{d}x.$$

The L^1-norm errors are given by

$$E_1^\beta(\mathcal{D}) = \frac{\sum_{k\in\mathcal{C}_\mathcal{D}} |U_{\beta,k}^\star - \overline{U}_{\beta,k}||c_k|}{\sum_{k\in\mathcal{C}_\mathcal{D}} |c_k|}, \beta = 1,2, \quad \text{and} \quad E_1^P(\mathcal{M}) = \frac{\sum_{i\in\mathcal{C}_\mathcal{M}} |P_i^\star - \overline{P}^\star - \overline{P}_i - \overline{P}||c_i|}{\sum_{i\in\mathcal{C}_\mathcal{M}} |c_i|},$$

and the L^∞-norm errors are given by

$$E_\infty^\beta(\mathcal{D}) = \max_{k\in\mathcal{C}_\mathcal{D}} |U_{\beta,k}^\star - \overline{U}_{\beta,k}|, \beta = 1,2, \quad \text{and} \quad E_\infty^P(\mathcal{M}) = \max_{i\in\mathcal{C}_\mathcal{M}} |P_i^\star - \overline{P}^\star - \overline{P}_i - \overline{P}|.$$

where $\overline{P^\star}$ is the mean value of the values gather in vector $\mathbb{P}^\star$ and $\overline{P}$ is the mean value of the values gather in vector $\overline{\mathbb{P}}$, given by

$$\overline{P^\star} = \frac{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} P_i^\star |c_i|}{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} |c_i|}, \qquad \overline{P} = \frac{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} \overline{P}_i |c_i|}{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} |c_i|},$$

respectively, to guarantee a solution with null mean value pressure.
We evaluate the convergence rate of the L^1-norm (and L^∞-norm error) between two different and successive finer primal meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, with I_1 and I_2 cells, respectively, as

$$O_1^P(\mathcal{M}_1,\mathcal{M}_2) = 2\frac{|\log(E_1^P(\mathcal{M}_1)/E_1^P(\mathcal{M}_2))|}{|\log(I_1/I_2)|}.$$

In the same way, we define the convergence order between two different and successive finer diamond meshes $\mathcal{D}_1$ and $\mathcal{D}_2$ with K_1 and K_2 cells, respectively, as

$$O_1^\beta(\mathcal{D}_1,\mathcal{D}_2) = 2\frac{|\log(E_1^\beta(\mathcal{D}_1)/E_1^\beta(\mathcal{D}_2))|}{|\log(K_1/K_2)|}.$$

In all the simulations we have carried out, the weights in functional (8) are set $\omega_{\beta,k\ell,q} = 3$, $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, $q \in S_{k\ell}$, $\beta = 1,2$, if $e_{k\ell}$ is an edge of c_q and $\omega_{\beta,k\ell,q} = 1$, otherwise, following [5].

The second-order case As a first test, we consider the classical polynomial reconstruction on the boundary given in Section 4.1 using the straithforward method computing the Dirichlet condition on the edge. We carry out simulations with successive finer regular triangular Delaunay meshes and the associated diamond meshes and report in Tables 1, 2, 3 the L^1- and L^∞-norm errors and the convergence rates using the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively. Notice that the number of unknowns (the same as degrees of freedom) is $DOF = K$ for U_1 and U_2 and $DOF = I$ for P. The notation E_1 is a generalization which stands for E_1^β or E_1^P depending on the variable we are dealing with (U_β or P, respectively). The same convention is valid for E_∞, O_1, and O_∞.
The $\mathbb{P}_1$ polynomial reconstruction provides a second-order approximation for the velocity and a first-order approximation for the pressure, as expected. The schemes based on the $\mathbb{P}_3$ and $\mathbb{P}_5$ polynomial reconstructions, also provides second-order convergence rates for the velocity since the reconstruction on dual boundary edges is of second-order. Inspite of this negative results, the scheme achieves a third-order covergence for the pressure with $\mathbb{P}_3$ and $\mathbb{P}_5$ polynomial reconstructions. Such situation arises since the null-velocity boundary condition guarantees that the divergence on $e_{k\mathrm{D}}$, $k \in \mathcal{C}_\mathcal{M}$ is very close to zero, the exact flux.

Table 1: Errors and convergence rates with $\mathbb{P}_1$ polynomial reconstructions where the polynomials on boundary edges are computed according to Section 4.1.

	DOF	E_1	O_1	E_2	O_2	E_∞	O_∞
U_1	1337	1.66E−02	—	2.03E−02	—	4.77E−02	—
	2486	8.83E−03	2.03	1.06E−02	2.08	2.97E−02	1.53
	9835	2.30E−03	1.96	2.80E−03	1.94	8.17E−03	1.88
	21337	1.02E−03	2.10	1.24E−03	2.10	3.70E−03	2.05
U_2	1337	1.67E−02	—	2.03E−02	—	4.73E−02	—
	2486	8.79E−03	2.07	1.07E−02	2.08	2.97E−02	1.50
	9835	2.30E−03	1.95	2.80E−03	1.95	8.44E−03	1.83
	21337	1.02E−03	2.11	1.24E−03	2.10	3.68E−03	2.14
P	870	7.89E−02	—	1.42E−01	—	6.21E−01	—
	1628	5.42E−02	1.20	9.83E−02	1.17	4.35E−01	1.13
	6498	2.21E−02	1.30	4.03E−02	1.29	2.48E−01	0.81
	14138	1.47E−02	1.06	2.58E−02	1.14	1.55E−01	1.20

Table 2: Errors and convergence rates with $\mathbb{P}_1$ polynomial reconstructions where the polynomials on boundary edges are computed according to Section 4.1.

	DOF	E_1	O_1	E_2	O_2	E_∞	O_∞
U_1	1337	3.07E−03	—	3.63E−03	—	7.41E−03	—
	2486	1.65E−03	2.00	1.95E−03	2.01	3.90E−03	2.07
	9835	4.18E−04	2.00	4.93E−04	2.00	9.85E−04	2.00
	21337	1.92E−04	2.00	2.27E−04	2.01	4.53E−04	2.01
U_2	1337	3.08E−03	—	3.63E−03	—	7.41E−03	—
	2486	1.65E−03	2.01	1.95E−03	2.01	3.90E−03	2.07
	9835	4.19E−04	1.99	4.93E−04	2.00	9.87E−04	2.00
	21337	1.92E−04	2.01	2.27E−04	2.01	4.53E−04	2.01
P	870	1.41E−03	—	2.23E−03	—	9.77E−03	—
	1628	5.08E−04	3.26	8.19E−04	3.20	3.74E−03	3.07
	6498	6.13E−05	3.06	1.01E−04	3.03	5.50E−04	2.77
	14138	1.87E−05	3.05	3.11E−05	3.03	2.29E−04	2.25

The high-order case We now consider the high-oder polynomial polynomial reconstruction on the boundary using the correction given in Section 4.2. We carry out simulations with successive finer regular meshes.

We report in Tables 4, 5, and 6 the L^1- and L^∞-norm errors and the convergence rates using the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively.

As in the previous case, the $\mathbb{P}_1$ polynomial reconstruction provides a second-order approximation for the velocity and a first-order approximation for the pressure. The scheme based on the $\mathbb{P}_3$ reconstruction achieves an effective fourth-order approximation for the

Table 3: Errors and convergence rates with $\mathbb{P}_5$ polynomial reconstructions where the polynomials on boundary edges are computed according to Section 4.1.

	DOF	E_1	O_1	E_2	O_2	E_∞	O_∞
U_1	1337	3.16E−03	—	3.73E−03	—	7.40E−03	—
	2486	1.68E−03	2.04	1.98E−03	2.05	3.93E−03	2.04
	9835	4.20E−04	2.02	4.95E−04	2.02	9.86E−04	2.01
	21337	1.93E−04	2.01	2.27E−04	2.01	4.53E−04	2.01
U_2	1337	3.16E−03	—	3.73E−03	—	7.40E−03	—
	2486	1.68E−03	2.04	1.98E−03	2.05	3.93E−03	2.04
	9835	4.20E−04	2.02	4.95E−04	2.02	9.86E−04	2.01
	21337	1.93E−04	2.01	2.27E−04	2.01	4.53E−04	2.01
P	870	3.23E−05	—	5.09E−05	—	2.54E−04	—
	1628	7.28E−06	4.76	1.23E−05	4.53	5.57E−05	4.84
	6498	3.15E−07	4.54	1.04E−06	3.57	2.56E−05	1.12
	14138	7.06E−08	3.85	3.33E−07	2.93	9.07E−06	2.67

Table 4: Errors and convergence rates with $\mathbb{P}_1$ polynomial reconstructions where the polynomials on boundary edges are computed according to Section 4.2.

	DOF	E_1	O_1	E_2	O_2	E_∞	O_∞
U_1	1337	2.01E−02	—	2.44E−02	—	5.50E−02	—
	2486	1.07E−02	2.03	1.28E−02	2.07	3.40E−02	1.55
	9835	2.77E−03	1.96	3.35E−03	1.95	9.26E−03	1.89
	21337	1.24E−03	2.08	1.49E−03	2.09	4.19E−03	2.05
U_2	1337	2.03E−02	—	2.44E−02	—	5.52E−02	—
	2486	1.07E−02	2.08	1.28E−02	2.07	3.40E−02	1.57
	9835	2.77E−03	1.97	3.35E−03	1.95	9.53E−03	1.85
	21337	1.23E−03	2.09	1.49E−03	2.09	4.17E−03	2.13
P	870	7.86E−02	—	1.41E−01	—	6.20E−01	—
	1628	5.42E−02	1.19	9.82E−02	1.15	4.36E−01	1.13
	6498	2.21E−02	1.30	4.03E−02	1.29	2.47E−01	0.82
	14138	1.47E−02	1.06	2.58E−02	1.15	1.55E−01	1.20

velocity and a third-order (slightly better) approximation for the pressure. The scheme based on the $\mathbb{P}_5$ reconstruction achieves a sixth- and fifth-order approximations for the velocity and the pressure, respectively, which demonstrates the effectiveness of the correction. Finally, we also mention that no oscillations or numerical locking are reported in all the experiences.

Table 5: Errors and convergence rates with $\mathbb{P}_3$ polynomial reconstructions where the polynomials on boundary edges are computed according to Section 4.2.

	DOF	E_1	O_1	E_2	O_2	E_∞	O_∞
U_1	1337	1.00E−04	—	1.25E−04	—	4.10E−04	—
	2486	2.69E−05	4.23	3.45E−05	4.15	1.13E−04	4.16
	9835	1.89E−06	3.86	2.35E−06	3.91	9.93E−06	3.54
	21337	2.90E−07	4.84	3.99E−07	4.58	2.53E−06	3.53
U_2	1337	9.18E−05	—	1.20E−04	—	4.36E−04	—
	2486	2.70E−05	3.94	3.36E−05	4.10	1.14E−04	4.34
	9835	1.81E−06	3.93	2.33E−06	3.88	1.26E−05	3.20
	21337	2.90E−07	4.73	3.97E−07	4.57	2.67E−06	4.00
P	870	1.40E−03	—	2.21E−03	—	9.54E−03	—
	1628	5.06E−04	3.25	8.15E−04	3.18	3.69E−03	3.03
	6498	6.12E−05	3.05	1.00E−04	3.03	5.49E−04	2.75
	14138	1.87E−05	3.05	3.10E−05	3.01	2.29E−04	2.25

Table 6: Errors and convergence rates with $\mathbb{P}_5$ polynomial reconstructions where the polynomials on boundary edges are computed according to Section 4.2.

	DOF	E_1	O_1	E_2	O_2	E_∞	O_∞
U_1	1337	4.18E−06	—	5.13E−06	—	1.38E−05	—
	2486	6.17E−07	6.17	7.87E−07	6.04	2.52E−06	5.48
	9835	1.30E−08	5.61	1.63E−08	5.64	4.91E−08	5.73
	21337	1.20E−09	6.15	1.51E−09	6.14	4.91E−09	5.94
U_2	1337	4.24E−06	—	5.15E−06	—	1.29E−05	—
	2486	6.14E−07	6.23	7.90E−07	6.05	2.50E−06	5.30
	9835	1.30E−08	5.61	1.63E−08	5.64	5.22E−08	5.63
	21337	1.20E−09	6.16	1.51E−09	6.14	5.87E−09	5.64
P	870	3.15E−05	—	4.81E−05	—	2.09E−04	—
	1628	6.98E−06	4.81	1.14E−05	4.60	4.73E−05	4.74
	6498	2.01E−07	5.13	3.35E−07	5.10	1.99E−06	4.58
	14138	2.93E−08	4.96	4.96E−08	4.91	3.01E−07	4.86

6 CONCLUSION

We have presented a powerful method to derived an effective sixth-order of approximation for the Stokes equations involving curved boundary. We highlight that straightforward approximation will provide at most a second-order approximation leading to a huge degradation of the accuracy. The method is very simple to implement and only requires the Gauss points on the curved boundary. It can be seen as a blackbox to improve the accuracy since the structure of the polynomial reconstruction (we mean the matrix which compute the polynomial coefficients) is independant of the position of the Gauss points.

Consequently, the approximation of curved boundaries is just a plug-in function which enable to use the polygonal domain and add a small correction provinding the optimal order.

ACKNOWLEDGEMENTS

This research was financed by FEDER Funds through Programa Operational Fatores de Competitividade — COMPETE and by Portuguese Funds FCT — Fundação para a Ciência e a Tecnologia, within the Projects PEst-C/MAT/UI0013/2014, PTDC/MAT/121185/2010 and FCT-ANR/MAT-NAN/0122/2012.

References

REFERENCES

[1] Barth, T.J., Frederickson, P.O., "Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction", *AIAA Paper* Vol. **90-0013**, 1990.

[2] Barth, T.J., "Recent developments in high order k-exact reconstruction on unstructured meshes", *AIAA Paper* Vol. **93-0668**, 1993.

[3] Boersma, B.J., "A 6th order staggered compact finite difference method for the incompressible Navier-Stokes and scalar transport equations", *J. Comput. Phys.* Vol. **230**, pp. 4940-4954, 2011.

[4] Boularas, A., Clain, S., Baudoin, F., "A sixth-order finite volume method for diffusion problem with curved boundaries", HAL preprint, https://hal.archives-ouvertes.fr/hal-01052517.

[5] Clain, S., Machado, G.J., Nóbrega, J.M., Pereira, R.M.S., "A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients", *Comput. Meth. in App. Mech. and Eng.* Vol. **267**, pp. 43-64, 2013.

[6] Ferrer, E., Willden, R.H.J., "A high order discontinuous Galerkin finite element solver for the incompressible Navier-Stokes equations", *Comput. & Fluids* Vol. **46**, pp. 224-230, 2011.

[7] Ferziger, J.H., Perić,, "Computational methods for fluids dynamics", Springer-Verlag, Berlin, 1996.

[8] Frochte, J., Heinrichs, W., "A splitting technique of higher order for the Navier-Stokes equations", *J. Comput. and App. Math.* Vol. **228**, pp. 373-390, 2009.

[9] Griffith, B.E., "An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as preconditioner", *J. Comput. Phys.* Vol. **228**, pp. 7565-7595, 2009.

[10] Guermont, J.L., Minev, P., Shen, J., "An overview of the projection methods for incompressible flows", *Comput. Meth.Appl. Engrg.* Vol. **195**, pp. 6011-6045, 2006.

[11] R.-S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. Vol. **19**, pp. 90-109, 1975.

[12] Kampanis, N.A., Ekaterinaris, J.A., "A staggerid grid, high-order accurate method for the incompressible Navier-Stokes Equations", *J. Comput. Phys.* Vol. **215**, pp. 589-613, 2006.

[13] Montlaur,A. , Fernandez-Mendez, S., Huerta, A., "Discontinuous Galerkin methods forthe Stokes equations using divergence-free approximations", *Inter. J. Numer. Meth. Fluids* **57**, pp. 1071-1092, 2008.

[14] Nigro, A., De Bartolo, C., Bassi, F., Ghidoni, A., "Up to sixth-order accurate A-stable implicit schemes applied to the discontinuous Galerkin discretized Navier-Stokes equations", *J. Comput. Phys.* Vol. **274**, pp. 136-162, 2014.

[15] Ollivier-Gooch,C. , Van Altena,M. , "A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation", *J. Comput. Phys. Arch.* Vol. **181(2)**, pp. 729-752, 2002.

[16] Patankar,S.V. , "Numerical heat transfer and fluid flow", Hemisphere, New-York, 1980.

[17] Rhie, C. M., Chow, W.L., "A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation", *AIAA J.* Vol. **21**, pp. 1525-1532, 1983.

[18] Sadd,Y., Schultz, M.H., "GMRES: a general minimal residual algorithm for solving nonsymmetric linear systems", *SIAM J. Sci. Stat. Comput.* Vol. **7(3)**, pp. 856-869,1986.

[19] Saad, Y., "Iterative methods for spares linear systems", Society for Industrial and Applied Mathematics, 2003.

[20] Shang, X., Zhao, V., Bayyuk, V., "Generalized formulations for the Rhie-Chow interpolation", *J. Comput. Phys.* Vol. **258**, pp. 880-914, 2014. Butterworth-Heinemann, Waltham, 2014.

SYMCOMP 2015
Faro, March 26-27, 2015

MOMENT-CURVATURE DIAGRAMS FOR REINFORCED CONCRETE SECTION DESIGN

T. Marques[1], C. C. Ferreira[2] and H. Barros[2]

1: COBA, Consultores de Engenharia e Ambiente, S.A.
Av. 5 de Outubro, 323, 1649-011 Lisboa
e-mail: t.marques@cobagroup.com, web: http://www.cobagroup.com

2: INESCC - Department of Civil Engineering
University of Coimbra
Rua Luis Reis Santos-Polo II, 3030-788 Coimbra
e-mail: carla@dec.uc.pt, web: http://www.dec.uc.pt

Keywords: Reinforced concrete, moment-curvature, stiffness, EC2.

Abstract *The moment-curvature (m-ρ) relationship produces a differential equation for the transverse displacement of a structural member. When the cracking of the reinforced concrete section is considered, the differential equation closed form solution can be solved and the m-ρ diagrams are useful to achieve the load-deflection behaviour. The following statements sustain this methodology: - for a given axial load there exist a section curvature, with an extreme fibre strain under compression, is in equilibrium with the applied axial load; - at this section curvature, a unique bending moment can be computed from the stress distribution. According to these statements, the m-ρ relationship can be obtained. The procedure is used in the present work with MAPPLE software support. The resulting curves are the moment-curvature (m-ρ) relationship for rectangular single reinforced concrete sections. The section stiffness (EI) is also derived. The lower bound of EI is the relevant section stiffness for the design and their values are compared to those advised by Eurocode 2(EC2)* [2] *in the nominal stiffness method.*

1. INTRODUCTION

In linear structures, such as beams or columns, there is a relation between the bending moment M and the curvature ($1/r$), disregarding the deformation produced by the tangential stresses. This moment (M)-curvature ($1/r$) relation takes the form of a differential equation for the transverse displacement of the structural member. It can be analytically obtained if the concrete is considered linear elastic and the steel either elastic or plastic. This is the case that is presented in this work, considering three different stages of the reinforced concrete section that are: uncracked concrete; cracked concrete and steel in the elastic or in the plastic domains.

2. REINFORCED CONCRETE SECTION BEHAVIOUR

The behaviour of singly reinforced concrete rectangular cross sections, defined by the width b, height h, concrete cover a (distance between steel centroid and concrete border) and steel area A_s (see Fig. 1), is described in the next sections.

2.1. Fundamental assumptions

The fundamental assumptions of the behaviour of flexural beams are [3]: the cross section, plane before application of the loading, remains plane after the deformation, see Fig.1; concrete and steel are considered to be perfectly adherent; the concrete stress-strain constitutive law is linear elastic in compression and in tension, with elasticity modulus E_c; in tension the maximum stress in concrete is f_{ctm}; the steel stress-strain constitutive law is linear elastic, with elasticity modulus E_s, up to the design yield stress f_{yd}.after this value a constant stress of f_{yd} is considered (plastic domain).

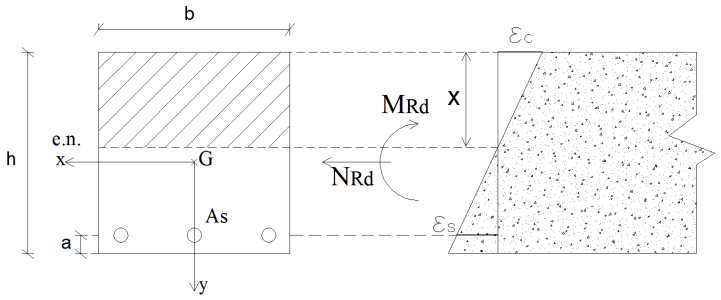


Figure 1. Cross section and deformation.

2.2. Static equations

The static equations traduce the resultant axial load N by the integration of normal stresses in concrete σ_c and steel σ_s , and the resultant moment M at the centroid of the concrete section (middle height in the rectangular section) by a similar procedure, that is:

$$N = \int_A \sigma_c \, dA + \sigma_s A_s \tag{1}$$

$$M = \int_A \sigma_c Y dA + \sigma_s Y_s A_s \quad (2)$$

where:
A – area of concrete;
σ_c, σ_s – concrete, steel stress;
A_s – steel area.
Y_s and Y – distances to the centroid (see Fig.1).

3. MOMENT CURVATURE EQUATIONS

The assumptions exposed in 2.2 lead to three stages of the reinforced concrete section and corresponding equilibrium equations (1) and (2), that are:

a) before concrete cracking and the steel in the elastic range;
b) after concrete cracking with the steel in the elastic range;
c) after concrete cracking and steel in the plastic domain.

In each stage the procedure is the following: 1-compute the axial load N; 2- the axial load value determines the neutral axis position x; 3- the bending moment M that equilibrates the axial load is found at the neutral axis or at middle height M^*. Section 3.1 presents the moment curvature relation for uncracked reinforced concrete section. Sections 3.2 and 3.3 resume the expressions for N, x, M or M^* after concrete cracking.

3.1. Uncracked concrete

Before concrete cracking, the steel is in the elastic domain and the section is homogenized in concrete through the factor $\alpha = E_s/E_c$, multiplying the steel area [3]. The moment M versus curvature $1/r$ relation is given by:

$$M = \frac{1}{r} E_c \left[\frac{bh^3}{12} + bh\left(\frac{h}{2} - x\right)^2 + \frac{1}{2} A_s \alpha (h - a - x)^2 \right] \quad (3)$$

The maximum moment before cracking, termed M_{cr}, that is the moment for which the maximum tensile stress in concrete is equal to the mean value of the tensile strength, f_{ctm}, and the corresponding curvature $(1/r)_{cr}$ are the following:

$$M_{cr} = \left[f_{ctm} + \frac{N}{bh + \frac{1}{2} A_s \alpha} \right] \frac{EI}{(h-x)E_c}; \; (1/r)_{cr} = \frac{M_{cr}}{EI} \quad (4)$$

with $EI = E_c \left[\frac{bh^3}{12} + bh\left(\frac{h}{2} - x\right)^2 + \frac{1}{2} A_s \alpha (h - a - x)^2 \right]$.

3.2. Cracked concrete

In the reinforced concrete section, after the cracking of the concrete under tension, the concrete supports the compressive stresses and steel the tensile stresses. At the initial stage, when the steel is yet in the elastic range, the static equation (1) is:

$$N = \frac{x^2 b E_c}{2r} + \frac{A_s E_s}{2r}(a - h - x) \tag{5}$$

Solved in x, gives the following result:

$$x = \left[-\frac{A_s \alpha}{2b} + \left(\frac{\alpha^2 A_s^2}{4b^2} + 2\frac{r}{bE_c} N + \frac{\alpha A_s}{b}(h - a) \right)^{1/2} \right] \tag{6}$$

Equation (2) than becomes:

$$M = \frac{x^3}{3r} b E_c + \frac{(a - h + x)^2}{2r} E_s A_s + \left(\frac{h}{2} - x \right) \left(N - \frac{x^2}{6r} b E_c \right) \tag{7}$$

3.3. Cracked concrete and steel in the plastic range

After the concrete cracking but with steel in the plastic range, that is the stress equal to f_{yd}, equations (5, 6 and 7) become:

$$N = \frac{1}{2\rho} x^2 b E_c - \frac{1}{2} A_s f_{yd} \tag{8}$$

$$x = \frac{r}{bE_c} \left[\frac{1}{r} b E_c (2N + A_s f_{yd}) \right]^{1/2} \tag{9}$$

$$M = \frac{E_c b x^3}{3r} - \frac{A_s f_{yd}}{2}(a - h + x) \tag{10}$$

The bending moment evaluated at the centroid of the concrete section (middle height) is:

$$M^* = M + N \left(\frac{h}{2} - x \right) \tag{11}$$

4. DERIVING STIFFNESS

The section flexural stiffness EI in the uncracked concrete in simply obtained by equation (4), where $E = E_c$ and I is the inertia of the homogeneized section. In the cracked concrete is obtained by deriving the moment M - curvature $1/r$ relation in order to $1/r$ (equations 7 and 10). The computed section stiffness can be used in design of columns or beams. For columns the (EC2) [2] approximates the flexural stiffness by the following:

$$EI = k_c E_c I_c + k_s E_s I_s \tag{12}$$

where k_c and k_s are factors, I_c and I_s are respectively the moments of inertia of concrete and steel about the centre of area of the concrete, defined in EC2.

5. NUMERICAL RESULTS

5.1. Moment curvature diagrams

The Figure 2 shows the moment curvature diagrams for a rectangular singly reinforced cross section with variable axial load ($N = 0; 100; 500; 1000 kN$). The cross section geometry is defined by: b= $0{,}2m$; $h = 0{,}4m$; $a = 0{,}02m$; $A_s = 0{,}0021m^2$. The material properties are: E_s =

$200GPa$; $E_c = 20GPa$; $f_{yd} = 400MPa$ $f_{ck} = 30MPa$.

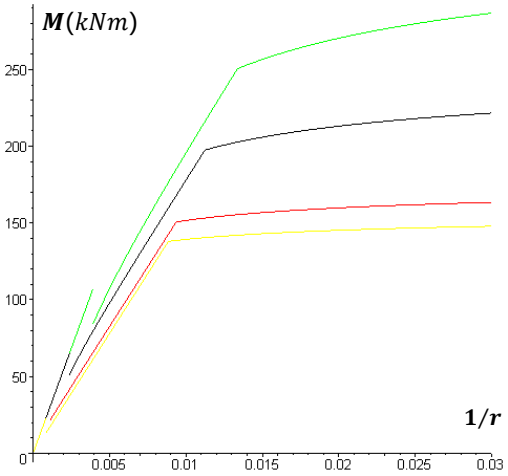


Figure 2. Moment curvature for $N = 0\ (yellow); 100\ (red); 500(black); 1000kN$(green).

5.2. Stiffness versus curvature diagrams

The Figure 3 shows the stiffness versus curvature diagrams for the same section and axial load ($N = 0; 100; 500; 1000kN$) obtained by deriving the moment M - curvature $1/r$ relation. The horizontal lines in the Figure 4 represent the advised stiffness values of the Eurocode 2 (EC2) [2], to be used in the evaluation of the second order effects by the nominal stiffness method. The values considered for the parameters in EC2 are

$$k_c = \frac{0.3N}{bhf_{ck}/1.5}\sqrt{\frac{f_{ck}}{20}}\ ;\ k_s = 1$$

It can be noted that the EC2 values lay between the previous results.

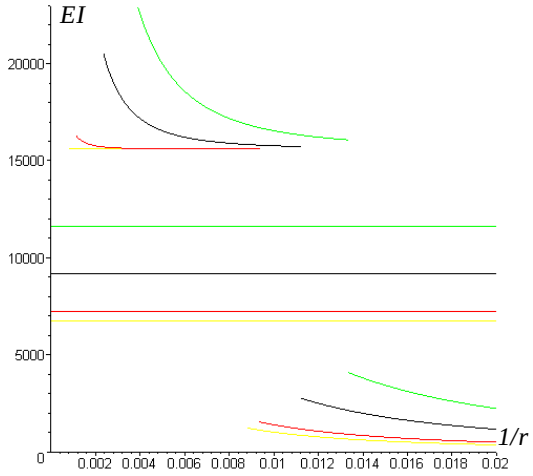


Figure 3. Stiffness by the present model and the (EC2) [2] for: $N = 0; 100; 500; 1000kN$.

12. CONCLUSIONS

The present work presents:

- the deduction of the moment curvature diagrams for reinforced rectangular singly reinforced concrete sections by MAPLE software;
- the three stages of the section: *i*) uncracked; *ii*) cracked and steel in elastic domain *iii*) cracked and steel in plastic domain;
- the effect of the variation of axial load in the flexural stiffness;
- the comparison of flexural stiffness with the nominal stiffness of the EC2 [2].

ACKNOWNLEGEMENTS

The financial support of ACIV is gratefully acknowledged.

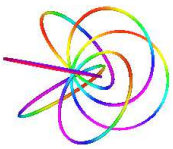
REFERENCES

[1] Silva V. D., Barros H., Julio E., Ferreira C., Closed form ultimate strength of multi-rectangle reinforced concrete sections under axial load and biaxial bending, Computers & Concrete, 6 (6): 505-521, DEC 2009.

[2] Silva V. D., Barros H., Ferreira C., *Computation of The Ultimate Load Capacity of Eccentrically Compressed Reinforced Concrete Columns Under Fire*, International Conference on Recent Advances in Nonlinear Models - Structural Concrete Applications, CoRAN 2011, Coimbra, Novembro, 2011.

[3] Nilson A. H., *Design of concrete structures* Mc-Graw-Hill, 1997.

[4] EN 1992-1-1; Eurocode 2 – *Design of concrete structures – Part 1-1: General rules and rules for buildings*, December, 2004.



SYMCOMP 2015
Faro, March 26-27, 2015

A GLOBAL OPTIMIZATION APPROACH BASED ON ADAPTIVE POPULATIONS

Tiago A. N. Silva[1,2], Maria A. R. Loja[1,2], Alda Carvalho[1,3], Nuno M. M. Maia[2] and Joaquim I. Barbosa[1,2]

1: GI-MOSM - Research Group on Modelling and Optimization of Multifunctional Systems
ADEM/ISEL - Mechanical Engineering Department,
Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal.
e-mail: tasilva@dem.isel.pt

2: LAETA, IDMEC, Instituto Superior Técnico,
Universidade de Lisboa,
Avenida Rovisco Pais, 1, 1049-001 Lisboa, Portugal.

3: CEMAPRE, ISEG,
Universidade de Lisboa,
Rua do Quelhas, 6, 1200-781 Lisboa, Portugal.

Keywords: Inverse problem, Extended Differential Evolution, Adaptive Empirical Distributions.

Abstract. *The solution of inverse problems based on experimental data is itself an important research issue. In this context and assuming that an experimental sample is available, rather than trying to find a specific deterministic solution for the inverse problem, one aims to determine the probabilistic distribution of the modelling parameters, based on the minimization of the dissimilarity between the empirical cumulative distribution function of an experimental solution and its simulation counterpart. The present paper presents an innovative framework, where Differential Evolution is extended in order to estimate not only an optimal set of modelling parameters, but to estimate their optimal probabilistic distributions. Additionally, the Adaptive Empirical Distributions optimization scheme is here introduced. Both schemes rely on the two samples Kolmogorov-Smirnov goodness-of-fit test in order to evaluate the resemblance between two empirical cumulative distribution functions. A numerical example is considered in order to assess the performance of the proposed strategies and validity of their solutions.*

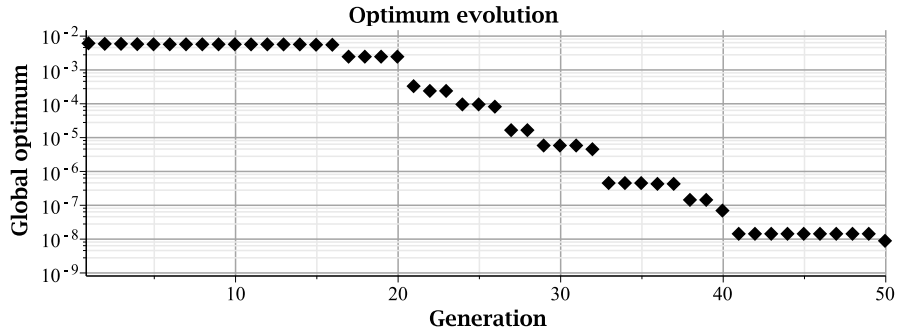
1 INTRODUCTION

The solution of inverse problems based on experimental data is itself an important research issue. In this context and assuming that an experimental sample is available, rather than trying to find a specific deterministic solution for the inverse problem, one aims to determine the probabilistic distribution of the modelling parameters, based on the minimization of the dissimilarity between the empirical cumulative distribution function (eCDF) of an experimental or reference solution and its simulated counterpart.
The present work presents an innovative framework, where the Differential Evolution (DE) algorithm is extended in order to estimate not only an optimal set of modelling parameters, but to estimate their optimal probabilistic distributions (Section 2). Additionally, the Adaptive Empirical Distributions optimization scheme is here introduced (Section 3). Both schemes rely on the Kolmogorov-Smirnov goodness-of-fit test (KS-test) for two samples, in order to evaluate the resemblance between two empirical cumulative distribution functions (CDFs). Note that this optimization process aims to maximize the p-value of the KS-test. However, for p-values above 0.05 there is no statistical evidence to reject the similarity of the two tested samples, although the higher the p-value the greater is the degree of similarity between the distributions from which the two samples are taken. Therefore, here, one chooses to constraint the maximum number of generation instead of using a stopping criteria based on performance.

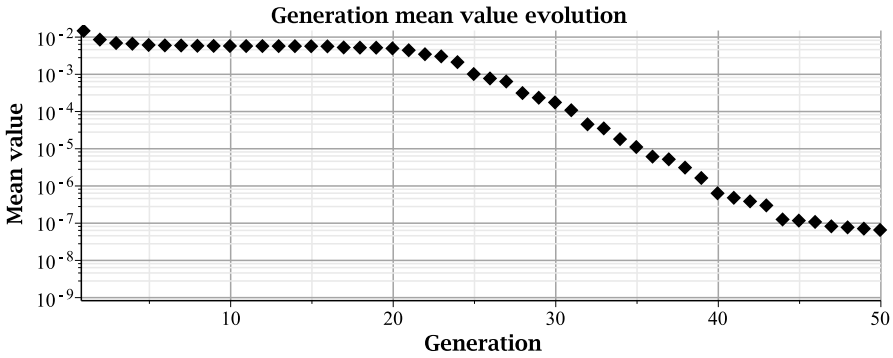
2 EXTENDED DIFFERENTIAL EVOLUTION (eDE)

2.1 Preliminary comments

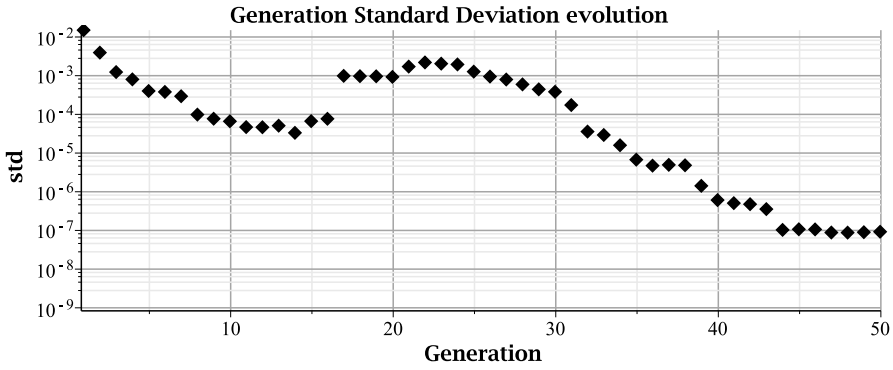
As detailed on [1], DE is an evolutionary computation algorithm based on generations of members, and therefore generations of solutions. DE allows for a statistical analysis of the population at each generation, in order to assess the degree of confidence on the optimal solution, based on statistical evidence. Hence, one can track the evolution of the mean value and standard deviation of each generation side by side with the optimal value. Figure 1 shows the tracking of the referred values for the experimental example given in [2], where the authors used DE to estimate a stiffness of elastic supports of shafts. It is interesting to evaluate the evolution of the mean and standard deviation values of each generation. Note that the standard deviation value has an expressive decrease until approximately the 15^{th} generation (Figure 1c), corresponding to a kind of stagnation of optimum value on a level (Figure 1a). However, when the DE algorithm is possibly trapped at local minima, it tries to explore further the design space in order to escape from it. Therefore, DE turns out to find a direction to the global minimum due to an enlargement of the search space, translated by the increase of the standard deviation value. Moreover, the analysis of the evolution of the mean value of each generation (Figure 1b) allows to assess the density or concentration of solutions near the optimum, in the example the minimum.



(a) Best solution value.

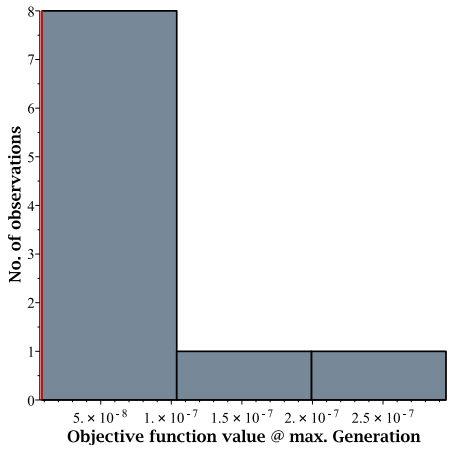


(b) Mean value of the population of solutions.

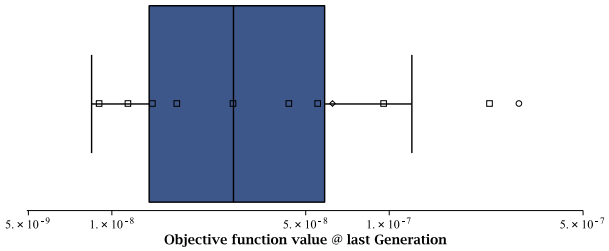


(c) Standard deviation of the population of solutions.

Figure 1: Evolution of the population of solutions over the generations.



(a) Frequency histogram, with the best solution marked in red.



(b) Boxplot.

Figure 2: The response population at the last generation.

This information is always available from any population based algorithm but it is often not used. Nevertheless, this kind of statistical tools can be considered to support a judicious decision on the validity of the computed values, or significance of the optimal value. Note that the previous comments on the results of Figure 1 can be further supported by the analysis of the population at the last generation, given in Figure 2. From Figure 2, it is possible to observe the referred density of solutions near the minimum. Note that the two classes on the right hand side of the frequency histogram (Figure 2a) are after all outliers (Figure 2b).
Despite the reduced sample size, one can stress out the significance of the current optimal value as the minimum of a population concentrated around it with a reduced scatter. Furthermore, if the sample size is sufficient, one may be able to compute a confidence interval for the optimal value, for instance.

2.2 Concept and implementation

As referred, with the extended DE (eDE), one aims for a more ambitious task, as the objective of eDE is to find the probabilistic empirical distribution of each design variable or parameter by the maximization of the degree of similarity between the eCDF of the reference response and its simulated counterpart.
The eDE can be presented as straightforward extension of the base DE algorithm, where a 3rd dimension is added to the base DE population. For the eDE, the definition of an individual is transformed in order to encompass a vector, instead of a single value. Hence, each individual is a vector with a sample of a given design variable, which is uniformly distributed at the first generation and mutated afterwards, accordingly to the DE implementation. Figure 3 illustrates a typical population of the eDE, where each

individual is schematically represented in Figure 4. Thus, each generation is composed by NP members of N_v design variables, with a sample size n.

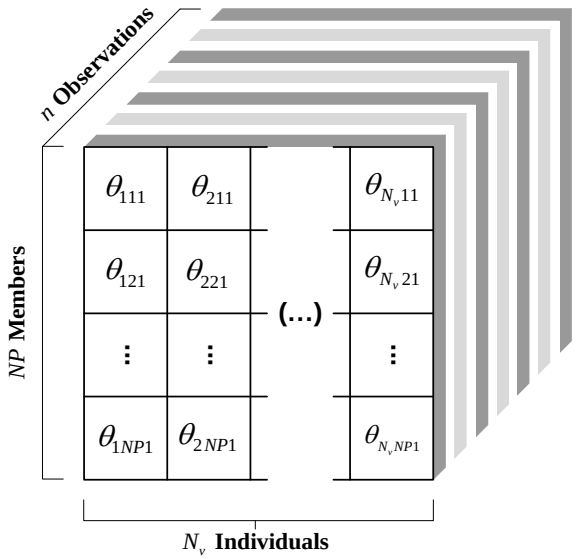
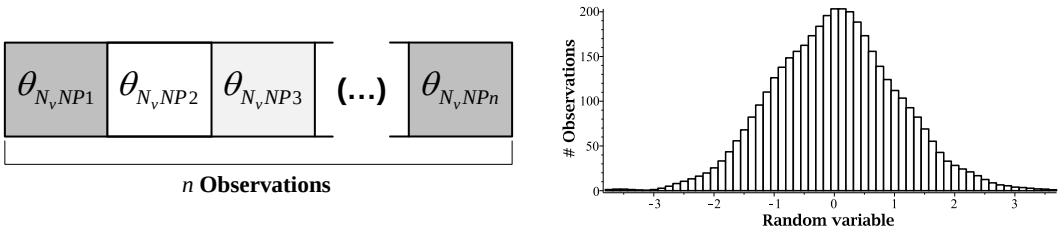


Figure 3: Extended DE population with NP members and N_v design variables at a given generation.



(a) An individual of the extended DE.

(b) A possible frequency histogram of an individual.

Figure 4: An illustrative sample of the individual $\theta_{N_v NP}$, with n observations, at a given generation.

Note that the base DE implementation is here adapted to deal with vectors instead of single value, but this is the principal difference introduced, beside the selection based on the KS-test. However, the solutions of eDE are completely distinct from the typical solutions of the usual evolutionary computation algorithms, as given in Section 4.

3 ADAPTIVE EMPIRICAL DISTRIBUTIONS (AED)

Based on the conceptual idea considered for the eDE, one introduces a different evolutionary strategy, the Adaptive Empirical Distributions (AED) algorithm. The AED algorithm also aims to find the probabilistic distribution of each design variable or parameter by the maximization of the degree of similarity between the eCDF of the reference response and its simulated counterpart. However, its formulation is completely different from the eDE, as it relies on an inverse transform sampling algorithm [3] to generate the population at each new generation. The inverse transform sampling or Smirnov transform is a general pseudo-random sampling generator here used with target eCDFs, the eCDFs of the design variables of the member which the response distribution presents the highest p-value. Solutions for a numerical example using the AED algorithm are given in Section 4.

4 NUMERICAL EXAMPLE

In order to assess the proposed adaptive populations algorithms introduced on the previous subsections, one considers one of the most widely used function for testing optimization algorithms, the Ackley's function [4],

$$z(\boldsymbol{\theta}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}\theta_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(c\theta_i)\right) + a + \mathrm{e} \tag{1}$$

Ackley's function is a continuous, multimodal test function for optimization, which poses complications to local optimization strategies based on gradients, as it has several local minima.
The global optimum z^* of the Ackley's function is attained for $\boldsymbol{\theta}^* = \mathbf{0}: \quad z^* = z\left(\boldsymbol{\theta}^*\right) = 0$, if the values $a = 20.0$, $b = 0.2$ and $c = 2\pi$ are used. Hence, for a two dimensional problem, one may recast eq. (1) as

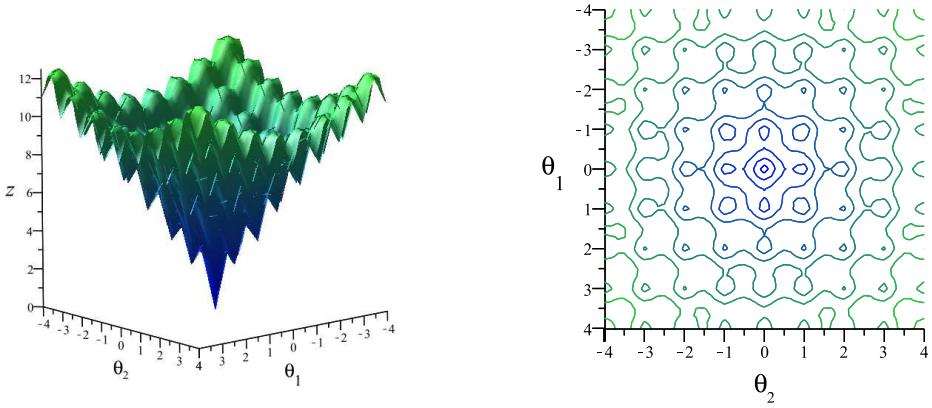
$$z(\theta_1, \theta_2) = -20 \exp\left(-0.2\sqrt{0.5\left(\theta_1^2 + \theta_2^2\right)}\right) - \exp\left(0.5\left(\cos(2\pi\theta_1) + \cos(2\pi\theta_2)\right)\right) + 20 + \mathrm{e} \tag{2}$$

Figure 5 shows the Ackley's function given by eq. (2), defined for $\theta_i = [-4\ ,\ 4]$ for $i = 1, 2$.

To simulate an inverse problem, one has produced a reference response data set using a random sample from a theoretical PDF. In the following subsections, one presents results for the two proposed adaptive populations algorithms, eDE and AED, considering different parametrization for the algorithms. An example of a reference data set is given in Figure 6, where a reference response set with $n = 100$ observations was simulated using a set of parameters sampled from a uniform distribution, such as, $z_m: \ \theta_i \sim N\left(0.0, 0.5\right)$.

4.1 Results and discussion

This subsection shows results for both eDE and AED for different control parameters. Note that in this work one assumes reference data set sampled from a normal distribution,



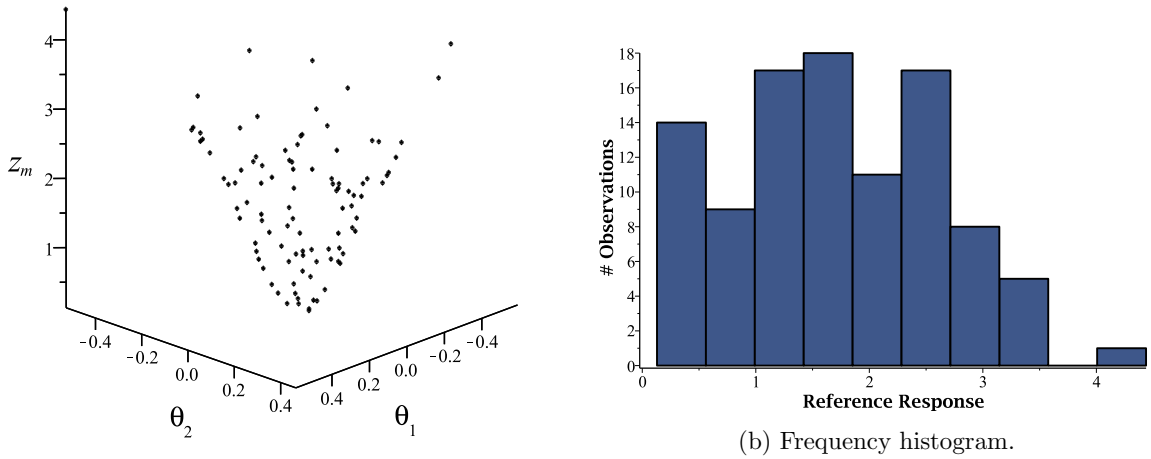
(a) Perpective. (b) Top view (contour plot).

Figure 5: A 2D Ackley's function.

such as $\theta_i \sim N(0.0, 0.5)$ for $i = 1, 2$. To close this subsection a comparison of results is presented (Table 1).
Note that a maximum of 30 generations is used in all the cases as the stopping criterion, in order to assess the behaviour of the algorithm, as the aim is to maximize the p-value of the two samples KS-test.

4.1.1 Case 1: Different eDE/Best/# schemes

Here, one presents a reference response set deemed to be close to the reality of physical tests, where usually one has a limited number of possible experimental tests, namely, when the testing of identical structures is addressed. Hence, a reference parameter set sampled from a normal distribution, $\theta_i \sim N(0.0, 0.5)$ for $i = 1, 2$ with just $n = 10$ observations is used to simulated the reference response set. For instance, Figure 6 shows an example of a reference response set with $n = 100$ observations simulated with $\theta_i \sim U(-0.5, 0.5)$ for $i = 1, 2$.
The initial parameter set is randomly sampled from an uniform distribution $\theta_i \sim U(-2.0, 2.0)$, for a population with 20 members and a sample size of $n = 100$ for each individual. Note that the sample size of the reference set is different from the estimated set. This is not a problem for the algorithm, although the degree of similarity between eCDFs is compromised, in the sense of an increased degree of difficulty in the comparison of the eCDFs.
In order to perform a preliminary parametric study, one presents the evolution of the maximum p-values obtained for different eDE schemes. Figure 7 shows results for eDE using a single difference vector (eDE/Best/1) and two difference vectors (eDE/Best/2), using an adaptive mutation scheme with the differential weight $F \in [0.1, 1.0]$. Comparing the evolution of the maximum p-value given in Figures 7a and 7b, one can observe a smoother evolution of the results of eDE/Best/2, as the increment on the number of dif-



(a) Sampled responses.

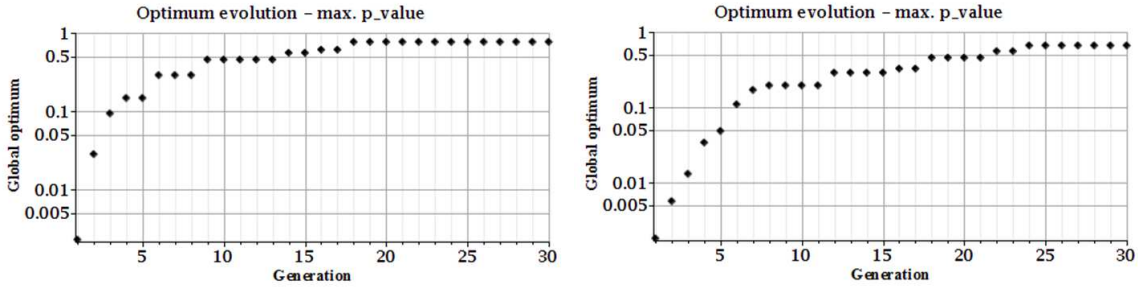
(b) Frequency histogram.

Figure 6: A reference response data set.

ference vectors results in a better support search direction, although its improved stability does not lead directly to a better solution in the end of the optimization process. Note that the maximum p-value obtained at the last iteration is around 0.77 for eDE/Best/1 and 0.66 for eDE/Best/2.

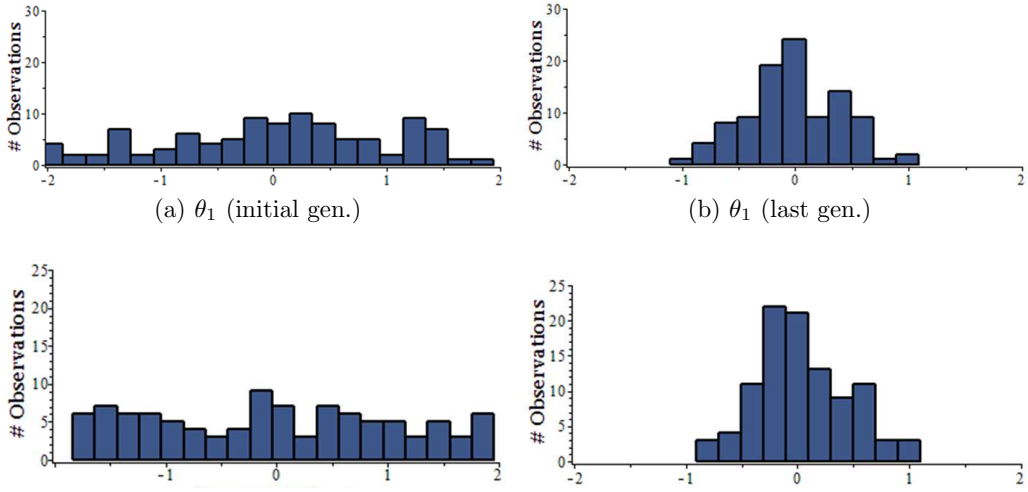
As the eDE/Best/2 scheme here considered presents the lower maximum p-value, the comparison of eCDF for this case is analysed. Note that the referred increased degree of difficulty in the comparison of the eCDFs is closely related to an insufficient sample size, which is evident in Figure 9. Note the lack of ability to fit the reference eCDF in a region where fewer observations are available. This clearly affect the attained maximum p-value, although its improvement is still quite good.

Within the framework of inverse problems, one aims to correct or find a set of modelling parameters, which enables to reproduce an experimental or reference response set. Figure 8 shows the set of parameters related to the best predicted response, the one with the maximum p-value, at the initial and last generations, for the described eDE/Best/2. From Figure 8, it is clear that the initial parameter set is randomly sampled from an uniform distribution, $\theta_i \sim U(-2.0, 2.0)$ (Figures 8a and 8c), and that at the last generation the parameter set assumes a normal configuration, at least approximately (Figures 8b and 8d).



(a) eDE/Best/1 (b) eDE/Best/2

Figure 7: Evolution of the maximum p-value for different eDE schemes - log. scale.



(a) θ_1 (initial gen.) (b) θ_1 (last gen.)

(c) θ_2 (initial gen.) (d) θ_2 (last gen.)

Figure 8: Set of parameters related to the best predicted response (eDE/Best/2).

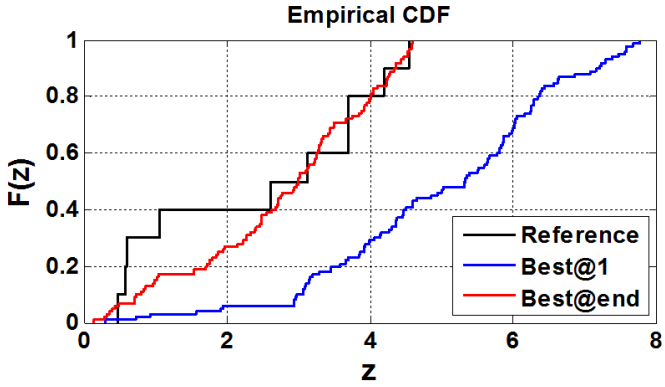
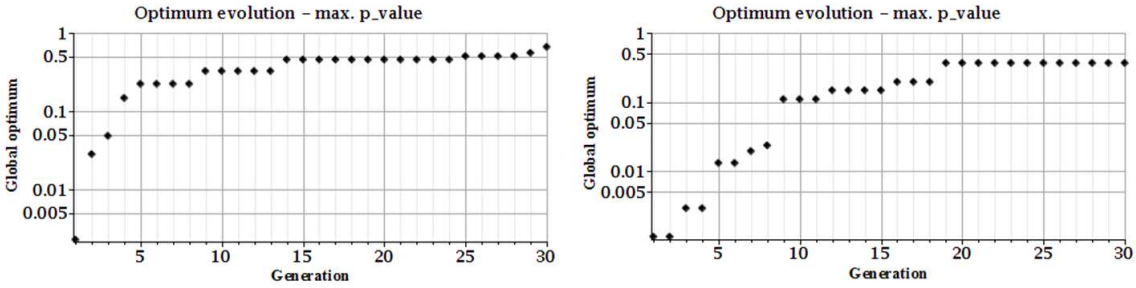


Figure 9: Comparison of the best member eCDFs (initial and final) with the reference one (eDE/Best/2).

From the comparison between eDE/Best/1 and eDE/Best/2, the former presents a higher maximum p-value. Therefore, the eDE for a single difference vector is elected for a study where a classical, non adaptive, mutation scheme is used. The eDE control parameter here considered to be changed is only the mutation differential weight. Thus, the results given in Figure 10a are obtained for eDE/Best/1 with a fixed $F = 1.0$, whereas Figure 10b presents results for $F = 2.0$. The results for a non adaptive mutation scheme are worst that the ones of Figure 7, although the maximum p-value for eDE/Best/1 with $F = 1.0$ is identical to the one obtained with eDE/Best/2.



(a) eDE/Best/1 ($F = 1.0$) (b) eDE/Best/1 ($F = 2.0$)

Figure 10: Evolution of the maximum p-value for different eDE/Best/1 mutation differential weights - log. scale.

4.1.2 Case 2: AED

With the initial and reference parameter sets defined for Case 1, so that the reference response set is generated using a parameter set sampled from a normal distribution, $\theta_i \sim N(0.0, 0.5)$ for $i = 1, 2$ with just $n = 10$ observations, and the initial parameter set is again defined for $\theta_i \sim U(-2.0, 2.0)$, for a population with 20 members and a sample size of $n = 100$ observations for each individual..

Figure 11 shows the evolution of the maximum p-value during the optimization process. The comments on the improvement of the degree of similarity between the reference response set and the predicted one are similar to the previous ones, where a quite good improvement is observed. Note that the maximum p-value is approximately 0.96, similarly to the value attained for a reference set with $n = 100$ observation. The memory issue is also patent here (Figure 12).

The recover from the initial random sampling from a uniform distribution is affected by referred memory issue (Figure 13). Therefore, the normal configuration is approximated (Figure 13b) but with less quality (Figure 13d), when compared to the results of Figure 8, attained for eDE/Best/2, although with a lower maximum p-value.

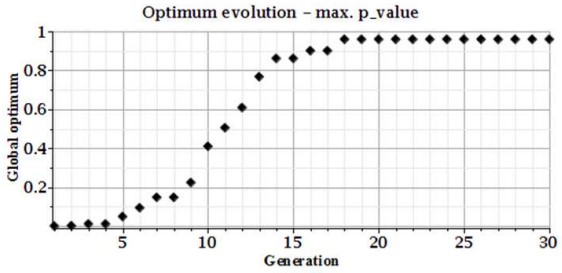


Figure 11: Evolution of the maximum p-value (AED) - log. scale.

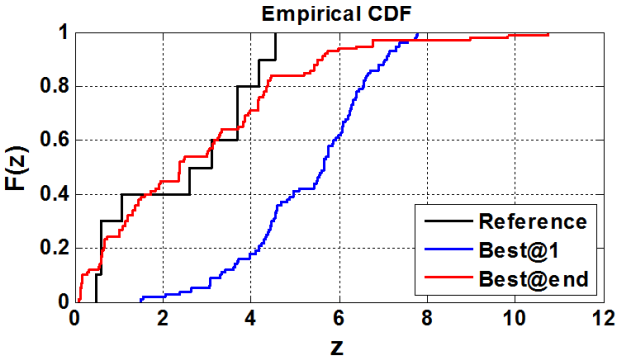
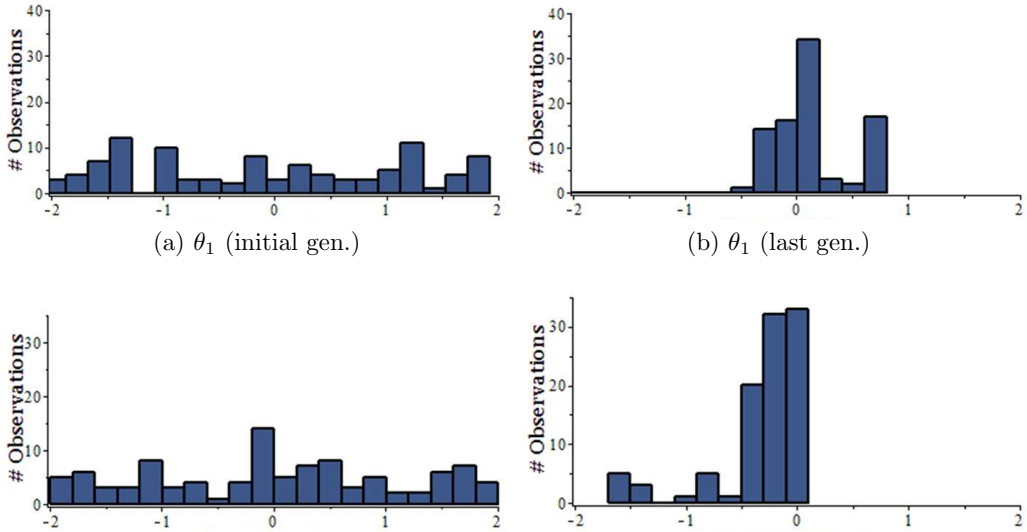


Figure 12: Comparison of the best member eCDFs (initial and final) with the reference one (AED).



(a) θ_1 (initial gen.) (b) θ_1 (last gen.)

(c) θ_2 (initial gen.) (d) θ_2 (last gen.)

Figure 13: Set of parameters related to the best predicted response (AED).

4.1.3 Comparison of results - summary

Table 1: Summary of the optimal values and computational times for all the considered cases and algorithmic schemes.

Reference set	Algorithm	Opt. p-value	Comp. Time
$z_m : \theta_i \sim N(0.0, 0.5)$ $n = 10$	**eDE/Best/1** AM: $F \in [0.1, 1.0]$ $n = 100$ — $NP = 20$	**0.76677**	$\sim$ 155s
	eDE/Best/2 AM: $F \in [0.1, 1.0]$ $n = 100$ — $NP = 20$	0.66179	$\sim$ 160s
	eDE/Best/1 M: $F = 1.0$ $n = 100$ — $NP = 20$	0.66179	$\sim$ 125s
	eDE/Best/1 M: $F = 2.0$ $n = 100$ — $NP = 20$	0.36603	$\sim$ 157s
	AED $n = 100$ — $NP = 20$	**0.95898**	**$\sim$ 37s**

Win7 @ CPU Intel Core2 Quad Q9000 @2GHz (4GB RAM)

The results given in Table 1 clearly show the computational advantage of AED. Despite the reported memory effect of AED, both algorithms attained very high p-values, which is a very good indication of the robustness of the proposed optimization strategies.

5 CONCLUDING REMARKS

Two innovative evolutionary computation approaches are proposed: the extended Differential Evolution and the Adaptive Empirical Distributions algorithms.
The theoretical application of both algorithms stressed out the very high quality of the estimated empirical response distributions with respect to the reference one. Moreover, it is shown that the agreement between predicted and reference eCDFs is achieved with a set of identified parameters, whose empirical distributions approximate their reference counterpart. Therefore, it is intended that the potential of both algorithms to deal with experimental applications is considerable, although improvements may be introduced.
The computational implementation of AED is simple and it performs faster than eDE at the same level of accuracy. Note that both algorithms use an integrated implementation of the commercials software packages *Maple* and *MatLab*.

Both algorithms are framed in the context of uncertainty quantification, as they may use all the available information to support engineering decisions. Thus, they are able to improve computational models in order that these can accurately reproduce a set of experimental results, considering the empirical probabilistic distribution of the optimal member in a population and not only a single optimal value.

ACKNOWLEDGEMENT

The authors acknowledge the support of the Portuguese Foundation for Science and Technology, FCT, under the projects PEst/OE/EME/LA0022/2011 and PTDC/ATP-AQI/5355/2012, and the PhD grant SFRH/BD/44696/2008.

REFERENCES

[1] Price, K., Storn, R., Lampinen, J. A., *Differential evolution: a practical approach to global optimization*, Springer-Verlag, 2006.

[2] Silva, T. A. N., Loja, M. A. R., Maia, N. M. M., Barbosa, J. I., "A Hybrid Procedure to Identify the Optimal Stiffness Coefficients of Elastically Restrained Beams", *International Journal of Applied Mathematics and Computer Science*, **25(2)**, 2015. (*in press*)

[3] Devroye, L., *Non-uniform random variate generation*, Springer-Verlag New York Inc., 1986.

[4] Bäck, T., *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, 1st edition, 1996.

SYMCOMP 2015
Faro, March 26-27, 2015

NOVEL METHODOLOGIES TO TRAIN EVOLUTIONARY NEURAL NETWORKS IN SUPERVISED MACHINE LEARNING. A CASE OF STUDY IN PRODUCT AND SIGMOID UNITS

Antonio J. Tallón-Ballesteros[1]*

1: Department of Languages and Computer Systems
Higher Technical School of Computer Science Engineering
University of Seville (Spain)
Reina Mercedes Av. 41012-Seville (Spain)
e-mail: atallon@us.es, web: http://www.lsi.us.es

Keywords: neural networks, evolutionary computation, supervised machine learning, product unit, sigmoid unit and evolutionary programming

Abstract *This paper gathers three contemporary proposals to train neural networks by means of a global meta-heuristic called evolutionary programming which simulates the evolution at species level. Broadly speaking, the baseline algorithm to train the neural networks, and therefore to get the models that will be utilised to evaluate the performance with the unseen data, follows a global optimization technique based on populations. As such this approach is founded on the principle that the population is undergone to the operators of replication and mutation. The neural network architecture is feed-forward -containing product or sigmoid neurons in the hidden layer- and two kinds of mutations named parametric and structural ones could change their elements such as the coefficients and the structure, respectively. The parametric mutation determines the new values for the coefficients via a simulated annealing algorithm which is a global optimization method based on trajectories. Each one of the new methodologies adds a particular ingredient to the general algorithm and is evaluated with a good number of binary and multi-class supervised machine learning problems. The results have been analysed with statistical tests. One of the outstanding conclusions from this work is that product units are more accurate, whereas sigmoid units are faster. Moreover, in order to accelerate the training of the product unit neural networks a data pre-processing with feature selection has been included leading to less complex classification models providing significantly more confidence. The single requirement to go through the data preparation phase is that the test error is close to a twenty percent with reference algorithms such as C4.5 or 1-nearest neighbour. The three new approaches are able to be applied either to another type of hidden units or unsupervised machine learning tasks.*

1. INTRODUCTION

Several approaches on machine learning tackle with the classification problem. Basically, these are algorithms which construct classifiers from sample data, such as neural networks, decision trees and rule-based classifiers, have attracted growing attention due to their wide applicability [1]. A deep review of them can be found in [2]. Depending on the kind of method the learning algorithms can be grouped into two categories: a) black-box methods like Bayesian classifiers or neural networks and b) knowledge-oriented methods, such as the models built by association rules or decision trees.

Regarding the neural networks, there are two well-known architectures: feed-forward and recurrent neural networks. The training algorithms are different according to the topology of the network and may be based on local or global optimization procedures. According to the year of introduction, the classical methods are older and have their foundations on local optimization. On the other hand, the modern approaches are global search techniques. Deeping in the global optimization strategies, we can find methods based on trajectories such as simulated annealing or taboo search and those based on populations like the techniques based on bio-inspired computation including Evolutionary Computation and/or Swarm intelligence. Neural networks that are trained by evolutionary algorithms are called evolutionary neural networks [3].

Our attention is focused on evolutionary artificial feed-forward neural networks (EAFFNN) that are a hot research field in the end of the previous century and in the current one. More concretely, the evolutionary algorithm learns simultaneously the architecture and the weights of the model. The units or neurons of the hidden layer will be product or sigmoid units.

This paper aims at summarizing in a detailed and single work the contemporary approaches that we have developed recently to train feed-forward neural network models via an evolutionary computation strategy.

This paper is organized as follows: Section 2 sketches the two kind of artificial neural networks that were majority utilised in this work; Section 3 details the three novel methodologies that are gathered in this paper; Section 4 collects the experimental results and the settings in order to promote the reproducibility of the results; finally, Section 5 summarizes the conclusions that achieve this research.

2. BACKGROUND

Pattern classification has been dealt with several kinds of artificial neural networks (ANN) [4], including, among others: multilayer perceptron (MLP) NNs, where the transfer functions are sigmoid unit basis functions; radial basis function NNs with kernel functions, where the transfer functions are usually Gaussian and a class of multiplicative NNs called product unit neural networks (PUNNs).

Figure 1 shows the scheme of two AFFNN models for a bi-classification problem, respectively with sigmoid (at part a)) and product units (at part b)). Each one is a *k:m:1* three-layer architecture, that is, *k* nodes in the input layer, *m* ones and a bias one in the hidden layer and one node in the output layer.

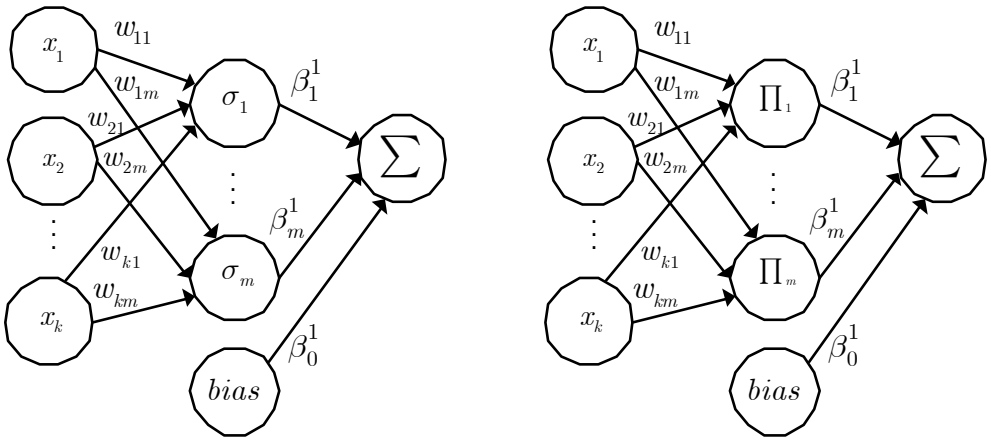


Figure 1. Scheme of two AFFNN models with sigmoid (a) and product (b) units for a bi-classification problem.

The transfer function of each node in the hidden and output layers is the identity function. Thus, the functional model obtained by each of the nodes in the output layer with J classes is given by:

$$f(x_1, x_2, ..., x_k) = \beta_0^l + \sum_{j=1}^{m} \beta_j^l B_j(\boldsymbol{x}, w_j) \qquad l = 1, 2, ..., J \tag{1}$$

where B_j follows different expressions depending on the type of unit:

- Product unit:

$$B_j(\boldsymbol{x}, w_j) = \prod_{i=1}^{k} x_i^{w_{ji}} \tag{2}$$

- Sigmoid unit:

$$B_j(\boldsymbol{x}, w_j) = \frac{1}{1 + \exp(-w_{j0} - \sum_{i=1}^{k} w_{ji} x_i)} \tag{3}$$

The base algorithm has been used as starting point in some previous works [5, 6, 7, 8, 9]. The search begins with a random initial population and, for each iteration, the population is modified using a population-update algorithm. The population is subjected to the operations of replication and mutation. Crossover is not used due to its potential disadvantages in evolving artificial networks [10, 11]. Two kind of mutation may be distinguished during the

evolution. The parametric mutation changes the value of the model coefficients (step 9) and consists of a simulated annealing algorithm. Parametric mutation is accomplished for each exponent w_{ji} and coefficient β_j^l of the model with Gaussian noise, where the variance depends on the temperature. The structural mutation implies a modification in the structure of the model (step 10) and allows different regions in the search space to be explored while helping to maintain the diversity of the population. There are five different structural mutations; the first four ones are similar to those in the GNARL model [11]: node addition, node deletion, connection addition, connection deletion and node fusion. All the above mutations are made sequentially in the given order, with a certain probability, in the same generation on the same network. If probability does not select a mutation, one of the mutations is chosen at random and applied to the network. Figure 2 overviews the pseudo-code of the base EA for classification tasks.

Program: Evolutionary Algorithm
Data: Training set
Input parameters: gen, neu, α_2
Output: Best ANN model
1: t ← 0
2: P(t) ← {ind$_1$, …, ind$_{10000}$} // Random initialisation of the population
3: f(P(t) {ind$_1$, …, ind$_{10000}$}) ← *fitness* (P(t) {ind$_1$, …, ind$_{10000}$}) // Calculate fitness
4: P(t) ← P(t) {ind$_1$, …, ind$_{10000}$} // Sort individuals by fitness: ind$_i$ > ind$_{i+1}$
5: P(t) ← P(t) {ind$_1$,… ind$_{1000}$} // Retain the 1000 best ones
6: **while** stop criterion not met **do** // main loop
7: P(t) {ind$_{901}$,…ind$_{1000}$} ← P(t) {ind$_1$, …, ind$_{100}$} // Best 10% replace the worst 10%
8: P(t+1) ← P(t) {ind$_1$, …, ind$_{900}$}
9: P(t+1) ← *pm* (P(t+1) {ind$_1$, …, ind$_{90}$}) // Parametric mutation (10% P(t+1))
10: P(t+1) ← *sm* (P(t+1) {ind$_{91}$, …, ind$_{900}$}) // Structural mutation (90% P(t+1))
11: f(P(t+1) {ind$_1$, .. ind$_{900}$}) ← *fitness* (P(t+1) {ind$_1$, …, ind$_{900}$}) // Evaluate
12: P(t+1) ← P(t+1) (ind$_1$, …, ind$_{900}$) ⋃ P(t) {ind$_{901}$, …, ind$_{1000}$}
13: P(t+1) ← P(t+1){ind$_1$, …, ind$_{1000}$} // Sort individuals
14: t ← t+1
15: last_generation ← t
16: **end while**
17: **return** *best* (P(last_generation) {ind$_1$})

Figure 2. Pseudo-code of the base EA for classification. Source: [9].

3. PROPOSALS

The three new approaches are introduced below. All of them take the base algorithm and customize it or add any novel element.

3.1. First methodology

Some parameters concerning either the topology of the EAFFNN model or the EA are distributed by the processing elements of the computation systems. Specifically these parameters are the maximum number of neurons in the hidden layer (*neu*), the maximum

number of generations (*gen*) and the α_2 value that is interrelated with the parametrical mutation. This approach was designed following a master-slave model where the master process fixes a base configuration that is distributed to the slaves which update the configuration once is received by changing a single parameter value. Once all the modifications have been carried out each node, including the master and the slave processes, will execute the new configuration. The first proposal was named Experimental Design Distribution (EDD). Initially, it was developed to train product units and then was extended to train sigmoid units and the adopted denomination was EDDSig (EDD with sigmoid units)

3.2. Second methodology

It is a specialization of the base algorithm. It diversifies the neural network architecture and is divided into two stages. The first stage operates with two populations with different properties that are evolved for a short time measured in number of generations. After that, the best halves of each population are integrated in a new population. The second stage evolves the new population for a complete evolutionary cycle. This approach was called two-stage evolutionary algorithm (TSEA) and designed for product units. The version for sigmoid units received the denomination TSEASig (TSEA with sigmoid units). The pseudo-code of the TSEA appears in Figure 3.

3.3. Third methodology

This is a natural extension of the previous approach to consider a data preparation phase prior to the training of the neural network models by the application of feature selection methods implemented as filter. A good number of feature selectors are applied in an independent fashion to the training set of the problem at hand. The result for every one is a list selected attributes that represent important features depending on the criterion or criteria managed by the filter. This methodology was introduced as two-stage evolutionary algorithm with feature selection (TSEAFS) and was created to operate with product units. Figure 4 depicts the third approach.

4. EXPERIMENTAL RESULTS

This section is devoted to the explanation of the experiments that were carried out along with their configurations.

```
Program: Two-Stage Evolutionary Algorithm
Data: Training set
Input parameters: gen, neu
Output: Best ANN model
1:                                          // First Stage
2:  t ← 0
3:                                          // Population P1
4:  P1(t) ← {ind1, …, ind10000}             // Individuals of P1 have neu nodes in the hidden layer
5:  f1 (P1(t) {ind1, …, ind10000}) ← fitness (P1(t) {ind1, …, ind10000}) // Calculate fitness
6:  P1(t) ← P1(t) {ind1, …, ind10000}       // Sort individuals
7:  P1(t) ← P1(t) {ind1,… ind1000}          // Retain the 1000 best ones
8:                                          // Population P2
9:  P2(t) ← {ind1, …, ind10000}             // Individuals of P2 have neu+1 nodes in the hidden layer
10: f2 (P2(t) {ind1, …, ind10000}) ← fitness (P2(t) {ind1, …, ind10000}) // Calculate fitness
11: P2(t) ← P2(t) {ind1, …, ind10000}       // Sort individuals
12: P2(t) ← P2(t) {ind1,… ind1000}          // Retain the 1000 best ones
13:                                         // Evolution of populations P1 and P2 until 0.1*gen generations
14: for each Pi
15:   current_generation ← 0
16:    while current_generation < 0.1*gen not met do
17:        Pi(t) {ind901,…ind1000} ← Pi(t) {ind1, …, ind100} // Best 10% replace the worst 10%
18:        Pi(t+1) ← Pi(t) {ind1, …, ind900}
19:        Pi(t+1) ← pm (Pi(t+1) {ind1, …, ind90})          // Parametric mutation (10% Pi (t+1))
20:        Pi(t+1) ← sm (Pi(t+1) {ind91, …, ind900})        // Structural mutation (90% Pi (t+1))
21:        fi(Pi(t+1) {ind1, .. ind900}) ← fitness (Pi(t+1) {ind1, …, ind900}) // Evaluate
22:        Pi(t+1) ← Pi(t+1) (ind1, …, ind900) ⋃ Pi(t){ind901, …, ind1000}
23:        Pi(t+1) ← Pi(t+1){ind1, …, ind1000}          // Sort individuals
24:        current_generation ← current_generation + 1
25:    end while
26: end for
27: P(t) ← P1{ind1, …, ind500} ⋃ P2{ind1, …, ind500}    // Individuals of P has [neu, neu+1]
28:                                                     // nodes in the hidden layer
29:  P(t) ← P(t) {ind1, …, ind1000}                     // Sort individuals by fitness: indi > indi+1
30:                                         // Second Stage
31:                                  // Input: gen, neu+1
32:  t ← 0
33:  while stop criterion not met do                    // main loop
34:        P(t) {ind901,…ind1000} ← P(t) {ind1, …, ind100} // Best 10% replace the worst 10%
35:        P(t+1) ← P(t) {ind1, …, ind900}
36:        P(t+1) ← pm (P(t+1) {ind1, …, ind90})        // Parametric mutation (10% P(t+1))
37:        P(t+1) ← sm (P(t+1) {ind91, …, ind900})      // Structural mutation (90% P(t+1))
38:        f(P(t+1) {ind1, .. ind900}) ← fitness (P(t+1) {ind1, …, ind900}) // Evaluate
39:        P(t+1) ← P(t+1) (ind1, …, ind900) ⋃ P(t) {ind901, …, ind1000}
40:        P(t+1) ← P(t+1){ind1, …, ind1000}            // Sort individuals
41:        t ← t+1
42:        last_generation ← t
43:  end while
44:  return best (P(last_generation) {ind1})
```

Figure 3. Pseudo-code of the TSEA for classification. Source: [7].

4.1 Data sets and setting parameters of the algorithms

In different moments of the experimentation we have utilised a great deal of data sets with the three presented methodologies. Most of them are publicly available in the international repository maintained by the University of California at Irvine [12] and others are regarding to real-world complex problems. Specifically, thirty data sets were used as is shown in Table 1: *Appendicitis*, Statlog (*Australian* credit approval), *Balance*, *Breast* Cancer, Breast Tissue (*Breast-t*), Breast *Cancer* Wisconsin, *Cardiotocography*, Statlog (*Heart*), Heart disease Cleveland (*Heart-c*), *HeartY*, *Hepatitis*, *Horse* colic, Thyroid disease (allhypo, *Hypothyroid*), *Ionos* (Ionosphere), *Labor* Relations, *Led24*, *Liver* disorders, *Lymphography*, Thyroid disease (*Newthyroid*), *Parkinsons*, *Pima* Indians diabetes, Steel *Plates* Faults, Molecular Biology (*Promoter* Gene Sequences), *SPECTF*, *Vowel*, *Waveform* database generator (version 2), Wine Quality (*Winequality-red*), *Yeast*, *BTX* and *Listeria* monocytogenes.

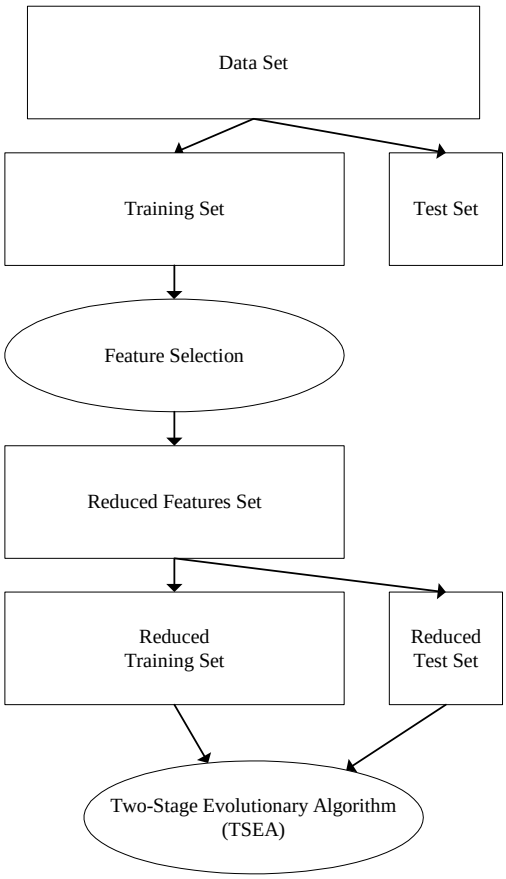


Figure 4. TSEAFS framework. Source: [8].

Data set	#Patterns	#Training patterns	#Test patterns	Attributes	Inputs	Classes
Appendicitis	106	80	26	7	7	2
Australian	690	517	173	14	51	2
Balance	625	469	156	4	4	3
Breast	286	215	71	9	15	2
Breast-t	106	81	25	9	9	6
Cancer	699	525	174	10	9	2
Cardiotocography	2126	1594	532	23	31	3
Heart	270	202	68	13	13	2
Heart-c	303	227	76	13	26	2
HeartY	270	202	68	13	13	2
Hepatitis	155	117	38	19	19	2
Horse	368	276	92	27	83	2
Hypothyroid	3772	2829	943	29	29	4
Ionos	351	263	88	34	34	2
Labor	57	43	14	16	29	2
Led24	3200	200	3000	24	24	10
Liver	345	259	86	6	6	2
Lymphography	148	111	37	18	38	4
Newthyroid	215	161	54	5	5	3
Parkinsons	195	146	49	23	22	2
Pima	768	576	192	8	8	2
Plates	1941	1457	484	27	27	7
Promoter	106	80	26	58	114	2
SPECTF	267	80	187	44	44	2
Vowel	990	528	462	12	11	11
Waveform	5000	3750	1250	40	40	3
Winequality-red	1599	1196	403	11	11	6
Yeast	1484	1112	372	8	8	10
BTX	63	42	21	3	3	7
Listeria	539	305	234	4	4	2

Table 1. Summary of the thirty data sets used in different moments of the experimentation

The experimental design uses the cross validation technique called hold-out that consists of splitting the data into two sets: a training and a test set. The former is employed to train the neural network and the latter is used to test the training process and to measure neural network generalization capability. In our case, the size of the training set is 3N/4 and that of the test set is approximately N/4, where N is the number of patterns in the problem. We have employed a stratified holdout where the two sets are stratified so that the class distribution of

the samples in each set is approximately the same as in the original data set. The proportions do not match in Listeria because the data is prearranged in two sets due to their specific features. The tables with the results report the accuracy of the test set that is the performance with the unseen data.
On relation to the algorithms we have experienced with C4.5 [13], 1-nearest neighbour (1-NN) [14, 15], SVM [16], PART [17], the traditional MLP model [18] with a learning Back-Propagation method (BP), the RBF model [19] and our proposals. Since ANN models like MLP, RBF, EDD, TSEA and TSEAFS depend on a seed the results were averaged from thirty repetitions, as usual, in order to achieve a reliable experimentation.

4.2 Feature selection methods

In the data preparation phase, four different feature selectors were applied that are described in Table 2. These are Correlation-based Feature Selection (CFS), CoNsistency-based feature Selection (CNS), Fast Correlation-Based Feature (FCBF), Selection by Ordered Projection (sp) and Best Incremental ranking search (BI).

Feature selector name	Ranking method	Subset evaluation
-	None	None
spBI_CFS	spBI	CFS
cnBI_CNS	cnBI	CNS
FCBF	Symmetrical Uncertainty	FCBF
BestFirst_CFS	BestFirst	CFS

Table 2. Feature selection methods utilised in TSEAFS

4.3 Results

4.3.1 First methodology

Table 3 reports the results of the EDD approach (last column) along with a good number of well-known machine learning algorithms. EDD gets the highest accuracy average and wins in 7 out of 25 problems, followed by RBF with 6 times.
Next, EDD is compared with EDDSig that includes sigmoid units in the hidden layer. Figure 5 represents the results for each data set with both approaches. According to the plot, EDD is slightly more accurate than EDDSig, whereas the latter is a bit faster than the former.

Data set	Classifier						
	C4.5	**1-NN**	**SVM**	**PART**	**MLP**	**RBF**	**EDD**
Australian	86.71	82.66	**88.44**	84.97	84.10	75.84	87.70
Balance	83.33	77.56	88.46	85.26	93.78	88.27	**95.62**
Breast	**70.42**	64.79	64.79	69.01	60.80	68.78	64.27
Breast-t	52.00	60.00	52.00	44.00	**63.20**	61.20	54.53
Cancer	97.13	97.13	98.28	97.13	97.81	97.20	**98.97**
Cardiotocography	82.71	76.32	**83.65**	82.52	80.75	81.80	81.25
Heart	70.59	73.53	76.47	73.53	74.85	**78.53**	76.23
Heart-c	75.00	76.32	82.89	80.26	84.82	**86.75**	82.36
Hepatitis	84.21	86.84	**89.47**	81.58	84.73	89.30	85.52
Horse	88.04	86.96	88.04	85.87	**88.51**	80.47	86.41
Hypothyroid	**99.15**	90.99	93.85	98.83	94.39	92.83	95.32
Ionos	92.05	90.91	88.64	**95.45**	89.12	92.46	93.06
Labor	85.71	71.43	78.57	**85.71**	69.52	71.67	84.28
Liver	68.60	61.63	58.14	61.63	65.65	57.17	**73.87**
Lymphography	75.68	83.78	**91.89**	75.68	86.58	70.99	76.85
Newthyroid	96.30	94.44	88.89	92.59	97.08	**98.27**	94.81
Parkinsons	71.43	77.55	75.51	75.51	77.62	70.27	**79.66**
Pima	74.48	73.96	78.13	74.48	75.94	77.34	**78.61**
Plates	39.05	49.17	57.02	46.69	53.50	**59.94**	52.82
Promoter	69.23	65.38	**88.46**	53.85	86.03	79.36	60.51
Waveform	74.80	68.96	86.24	76.88	84.85	**87.29**	84.32
Winequality-red	53.85	49.88	59.55	51.36	56.35	57.11	**61.16**
Yeast	54.84	48.39	55.91	56.72	**59.94**	58.31	59.62
BTX	80.95	76.19	61.90	80.95	54.12	**80.95**	79.04
Listeria	85.93	83.70	80.74	86.67	84.49	83.70	**87.45**
Average	86.32	83.18	84.59	85.82	84.30	79.77	**87.58**

Table 3. Global results with a good number of algorithms and the EDD proposal

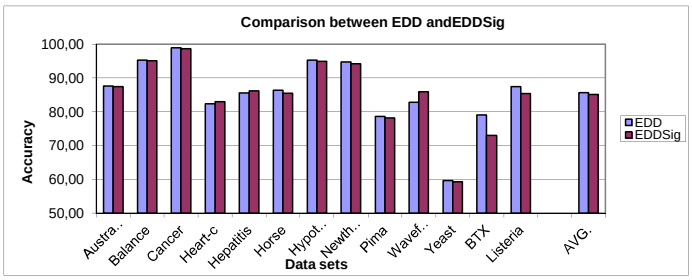


Figure 5. Comparison between EDD and EDDSig.

4.3.2 Second methodology

Table 4 overviews the results using TSEA classifier versus a good number of classical/modern machine learning algorithms. In 11 out of 30 data sets TSEA is the best approach and also the highest accuracy average is achieved.
Figure 6 plots the individual and global results with TSEA and TSEASig. Again, the AFFNN based on product units are more accurate than sigmoid units.

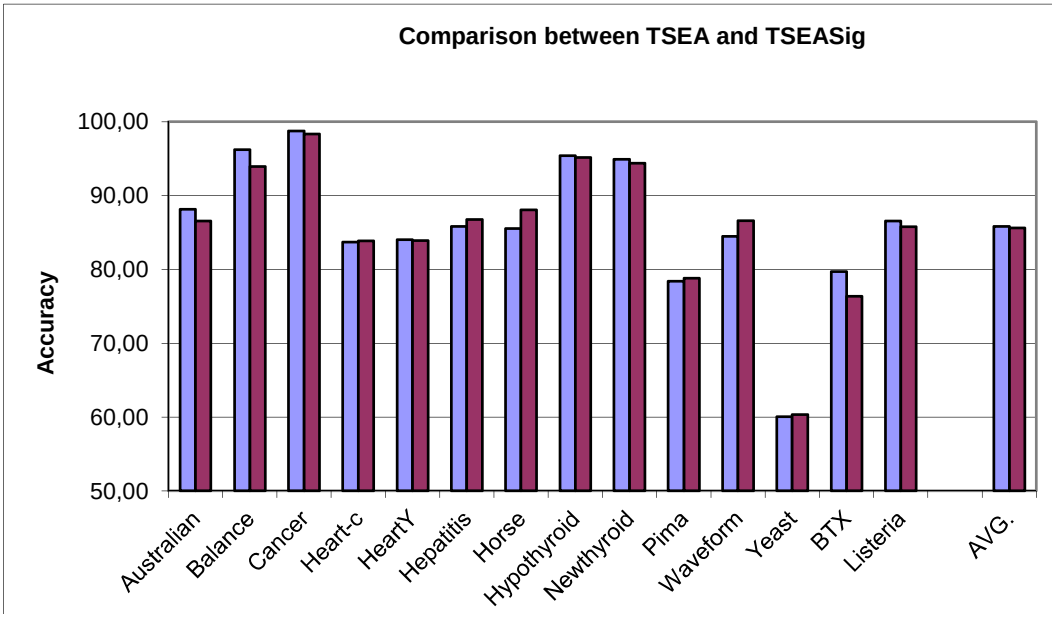


Figure 6. Comparison between TSEA and TSEASig.

4.3.3 Third methodology

Table 5 shows the results with (from second to fifth row) and without (first row) for each dataset by means of the TSEAFS and other typical supervised data mining algorithms. The last part of the table summarizes the averages for each classifier and feature selection method (FSM). The best average for each FSM appears in boldface and the second best in italics. TSEAFS with any of the FSM reaches to better results compared with those obtained without feature selection. However, the best result is provided by spBI_CFS filter which is based on a correlation measure. For the remaining classifiers some improvements may be got and the best trend looks like to be in CFS via spBI or BestFirst as ranking method.

Data sets	Classifier						
	C4.5	**1-NN**	**SVM**	**PART**	**MLP**	**RBF**	**TSEA**
Appendicitis	73.08	69.23	**84.62**	73.08	76.92	74.67	81.66
Australian	86.71	82.66	88.44	84.97	84.10	75.84	**88.68**
Balance	83.33	77.56	88.46	85.26	93.78	88.27	**96.20**
Breast	**70.42**	64.79	64.79	69.01	60.80	68.78	65.96
Breast-t	52.00	60.00	52.00	44.00	**63.20**	61.20	55.33
Cancer	97.13	97.13	98.28	97.13	97.81	97.20	**98.98**
Cardiotocography	82.71	76.32	**83.65**	82.52	80.75	81.80	81.69
Heart	70.59	73.53	76.47	73.53	74.85	**78.53**	77.45
Heart-c	75.00	76.32	82.89	80.26	84.82	**86.75**	83.68
HeartY	86.76	79.41	**86.76**	83.82	84.29	83.79	84.01
Hepatitis	84.21	86.84	**89.47**	81.58	84.73	87.01	87.01
Horse	88.04	86.96	88.04	85.87	**88.51**	80.47	86.59
Hypothyroid	**99.15**	90.99	93.85	98.83	94.39	92.83	95.37
Ionos	92.05	90.91	88.64	**95.45**	89.12	92.46	93.22
Labor	85.71	71.43	78.57	85.71	69.52	71.67	**86.90**
Led24	**65.67**	39.43	58.97	55.80	57.48	55.14	51.03
Liver	68.60	61.63	58.14	61.63	65.65	57.17	**74.61**
Lymphography	75.68	83.78	**91.89**	75.68	86.58	70.99	79.37
Newthyroid	96.30	94.44	88.89	92.59	97.08	**98.27**	94.88
Parkinsons	71.43	77.55	75.51	75.51	77.62	70.27	**78.09**
Pima	74.48	73.96	78.13	74.48	75.94	77.34	**79.21**
Plates	39.05	49.17	57.02	46.69	53.50	**59.94**	51.46
Promoter	69.23	65.38	**88.46**	53.85	86.03	79.36	68.20
SPECTF	67.91	61.50	72.19	70.59	71.28	**76.19**	61.56
Vowel	39.39	**48.48**	45.45	38.53	45.87	47.25	47.18
Waveform	74.80	68.96	86.24	76.88	84.85	**87.29**	84.46
Winequality-red	53.85	49.88	59.55	51.36	56.35	57.11	**61.11**
Yeast	54.84	48.39	55.91	56.72	59.94	58.31	**60.16**
BTX	80.95	76.19	61.90	80.95	54.12	80.95	**81.11**
Listeria	85.93	83.70	80.74	86.67	84.49	83.70	**87.68**
Average	74.83	72.22	76.80	73.97	76.15	76.09	**77.39**

Table 4. Global results with a good number of algorithms and the TSEA proposal

Data set	Filter	Classifier						
		C4.5	**1-NN**	**SVM**	**PART**	**MLP**	**RBF**	**TSEAFS**
Appendicitis	-	73.08	69.23	84.62	73.08	76.92	74.67	81.66
	spBI_CFS	80.77	69.23	76.92	80.77	78.85	80.00	82.82
	cnBI_CNS	76.92	57.69	76.92	76.92	77.95	77.05	80.89
	FCBF	80.77	80.77	80.77	80.77	80.77	74.49	80.38
	BestFirst_CFS	80.77	65.38	76.92	80.77	79.23	79.36	81.79
Breast	-	70.42	64.79	64.79	69.01	60.80	68.78	65.96
	spBI_CFS	69.01	70.42	66.20	71.83	69.01	67.46	69.85
	cnBI_CNS	69.01	70.42	64.79	69.01	69.01	69.01	69.01
	FCBF	69.01	70.42	64.79	69.01	69.53	67.65	69.10
	BestFirst_CFS	69.01	70.42	66.20	71.83	69.01	67.46	69.01
Breast-t	-	52.00	60.00	52.00	44.00	63.20	61.20	55.33
	spBI_CFS	56.00	52.00	60.00	44.00	65.33	58.67	54.40
	cnBI_CNS	52.00	52.00	64.00	52.00	67.20	61.20	57.87
	FCBF	48.00	48.00	56.00	48.00	65.60	60.40	60.93
	BestFirst_CFS	68.00	56.00	60.00	56.00	65.47	60.67	59.47
Cardiotocography	-	82.71	76.32	83.65	82.52	80.75	81.80	81.69
	spBI_CFS	77.07	81.77	81.20	82.52	81.94	83.40	85.26
	cnBI_CNS	75.19	63.91	75.19	75.00	68.29	65.91	76.71
	FCBF	77.82	81.20	81.20	77.26	80.13	80.50	81.55
	BestFirst_CFS	78.38	80.45	81.39	81.20	80.86	84.12	82.38
Heart	-	70.59	73.53	76.47	73.53	74.85	78.53	77.45
	spBI_CFS	73.53	73.53	76.47	77.94	72.50	78.24	77.69
	cnBI_CNS	72.06	75.00	76.47	75.00	74.85	77.60	78.57
	FCBF	73.53	70.59	77.94	75.00	74.90	76.37	75.34
	BestFirst_CFS	73.53	73.53	76.47	77.94	72.50	78.53	77.69
Hepatitis	-	84.21	86.84	89.47	81.58	84.73	89.30	87.01
	spBI_CFS	84.21	89.47	86.84	84.21	87.28	89.30	90.78
	cnBI_CNS	89.47	84.21	89.47	84.21	84.21	88.42	87.45
	FCBF	89.47	84.21	89.47	86.84	87.72	90.79	91.05
	BestFirst_CFS	84.21	89.47	86.84	84.21	87.28	89.30	90.78
Labor	-	85.71	71.43	78.57	85.71	69.52	71.67	86.90
	spBI_CFS	85.71	71.43	78.57	85.71	64.29	71.43	96.19
	cnBI_CNS	85.71	64.28	78.57	78.57	78.57	64.29	88.33
	FCBF	85.71	78.57	71.43	78.57	71.43	64.29	90.95
	BestFirst_CFS	85.71	64.29	71.43	85.71	57.62	71.43	89.76
Led24	-	65.67	39.43	58.97	55.80	57.48	55.14	51.03
	spBI_CFS	68.10	67.90	67.93	68.50	68.44	67.42	68.30
	cnBI_CNS	68.10	67.90	67.93	68.50	68.44	67.42	68.30
	FCBF	68.10	67.90	67.93	68.50	68.44	67.42	68.30
	BestFirst_CFS	68.10	67.90	67.93	68.50	68.44	67.42	68.30
Lymphography	-	75.68	83.78	91.89	75.68	86.58	70.99	79.37
	spBI_CFS	88.29	78.38	83.78	70.27	73.24	68.92	79.09
	cnBI_CNS	75.68	70.27	78.38	64.86	71.89	75.77	80.36
	FCBF	81.08	75.68	81.08	70.27	74.50	69.64	80.61

	BestFirst_CFS	81.08	81.08	81.08	64.86	80.45	69.16	80.90
Parkinsons	-	71.43	77.55	75.51	75.51	77.62	70.27	78.09
	spBI_CFS	75.51	79.59	75.51	77.55	81.56	77.75	78.77
	cnBI_CNS	79.59	79.59	75.51	81.63	75.65	73.47	80.13
	FCBF	81.63	73.47	79.59	77.55	84.83	80.27	82.79
	BestFirst_CFS	73.47	81.63	75.51	79.59	83.13	77.55	79.25
Pima	-	74.48	73.96	78.13	74.48	75.94	77.34	79.21
	spBI_CFS	76.04	74.48	77.60	76.04	78.18	79.17	79.72
	cnBI_CNS	74.48	67.19	78.65	74.48	76.89	75.64	78.54
	FCBF	76.04	67.71	79.17	76.04	79.01	80.28	79.53
	BestFirst_CFS	76.04	67.71	79.17	76.04	78.73	80.28	79.53
Plates	-	39.05	49.17	57.02	46.69	53.50	59.94	51.46
	spBI_CFS	40.50	51.24	51.03	46.90	56.71	64.08	53.81
	cnBI_CNS	38.22	50.62	55.17	44.63	55.24	62.17	56.93
	FCBF	44.63	43.18	45.04	49.79	52.85	55.88	51.87
	BestFirst_CFS	54.75	47.31	51.65	51.65	57.33	59.88	48.84
Promoter	-	69.23	65.38	88.46	53.85	86.03	79.36	68.20
	spBI_CFS	73.08	57.69	84.62	80.77	84.49	83.46	85.64
	cnBI_CNS	80.77	57.69	84.62	76.92	75.64	85.00	80.00
	FCBF	73.08	76.92	73.08	80.77	78.21	79.74	75.12
	BestFirst_CFS	73.08	69.23	73.08	80.77	76.28	80.00	74.74
SPECTF	-	67.91	61.50	72.19	70.59	71.28	76.19	61.56
	spBI_CFS	66.84	59.36	72.19	72.19	73.67	76.24	73.85
	cnBI_CNS	65.78	60.96	70.05	65.78	70.02	74.60	72.07
	FCBF	67.91	59.36	65.24	64.71	69.57	74.58	73.99
	BestFirst_CFS	66.84	57.75	73.26	70.05	72.26	74.63	73.76
Vowel	-	39.39	48.48	45.45	38.53	45.87	47.25	47.18
	spBI_CFS	45.24	46.54	54.33	44.59	52.79	43.12	54.31
	cnBI_CNS	38.53	51.52	48.48	40.04	52.05	44.73	47.65
	FCBF	41.56	46.97	41.34	36.58	44.97	46.95	49.45
	BestFirst_CFS	45.24	46.54	54.33	44.59	52.79	43.12	54.31
Waveform	-	74.80	68.96	86.24	76.88	84.85	87.29	84.46
	spBI_CFS	74.40	75.36	86.88	77.04	83.21	82.24	86.89
	cnBI_CNS	74.40	76.64	87.12	79.68	86.27	82.22	86.02
	FCBF	74.72	69.12	78.80	74.00	77.57	76.88	80.67
	BestFirst_CFS	74.40	75.36	86.88	77.04	83.21	82.24	86.89
Winequality-red	-	53.85	49.88	59.55	51.36	56.35	57.11	61.11
	spBI_CFS	50.87	48.88	59.80	52.11	59.36	59.00	61.63
	cnBI_CNS	50.12	49.63	58.81	52.85	57.04	59.19	61.47
	FCBF	51.36	50.37	59.31	49.13	59.64	59.17	61.65
	BestFirst_CFS	50.87	48.88	59.80	52.11	59.36	59.00	61.63
Yeast	-	54.84	48.39	55.91	56.72	59.94	58.31	60.16
	spBI_CFS	53.49	48.92	54.03	54.84	60.20	58.48	60.06
	cnBI_CNS	54.03	49.46	54.84	54.30	60.20	58.91	60.78
	FCBF	52.69	48.12	51.61	52.96	58.96	58.78	58.29
	BestFirst_CFS	54.03	49.46	54.84	54.30	60.20	58.91	60.78

Average	-	66.95	64.92	72.16	65.86	70.34	70.29	69.88
	spBI_CFS	68.81	66.46	*71.88*	69.32	71.73	71.58	**74.39**
	cnBI_CNS	67.78	63.83	*71.39*	67.47	70.52	70.14	**72.84**
	FCBF	68.73	66.25	69.10	67.54	*71.03*	70.23	**72.87**
	BestFirst_CFS	69.86	66.25	70.93	69.84	*71.34*	71.28	**73.32**

Table 5. Global results with a good number of algorithms and the TSEAFS proposal

5. CONCLUSIONS

This paper introduced three approaches to trained artificial feed-forward neural networks with an evolutionary strategy. Roughly speaking, the two first ones (EDD and TSEA, or their extensions such as EDDSig or TSEASig) operate in the raw data without any kind of pre-processing based on feature selection. On the other hand, the third methodology (TSEAFS) includes at the beginning of the algorithm a data preparation stage to choose a subset of the original features' space.

The results reported that models containing product units such as EDD and TSEA have a better behaviour in terms of performance than the approaches based on sigmoid units. As a counterpart, the latter ones are a bit faster than the former ones. TSEAFS is a very powerful classifier which includes a pre-processing state to select the most relevant features according to linear correlation or consistency measures. However, the correlation is a metric that fits better to the product unit neural networks trained with the two-stage evolutionary programming algorithm that was detailed throughout the pages of this work.

REFERENCES

[1] Michie, D., Spiegelhalter, D.J., Taylor, C.C. Machine Learning, Neural and Statistical *Classification*, Ellis Horwood, 1994.

[2] Duda, R.O., Hart, P.E., Stork, D. *Pattern Classification*, second ed., Wiley, 2001

[3] Yao, X. "Evolving artificial neural networks". *Proceedings of the IEEE* Vol. 87(9), pp. 1423-1447, 1999.

[4] Lippmann, R.P. Pattern classification using neural networks, *IEEE Communications Magazine* Vol. 27, pp. 47-64, 1989.

[5] Tallón-Ballesteros, A.J., Gutiérrez-Peña, P.A., Hervás-Martínez, C., "Distribution of the search of Evolutionary Product Unit Neural Networks for Classification", *AC 2007, Proceedings of the IADIS Internacional Conference Applied Computing*. IADIS, Salamanca - Spain, pp. 266-273, 2007.

[6] Tallón-Ballesteros, A.J., Hervás-Martínez, C., Riquelme, J.C., Ruiz, R. "Improving the accuracy of a two-stage algorithm in evolutionary product unit neural networks for classification by means of feature selection". *IWINAC 2011, Proceedings of the 4th International work-conference on the Interplay Between Natural and Artificial Computation*. Lecture Notes in Computer Science (LNCS) 6687 (vol. II). Springer, La Palma - Spain, pp. 381-390, 2011.

[7] Tallón-Ballesteros, A.J., Hervás-Martínez, C. "A two-stage algorithm in evolutionary product unit neural networks for classification", *Expert Systems with Applications* Vol. 38(1), pp. 743-754, 2011.

[8] Tallón-Ballesteros, A.J., Hervás-Martínez, C., Riquelme, J.C., Ruiz, R. “Feature selection to enhance a two-stage evolutionary algorithm in product unit neural networks for complex classification problems”, *Neurocomputing* Vol. 114, pp. 107-117, 2013.
[9] Tallón-Ballesteros, A.J., Hervás-Martínez, C., Gutiérrez, P.A. “An extended approach of a two-stage evolutionary algorithm in artificial neural networks for multiclassification tasks”. *Innovations in Intelligent Machines – 3: Contemporary Achievements in Intelligent Systems*. Series: Studies in Computational Intelligence, vol. 442. Springer-Verlag Berlin Heidelberg, Germany, pp. 139-153, 2013.
[10] Yao, X., Liu, Y. “A new evolutionary system for evolving artificial neural networks”, *IEEE Transactions on Neural Networks* Vol. 8(3), pp. 694-713, 1997.
[11] Angeline, P.J., Saunders G.M., Pollack, J.B. “An evolutionary algorithm that constructs recurrent neural networks”, *IEEE Transactions on Neural Networks* Vol. 5(1), pp. 54-65, 1994.
[12] Bache, K., Lichman, M. "UCI machine learning repository", *URL: http://archive. ics. uci. edu/ml*, 2013.
[13] Quinlan, J. *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, CA, 1993.
[14] Cover, T., Hart, P. “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory* Vol. 13(1), pp. 21-27, 1967.
[15] Aha, D., Kibler, D., Albert, M.K. ”Instance-based learning algorithms”, *Machine Learning* Vol. 6, pp. 37-66, 1991.
[16] Vapnik, V. *The Nature of Statistical Learning Theory*, Springer, New York, 1995.
[17] Frank, E., Witten, I.H. (1998) Generating accurate rule sets without global optimization. *ICML 1998, Proceedings of the fifteenth international conference on machine learning*. Madison, Wisconsin, USA: Morgan Kaufmann, pp. 144-151, 2010.
[18] Bishop, C.M. *Neural networks for pattern recognition*, Oxford University Press, New York, 1995.
[19] Howlett, R.J., Jain L.C. *Radial Basis Function Networks 1: Recent Developments in Theory and Applications*, Springer, Heidelberg, Germany, 2001.

SYMCOMP 2015
Faro, March 26-27, 2015

DECISION TREES UNDER FEATURE SELECTION VIA SCATTER SEARCH IN CLASSIFICATION PROBLEMS

Alberto Ibiza-Granados[1] and Antonio J. Tallón-Ballesteros[2]*

1: Higher Technical School of Computer Science Engineering
University of Seville (Spain)
W/o number Reina Mercedes Av. 41012-Seville (Spain)
e-mail: albibigra@alum.us.es, web: http://www.informatica.us.es

2: Department of Languages and Computer Systems
Higher Technical School of Computer Science Engineering
University of Seville (Spain)
W/o number Reina Mercedes Av. 41012-Seville (Spain)
e-mail: atallon@us.es web: http://www.lsi.us.es

Keywords: decision trees, feature selection, scatter search, classification and data preparation

Abstract *This paper explores the performance of some classifiers based on decision trees with a prior data preparation that is carried out by a scatter search procedure to get promising feature subsets using measures with regard to the correlation or consistency between each feature and the class. The experimentation has been performed with six binary data sets from different domains such as Physics, Finances or Medicine, with a size up to approximately eight hundred samples and a dimension between eight and eighty three. Two well-known classifiers based on decision trees have been used after the data pre-processing phase. The behaviour of the filter may depend in the population size of the reference set containing the solutions that in our case are the subsets of attributes, thus a group of three values has been considered for this parameter that is common to all the problems at hand. According to the obtained results in terms of accuracy, the main conclusion is that this parameter is sturdy and presents a homogenous operation with different attribute selection approaches. The most suitable value for the aforementioned parameter is ten for each decision tree in conjunction with the filter based on correlation, whereas the default value is pertinent in the case of the consistency measure. From a global point of view, the feature selection using correlation measures reached slightly better average values versus that based on consistency.*

1. INTRODUCTION

Classifiers can be divided into several kinds [1] such as Decision trees, Bayes classifiers, rule-based classifiers [2], artificial neural networks [3] or classifiers based on nearest neighbours [4]. Some issues like the interpretability, the noise tolerance or the prediction ability are some criteria that may help the user to choose one specific type. According to the literature, the decision trees are very widespread due to their power and the simplicity of the classification models that are able to get [5]. This fact motivated us to tackle this research. On the other hand, the increase of the number of instances of the problem may cause that the required time to create the tree would be computational expensive or the size of it could be a cons. Data preparation could aid to diminish the input data to the algorithm by reducing the dimensionality or the sample size regarding the case of study at hand.

A decision tree [6] may be defined as a simple structure based on a tree that can be used as a classifier. Each non-leaf or internal node is associated with a decision and the leaf nodes are generally associated with an outcome or class label. Each internal node tests one or more attribute values leading two or more links or branches. Each link in turn is associated with a possible value of the decision. These links are mutually distinct and collectively exhaustive. This means that it is possible to follow only one of the links and all possibilities will be taken care of; there is a link for each possibility.

This paper focuses on the application of feature selection previously to the training of the classifiers. The selection of relevant attributes may be a complex task and the full exploration of the search space could be prohibitive. This is the reason to use an evolutionary algorithm to get solutions with good quality in a reasonable amount of time.

The rest of this paper is organized as follows. Section 2 introduces the feature selection and the scatter search. Section 3 details the experimentation. Then, Section 4 reports the results. Finally, Section 5 draws some conclusions.

2. FEATURE SELECTION AND SCATTER SEARCH

Basically, there are two possibilities for feature selection. The former is to select the best individual features and the latter is to choose a subset of features with good quality. This paper lays emphasis on the last approach. Different kind of measures to evaluate the goodness level of a subset may be used. However, two of the best well-known are based on correlation or consistency measures.

Discovering the best subset may be a task with a high computational cost, especially when the number of features is very high. Instead of doing an exhaustive exploration in the search space a meta-heuristic could guide the steps to follow. Scatter search is an evolutionary algorithm or population-based algorithm, which was first proposed by F. Glover in the seventies of the previous century. Unlike most of other evolutionary algorithms, scatter search stores solutions in a so called Reference Set and constructs new solutions by combining existing ones. There are a good number of pros to use scatter search but we would like to stress that are able to solve both combinatorial and continuous optimization problems.

3. EXPERIMENTATION

Table 1 depicts the data sets coming from different domains such as Physics, Finances or Medicine that were considered to conduct the experiments. All of them are freely available in the UCI (University of California at Irvine) machine learning repository (http://archive. ics. uci. edu/ml). The experimental design follows a stratified hold-out cross validation procedure which splits the problem in two sets: one for training and another for test containing one and three quarters of the samples, respectively.

Table 2 represents the methods of feature selection that were used in the experimentation. Last column includes an abbreviated name that is referred hereinafter. In order to do a fair experimentation we only select attributes of the training set. On relation with the parameters of the scatter search, the single parameter that we have changed is the population size that means the number of solutions to be saved in the reference set. Three possible values where tested for this parameter, namely, -1 (by default), 5 or 10. In the default case, as many solutions as the half of the number of attributes are stored in the reference set. Table 3 includes the concrete number of solutions that are kept in each situation.

Problem	Instances	Features	Labels
Card	690	51	2
Heart	303	26	2
Hepatitis	155	19	2
Horse	368	83	2
Ionosphere	351	33	2
Pima	768	8	2

Table 1. Classification problems

Feature Selector	Search method	Abbreviation
CFS	Scatter Search	CFS-SS
CNS	Scatter Search	CNS-SS

Table 2. Feature selectors

Concerning the classifiers, ADTree (Alternating Decision Tree, [7]) and BFTree (Best-First decision Tree classifier, [8-9]) were applied to every problem once the data preparation phase took place. We have references of the latter classification algorithm from a previous work [10] and the conclusions that there were achieved are concretely that is very fast and got promising results in the context of feature selection. For each classifier we include in the forthcoming tables the accuracy measure obtained in the test set.

Problem	Features	Parameter values		
		P=-1	P=5	P=10
		Solutions		
Card	51	25	5	10
Heart	26	13	5	10
Hepatitis	19	9	5	10
Horse	83	41	5	10
Ionosphere	33	16	5	10
Pima	8	4	5	10
Average	36.7	18	5	10

Table 3. Translation of parameter values in each problem

4. RESULTS

Table 4 reports the test results obtained with BFTree that is implemented in Weka tool. On the left side, there are the results concerning CFS-SS and on the right side those about CNS-SS. Accuracies and rankings are specified for each type of feature selection algorithm. In the mid of the table averages are included. In the next two rows, it is represented the number of times, excluding ties, that a dataset within a column gets the best result inside each group of three with the same feature selection method. Best results are in boldface. In the following row the best approach for each side is stressed. And the last row shows the number of wins that each of the aforementioned best options is better than the other. The individual results that fulfil this condition are written in italics and underlined. On the one hand, for CFS-SS the option with a best average and ranking is obtained with the value 10 for P parameter. On the other hand, regarding CNS-SS the best option is P=-1 because the global average is the highest and the ranking is very close to those got with P=10. The comparison between the best ones is in favour of CFS-SS due to the number of wins and the average.

Table 5, following the same criteria as the previous one, shows the test results for ADTree classifier. For CFS-SS the best ranking and average is P=10, by coincidence the same as in the previous classifier. CNS-SS achieves the best ranking and average for the case P=-1; now, the times that get the best result is greater than the two remaining configurations. The best of the two best ones is CNS-SS with P=-1 reaching three times the best individual result.

Table 6 compares the two best approaches that were obtained so far. According to the average, both pairs (classifier, feature selection) with their suitable values for the parameters are just very close. However, the number of wins is greater for the pair (BFTree, CFS-SS with P=10).

Problem	**CFS-SS**						**CNS-SS**					
	P=-1	P=5	P=10	P=-1	P=5	P=10	P=-1	P=5	P=10	P=-1	P=5	P=10
	Accuracy			Ranking			Accuracy			Ranking		
Card	57.22	89.59	*89.59*	3.0	1.5	1.5	86.70	**89.59**	85.55	2.0	1.0	3.0
Heart	77.63	77.63	*77.63*	2.0	2.0	2.0	76.31	73.68	76.31	1.5	3.0	1.5
Hepatitis	84.21	84.21	84.21	2.0	2.0	2.0	*86.84*	**89.47**	86.84	2.5	1.0	2.5
Horse	90.22	90.22	*90.22*	2.0	2.0	2.0	83.69	81.52	**86.96**	2.0	3.0	1.0
Ionosphere	92.04	90.91	92.04	1.5	3.0	1.5	***93.18***	86.36	87.50	1.0	3.0	2.0
Pima	75.52	75.52	*75.52*	2.0	2.0	2.0	72.39	74.48	74.48	3.0	1.5	1.5
Average	79.47	84.68	84.87	2.1	2.1	1.8	83.19	82.52	82.94	2.0	2.1	1.9
#Best	-	-	-				1	2	1			
#Worst	-	-	-				1	3	1			
Global quality within CFS/CNS			Best				Best					
Wins CFS vs CNS			4				2					

Table 4. Test results with BFTree via Scatter Search

Problem	**CFS-SS**						**CNS-SS**					
	P=-1	P=5	P=10	P=-1	P=5	P=10	P=-1	P=5	P=10	P=-1	P=5	P=10
	Accuracy			Ranking			Accuracy			Ranking		
Card	67.05	88.44	*88.44*	3.0	1.5	1.5	83.24	**89.59**	86.70	3.0	1.0	2.0
Heart	78.95	78.95	78.95	2.0	2.0	2.0	*84.21*	81.58	84.21	1.5	3.0	1.5
Hepatitis	86.84	86.84	*86.84*	2.0	2.0	2.0	84.21	84.21	84.21	2.0	2.0	2.0
Horse	88.04	88.04	88.04	2.0	2.0	2.0	***89.13***	81.52	85.87	1.0	3.0	2.0
Ionosphere	92.04	90.91	92.04	1.5	3.0	1.5	**92.04**	87.5	85.23	1.0	2.0	3.0
Pima	74.48	74.48	74.48	2.0	2.0	2.0	***75.52***	73.96	73.96	1.0	2.5	2.5
Average	81.23	84.61	84.80	2.1	2.1	1.8	84.73	83.06	83.36	1.6	2.3	2.2
#Best	-	-	-				3	1	-			
#Worst	1	1	-				1	2	1			
Global quality within CFS/CNS			Best				Best					
Wins CFS vs CNS			2				3					

Table 5. Test results with ADTree via Scatter Search

Problem	**BFTree**	**ADTree**
	CFS-SS (P=10)	CNS-SS (P=-1)
Card	**89.59**	83.24
Heart	77.63	**84.21**
Hepatitis	84.21	84.21
Horse	**90.22**	89.13
Ionosphere	92.04	92.04
Pima	75.52	75.52
Average	84.87	84.73
#**Best**	2	1

Table 6. Comparison results: BFTree versus ADTree

5. CONCLUSIONS

This paper assessed two classification algorithms based on decision trees, BFTree and ADTree, with a prior data pre-processing phase using a scatter search method in the context of the feature selectors CFS and CNS on six problems from the UCI data mining repository with a number of attributes between 8 and 83. Broadly speaking, given a classifier based on decision tress the parameter regarding the population size of the reference set has a heterogeneous behaviour for different feature selection algorithms. On the other hand, bearing in mind BFTree and ADTree classifiers, the best parameter values are the same for a particular kind of feature selection strategy such as CFS or CNS. Finally, the two best configurations were achieved, in this order, with the pairs (BFTree, CFS-SS, P=10) and (ADTree, CNS-SS, P=-1).

REFERENCES

[1] Dougherty, G. *Pattern recognition and classification: an introduction*. Springer Science & Business Media, 2012.

[2] Frank, E., Witten, I.H. Generating accurate rule sets without global optimization. *ICML 1998, Proceedings of the fifteenth international conference on machine learning*. Morgan Kaufmann, Madison, Wisconsin - USA, pp. 144-151, 1998.

[3] Bishop, C.M. *Neural networks for pattern recognition*, Oxford University Press, New York, 1995.

[4] Aha, D., Kibler, D., Albert, M.K. ”Instance-based learning algorithms”, *Machine Learning* Vol. 6, pp. 37-66, 1991.

[5] Michie, D., Spiegelhalter, D.J., Taylor, C.C. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[6] Murty, M.N., Devi, V.S. *Pattern recognition: An algorithmic approach*. Springer Science & Business Media, 2011.

[7] Manber, U., Tompa, M. “Probabilistic, nondeterministic, and alternating decision trees (preliminary version)”. *Proceedings of the fourteenth annual ACM symposium on Theory of*

computing. ACM, pp. 234-244, 1982.

[8] Shi, H. *Best-first decision tree learning*. PhD thesis, University of Waikato, Hamilton, NZ, 2007.

[9] Friedman, J., Hastie, T., Tibshirani, R. "Additive logistic regression: A statistical view of boosting", *Annals of statistics* Vol. 28(2), pp. 337-407, 2000.

[10] Tallón-Ballesteros, A.J., Benavides-Vallejo, J.E. "Attribute selection from Naïve Bayes classifiers". *CMMSE 2014, Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering*. CMMSE, Rota – Spain, pp. 1417-1423, 2014.

SYMCOMP 2015
Faro, March 26-27, 2015

Reconstruction of Surfaces from Unstructured Points Clouds, using Compactly-Supported Radial Basis Functions

G.M.S. Bernardo[1,2]* and M.A.R. Loja[1,2]

1: GI-MOSM, Grupo de Investigação em Modelação e Optimização de Sistemas Multifuncionais
ISEL, IPL - Instituto Superior de Engenharia de Lisboa
Av. Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal

2: LAETA, IDMEC - Instituto Superior Técnico
Universidade de Lisboa,
Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal

e-mail: goncalo.bernardo@tecnico.ulisboa.pt , amelialoja@dem.isel.ipl.pt

Keywords: Points cloud, laser scanning, surface reconstruction, meshless method, compactly-supported radial functions, partition of unity

Abstract *The need for a relevant viable approach to fit point clouds obtained by 3D laser scanning, to a desirable surface, has been object of a substantial research effort and progress in the past two decades in a wide range of scientific and technological fields. However, this task is far from being a trivial task. First, because of the randomness of the sampled points obtained, which in most cases count with additional noise points. Secondly, in point clouds it is frequent to find lacks of data, leading to the existence of holes in the surface. As far as it is possible to know, all the methods used to achieve the fitting surfaces, present different undesirable behaviours, under different conditions.*

In the present work we present a hybrid method to reconstruct the surfaces associated to synthetic point clouds randomly generated. Parametric studies are carried out to illustrate and characterize the performance of the different techniques implemented..

1. INTRODUCTION

The 3D acquisition devices availability has been increasing substantially in the last years, which results in the widespread dissemination of 3D point clouds of sampled real-world objects. This reality leads and denotes the importance of giving more and more attention to the research of efficient and robust approaches related to surface reconstruction from 3D point

clouds. Solving those kind of problems has become relevant in computer graphics and computer vision, which, in many cases, are not trivial problems, due to the fact of the 3D acquisition process typically produce point clouds which are incomplete, noisy, and non-uniformly sampled. That nature of 3D point clouds grant to the modelling processes and methods requirements associated with pre-processing techniques used to eliminate as much as possible the sampling errors, varying sampling density, and registration errors, i.e., find feasible modelling processes departing from unorganized 3D laser scanning point clouds. To note that concerning to the actual state of the art, one is far from getting an immediate digital representation of the physical surface/component by using an entirely automatic procedure.

Concerning to 3D point clouds acquisition techniques, we can consider digital images, laser scanning point clouds or other optical data sources. Considering the availability of this geometrical information, the subsequent matter is then related to its processing.

From the literature research carried out it is possible to notice the growing attention paid to surfaces modelling issues, through the use of mathematical models to approximate the physical object/structure [1]. In some cases when the object requires a high level of detail, a large number of small primitives, is needed to model a small region of the surface description [2,3]. Achieving these surface's models is fundamental in different scientific areas that can go from the architectural heritage conservation to numerous science and engineering fields, as they provide the possibility to reconstruct real structures [4].

There are several possible techniques to obtain surfaces from ordered point clouds which, typically, find a single curve segment that approximates or interpolates the given points, fitting the curve to the points by minimizing an error criterion [5]. If points are ordered, piecewise polynomial curves can also be fitted to them. Difficulties arise however when the points are not ordered. In recent years some researchers have proposed approaches to obtain surfaces from unorganized points. Among them, we can refer [6] which developed direct methods that are dimension-independent, fitting algebraic surfaces to a set of points. [3] proposed an approach that uses implicit polynomial curves and surfaces to represent 2D and 3D data and objects, showing to be able to smooth noisy data and to interpolate through sparse or missing data. Another work focused on the reconstruction of point set surfaces from point clouds, based on the method of moving least squares was proposed by [7]. Further approaches following the idea of locally polynomial surface patches to confined point neighborhoods are proposed in [8].

Delaunay/Voronoi diagrams were considered in the context of surface reconstruction considering noise existence. This was done by [9] using some of the principles of the power crust algorithm. Also using Delaunay triangulation, [10] considered a volume-based method to obtain shape minimal representations. It is known however, that methods that use polynomials to fit the data, can present inconvenient because an invertible system which uniquely defines the interpolating function is not guaranteed for all positions of the interpolation nodes. Additionally it can also present spurious bumps and wiggles.

Hoppe et al. [11] described an algorithm for the reconstruction of a polyhedral surface from an unorganized set of points, which is based on region growing. They used a plane that is fitted to a neighbourhood around each data point, providing an estimate of the surface normal for the point. The surface normals are propagated using a minimal spanning tree, and

then a signed distance function is constructed in small vicinities around the data points. Also based on region growing, we can refer Lee and Medioni's [12] tensor voting method, which is similar in that neighbouring points are used to estimate the orientations of data points. In [13], the surface is reconstructed by growing planar, edge, and point features until they encounter neighbouring features. Both methods described above are sensitive to noise in the data, because they rely on good estimates for the normal vector at each data point.

Muraki and his colleagues [14] propose to apply a "blobby" model to fit scattered points. This model describes an isosurface of a scalar field produced by a number of potential field generating primitives. The number and location of the primitives are found by minimization of an "energy function". This method is extremely computationally expensive.

When we refer to surface reconstruction, including range data merging and mesh reconstruction, region growing, and algorithms based on algebraic fitting, one can be mentioned the work develop by [15] and [16], where they constructed signed distance functions from range images and obtained a manifold surface by isosurface extracting. [17] and [18] merged triangulations of the range points, but it is important to note that their methods require range data using structured light that is much more accurate than can be measured passively using photographs alone.

Regarding to the issue related to seeking an interpolant that fit well in an amount of unorganized data points, another several algorithms based on computational geometry construct a collection of simplexes that form the shape or surface from a set of unorganized points can be referred. These methods exactly interpolate the data - the vertices of the simplexes consist of the given data points. A consequence of this is that noise and aliasing in the data become embedded in the reconstruction surface. Of such methods, one can mention the Alpha Shapes [19], the Crust Algorithm [20] and the Ball-Pivoting algorithm [21]. In Alpha Shapes, the shape is carved out by removing simplexes of the Delaunay triangulation of the point set. A simplex is removed if its circumscribing sphere is larger than the alpha ball. In the Crust algorithm, Delaunay triangulation is performed on the original set of points along with Voronoi vertices that approximate the medial axis of the shape. The resulting triangulation distinguishes triangles that are part of the object surface from those that are on the interior because interior triangles have a Voronoi vertex as one of their vertices. Both the Alpha Shapes and Crust algorithms need no other information then the locations of the data points and perform well on dense and precise data sets. The collection of simplexes is not manifold surface, and extraction of such a surface is a non-trivial post-processing task. The Ball-Pivoting algorithm is a related method that avoids non-manifold constructions by growing a mesh from an initial seed triangle that is correctly oriented. Starting with the seed triangle, a ball of specific radius is pivoted across edges of each triangle bounding the growing mesh. If the pivoted ball hits vertices that are not yet part of the mesh, a new triangle is instantiated and added to the growing mesh.

Taubin [22,23], Gotsman and Keren [24,25], in their work, reconstructed the surfaces by global algebraic fitting. In [22], the author fit a polynomial implicit function to a point set by minimizing the distance between the point set and the implicit surface, and developed a first order approximation of a Euclidean distance and improves the approximation later. Gotsman and Keren created parametrized families of polynomials that satisfy desirable properties, such

as fitness to the data or continuity preservation. Such a family must be large so that it can be include as many functions as possible. This technique leads to an over-representation of the subset, in that the resulting polynomial will often have more coefficients for which to solve than the simpler polynomials included in the subset, thus requiring additional computation.

Alternative approaches based on implicit representation of object surfaces with radial basis functions (RBFs) were presented by [26-30]. In the work carried out by [26], the authors used these functions to reconstruct cranial bone surfaces from 3D CT scans. Data surrounding large irregular holes in the skull were interpolated using thin-plate spline RBF. These functions offer several advantages over piecewise polynomial interpolants and may simplify problems related to the process of smoothing and re-meshing existing noisy surfaces. Additionally, an RBF offers a unified framework that is simple, leading to a compact functional description of a set of surface data. Interpolation and extrapolation are inherent in the functional representation. The RBF associated with a surface can be evaluated anywhere to produce a mesh at the desired resolution without an augmentation of the computation time and the memory requirements. This is due to the fact that an RBF approximation provides the ability to approximate the input data using significantly fewer centers comparing to the whole number of data points obtained from laser scanning [27]. RBF representation is inherently a 3D modeling approach, which can be further manipulated through a series of Boolean unions and intersections with other objects in a manner similar to simpler geometric primitives are currently used to construct more complicated objects.

In 2002, [31] introduced some techniques for modelling with interpolating implicit surfaces, where a 3D implicit surfaces are described by specifying locations in 3D through which the surface should pass, and also identifying locations that are interior and exterior to the surface. A 3D implicit function is created from theses constraints using a variational scattered data interpolation approach, and isosurface of this function describes a surface. In their formulations, they also implemented a simple method for converting a polynomial model to a smooth implicit model, as well as new way to form blends between objects.

Dihn and his colleagues [32] presented a new method of surface reconstruction that generates smooth and seamless models from sparse, noisy, non-uniform, and low resolution range data. Contrary to the traditional reconstruction algorithms, designed for dense and precise data which do not produce smooth reconstructions when applied to vision-based data sets, they implemented a method that constructs a 3D implicit surface formulated as a sum of weighted radial basis functions, which grant the achievement of multiple orders of smoothness as well as it was introduced a new perspective over those type of functions which leaded to a enhancement of fine detail and sharp features that are often smoothed-over by the variational implicit surfaces.

Respecting to the development of new approaches in the interpolation of radial basis functions, Morse and his colleagues [33] proposed a algebraic method for creating implicit surfaces using linear combinations of radial basis interpolants to form complex models from scattered surface points. In their work, they explored and extended the implicit interpolations methods of [34] and [35] in order to develop approaches computationally less complex, making them suitable for systems of large numbers of scattered surface points by using compactly supported radial basis interpolants, which generate a sparse solution space, making

the technique practical for large models.

The present work focus the reconstruction of 3D surfaces from synthetic point clouds, based on the three main steps:

- Construction of a signed-distance function;
- Fitting an RBF to the resulting distance function;
- Iso-surfacing the fitted RBF.

The most relevant differences between this approach and [27] consist on the type of RBFs used to interpolate the desired function, but also on the methods considered to create the off-surface points, leading to a very distinct way of handle with the problem related to the computational complexity of the algorithm.

The remainder of this paper is organized as follows. In section 2, a reference to some basic concepts related to implicit functions interpolation using radial basis functions is carried out. It is also described the construction of the signed-distance function to be interpolated, as well as the Principal Component Method used to estimate the normals to each of in-surface points. The interpolation using compactly-supported radial basis functions considering the Shepard Method and the construction of Multidimensional Binary Search Trees is addressed in Section 3. The presentation of results and its analysis is developed within Section 4 where a parametric study is made to enable a better understanding of the influence of different parameters involved in the reconstruction steps. Final remarks and conclusions are withdrawn on the performance of this combined techniques approach.

2. BASIC CONCEPTS AND RELATED WORKS

2.1 Introduction

Concerning to the construction of the signed-distance function, the present work adopted a similar procedure to that carried out by Carr and his colleagues [27], i.e., if we have a set of distinct points, the problem to find an interpolant that fits as best as possible can be expressed as: *given n distinct points* $\{(x_i, y_i, z_i)\}_{i=1}^{n}$ *on a surface M in* $\Re^3$*, find a surface M' that is a reasonable approximation to M.* This corresponds to modelling the surface implicitly with a function *f(x,y,z)*. If the surface *M* consists of all the points *(x,y,z)* that satisfy the equation:

$$f(x, y, z) = 0 \qquad \textbf{(2.1)}$$

then we say that *f* implicitly defines *M* [36]. Here we want to approximate the signed-distance function *f(x)* given a set of zero-valued surface points and non-zero off-surface, leading to a problem that can be stated formally as follows: *given a set of distinct nodes,* $X = \{x_i\}_{i=1}^{N} \subset \Re^3$*, find an interpolant s:* $\{f_i\}_{i=1}^{N} \subset \Re$ *such that*

$$s(x_i) = f_i, i = 1...N \tag{2.2}$$

Carr et al. [27] and Turk and O'brien [31] wanted to approximate the signed-distance function *f(x)*, given a set of zero-valued surface points and non-zero off-surface. Carr's interpolant was chosen from *BL(2)* ($\Re^3$), the Beppo-Levi space of distributions in $\Re^3$ with square integrable second derivatives and is equipped with the rotation invariant semi-norm, that measure the energy or "smoothness" of functions, defined by:

$$E(s) = \int_{\Omega} \left(s_{xx}^2(x) + 2s_{xy}^2(x) + s_{yy}^2(x) \right) dx \tag{2.3}$$

where the notation s_{xx} means the second partial derivative in the x direction, and the other two terms are similar partial derivatives, one of them mixed. This energy function is basically a measure of the aggregate curvature of *s(x)* over the region of interest Ω (a portion of plane). Any creases or pinches in a surface will result in a larger value of E. According to [31], smooth function that as no such regions of high curvature will have a lower value of E. Note that because there are only squared terms in the integral, the value of E can never be negative. The thin-plate solution to that interpolant problem is the function *s(x)* that satisfies all of the constraints and has the smallest possible value of E. Note that thin-plate surfaces are height fields, and thus they are in fact parametric surfaces.

Additionally, [37] mentioned that the smoothest interpolant *s(x)* has the simple form (particular example of an RBF) given by:

$$s(x) = p(x) + \sum_{i=1}^{N} \lambda_i \phi\left(\left| x - x_i \right| \right) \tag{2.4}$$

where the second part of the equation can be one of the functions showed in Table I (for example) and p is low degree and the RBF is a real valued function on [0, ∞), usually unbounded and of on-compact support [38]. In this context the points x_i are referred to as the centers of the RBF.

TABLE I. EXAMPLES OF TYPICAL RADIAL BASIS FUNCTIONS.

RBF type		Params.
Multiquadric	$R_i(x,y) = \left(r_i^2 + \left(\alpha_c d_c \right)^2 \right)^q$	$\alpha_c \geq 0, q$
Gaussian	$R_i(x,y) = exp\left(-\alpha_c . \left(r_i / d_c \right)^2 \right)$	α_c
Thin plate spline	$R_i(x,y) = r_i^n$	η
Logarithm	$R_i(x,y) = r_i^n \log(r_i)$	η

Radial basis functions are popular for interpolating scattered data as the associated system of linear equations is guaranteed to be invertible under very mild conditions on the locations of the data points [34],[38]. In general, if the polynomial in equation **(2.4)** is of degree *m* then it has to be implied that the side conditions (orthogonality) imposed on the coefficients λ_i are

$$\sum_{i=1}^{N} \lambda_i q(x_i) = 0 \qquad \textbf{(2.5)}$$

for all polynomials q of degree at most m. These side conditions along with the interpolation conditions of equation **(2.2)** lead to a linear system to solve for the coefficients that specify the RBF.

Let $\{p_1,...,p_l\}$ be a basis for polynomials of degree at most m and let $c=\{c_1,...,c_l\}$ be the coefficients that give p in terms of this basis. Then equations **(2.2)** and **(2.5)** may be written in matrix form as

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix}\begin{pmatrix} \lambda \\ c \end{pmatrix} = B\begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \qquad \textbf{(2.6)}$$

where

$$A_{i,j} = \phi\left(\left|x_i - x_j\right|\right), \quad i,j = 1..N \qquad \textbf{(2.7)}$$

$$P = \begin{pmatrix} 1 & x_1 & y_1 & z_1 & \cdots p_l(x_2) \\ 1 & x_2 & y_2 & z_2 & \cdots p_l(x_2) \\ 1 & x_3 & y_3 & z_3 & \cdots p_l(x_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots \ \vdots \\ 1 & x_n & y_n & z_n & \cdots p_l(x_2) \end{pmatrix} \qquad \textbf{(2.8)}$$

Solving the linear system **(2.6)** determines λ and c, and hence $s(x)$. However, the matrix B in equation **(2.6)** is typically bad conditioned as the number of data points N gets larger. This means, that substantial errors will easily creep into any standard numerical solution. In our work we used compactly-supported radial functions, and the terms $p_{i,j}$ are not considered.

Besides the thin-plate radial basis functions used by [27], does indeed minimize the smoothness functional in equation **(2.3)**, its drawbacks can be enumerated as follows:

-$O(n^2)$ computation is required to build the system of equations;

-$O(n^2)$ storage is required (for the nearly-full matrix) to represent the system;

-$O(n^2)$ computation is required to solve the system of equations;

-$O(n)$ computation is required per evaluation;

Because every known point affects the result, a small change in even one constraint is felt

throughout the entire resulting interpolated surface, which is an undesirable property for shape modelling. According to [33] a significant portion of the computational cost involved in calculating implicit surfaces is the cost required to construct the matrix (or submatrix) $\phi_{ij} = \phi\left(\left\|\overline{c}_i - \overline{c}_j\right\|\right)$. For example, if it is used the thin-plate radial basis function, one can easily conclude that the matrix is entirely non-zero except along the diagonal, requiring the calculation of all inter-point distances within the set $\{\overline{c}_i\}$. Although the symmetry of the matrix cuts the computational cost in half, the complexity is still $O(n^2)$. Furthermore, storage of such a matrix requires $O(n^2)$ floating-point values, which is potentially a more prohibiting factor than the computational complexity.

Although [39] use LU factorization (an $O(n^3)$ algorithm to solve the system of equation **(2.6)**, they correctly point out that it is possible to solve this system in $O(n^2)$ by iterative means. Thus, while solution of the system may appear to be a limiting step, it needs only to be as computationally expensive as constructing the system.

2.2 Fitting an implicit function to a surface

2.2.1 Construction of a signed-distance function to fit to a surface

To fit an implicit function to a surface, one can say that is wished to find a function *f* which implicitly defines a surface *M'* and satisfies the equation

$$f(x_i, y_i, z_i) = 0 \quad , \qquad i = 1,..., \quad n \tag{2.11}$$

Where $\{(x_i, y_i, z_i)\}_{i=1}^{n}$ are points lying on the surface. In order to avoid the trivial solution that *f* is zero everywhere, off-surface points are appended to the input data and are given non-zero values. This gives the interpolation problem: *find f such that*

$$f(x_i, y_i, z_i) = 0 \quad , \; i = 1,..., \quad n \quad \text{(on-surface points)} \tag{2.12}$$

$$f(x_i, y_i, z_i) = d_i \neq 0 \quad , \; i = n+1,..., \quad N \quad \text{(off-surface points)} \tag{2.13}$$

This still leaves the problem of generating the off-surface points $\{(x_i, y_i, z_i)\}_{i=n+1}^{N}$ and the corresponding values d_i.

An obvious choice for f is a signed-distance function, where the d_i are chosen to be the distance to the closest on-surface point. Points outside the object are assigned positive values, while points inside are assigned negative values. Similar to [27] and [39], these off-surface points are generated by projecting along surface normals. Off-surface points may be assigned either side of the surface as illustrated in figure I.

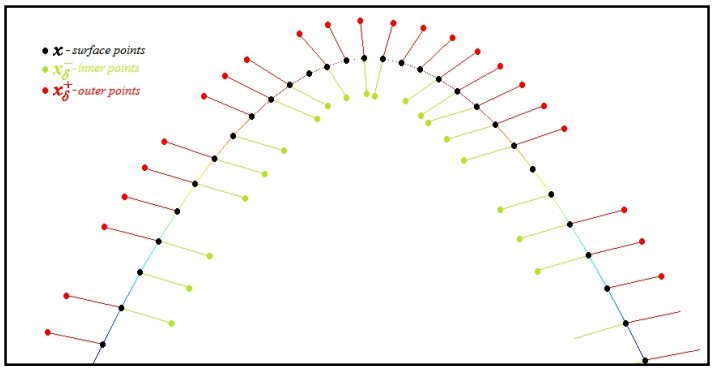


Figure I -Illustration of the Off and On-surface points.

Concerning to the augment of the data points, actually there exists a several considerations associated to the dimension of this augment, i.e., some authors consider that is better to create two off-surface points for each point lying in the surface, one either side of the surface, marked with negative and positive signals.

However, some other authors, from their experience, showed that is needed just to use interior constraints, or exteriors constraints and some of them proposed to create a number of exterior points just about 10% of the total data points. With respect to the distance δ of the off-surface points to the zero-valued points toward the estimated normals of each point, in the present work, it was implemented as 1% of the greater maximum distance of the upper and down boundaries of each x y z coordinates [40].

Due to those augment of data points, we face a problem that has to be solved: estimate surface normals and determine the appropriate projection distance.

If we have a partial mesh, then is straightforward to define off-surface points since normals are implied by the mesh connectivity at each vertex. In the case of unorganized points-cloud data, normals may be estimated from a local neighbourhood of points.

2.2.2 Normals estimation via Principal Component Analysis

In our work we estimate those normals, at each point by choosing a set of closest points from the point, which are chosen considering a given search radius. That search is performed by a algorithm based on the construction of a *kd*-tree [41], containing the information of the nearest points of each point p_i, structured in a tree based on the positions of the points (right, left, top or down).

Further, we find the unitary normal vector of the plane via Principal Component Analysis (PCA), which can be described based on the work developed by [42] and [11], where one can say that the estimation of the normal is done following three main steps:

- Find all points within a certain neighbourhood of each point $p_i \epsilon P$;
- Estimate the direction of the normal n_i for p_i via PCA (as previously referred);

- Orient the generated normals n_i consistently.

With respect to the first step, one can say that the neighbourhood of each point $p_i \in P$ consists of k nearest neighbours in the data set, i.e.,

$$P_i = \left\{ p_{i,0}, ..., p_{i,k-1} \right\} \tag{2.14}$$

In the next step, it is used the *Principal Component Method* to compute the normal vector for each point p_i.

Let the point p_i be the centroid of neighbourhood P and considering the points referred in equation **(2.14)**, we can compute n_i forming the covariance matrix of the neighbourhood P. This is the symmetric 3×3 positive semi-definite matrix:

$$CV = \sum_{j=0}^{k-1} \left(p_{i,j} - p_{i,0} \right) \otimes \left(p_{i,j} - p_{i,0} \right) \tag{2.15}$$

where $\otimes$ denotes the outer vector operator. If $\lambda_i^1 \geq \lambda_i^2 \geq \lambda_i^3$ denote the eigenvalue of CV associated with unit eigenvectors $\hat{v}_i^1 \geq \hat{v}_i^2 \geq \hat{v}_i^3$, respectively. We choose n_i to be either $\hat{v}_i^3$ or $-\hat{v}_i^3$.

Finally, similar to [42], we choose the quantity $\cos\left(\gamma_{ij}\right) = n_i \otimes n_j$ as criterion to check if the normals are well oriented comparing to the nearest neighbourhood points, i.e., if the value $\cos\left(\gamma_{ij}\right)$ is far from the unit value, the normal of this point is not computed and consequently the normal points to the point are not computed either.

The correction of the normals is yet an issue that is being discussed.

However, we should be careful, because this method of seeking the normal vectors could lead to some significant errors if the chosen points to estimate the normals are not so close to the candidate points. In general, it is difficult to robustly estimate normals everywhere.

If normal direction or sense was ambiguous at a particular point then they did not fit to a normal at that point and instead, they let the data point as a zero-point (lied in the surface) tied down the function in that region.

Additionally, given a set of surface-normals care must be taken when projection off-surface points along the normals to ensure that they do not intersect other parts of the surface. The projected point is constructed so that the closest surface point is the surface point that generates it. Provided this constraint is satisfied, the constructed surface is relatively insensitive to the projection distance d_i.

Sometimes it happens to project inappropriately the off-surface points and the effect of this lead to a resulting surface, where f is zero, that is distorted in the vicinity of some particular part of the surface where opposing normal vectors have intersected and generated off-surface points with incorrect distance-to-surface value, both in sign and magnitude.

Once the normals are computed, the position of the off-points created toward the direction of those normals can be calculated as in equations **(2.16-17)**.

$$\left(x_{N+i}, y_{N+i}, z_{N+i}\right) = p_i + \delta no_i = \left(x_i + \delta no_i^x, y_i + \delta no_i^y, z_i + \delta no_i^z\right) \quad \textbf{(2.16)}$$

$$\left(x_{2N+i}, y_{2N+i}, z_{2N+i}\right) = p_i - \delta no_i = \left(x_i - \delta no_i^x, y_i - \delta no_i^y, z_i - \delta no_i^z\right), \quad i = 1..n \quad \textbf{(2.17)}$$

where δ is the distance considered, between the in-surface points and the corresponding off-surface points and $no_i^{x,y,z}$, $i=1,...,n$, the normals estimated for each coordinate axis.

2.2.3 Interpolant model identification on the extend dataset

This step consists on determining the value of the function f expressed in equations **(2.18)** whose zero contour (isosurface $f=0$) interpolates the given point cloud data $x_{1,...,n}$, and whose isosurface $f=1$ and $f=-1$ interpolate $x^+_{n+1,...,2n}$ and $x^-_{2n+1,...,3n}$, respectively, i.e.,

$$f\left(x_i\right) = \begin{cases} 0 & i = 1..n \\ 1 & i = n+1..2n \\ -1 & i = 2n+1..3n \end{cases} \quad \textbf{(2.18)}$$

The values of ±1 for the auxiliary data are assigned in an arbitrary way. Such choice does not affect the quality of the results. Here we are interested in the zero isosurface of f.

3. COMPACTLY-SUPPORTED-RADIAL BASIS FUNCTION INTERPOLATION (CS-RBF)

3.1 Construction of the linear system of equations with CS-RBFs

As mentioned before, in the present work we use CS-RBFs as interpolant functions to construct the linear system of equations and by the literature research carried out, Wendland CS-RBFs functions [43] represent feasible choices as interpolant functions once they guarantee the creation of sparse and positive-definiteness matrices of the linear system. In general, the solutions of the minimum-degree polynomial for compact, locally-supported radial basis functions have the form:

$$\phi(r) = \begin{cases} (1-r)^p P(r) & r<1 \\ 0 & otherwise \end{cases} \quad \textbf{(3.1)}$$

In this work, the computation of $\phi(r)$ is performed using the Shepard Method, which correspond to one of the existents partition of unity (PU) approaches. This approach is used to construct interpolations and approximations and possess good properties, such as the fact that the grid is composed by scattered nodes, the basis reproduces exactly complete linear polynomials and the method is applicable in any number of spatial dimensions.

Shepard Method is used here due its benefits that include better conditioning of discrete equations and easier handling of essential boundary conditions in applications to PDE's. Furthermore, compared to moving least squares approximations, the construction of the basis is quite fast [44].

In Shepard Method, we consider the discretization of the domain Ω with an approximation based on a set of scattered nodes. Each node affects the approximation in its neighbourhood, or domain of influence Ω_I.

The domains of influence can be of any shape: square, circular, etc. A weight function $W_I(x)$ is associated with each node I. It is non-negative inside Ω_I, vanishes on the boundary $\delta\Omega_I$, and is non-zero at node I. the approximation can be written as:

$$u_h(x) = \sum_{I=1}^{N} w_I(x) u_I \tag{3.2}$$

where u_I are the nodal parameters, and $w_I(x)$ are the basis functions of compact support, which are constructed from the weight functions associated with the nodes, $W_I(x)$, by:

$$w_I(x) = \frac{W_I(x)}{\sum_{k=1}^{N} w_k(x)} \tag{3.3}$$

Being trivially shown that:

$$\sum_{I=1}^{N} w_I(x) = 1 \tag{3.4}$$

Equation **(3.4)** expresses the fact that the functions $w_I(x)$ represent a partition of unity: a constant function $u(x)=C$ is reproduced exactly ("constant precision"). Id $u_I=C$, for $I=1,...,N$, it follows from **(3.2)** that:

$$u_h(x) = \sum_{I=1}^{N} w_I(x) u_I = \sum_{I=1}^{N} w_I(x) C = C \sum_{I=1}^{N} w_I(x) u_I = 1 \tag{3.5}$$

Figures II(c-d) show the plot of two examples of 3D compactly-supported functions and in table II, it can be observed an example of that type of functions, divided by their dimensions

in space.

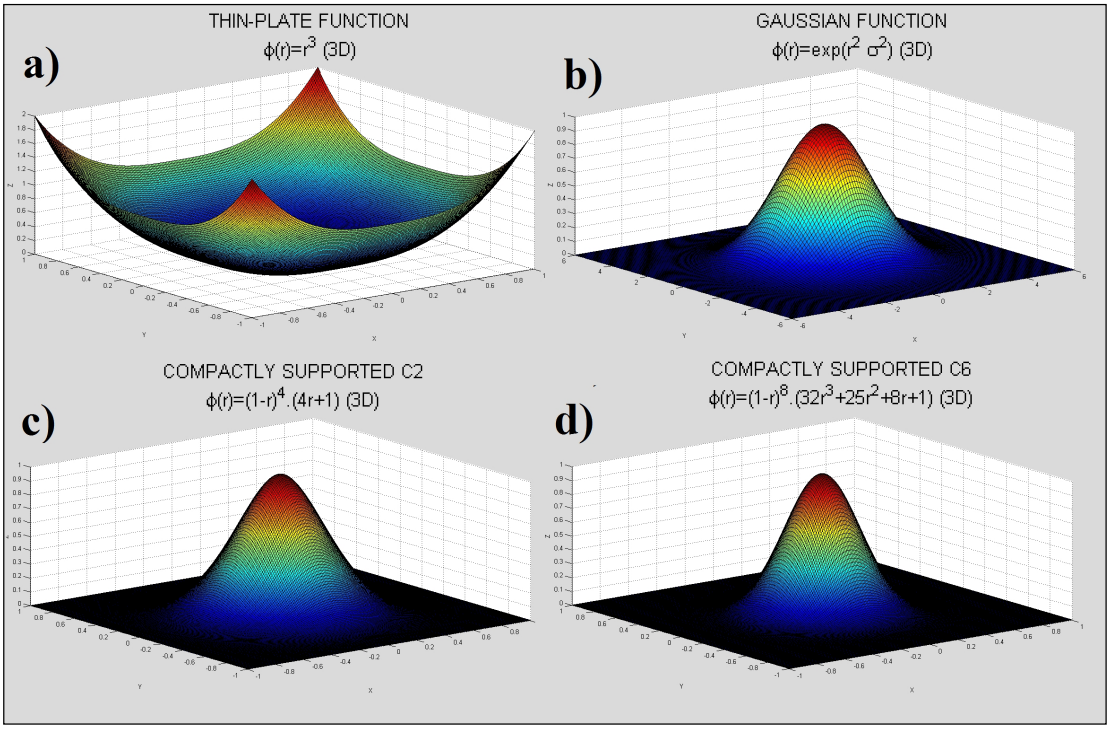


Figure II - Plot and comparison of different radial basis functions.

For various degrees of desired continuity (C^k) and dimensionality (d) of the interpolated function, Wendland [43] has derived the radial basis functions presented in table II.

These functions have a radius of support equal to 1.

Scaling of the basis functions (i.e, $\phi(r/\alpha)$) allow any desired radius of support α.

The system of equations using the CS-RBFs has the same form as equations **(2.7)**, but the construction of the linear system considering those type of interpolant make use of an important feature of RBF functions, namely the fact that they have finite support $\phi\left(\left\|\overline{c}_i,\overline{c}_j\right\|\right)=0$ for all $\left(\overline{c}_i,\overline{c}_j\right)$ farther apart than the radius of support.

By using a *kd*-tree [41], the set of all points within distance r of a particular point $\overline{c}_i$ can be determined in $O(\log\ n)$ time.

TABLE II. EXAMPLES OF COMPACTLY RADIAL BASIS FUNCTIONS, THEIR DIMENSIONS AND CONTINUITIES (MORSE ET AL. (2001)).

Dimension d	Continuity C^k	RBF
1	$(1-r)_+$	C^0
	$(1-r)_+^3(3r+1)$	C^2
	$(1-r)_+^5(8r^2+5r+1)$	C^4
3	$(1-r)_+^2$	C^0
	$(1-r)_+^4(4r+1)$	C^2
	$(1-r)_+^6(35r^2+18r+3)$	C^4
	$(1-r)_+^8(32r^3+25r^2+8r+1)$	C^6
5	$(1-r)_+^3$	C^0
	$(1-r)_+^5(5r+1)$	C^2
	$(1-r)_+^7(16r^2+7r+1)$	C^4

As a result of that feature, in the present approach, the linear system of equations is solved interactively, considering in each loop a particular center $\bar{c}_i$, for which vicinity the system is solved.

3.2 Multidimensional binary search tree

A *kd*-tree is a multidimensional binary tree with the following sorting property for a tree with point $\bar{x}$ at the root and subtrees T_{left} and T_{right}.

$$\forall \bar{y} \in T_{left} : \ \bar{y}^d \leq x^d ,$$
$$\forall \bar{y} \in T_{right} : \ \bar{y}^d \geq x^d \qquad \textbf{(3.6)}$$

where the sorting dimension d changes at each level of the tree. *kd*-trees can be used to find all points within distance r of a particular constraint in $O(n \log n)$ time.

While a number of points must still be tested explicitly, the multidimensional sorting nature of the *kd*-tree allows a large number of points to be rejected at each level of the tree [33]. The resulting matrix is extremely sparse, as illustrated in the figure III.

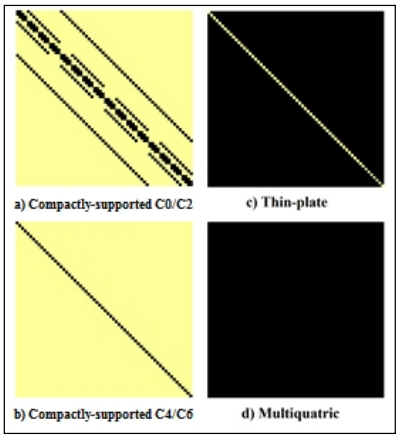


Figure III - Structure of the matrices produced by four different CS-RBFs (a and b), thin-plate (c) and multiquadric (d). The compactly-supported basis function produce a matrix that is sparse (yellow), while the thin-plate basis functions as well multiquadrics produce a matrix that is nearly and completely full, respectively (black).

In figure III (a-b), the matrices used to illustrate the sparsity of the CS-RBFs where obtained considering the same radius of support and it is important to mention that the order of magnitude of the "non-principal diagonal cells" values, in figure III (a), is greater for C0. Additionally it was observed that, with growth of the radius of support, those values in all CS-RBFs growth as well, being always (for the same radius) greater to functions with a lower continuity k. This is easily observed, for example, if we observe the behavior between the plots illustrated in figure II c) and d), for CS-RBFs C2 and C6.

According to [41], there already exists numerous methods for building an information retrieval system capable of handling such associate queries, but they fall short in some very important way, either in having only a small class of queries easily performed, large running time, large space requirements, horrible worst cases, or some other adverse properties.

In our paper, we use their method to perform the queries of neighbourhood points search. As referred, the *kd*-tree structure grants good flexibility on performing different types of queries. In the present work, we perform a query that search, for each surface point considered, the nearest neighbour points by considering the distance between the points (represented by a radius).

However, if we want, it is extremely easy to perform other types of queries, which can be for example, queries based on the distance of a specific coordinate axis, or queries based on a specific position between the points, as right/ left, top/bottom or in some desired direction. In addition to that type of queries, nearest neighbour queries, the *kd*-tree is also able to perform intersection queries.

Summarising, one can say that a *kd*-tree represents an information file, where each record in the file is stored as node in the tree. In addition to k keys which comprise the record, each node contains two pointers, which are either null or point to another node in the *kd*-tree. Associated with each node, though not necessarily stored as a field, is a discriminator, which

is an integer between *0* and *k-1*, inclusive.

3.3 Selection of radius of support and isosurface extraction

Due to the finite extent of CS-RBFs, only those points within the radius of support of one of the original positions have non-zero values. Hence, it is crucial to properly select that radius of support to achieve optimal efficiency of computation and results. Too small a radius can produce basis functions that are unable to span the inter-constraint gaps. Too large a radius does not adversely affect the results but reduces the sparseness of the matrix, thus increasing the computation required. It is necessary to select a radius of support that is both large enough to produce effective results and not so large that the computation becomes impractical. In section 4, some examples of considering different radius of support to reconstruct the same surface are demonstrated.

For all points outside the radius of support considered in each iteration of the surface reconstruction, all the terms in equation **(2.5)**, with exception to the polynomial term, are zero. In this way, these embedding functions are not the same as those normally used for implicit surfaces - the implicit surface represented is not the only set of zero-valued points in the space. However, the implicit surface does form a unique contiguous locus of zero-valued points passing through the constraints. An isosurface extractor may be used to extract this surface by seeding it with any one of the initial constraints. However, care must be taken so that the step size of the extractor does not cause it to jump outside the band of non-zero points. It is easy to explicitly recognize when no non-zero terms are found in equation **(2.5)** (none of the constraint point lie within the radius of support of the point being evaluated).

As the zero set of embedding function is not uniquely the surface of interest, the resulting embedding function has limited applications in interpolation [35], or similar applications.

For the purpose of isosurface extraction, in this work we used the MATLAB function *isosurface()*, which provides good results, being also used in the work of [40] and [46]. From the experience obtained, the results provided by this function showed to be similar to the one obtained using Marching Cubes approach.

3.4 Pseudo-algorithms

For illustrative purposes, one presents schematically in this sub-section the most relevant pseudo-algorithms considered in this work, reflecting the considerations made in the previous sections. The first one is related to the main routine associated to the surface reconstruction.

Pseudo-algorithm 1 - Main Routine

Computes the reconstruction of a surface represented by a synthetic point cloud and represents it in the space
Input: $p(p_1,p_2,...,p_N)$; N_{eval}; N_{cell}; w_{func}; rbf; R_{no}; R; Ln_{min}; Ln_{max}; Val_{crit}; n_{check} (integer)
Output: Surface iso-plot; *vertices* %Final fitted points

Construct Distance Matrix D of p and set N as the number of points (rows) of p;
Construct the Multidimensional Binary Tree (*Datatree*) of p;
Initialize *normals* %Size p
Initialize $count_1$
For i **from** 1 **to** N **do**
[*normals*,$count_1$,$count_2$] ← Principal_Component_Analysis(*normals*,$count_1$,Ln_{min},Ln_{max},Val_{crit},p,N,i) %Algorithm 2
endfor
Δ ← max(max(p,[],1)-min(p,[],1))/100
Create the off-surface points, with the distance of Δ to the on-surface points p;
Set the values 1 and -1 to the function vector f, with respect to the inner and outer surface points, respectively;
Create the grids of validation points (p_{eval}) and center points (p_{pu}) with sizes N_{eval}^3×3 and N_{cell}^3×3, respectively;

Construct Distances Sparse Matrix (D_{csrbf}) to construct interpolation function via Shepard Method, concerning the radius of support R, evaluation grid p_{eval} and center points p_{pu}; uses kD_tree and kD_search
Evaluate W_{func} ← w_{func}(D_{csrbf})
Construct the Multidimensional Binary Tree (*evaltree*) of *peval*;
Initialize P_f %Size N_{cell}^3×3
p_{ctrs} ← p
for j **from** 1 **to** N_{cell}^3 **do**
Search for the points $pt(pt_1,pt_2,...,pt_{Nc})$, their indices $idx(idx_1,idx_2,...,idx_{Nc})$and distances $d(d_1,d_2,...,d_{Nc})$, on the neighborhood of $p_{pu}(j,:)$, considering the *Datatree* and R;
if not empty(idx)
Construct Distances Sparse Matrix (D_{data}), concerning R, p(idx,:) and p_{ctrs}(idx,:)
RBF_{data} ← rbf(D_{data})
Search for the points $pt_e(pt_{e1},pt_{e2},...,pt_{eNc})$, their indices $idx_e(idx_{e1},idx_{e2},...,idx_{eNc})$ and distances $d_e(d_{e1},d_{e2},...,d_{eNc})$, on the neighborhood of $p_{pu}(j,:)$, considering the *evaltree* and R;
Construct Distances Sparse Matrix (D_{eval}), concerning R, p_{ctrs}(idx,:) and p_{eval}(idx,:)
RBF_{eval} ← rbf(D_{eval})
Fit_{local} ← RBF_{eval}×(RBF_{data}\f(idx))
Pf(idx_e) ← Pf(idx_e)+Fit_{local}×D_{csrbf}(idx_e,j)
Iso-plot Pf %Progressive iso-plot of the surface
endif
endfor
Iso-plot Pf %Final iso-plot of the surface

The second pseudo-algorithm respects to the principal component analysis method.

Pseudo-algorithm 2 - Principal_Component_Analysis

Computes the normal vectors to each point of the point cloud, via Principal Component Analysis Method
Input: p; R_{no}; R; Ln_{min}; Ln_{max}; Val_{crit}; $normals$; $count_1$; N; i.
Output: $normals$; $count_1$; $count_2$.

Search for the points $pt(pt_1,pt_2,...,pt_{Nc})$, their indices $idx(idx_1,idx_2,...,idx_{Nc})$and distances $d(d_1,d_2,...,d_{Nc})$, on the neighborhood of $p(i,:)$, considering the *Datatree* and R_{no};
$p_c \leftarrow p(idx,:)$
$N_{cc} \leftarrow$ round(N_c/n_{check}) %Choose a set of neighbors to check normals
Initialize CV %Size $3\times3\times N_{cc}$
for j **from** 1 **to** N_{cc} **do**
for k **from** 1 **to** N_c **do**

$$CV(:,:,j) <- \frac{1}{n_c}\sum_{j=1}^{n_{cc}}\sum_{k=1}^{n_c}\left(pt(k,:)'-p_c(j,:)'\right)\left(pt(k,:)-p_c(j,:)\right)$$

endfor
endfor
Compute eigenvalues $lambda$ and eigenvectors ei_{vec} with size $3\times3\times N_{cc}$ for each $p_c(pt_1,pt_2,...,pt_{Ncc})$;
Set the value of normal vectors n_u with size $1\times3\times N_{cc}$ for each point $p_c(pt_1,pt_2,...,pt_{Ncc})$ as the the eigenvector corresponding to smallest eigenvalue each point $p_c(pt_1,pt_2,...,pt_{Ncc})$
Calculate the vector $v_{sign}(v_{sign1},v_{sign2},...,v_{signNcc})$ corresponding each cell to the value for outer product between each point $p_c(pt_1,pt_2,...,pt_{Ncc})$ normal vector and the point $p(i)$ normal.
Initialize $count_2$
Increment 1 to $count_2$ for each $v_{sign}(v_{sign1},v_{sign2},...,v_{signNcc})$ between Ln_{min} and Ln_{max};
Consider the computation of normal of $p(i)$ if $count_2/Ncc>Valcrit$; otherwise, increment 1 to $count_1$.

4. RESULTS

To enable the evaluation of the methods considered to reconstruct the 3D surfaces, and their implementation, it was found important to develop a parametric study concerning different factors involved. To this purpose a set of primitive surfaces was considered in order to characterize the influence of parameters and their effect on the final results. These analyses are considered very important and pertinent as they are not found in the literature, as far as it was possible to understand from the extensive review carried out. The parameters that will be analyzed are the radial basis function used for interpolation (C2 or C6, see table II), the set dimension N of the synthetic point cloud, the size of the validation grid N_{eval}^3 as well the size of centers grid N_{cell}^3, the radius of support for interpolation R and for normals estimation R_{no}, and the criterion for the considering if they should be computed. The 3D surfaces considered for reconstruction were the surface of a unitary cube, that contain sharp edges, the surface of a unitary radius sphere and the surface of a torus with a unitary bend radius and an outer radius of 0.3.

4.1 Cube

Concerning to the cube surface, it has to be mentioned, for the simulations carried out, that the results do not differ much for different input parameterization, with the exception of the surface in the edges neighborhood. Due to this observation, we set the parameters to focus in the study of those regions of the surface. Table III shows the parameters considered for three

different cube surfaces obtained and in the figure IV that surfaces can be observed. It should be noted the mean value of the interval showed in the Normal Criterion column. That criterion is presented in the pseudo-algorithm 2 (sub-section 3.4), and intends to compare the value v_{sign} within the interval, i.e., the normal vector for the point is considered just if a certain percentage of the values v_{sign} lies in the interval. It becomes obvious that with a smaller interval it is expected that the number of normals considered, hence all points of the surface are visited, is also smaller.

TABLE III. PARAMETERS IMPOSED TO RECONSTRUCT THE CUBES SURFACES

Number	CS-RBF	*N*	Add. Points (/2)	N_{eval}/N_{cell}	Normal Criterion	R/R_{no}
1	C6	9000	3058	15/15	1	0.2/0.2
2	C6	4500	3341	20/20	[0.98-1.02]	0.3/0.1
3	C2	9000	8895	15/15	[0.98-1.02]	0.2/0.2

By observing the three cubic surfaces obtained, at first place it is clearly noticed the undesirable shapes that appear in the reconstruction of the cube (1) edges. If we compare the combinations of parameters in the table III, it can be concluded that the responsible for the lack of quality of the edges is the strict normal criterion for normal computation, i.e., the normals for each point of the surface was computed only if the cells values of vector v_{sign} was unitary.

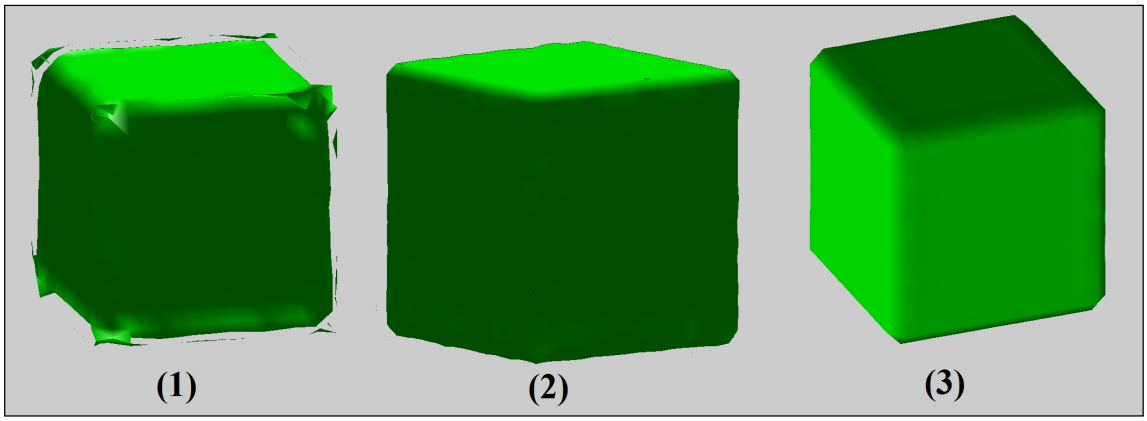


Figure IV - Representation of the cubes surfaces (1), (2) and (3), obtained considering the parameters showed in table III.

That fact lead to a computation of just a few number of normals in the edges, because the normals of a large number of points in its neighborhood lay within a wide angle region which points possess completely different normals when compared to the normal of the point.

Additionally, although it is not so clear, if we observe the representation of cube (1), one can see that some of vertices of that surface present a small hole, due, precisely, to lack of points created in that zones.

Focusing in the Additional Points column in table III, if we compare the percentage of normal points created with respect to the number *N* of the cloud point, for the three cubic surfaces, we found for cube (1) just about 30%, for cube (2), 75%, and for the last, 98%. Considering that, it is obvious that, for the interpolation carried out to reconstruct the surface, the difference between the numbers of points in the cube faces and in the edges is enormous.

Comparing the latter two surfaces, in a first view, one can conclude that the results are not so much different, but in fact, faces of cube (3) are flatter than cube (2), which result in a better quality of that zones of the cube. From table III, it can be concluded that even using a bigger validation and centers grid in the computation of surface (2), the dimension of point cloud of the latter is half of the cube (3). The fact that the percentage of normal points added in the cube (2) is smaller than that of the cube (3) is related with the criterion of normals creation. This happens because on the calculation of the normals, in the cube (3) case, there is a larger number of points in the neighborhood, hence a wide variety of normals are available for comparison purposes.

Finally it is important to refer that even having yet a good quality, the cubes surfaces (2) and (3) still not have the "perfect" approximation in the edges. That fact is related precisely with the computation of normals vectors, as it was referred before, is a issue that is being studied, because actually, by the literature research carried out, it can be said that, besides, exists a relatively large number of approaches to compute that normals, all of them has their drawbacks and should not work well for some kind of surfaces features. Regardless that fact, the most relevant consideration when the normals are calculated is the orientation of those normals, which in most cases is not consistent, and until now it has attracted relatively little research interest [45].

4.2 Sphere

With respect to the study of several parameters effects on the quality of the reconstructed surface of a unitary sphere, that has different geometric features when compared with the cube surface, it was concluded that the most relevant parameters to be studied were the size of validation grid, the radius of support to interpolate the function as well the size of centers grid.

In table IV it can be observed the parameters considered for the reconstruction of the nine spheres presented in figure V. As it can be seen, the dimension of the point cloud sets was the same for all cases and in most of them it was considered the C6 as the CS-RBF. In fact, the consideration of different radial functions showed to have not a relevant effect on the results obtained, except if it is carefully studied the variation of the radius of support for each CS-RBF.

TABLE IV. PARAMETERS IMPOSED TO RECONSTRUCT THE SPHERES SURFACES

Number	CS-RBF	*N*	Add. Points (/2)	N_{eval}/N_{cell}	Normal Criterion	R/R_{no}
1	C6	4000	1985	7/7	[0.98-1.02]	0.1/0.2
2	C6	4000	2156	7/7	[0.98-1.02]	0.4/0.2
3	C6	4000	1875	40/10	[0.98-1.02]	0.1/0.2
4	C6	4000	2109	15/15	[0.98-1.02]	0.1/0.1
5	C6	4000	1839	35/10	[0.98-1.02]	0.15/0.1
6	C2	4000	406	15/15	1	0.3/0.2
7	C2	4000	1929	15/15	[0.98-1.02]	0.3/0.2
8	C6	4000	1992	25/10	[0.98-1.02]	0.3/0.2
9	C6	4000	1852	35/10	[0.98-1.02]	0.5/0.15

First of all, the worst two results obtained for the reconstruction of sphere surface must be discussed. It can easily be concluded from table IV that the parameter with more relevant effect on the surface quality was the dimension of the grids sizes.

Obviously, when we have a small centers grid, just a few numbers of centers is considered to find the in/out surface points, leading to a low precision in the interpolation of the surface function. In addition, if the validation grid is small, it will reinforce the achievement of bad results, once the corresponding validation is done considering just a few points.

The difference between those two surfaces is related with the size of radius of support considered, i.e., regardless the fact that it is obvious that the surface (2) has a miserable quality, at least, when it is seen, it can already be observed some kind of a sphere shape.

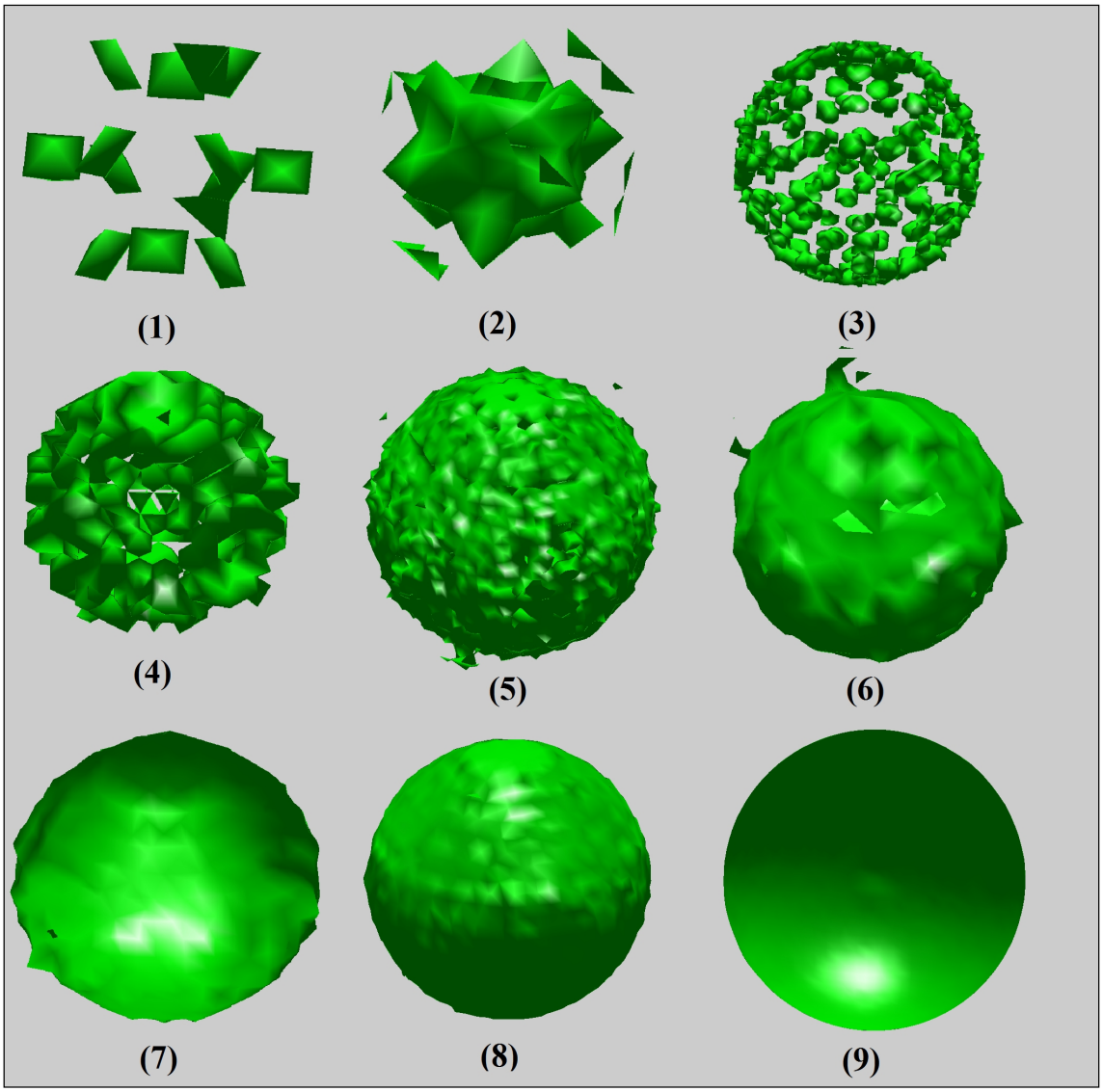


Figure V - Representation of the spheres surfaces (1) to (9), obtained considering the parameters showed in table IV.

In sphere (1), that idea is far from being obvious. That sphere was reconstructed considering a radius of support of 0.1, i.e, just 10% of the radius of the sphere, and when combined with small grids, each center visited during the iteration process will find just a few points in the neighborhood, compromising the interpolation and the validation of the function.

Surface (3) is represented by a bigger number of fragments of minor dimensions, contrarily to what happened to the first surface in this case. This is due to the increase of the validation

grid, being an expected pattern, as the number of points where the interpolated function will be evaluated is larger. Despites this, the surface obtained is not a feasible representation for the surface effectively represented by the synthetic point cloud.

By observing sphere (4), we still have a bad quality surface, but the results are better than the obtained in the representation of the previous surfaces. The size of the validation grid is much smaller, but the size of the centers grid considered is bigger, showing that the consideration of that latter grid size is also an important aspect.

In the case of sphere (5) we still have a surface with several holes, but when compared with the 3rd sphere, we can observe in special, the effect of the radius of support. The validation grid in case (5) is small, but slightly increasing the radius of support, the quality of the results gets better and this is due to the fact that in each iteration on the interpolation and validation routine, the function is interpolated using a large set of points, even that the number of evaluated points is smaller.

The next two spheres presented in figure VI, do not contain holes, but still do not fit with a reasonable quality the desirable surface. Those surfaces were obtained considering a radius of support of 30% of the sphere radius, showing again the importance of that parameter. The difference on the quality of those two surfaces is related with the normal criterion, having sphere (6) some undesirables shapes in certain zones of the surface, which must represent zones where just a few number of normals were computed.

Finally, the spheres (8) and (9) go on clearly improving until reaching a high quality. The improvement from sphere (7) to (8) is related to the increase of the validation grid, and those two surfaces already show that it is more important to consider a greater size for that grid than for the centers grid. The last sphere shown in the figure confirm that conclusion and also that the radius of support considered is a very influent parameter that grant to the evaluated function a much better fitting, as in each iteration, a greater number of points are considered.

Considering all the results, in general we can conclude that the validation and centers grids grant to the interpolation function a bigger space range that will be explored to proceed with the interpolation and evaluation process and a bigger radius of support gives the possibility (in each iteration carried out during the reconstruction process) of visiting a large number of points, grating a greater smoothness to the desirable surface represented by the point cloud.

4.3 Torus

The same procedures carried out in the study of parameters effects in the surface of a sphere were considered for the torus, i.e., the dimension of the point cloud generated was the same, except in the last case, where it was considered a set of 2000 points.

Table V shows the parameters considered on the reconstruction of each torus represented in figure VI.

TABLE V. PARAMETERS IMPOSED TO RECONSTRUCT THE TORUS SURFACES

Number	CS-RBF	*N*	Add. Points (/2)	N_{eval}/N_{cell}	Normal Criterion	R/R_{no}
1	C6	4000	3425	15/15	[0.98-1.02]	0.1/0.3
2	C6	4000	1605	15/15	[0.99-1.01]	0.4/0.1
3	C6	4000	3556	20/20	[0.98-1.02]	0.25/0.2
4	C6	4000	1709	20/20	[0.99-1.01]	0.4/0.1
5	C6	4000	3431	20/20	[0.98-1.02]	0.2/0.2
6	C6	2000	1798	30/10	[0.98-1.02]	0.2/0.2

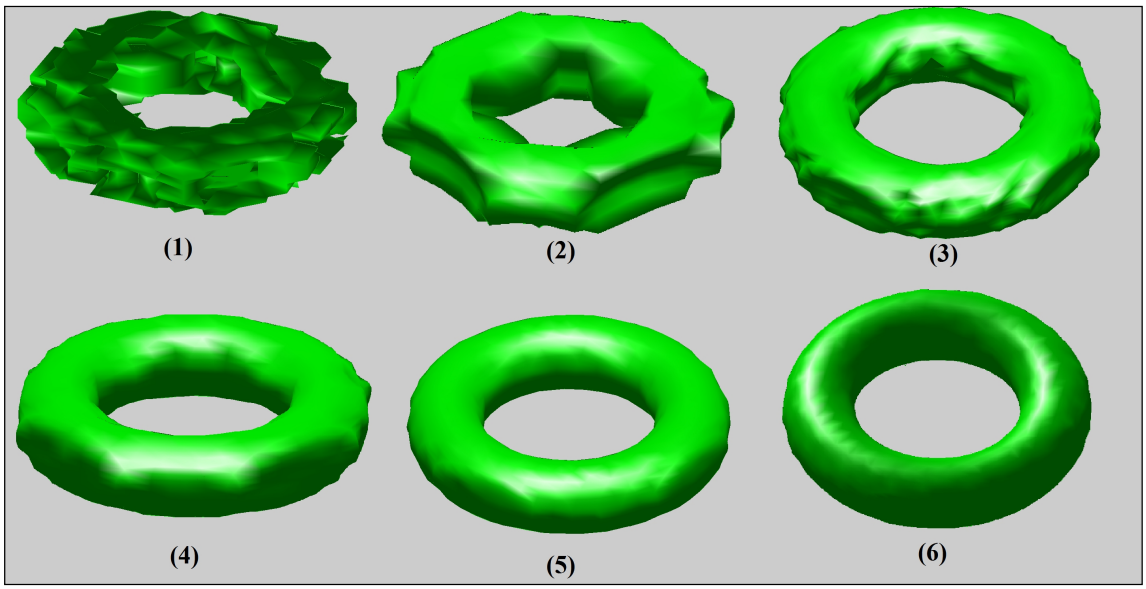


Figure VI - Representation of the torus surfaces (1) to (6), obtained considering the parameters showed in table V.

From the observation of the results for the torus surface, one can obtain similar conclusions to the ones mentioned in the case of the sphere surfaces. On these new simulations, there weren't considered grids of center and validation points smaller than 15^3, due to the fact that it would be expected a reconstructed surface with very bad quality.

For the simulations carried out for the reconstruction of the torus it is relevant to mention the difference between surfaces (3) and (4), considering the second case a stricter criterion for the creation of normals points to the surface, and consequently computing a smaller number of normals. It is interesting to observe that surface (4) present a higher smoothness when compared to (3), especially if we see that in the latter case more off-surface points were created. The explanation for those results is related with the radius of support, which was

considered to be bigger in the case of torus (4), leading to the conclusion that, at least, for surfaces with those features, the radius of support has a most visible effect on the surface quality than the criterion for the surface normals computation. When we observe the last two surfaces, it can be reinforced the idea that the size of the validation grid is the most relevant parameter in the cases studied, not only because of the fact that the grid is bigger in the torus (4), which present a very good quality, but also due to the fact that the size of the point cloud is half of the one considered for the reconstruction of the other surfaces.

5. CONCLUSIONS

This work considered the use of a meshless approach based on compactly-supported radial basis functions interpolation to reconstruct object surfaces from unorganized point clouds. In addition, these functions, which shown to be appropriate and trustworthy for the purpose, are constructed using the Shepard Partition of Unity approximation allied to an efficient algorithm that structure the data in a Multidimensional Dimensional Tree for a quick search within points neighborhood.

These methods grant to the approach implemented not only the achievement of surfaces with several levels of detail and smoothness, but also gives the possibility of reconstructing surfaces considering points clouds with relatively large sizes in acceptable periods of time, depending on the computational complexity imposed by the input parameters.

These methods were tested for a set of primitive surfaces with distinct geometrical features, in order to evaluate the influence of different parameters on the final results. The information obtained through these tests is an important issue although it is not possible to find it in the literature, as far as it was possible to understand from the extensive review carried out.

A main area of future work is related to the research and development of efficient methods devoted to the calculation of consistent normal vectors of on-surface points, especially concerning to the orientation of these normals, which must be well defined to distinguish among inner points or outer points created.

Acknowledgments

The authors wish to acknowledge the financial support given by FCT/MEC through Project PTDC/ATP-AQI/5355/2012 and Project LAETA - UID/EMS/50022/2013.

REFERENCES

[1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, The digital michelangelo project: 3d scanning of large statues, Proceedings of SIGGRAPH 2000:131-144, July 2000.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin and C. T. Silva, Computing and rendering point set surfaces, IEEE Transactions on Visualization and Computer Graphics 9(1):1-12, January 2003.

[3] M. Blane, Z. Lei and D. B. Cooper, The 3L algorithm for fitting implicit polynomial curves and surfaces to data, June 1996.

[4] H. Pfister, M. Zwicker, J. Van Baar and M. Gross, Surfels: Surface elements as rendering primitives, Proceedings of SIGGRAPH 2000:335-342, July 2000.

[5] Rusinkiewicz and M. Levoy, Qsplat: A multiresolution point rendering system for large meshes, Proceedings of SIGGRAPH 2000:343-352, July 2000.

[6] V. Pratt, Direct Least-squares fitting of algebraic surfaces, Proceedings of SIGGRAPH Computer Graphics.:1:19, April 1987.

[7] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin and C. T. Silva, Point Set Surfaces, IEEE Visualization 2001:21-28, October 2001.

[8] A. Nealen, An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation, Technical Report, TU Darmstadt.

[9] T. K. Dey and S. Goswami, Provable surface reconstruction from noisy samples, Proceedings of the Annual Symosium. Computational Geometry (2004):428-438, June 2004.

[10] J.-D. Boissonnat, Geometric structures for three-dimensional shape representation, ACM Trans. Graph. 3(4):266-286, October 1984.

[11] H. Hoppe et al., Surface reconstruction from unorganized point clouds, SIGGRAPH Proceedings, 71-78, 1992.

[12] M. Lee and G. Medioni, Inferring segmented surface description from stereo data, Computer Vision and Pattern Recognition Proceedings, 346-352, 1998.

[13] C. Tang and G. Medioni, Inference of integrated surface, curve, and junction descriptions from sparse 3d data, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11), 1998.

[14] S. Muraki, Volumetric shape description of range data using blobby model, SIGGRAPH Proceedings, 227-235, 1991.

[15] B. Curless and M. Levoy, A volumetric method for building complex models from range images, SIGGRAPH Proceedings, 303-312, 1996.

[16] A. Hilton et al., Reliable surface reconstruction from multiple images, Proceedings of Fourth European Conference on Computer Vision, 1996.

[17] M. Soucy and D. Laurendeau, A general surface approach to the integration of a set of range views, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(4), 344-358, 1995.

[18] G. Turk and M. Levoy, Zippered polygon meshes from range images, SIGGRAPH Proceedings, 311-318, 1994.

[19] H. Edelsbrunner and R. N. Mucke, Three-dimensional alpha shapes, ACM Transactions on Graphics, 13(1), 43-72, 1994.

[20] N. Amenta et al., A new Voronoi-based surface reconstruction algorithm, SIGGRAPH Proceedings, 415-420, 1.

[21] F. Bernardini et al., The ball-pivoting algorithm for surface reconstruction, IEEE Transactions on Visualization and Computer Graphics, 5(4), 349-359, 1999.

[22] G. Taubin, Estimation of planar curves, surfaces, and nonplanar spaces curves defined by implicit equations with applications to edge and range image segmentation, IEEE

Transactions on Pattern Analysis and Machine Intelligence, 13 (11), 1991.

[23] G. Taubin, An improved algorithm for algebraic curve and surface fitting, Proceedings Fourth International Conference on Computer Vision, pp. 658-665, 1993.

[24] D. Keren and C. Gotsman, Tight fitting of convex polyhedral shapes, International Journal of Shape Modeling (special issue on Reverse Engineering Techniques), pp. 111-126, 1998.

[25] D. Keren and C. Gostman, Fitting curve and surfaces with constrained implicit polynomials, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(1), pp. 21-31, 1999.

[26] C. Carr, W. R. Fright and R. K. Beatson, Surface interpolation with radial basis functions for medical imaging, IEEE Transactions on Medical imaging 16(1):96-107, February 1997.

[27] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum and T.R. Evans, Reconstruction and representation of 3D objects with radial basis functions, SIGGRAPH (2001) Conference Proceedings:67-76, August 2001.

[28] R. K. Beatson, J. B. Cherrie and C. T. Mouat, Fast fitting radial basis functions: methods based on preconditioned gmres iteration, Advances in Computational Mathematics 11 (1999):253-270, July 1999.

[29] R. K. Beatson, W. A. Light and S. Billings, Fast solution of the radial basis function interpolation equations: Domain decomposition methods, SIAM J. Sci. Comput 22(5):1717-1740, 2000.

[30] R. K. Beatson, J. B. Cherrie and D. L. Ragozin, Fast evaluation of radial basis functions: methods for four-dimensional polyharmonic splines, SIAM J. Sci. Math. Anal. 32(6):1272-1310, 2001.

[31] G. Turk and J. O'brien, Modelling with implicit surfaces that interpolate, ACM Transactions on Graphics, 21(4), 2002.

[32] H. Dinh et al., Reconstructing surfaces by volumetric regularization using radial basis functions, IEEE Pattern Analysis and Machine Intelligence, 24(10), pp. 1358-1371, 2002.

[33] B. Morse et al., Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, IEEE International Conference on Shape Modeling and Applications, pp. 89-98, 2001.

[34] V. Savchenko et al., Function representation of solids reconstructed from scattered surface points and contours. Computer Graphics Forum, 14(4), pp. 181–188, 1995.

[35] Turk, Greg and James F. O'Brien, Shape transformation using variational implicit surfaces., Computer Graphics Proceedings, Annual Conference Series, 1999.

[36] J. Bloomenthal, Introduction to implicit surfaces, Morgan Kaufmann, San Francisco, California, 1997.

[37] J. Duchon, Spline minimizing rotation-invariant semi-norms in sobolev spaces, W. Schemps and K. Zeller, editors, Constructive Theory of Functions on Several Variables, Lecture Notes in Mathematics 571, Berlin, 1977.

[38] E. Cheney and W. Light, A course in approximation theory, Brooks Cole, Pacific Grove, 1999.

[39] Turk, Greg and James F. O'Brien, Variational implicit surfaces, Tech Report GIT-

GVU-99-15, Georgia Institute of Technology, 1999.

[40] S. Cuomo et al., Surface reconstruction from scattered point via rbf interpolation on gpu, Proceedings of the 2013 Federated Conference Computer Science and Information Systems, pp. 433-440, 2013.

[41] J. Bentley, Multidimensional binary search trees used for associate searching, CACM, 18(9), pp. 509-517, 1975.

[42] P. Chalmovianský and B. Juttler, Filling holes in point clouds, Math Surfaces X, M. Wilson and R. Martin, eds., pp. 196-212, Springer-Verlag, 2003.

[43] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree, AICM, 4, pp. 389-396, 1995.

[44] P. Krysl and T. Belytshko, An efficient linear-precision partition of unity basis for unstructured meshless methods, Communications in Numerical Methods in Engineering, 16, pp. 239–255, 2000.

[45] M-S Kim and K. Shimada, Geometric modeling and processing - GMP 2006, 4th International Conference Pittsburgh Proceedings, PA, USA, Springer, 2006.

[46] L. Sang-Hyun et al., 2D Image to 3D Based on Squeeze function and Gradient Map, International Journal of Software Engineering and its Applications, vol.8, No. 2, pp. 27-40, 2014;

SYMCOMP 2015
Faro, March 26-27, 2015

MECHANICAL DESIGN IMPROVEMENT USING SYMBOLIC AND NUMERICAL COMPUTATIONAL TOOLS

Costa, D.M.S.[1]* and J. Infante Barbosa[1,2]

1: GI-MOSM, Grupo de Investigação em Modelação e Optimização de Sistemas Multifuncionais
ISEL, IPL - Instituto Superior de Engenharia de Lisboa
Av. Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal

e-mail: a38293@alunos.isel.pt

2: LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal

e-mail: jib@dem.isel.ipl.pt

Keywords: Finite Element Analysis, Composite Materials, Symbolic Computation, 3D Geometric Modelling, Trailer, Numerical Models.

Abstract

This paper refers to the improvement of the mechanical design of two wheels (one axle) trailers, for the carriage of hounds and miscellaneous equipment. An introduction in which the objectives of the work are exposed, followed by presentation of the motivation that led to the choice of this project is performed. After that, the documentation/literature used in the project is described, to make easier the comprehension of subsequent analytical and numerical models. A detailed description of all models built to carry out the work – Aluminum Chassis, Fairing Aerodynamics in Composite Materials, Steel Ball Hitch and Weight Support Plates in Composite Materials are presented. After having done all the required calculations and simulations, the results obtained are analysed in order to carry out all the comparative studies necessary to understand the product, to improve it and ensure its theoretical security. The use of symbolic computation in the development of analytical models and the use of the appropriate numerical tools, using specialized software to perform finite element analysis, as well as 3D geometric modelling and simulation, are widely demonstrated by presenting a several number of studies on the draft of the trailer and its improvement process in order to increase their efficiency to the service conditions for which it was designed. Appropriate conclusions, followed by further developments are made, due to the size of the project and consequently the inability to realize it completely.

1. INTRODUCTION

This paper refers to a project developed under the Course of Mechanical Design of the School of Engineering of the Polytechnic Institute of Lisbon. This design was developed in part with a trailer comprising five separate compartments, to carry hunting dogs and miscellaneous equipment.

The main objective of the present work is to show that the conjunction between the commercial finite element method code and the development of analytical models, with use of symbolic computation, can significantly improve the design of solutions that are being developed. In addition to the easiness of implementation compared with the optimization methods, should point out the very favourable environment for the learning process and the assimilation of concepts, using real problems and solutions available in the market.

For the structure's / chassis' trailer is provided the use of non-ferrous material, aluminum (several beams welded) with five independent compartments: four for the transport of hunting dogs and one for the transport of miscellaneous equipment (storage area).

It was also considered to incorporate an outer aerodynamic fairing to the chassis, in order to minimize the fuel consumption, thereby reducing the drag coefficient of the trailer. This component is produced from composite materials, in order to reduce the weight of the product and to increase its mechanical strength.

The introduction of the load support plates, also made of composite materials, has also been considered in order to prevent the corrosion due to the transport of animals (moisture, etc.), reducing the weight of the product and ensuring easy disassembly (to make the cleaning of the interior easier, for example).

In this paper we only present some of the fundamental aspects of the full trailer project. One aspect that arises is the analysis and the design of the structural part of the product, the following components: aluminum chassis, both aerodynamic fairing and support plates in composite materials. It is also studied the influence of the trailer on the hitch ball made of steel, to check if the product would not compromise the use of a conventional ball and hitch available on the market. In addition, several simulations are carried out to study the influence of the hitch ball's geometry on the stress concentrations – more specifically, its coupling radius and angle.

To complement the structural analysis of the trailer was also performed a CFD (Computational Fluid Dynamics) analysis of the external flow on the fairing.

In order to design the product's geometry, SolidWorks was used. The result of this modelling can be visualized in Figure 1.

Figure 1 – 3D trailer modelling with SolidWorks.

In order to carry out the study of the components, commercial software was used to enable the displacements, stresses, safety factors and failure criteria (orthotropic materials) calculation, through the use of Finite Element Method (FEM). The simulations were based on numerical models created especially for this application, divided into several versions corresponding to each particular situation (trailer in acceleration, braking, cornering, etc.).

All numerical models were validated by analytical models, anticipating critical situations in the operation of the trailer. The Maple software was used in the preparation of symbolic computation of analytical models. In FEM simulations, ANSYS MECHANICAL APDL code was used, to program the models' routines.

The results were analysed and supported by comparative studies, which lead to the conclusions of the work and the proposed future developments.

2. DESCRIPTION OF STANDARDS AND TECHNICAL DOCUMENTATION

During the project of the trailer several standards and technical documentation were used, in order to carry out checks of some components. This section presents that literature, standards and technical documentation. These documents have been essentially used to study the hitch ball: *ISO (International Organization for Standardization)*; *ISO 1103: 2007 - Road vehicles - Coupling Balls for Caravans and Light Trailers - Dimensions*; *ISO 3853: 1994 - Road vehicles - Towing vehicle coupling device to tow caravans or light trailers - Mechanical strength test.*

The *ISO 7641: 2012 - Road vehicles - Trailers up to 3.5 t - Calculation of the mechanical strength of steel drawbars* standard was used to analyse the trailer's drawbar.

Was also used *UN-ECE Regulations (United Nations Economic Commission for Europe)*: *UN-ECE Regulation No. 55; Review 1: 2001 - Uniform Provisions Concerning the approval of mechanical couplings components of combinations of vehicles*.

In order to analyse the trailer's chassis, several *Eurocodes* were used: *EN 1990- Eurocode 0 - Basis of Structural Design*; *EN 1991 - EC1 - Eurocode 1 - Actions on Structures*; *EN 1999 - EC9 - Eurocode 9 - Design of Aluminium Structures*.

3. ANALYTICAL AND NUMERICAL MODELS

3.1. INTRODUCTION

In the present section a brief description of all analytical and numerical models developed in this project is done. This type of product is unavoidably exposed, during its lifetime, to different types of actions, including:

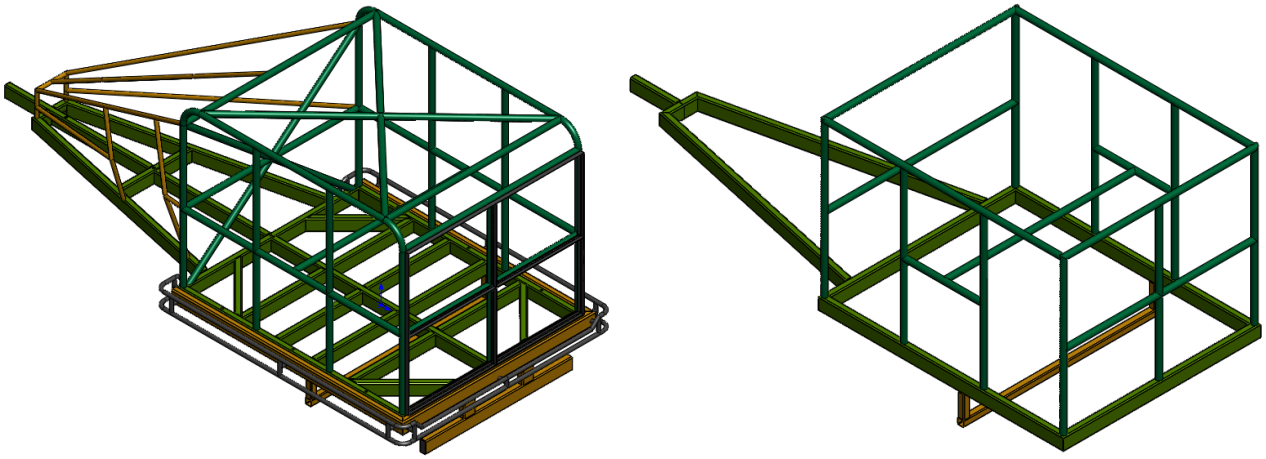
1. Static Actions – "Actions that do not cause significant accelerations of the structure or structural elements." [1]. Examples of such actions are: Weight of the dogs, weight to be transported into the storage zone, own weight of the structure, the weight of the trailer itself, as the fairing components, the weight support plates, the snow (additional weight to the structure), etc.

2. Dynamic Actions – "Actions that cause significant accelerations of the structure or structural elements." [1]. Examples of these actions are: longitudinal accelerations caused by the towing vehicle and transmitted through a force imposed by the hitch ball on the trailer, lateral acceleration due to wind, lateral acceleration due to centrifugal forces that arise in curves, vertical accelerations due to bumps, impacts on holes or irregularities of the road, impacts in case of accidents with other objects (cars, rails, walls), snow, fire, etc.

3.2. STATIC ANALYSIS OF THE ALUMINUM CHASSIS

3.2.1. NUMERICAL MODEL: ITERATIONS USING SOLIDWORKS SIMULATION

The introductory numerical model of the chassis was made in SolidWorks Simulation, wherein the component is simplified, reducing the number of beams and keeping in mind that the initial release (see Figure 2 a)) had a clear oversizing.



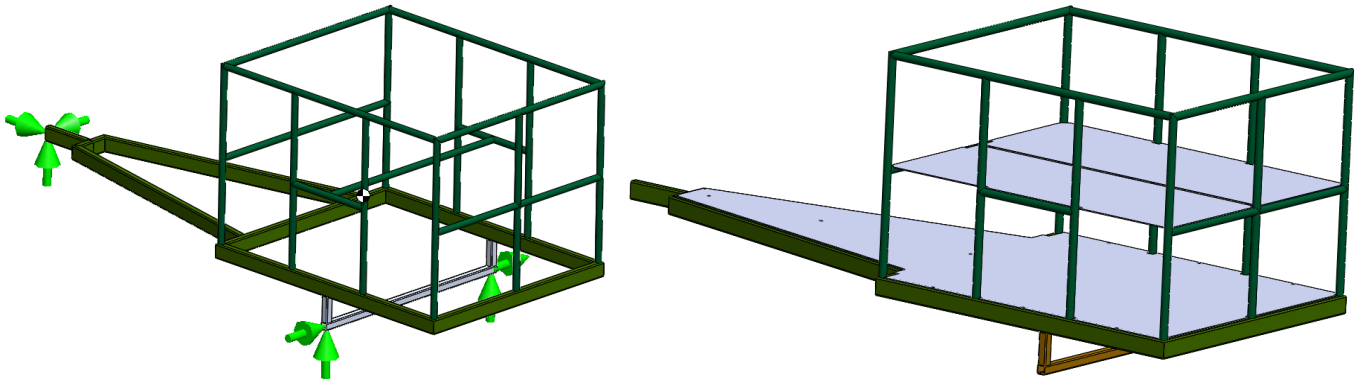
a) b)

Figure 2 – Initial release a) and version 2 b) of the chassis.

This first phase of the static chassis study involved various settings until it reaches a more simplified model (see Figure 2 b)), which meets the requirements of conditions of service, with a metallic structure much less complex.

The summary of the key variables used in this simplified version is:

1. Down Structure: Rectangular Tube Section Normalized ISO 657-14 - 80x40x3.2 [mm];
2. Top Frame ("Rear Box"): Circular Tube Section Normalized ISO 657-14 - 33.7x2.6 [mm];
3. Wheels' Axle: Squared Tube Section Normalized ISO 657-14 - 40x40x2.6 [mm];
4. Software: SolidWorks Simulation 2014; Material: Aluminium EN-AW 5454; Mesh: Beam - Beam Element;
5. Attachments: Hitch (Uall = 0), wheels (Uy = 0, Uz = 0) - (Fig 3 a.); Actions: Weight of Dogs + Equipment Weight - Y Direction - 3008 N.



a) b)

Figure 3 – Constraints considered a) and simply supported plates b).

In order to carry the dogs and goods, rectangular plates must be used to transmit the loads to the structure. In the first version of the trailer, the rectangular plates are bolted to the chassis, but in the second version it is considered that the rectangular plates are simply supported on beams (see Figure 3 b)). Figure 3 a) shows the considered constraints for this particular simulation. In a first approach, to simplify the problem, it is assumed that the weights are concentrated forces in the calculation of the reactions at the edges of the rectangular plates (see Figure 4).

Before performing the simulation itself, a quick simulation was done to predict the appropriate trailer jack location. As is known, the trailer must have stability when it is loaded or not, that is, if it is parked, it weight distribution (centre of gravity) must ensure that it will tend to "fall" to the side of the coupling so that not tip back and rest on the jack device. On the other hand, when loaded must also ensure stability.

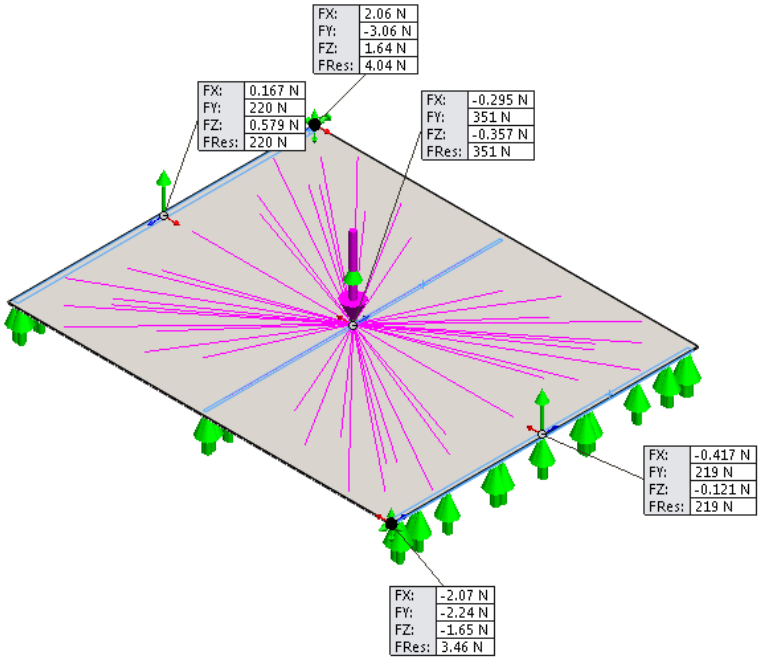


Figure 4 – Applied force (785 N) and reactions in the upper back plate.

In Figure 5 a), b) and c) the applied loads to the chassis are presented. On the other hand, the mesh is shown in Figure 6, comprising 544 nodes and 530 beam elements.

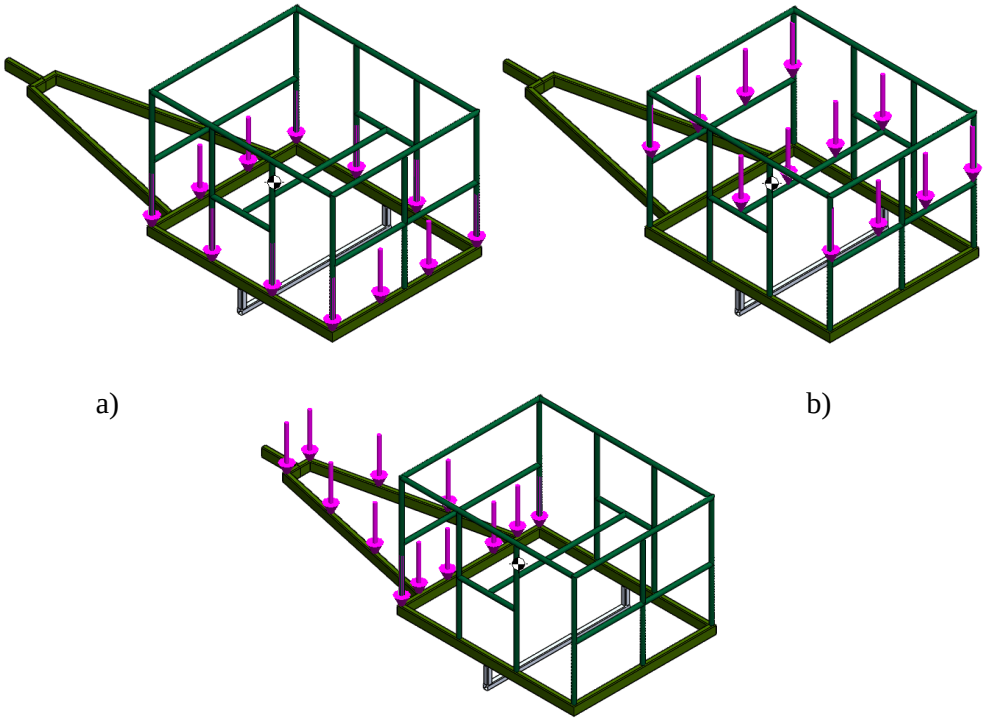


Figure 5 – Vertical forces applied to the chassis lower a) and upper b) sections and drawbar c).

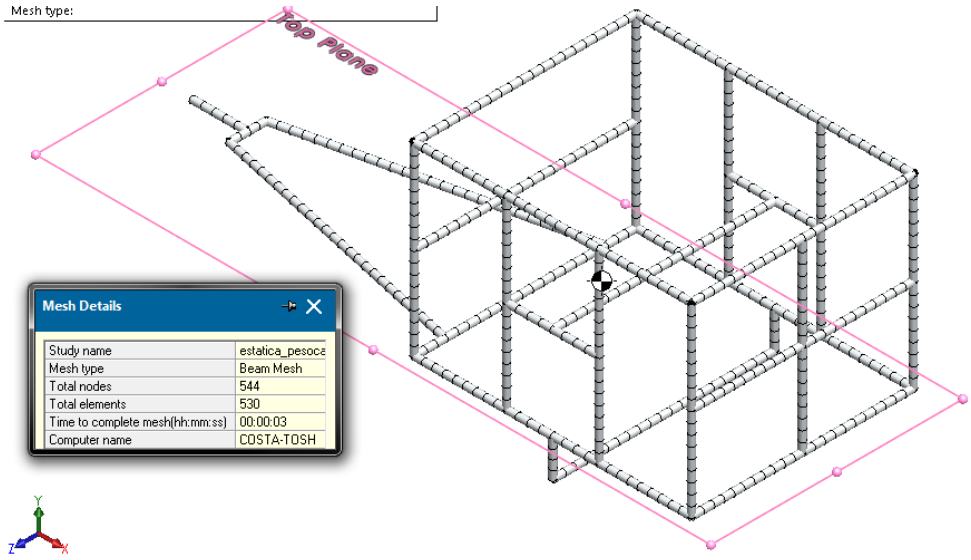
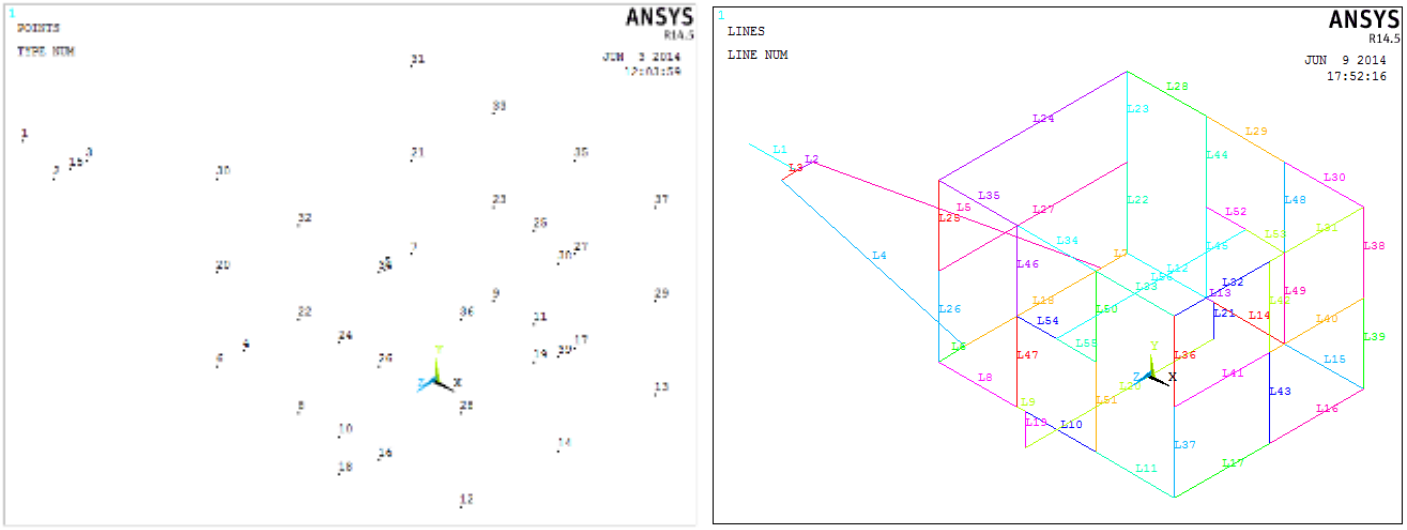


Figure 6 – Type of mesh used in the chassis, beam element (SolidWorks).

3.2.2. NUMERICAL MODEL: ITERATIONS USING ANSYS MECHANICAL APDL

In the second version it was used for the finite element analysis the commercial program ANSYS MECHANICAL APDL. In order to facilitate future modification of the model in ANSYS, a programming routine associated with it has been developed. This model is the same as the described above for the SolidWorks Simulation, for comparison proposes. First of all, it was introduced all keypoints needed to model formulation – 39 keypoints (see Figure 7 a)). Later were created the necessary lines to join the keypoints – 56 lines (see Figure 7 b)).

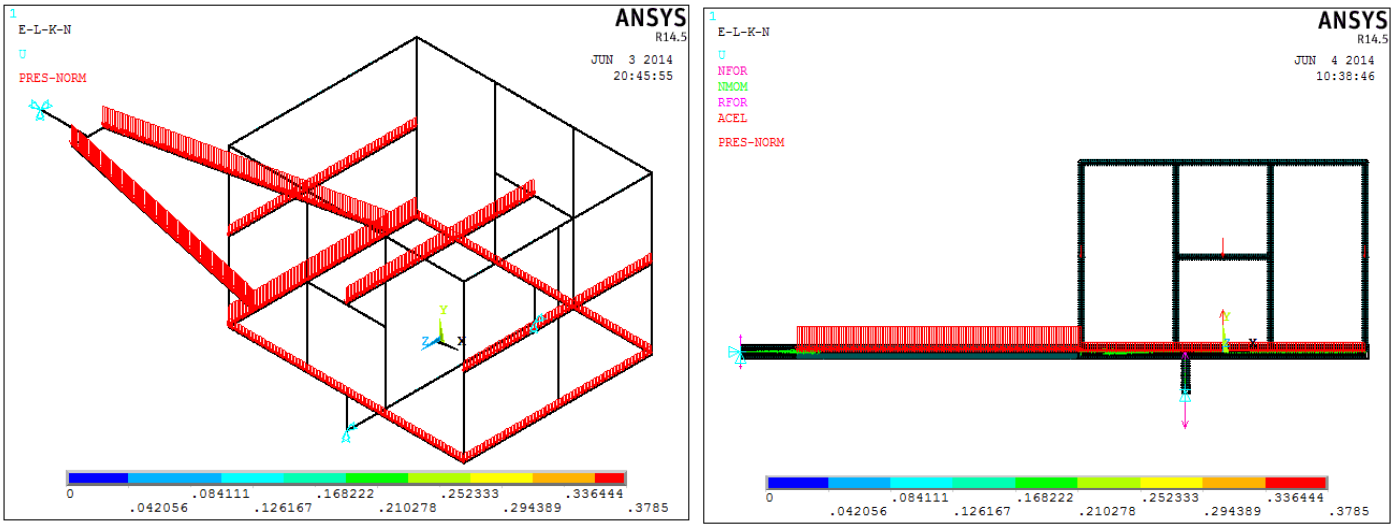
In the finite element discretization it was used a beam element with 3 nodes, BEAM189. As in the previous model, the aluminum has an elastic modulus of 70 GPa and Poisson's ratio of 0.3897.



a) b)

Figure 7 – Chassis' keypoints a) and chassis' lines b), in ANSYS MECHANICAL APDL.

The vertical forces applied to the beams had to be converted into distributed loads along the lines, i.e. pressures (see Figure 8) (force per unit length of beam).



a) b)

Figure 8 – Distributed loads applied on the chassis' beams a) and the shifted axle b).

Due to the complexity of the project and the huge amount of checks and simulations, we came to the conclusion that the most cost effective way of achieving it is using the possibilities of ANSYS APDL. The time used to program all versions made was very considerable, but then the successive changes are much easier to program (sections' changes, geometry, material, laminate stacking, etc.). The fact that it is possible to use a single solution for various types of elements at the same time has simplified the entire project.

As in the simplified version of the chassis displacement is very high, it was made an iterative process to reduce it. Be introduced to a new version three changes. The first change is to shift the axle of the support structure about -200 mm in the X direction, to bring the support of the wheels of the trailer to the centre of gravity of it. The changes of 750 mm distance between the rear wheels and trailer to 950 mm. This easily models up by changing four keypoints in routine programming, relating to axle bearing support (k10, k11, k18 and K19), adding -200 mm to its X position (attached routine). A second change involves the introduction of the density of aluminum mass MP, DENS, 1, 2.7e-6. Finally began to consider the severity toward -Y to consider the own weight of the beams. Through the ACEL command, 0, 9.81,0. Through the details provided by SolidWorks is possible to know that there are about 544 nodes and 530 elements. It is to be noted that in this solution it was not changed the mesh parameters.

Under the used process improvement it was introduced the fairing component. The main objective of this component is to reduce the drag coefficient when the trailer is towed by a towing vehicle. In order to reduce this parameter is necessary to ensure that the depression that comes on the back of the trailer decrease and consequently also reduce the same to the movement opposing force. Thus can be guaranteed a reduction in fuel consumption by reducing the movement resistance of the fair, as opposed to a solution without this component, or if a simple "box". Most of the trailers for transport does not address this situation.

By adding fairings to the trailer, not only influences the aerodynamic vehicle, but also introduces an increased structural rigidity to the frame. It is a component with a very large area (about 12 m^2), which when properly secured to the structure, reducing dislocations at the outset, due to its high moment of inertia. The fairing is made of composite material to reduce weight and increase mechanical resistance. On the other hand, necessarily need to resist atmospheric conditions to be the outside "shell" of the product (UV rays from the sun, rain, wind, snow, rock impacts, etc.).

In Figure 9 is shown the new model. The most important variables used in this numerical model are used composite material is fiberglass-type E with epoxy matrix; the laminate is symmetrical sequence [+ 45 / -45 / 0] 3S; the mass equals to 100 kg; the mesh used is the SHELL181 element; actions in the direction Y are 4329 N.

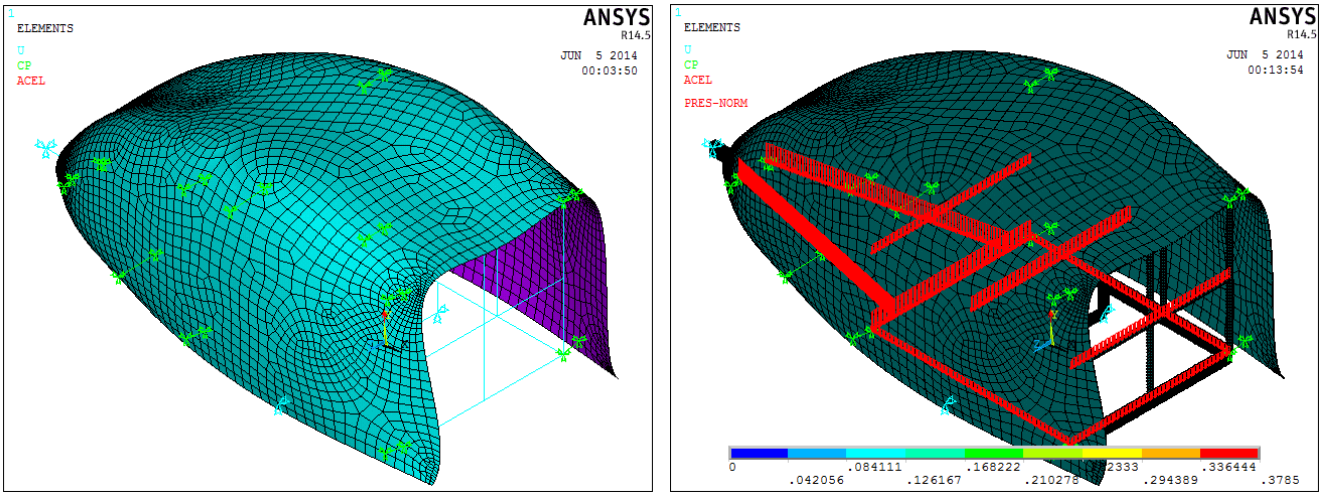


Figure 9 – Modelling the fairing (SHELL181 element.

To consolidate the improvement process of the chassis were used a total of 10 versions with different assumptions loads, geometry and loading conditions. Were also carried out several analytical studies considering the different possibilities of static and dynamic loads (accelerations in straight and in curve lines).

In this model considers the actions associated to version 6 and introduces a longitudinal acceleration to the trailer, in order to consider the inertia of the chassis and fairing.

In terms of animal comfort, the trailer's user cannot / should perform very high accelerations with the towing vehicle during movement. In figure 10 it is shown a typical acceleration curve for this equipment.

Summary of the most important variables used in this numerical model: Longitudinal Acceleration, Start; Acelx = 3.5 m / s2; Actions: direction Y - 4716 N, X direction - 615 N.

In version 8 we considered the situation in which the motor vehicle is traveling in a straight line and hard braking. Acelx considered a deceleration = -10 m / s2 and Y direction actions - 4716 N, the X direction - 615 N. In the simulated version 9, the trailer without longitudinal acceleration, carries out a lateral acceleration curve Acelz = 2.5 m / s2 with actions in the Y direction - 4716 N, X direction - 142 N and Z direction - 338 N. Finally made a version 10 which undertook to trailer behaviour analysis loaded using a buffer in each wheel support and considering action in the Y direction - 4329 N.

Figure 10 rough Figure 14 represent the different loads considered on the several cases analysed.

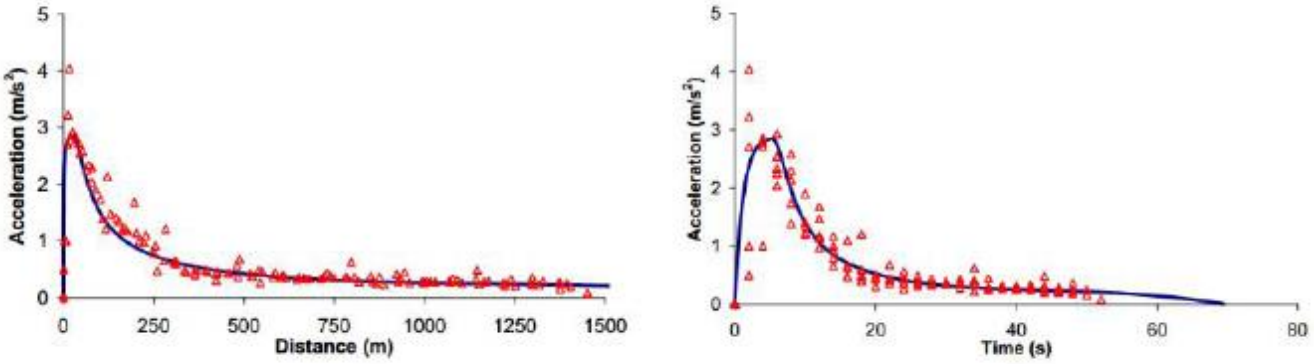
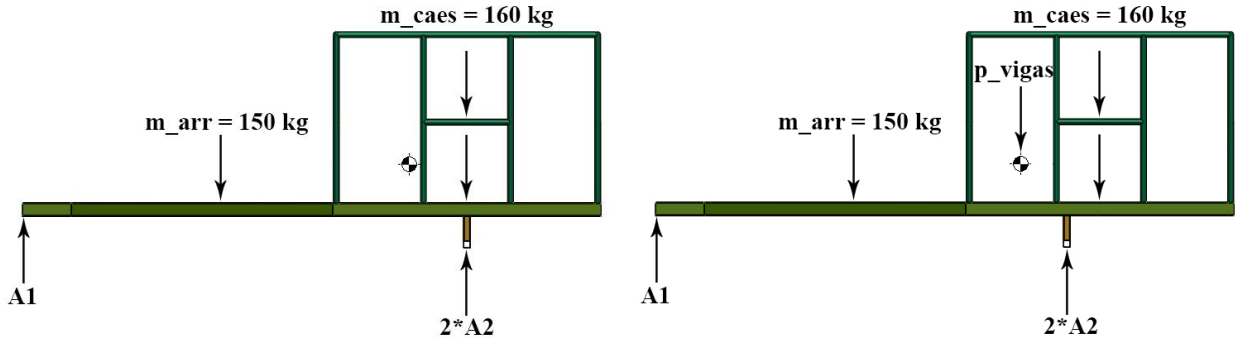
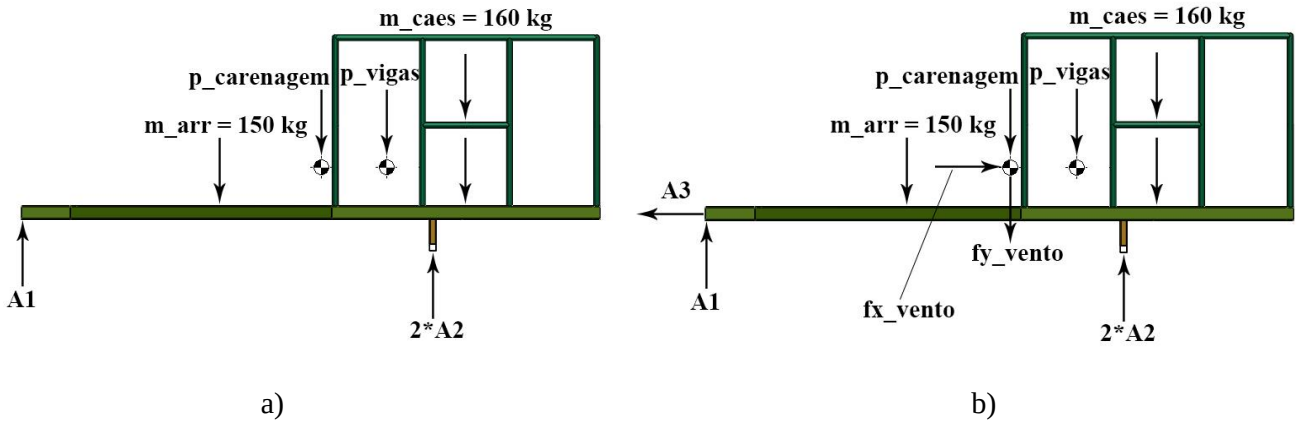


Figure 10 – Typical curves of acceleration, vehicle Chevy S-10 [1].



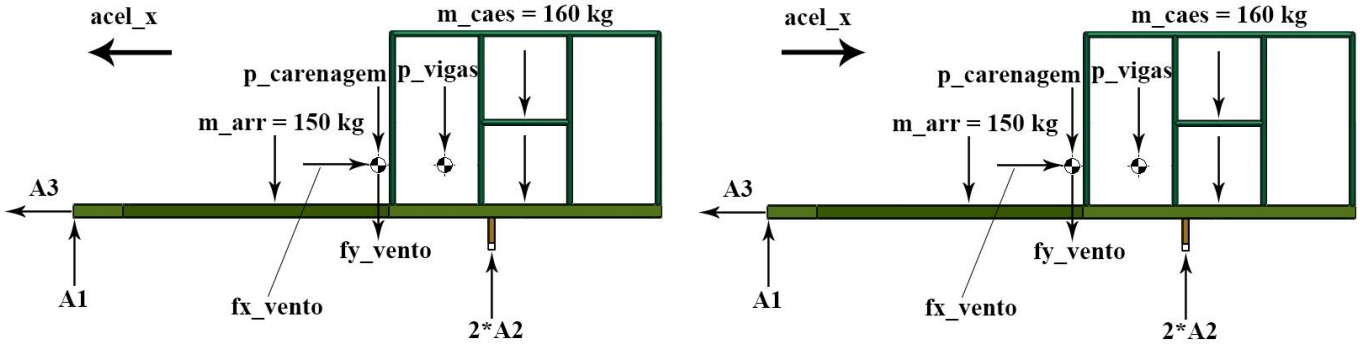
a) b)

Figure 11 – Free body diagram of the chassis - load case 2 a) and load case 3 b).



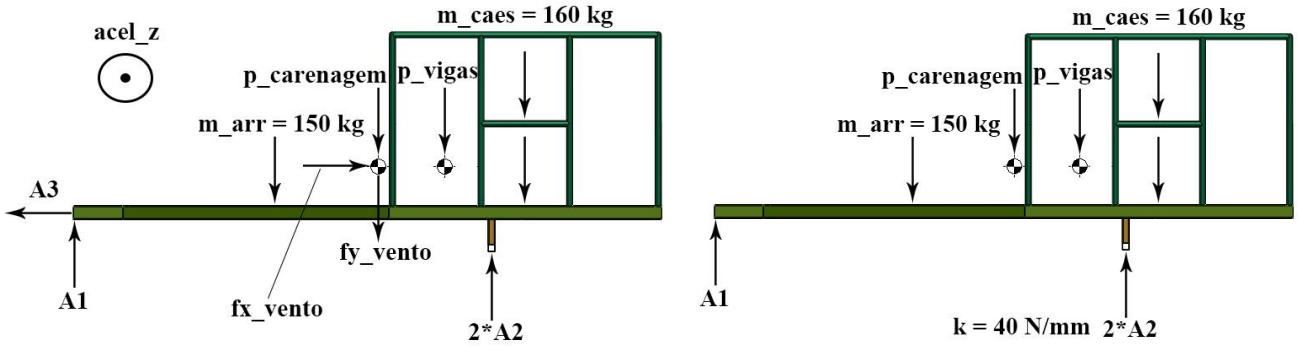
a) b)

Figure 12 – Free body diagram of the chassis - load case 4 a) and load case 5 b).



a) b)

Figure 13 – Free body diagram of the chassis - load case 6 a) and load case 7 b).



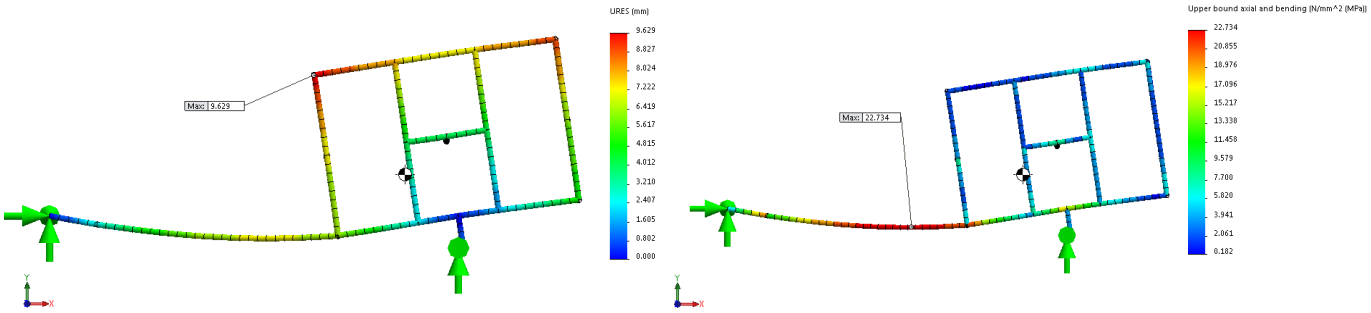
a) b)

Figure 14 – Free body diagram of the chassis - load case 8 a) and load case 9 b).

During the improvement process the maximum Von Mises stress value was reduced to 18.3 MPa, starting from a starting value of 33 MPa. This was a high reduction. This happened because the moving of the axle position closer to the centre of gravity, bending the drawbar reduces the bending moment and therefore also the stresses in the beams. The fact that it is to apply the own weight of the low beams influences the outcome, given that the chassis weight

is greatly reduced.

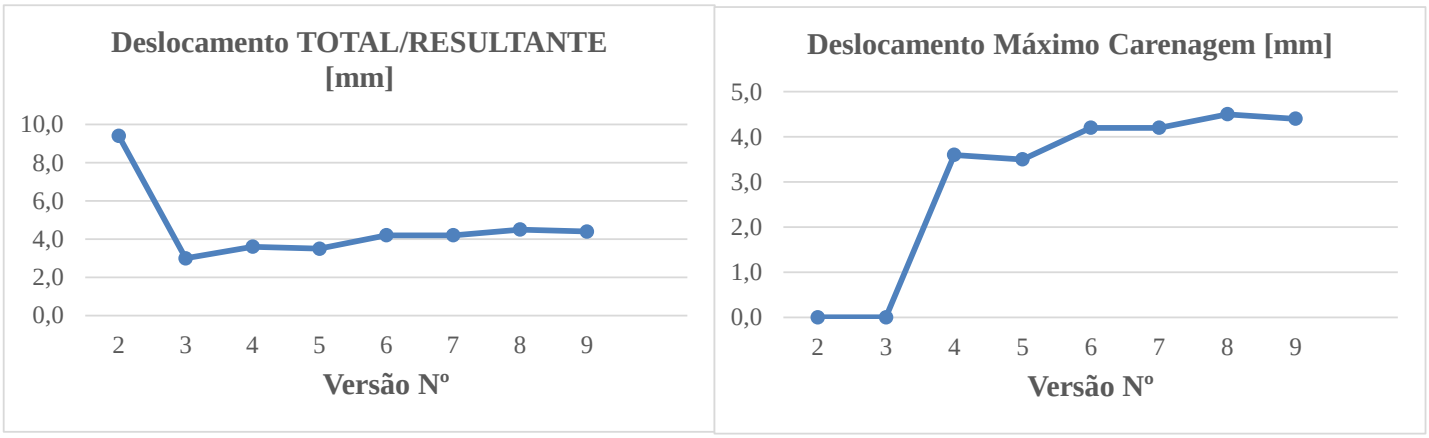
Note that in the comparative studies between analytical models and numerical models developed in ANSYS, the maximum difference noted was 1%.



a) b)

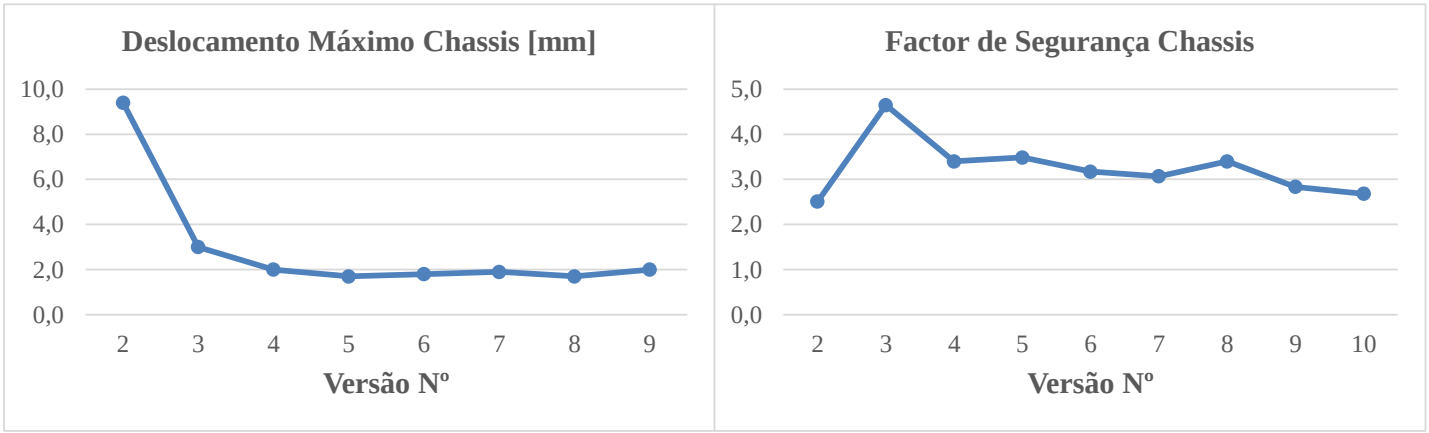
Figure 15 – Maximum displacement in the chassis a) and maximum axial and bending stresses b).

In Figure 16 through Figure 17 are shown the results for total displacements and safety factors for different load cases analysed (version 2 to version 9).



a) b)

Figure 16 – Total displacement to the chassis a) and the total displacement on fairing b).



a) b)

Figure 17 – Maximum displacement on the frame a) and the chassis safety factor b).

3.3. DYNAMIC ANALYSIS OF THE ALUMINUM CHASSIS

3.3.1. NUMERICAL MODEL: CFD USING SOLIDWORKS FLOWSIMULATION

In order to be possible to predict what efforts the wind causes in the fairing when the trailer is towed by the towing vehicle, there was made an external flow simulation in SolidWorks FlowSimulation. Not considered flow rates in the lateral direction or vertical to the trailer, only longitudinal.

As the trailer is mandatory towed by a motor vehicle, the flow behaviour in the back of it is always dependent on its type, that is, if a vehicle type "sedan" the wind focuses much more on the fairing, unlike the situation is a jeep, for example. In the case of jeep, due to their high volume and geometry of the rear of the form is a far superior depression, characterized by the re-circulated air (vortex effect). As the towing vehicle has as much influence on the trailer in terms of aerodynamics, is modelled, in addition to the fairing, a jeep Mitsubishi Pajero TR4 2.0 16V (2003) block. In the simulation it was placed in front of the fairing, to analyse the combination of both vehicles.

Although the fairing being shaped surface, a block was formed from this 3D geometry. The blocks are used instead of real models, because that reduces the simulation time and any errors that may appear – are eliminated automatically holes, internal cavities, etc., simplifying much the model (and it's okay to do this, taking into account which are the exterior surfaces of the blocks to be analysed in terms of flow). In Figure 18, are shown the arrangement of the jeep and the fairing.

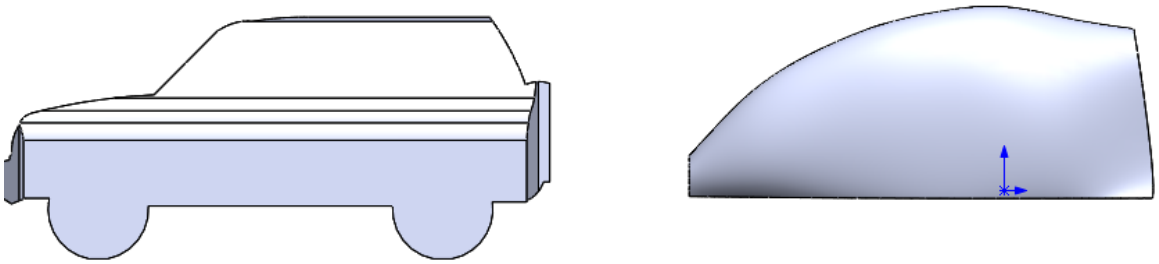


Figure 18 – Modelled blocks used in FlowSimulation.

The fairing 3D modelling was performed and was gradually up iterating from complex solutions and solutions for simple but efficient (Figure 19). After the modulation of the component the data obtained it was exported to IGES format to allow their importation into ANSYS APDL. For analysis of this component in the composite material it was used an elastic orthotropic linear model using the SHELL181 element.

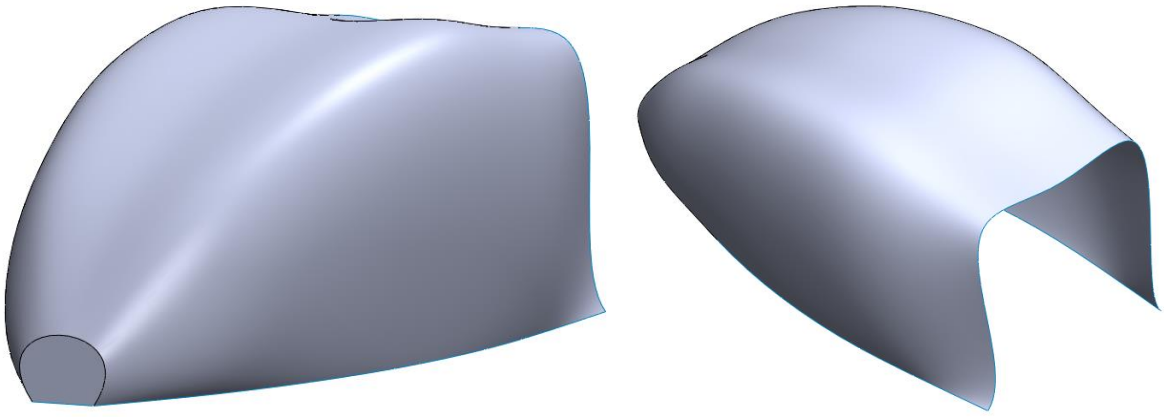


Figure 19 – Fairing modelled in SolidWorks.

It was used for fairing the stacking sequence [+ 45 / -45 / 0] 3S with a total of 18 layers (Figure 20)

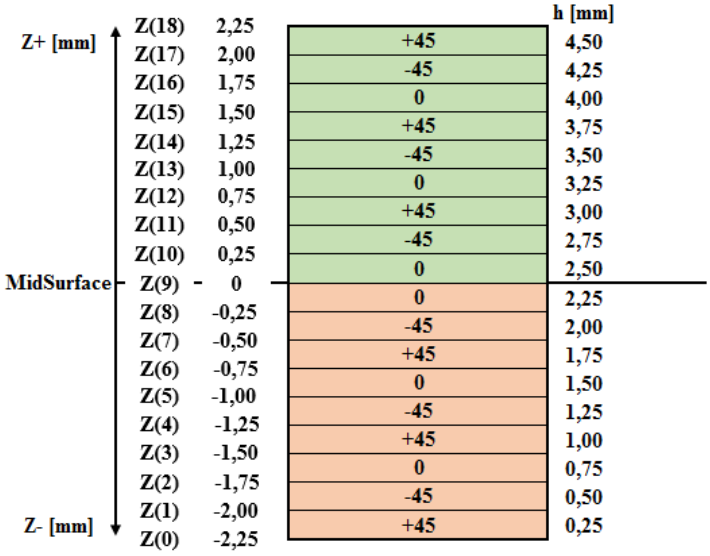


Figure 20 – Stacking sequence of layers and thickness of the fairing.

Figure 21 shows the view of the mesh flow created in SolidWorks FlowSimulation, using 3D modelling and assembly blocks.

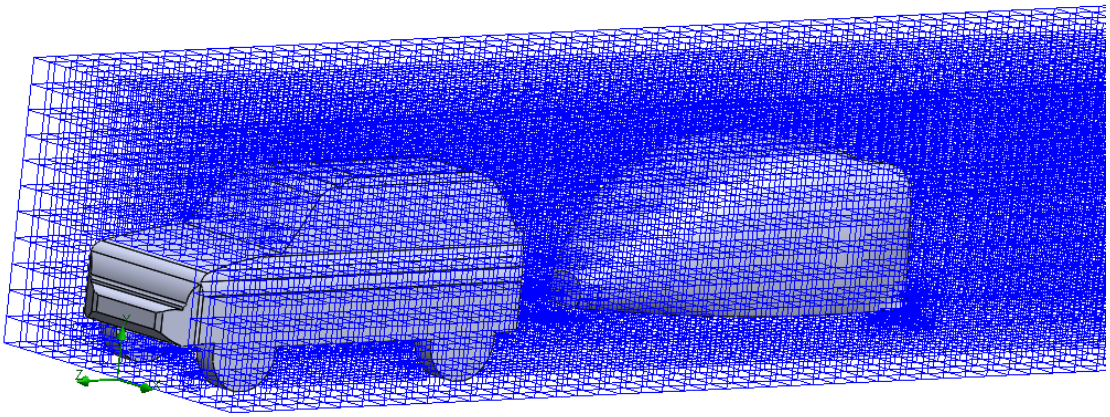


Figure 21 – Definition of the flow loop in SolidWorks FlowSimulation.

In this numerical model apply to wind action on the fairing, which were obtained in the previous model by external flow simulation. After having done extensive research and made several attempts at ANSYS, came to the conclusion that the APDL applying pressures through a complex surface components is harder than it looks. To resolve this issue we selected the area of the fairing where you wanted to apply forces, we selected all the constituent nodes that area and gave their total number - about 3586 nodes. The best approach, although not the most realistic possible, that we managed to do was to divide the forces by the number of nodes and apply them to each as concentrated forces: Towards Y, applied in each node a concentrated force equal to 387/3586 = 0.1079 N and X direction, apply in each node a concentrated force equal to 142/3586 = 0.0396 N.

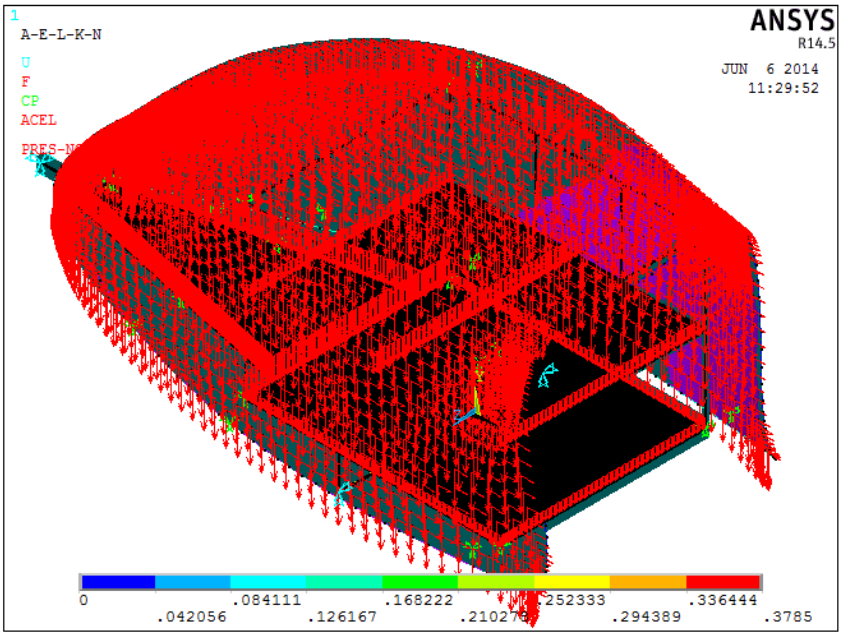


Figure 22 – Model loads applied to the chassis.

The main objective of this simulation is to predict the loads that air exerts on the fairing when the trailer is towed, so you can move to the Chassis Version 6, with these loads applied instead of the optimization of the drag coefficient of geometry. However we can also observe, through graphics, the moving assembly behaviour through the pressure of view, speed and flow lines. Figure 22 shows the resulting graph of speed during towing considering only the fairing. There is a high reduction speed behind the fairing, which means that there recirculation flow due to the mat. With optimum geometry this issue could be minimized and further. It also appears in Figure 23 high mat just behind the jeep, much higher than the fairing, due to its high volume.

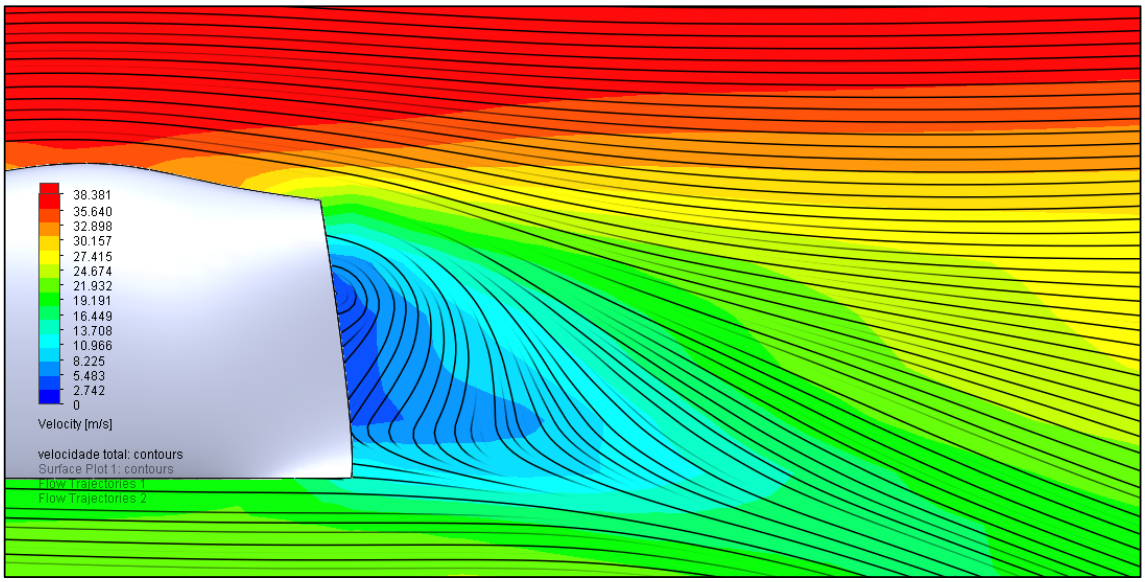


Figure 23 – Resulting graph of speed when towing (only fairing).

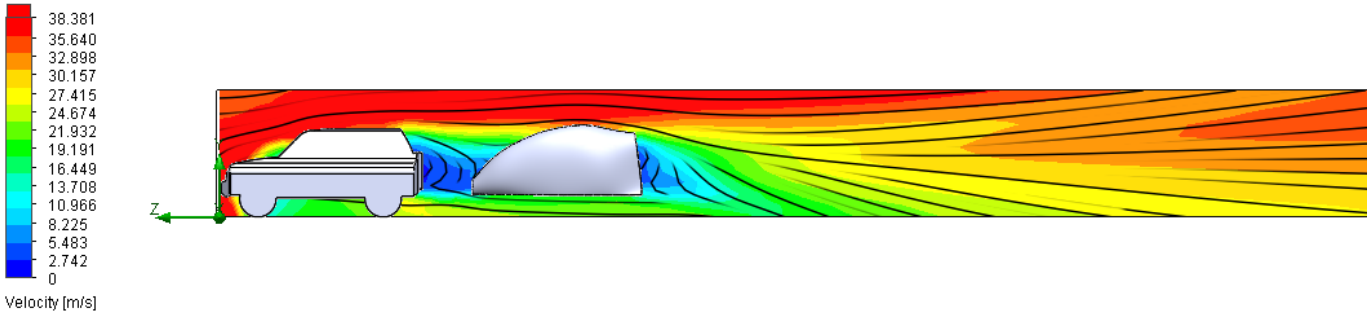


Figure 24 – Resulting graph of speed when towing (both vehicles).

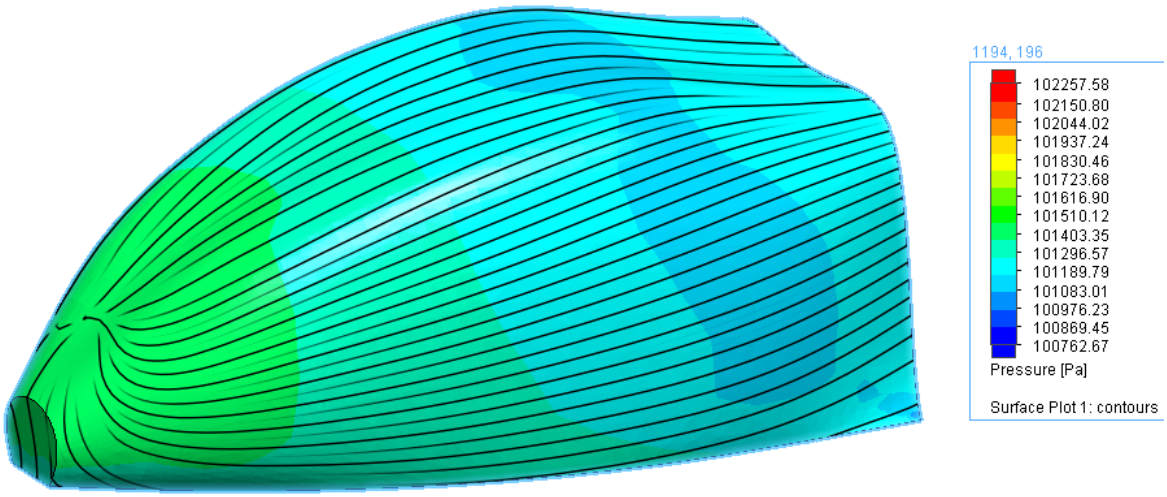


Figure 25 – Current lines on the surface of the fairing (pressure).

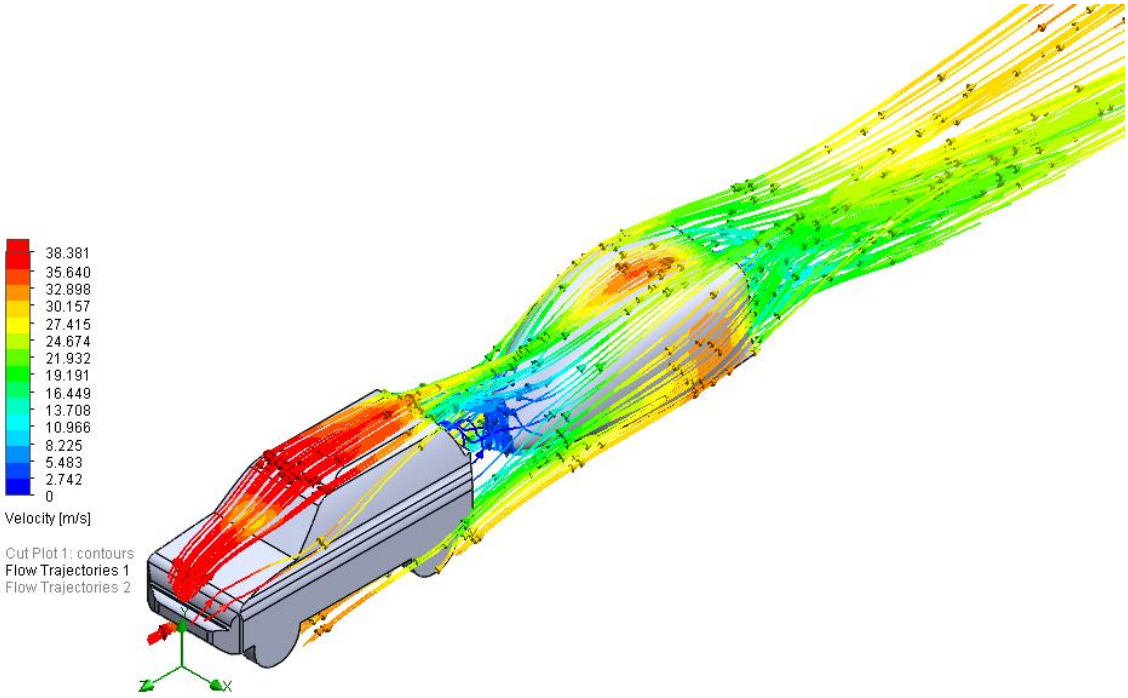


Figure 26 – Display of flow velocity in SolidWorks FlowSimulation.

Through the analysis of the results it is concluded that the force applied on the fairing in the longitudinal direction of the trailer is 142 N and the vertical direction is 387 N.

Another conclusion that can be carried out is due to the fact that the fairing is behind the jeep in the longitudinal direction of the trailer the force is much smaller than the force in the vertical direction. Therefore, the chassis will be more requested in the Y direction than in the X direction, the action-reaction principle (the fairing is attached to the chassis). According to the undertaken and presented in the project report results the drag coefficient is approximately 0.1, which is quite positive. A common carrier, as explained above, has a drag coefficient close to 0.3. In this solution the trailer, with the built fairing has a drag coefficient 3 times lower.

3.4. STEEL HITCH BALL – STATIC ANALYSIS USING SOLIDWORKS

One of the goals of the model developed for the trailer connection was the analysis of the steel hitch ball with respect to the trailer action in order to draw conclusions about its safety in the most critical situations. In addition to this check, another objective was to study the influence of the curvature geometry in the study of the stress concentrations in the drawbar. Using the ISO 1103 standard and the UN-ECE Regulation No. 55 hitch ball to modelled using the SolidWorks Simulation, taking into account the related requirements. In Figure 26 there is shown the first version analysed.

Summary of the most important variables used in this numerical model: Material: Steel AISI A3; Mesh: Solid Mesh; Constraints: embedding in support; Actions: Y Direction - 1151 N Directing X - 1260 N (sudden braking).

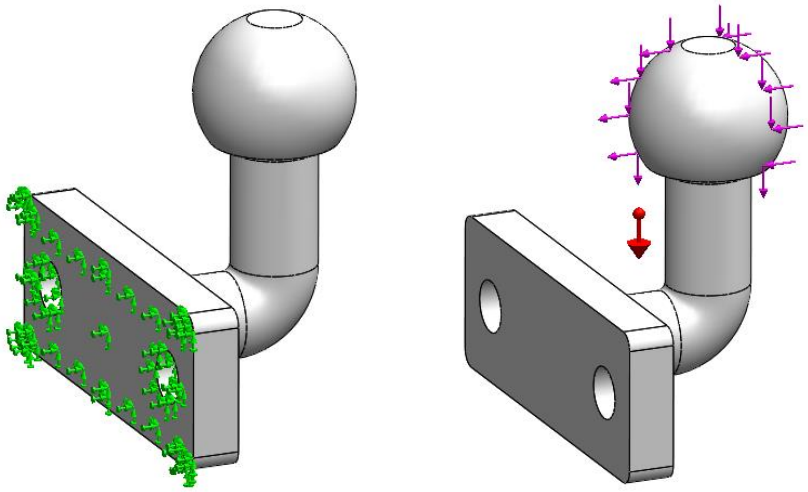


Figure 27 – Steel hitch ball in version 1, constraints (left) and applied loads (right).

In the same configuration, loads and the boundary conditions, was conducted a second analysis (version 2) that for the same type of mesh (solid mesh) was applied Adaptive Method H using the mesh coarsening option to refine the mesh in higher stress concentration zone, in the curved bar.

3.4.1. INFLUENCE OF THE CURVATURE RADIUS OF THE STEEL HITCH BALL

In the numerical model, taking into account the tests required by the ISO 3853 standard, is performed fatigue test. Figure 28 is the model of actions and boundary conditions using the SolidWorks Simulation 2014. The most important variables used in this numerical model (version 3): Attachments: embedding in support; Actions: Y Direction - 615 N Directing X - 2294 N; No Cycles: 2 * 10 ^ 6, Sinusoidal Fully Reversed.

In the numerical model, we performed a study on the variation of the stress concentration of the hitch ball with radius of curvature (see Figure 29). It was analysed the radius variation and the inclusion of a fillet. Performing simulations of fatigue accompanied by the application of Adaptive Method H, take up conclusions about the influence of the radius of curvature in the stress concentration in component (depending on geometry). From a radius R15 in previous versions were carried out various simulations (R15, R20, R25 and R30) and it was found that the lower equivalent stress is obtained with an R30 radius (version 4).

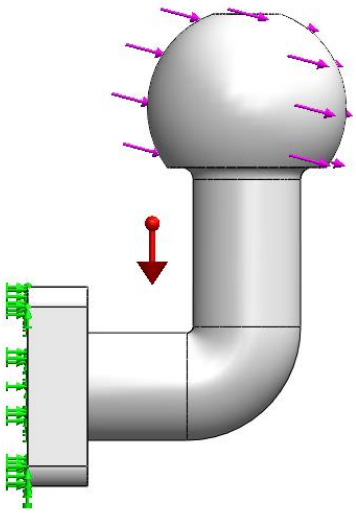


Figure 28 – Numerical Model: Study of fatigue (Hitch Ball Version 3).

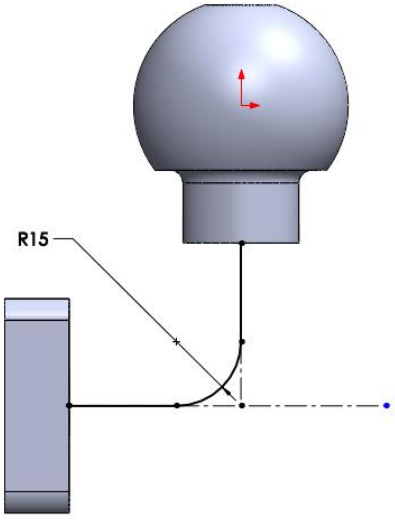


Figure 29 – Effect of the radius of curvature R (steel hitch ball).

3.4.2. INFLUENCE OF THE COUPLING ANGLE OF THE STEEL HITCH BALL

In this numerical model was carried out to study the influence of the angle (50, 100, 150 and 200) of the bar (Figure 14) in the stress concentration. Remained the R30 radius and there were the fatigue simulations accompanied by the application of the h-adaptive method in order to take up conclusions about his influence on the stress concentrations in component (as a function of the geometry). Remained the radius of curvature equal to 30 mm, the frame in the holder, shares: Y direction - 615 N, X direction - 2294 N and the Cycles NO: 2 * 10 ^ 6, Sinusoidal Fully Reversed.

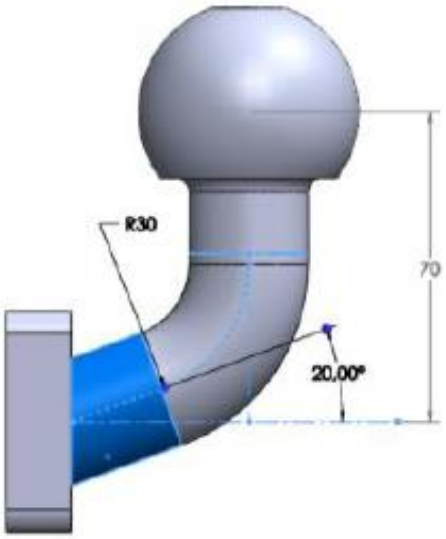


Figure 30 – Effect of the angle of the steel hitch ball (version 6).

4. CONCLUSIONS

The dimensions, geometry and material of the chassis were quite adequate, that is, models, without much change can meet much of its function. In static analysis terms, the displacements were quite acceptable, as the safety factors. There have been significant improvements over the initial models adopted.

In summary can be summarized as the most important findings with respect to chassis the utilization in the present project of the appropriate materials and dimensions. To highlight the efficiency of the fairing marked reduction process running resistance and the effect of reducing the fuel consumption in the trailer movements. Also noteworthy improvements made to the hitch ball reducing stress concentration through an appropriate correlation of geometric distance and inclination angles used.

In conclusion it can be noted that the use of modern computing means, using numerical and / or symbolic computation, shown to be essential tools in improving the mechanical design process.

ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support given by FCT/MEC through Project PTDC/ATP-AQI/5355/2012 and Project LAETA - UID/EMS/50022/2013

REFERENCES

[1] LNEC. NP EN 1991 - Eurocode 1 - Actions on Structures. 2009.

[2] Vehicle Dynamics Model for Estimating Maximum Light Duty, Vehicle Acceleration Levels. Rakha, Hesham, Snare, Matthew e Dion, François. 2003.

SYMCOMP 2015
Faro, March 26-27, 2015

PHOTOGRAMETRIC TECHNIQUES TO HEALTH MONITORING CONTROL OF BREAKWATER'S STRUCTURE USING SCILAB

Rute Lemos[1], Amélia Loja[2], João Rodrigues[3] and José A. Rodrigues[4]

1: Núcleo de Portos e Estruturas Marítimas
Laboratório Nacional de Engenharia Civil,
Av. Brasil, Lisboa
e-mail: rlemos@lnec.pt

2: Área Departamental de Engenharia Mecânica
Instituto Superior de Engenharia de Lisboa,
Rua Conselheiro Emídio Navarro Lisboa
e-mail: amelialoja@dem.isel.ipl.pt

3: Área Departamental de Engenharia Civil
Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro Lisboa
e-mail: joaocarloslr@gmail.com

4: Área Departamental de Matemática
Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro Lisboa
e-mail: jrodrigues@adm.isel.pt

Keywords: Photogrametry, health monitoring, breakwater, image processing, Scilab

Abstract. *The need for shelter zones in coastal areas leads to the construction of structures for shore protection such as breakwaters. Its design is mainly made based in semi-empirical formulas [1] [2] [3] and in the experience of the project designer, being the effectiveness of the breakwater carried out through physical modelling.*
The Harbours and Maritime Structures Division of National Laboratory (NPE, portuguese acronym) for Civil Engineering (LNEC, portuguese acronym) have developed several experiments in bidimensional and tridimensional models. Through this experiments, the stability of the breakwater is tested and the evaluation of damage is made by counting the armor units dislocated (movements or displacements). This damage evaluation is currently done by visual inspection thus being a subjective method where some relevant movements may go unnoticed.

Additionally, this visual method is time expensive. Therefore it is important to develop and implement an automated health monitoring and detection method.
With this study innovative photogrammetric analysis techniques are proposed to process digital photographs of physical modelling experiments made in LNEC. Mathematical techniques were applied in the procedures leading to assess changes between two photos taken at different time instants.
The procedures were developed using Scilab [4]. With these procedures, we are able to identify and locate the modified areas in the breakwater and even make an assessment of the translational movement of Antifer cubes.

1 INTRODUCTION AND PROBLEM DESCRIPTION

The design of a breakwater is mainly made based in semi-empirical formulas [1] [2] [3] and in the experience of the project designer. So the effectiveness of the breakwater is often made by physical modelling. At NPE (Harbours and Maritime Structures Division of LNEC) one develops extensive experiments in bidimensional and tridimensional models of breakwaters, to evaluate their effectiveness to the waves actions. Through these experiments, where *in-situ* condition of the real structure are simulated, namely concerning the waves characteristics, it is possible to test the stability of the breakwater, being the evaluation of damage done by counting the armor units dislocated (movements or displacements).

Currently the identification of these situations is completely done by visual inspection, and it's made by counting the noticed movements and displacements of the rocks or concrete units [5]. This process has effective and known limitations because it is highly dependent on the experience of the technician, and also on its fatigue which easily happens after a long time doing this job. Additionally it can be influenced by individual skills if more than one person is involved in the process.

With this study it was intended to develop innovative photogrammetric analysis techniques based on mathematical methods, for digital photographs comparative analysis of breakwaters status, before and after physical modelling experiments.

These techniques are considered in the context of the procedures that assess changes between two photos taken at different time instants. Procedures were developed using Scilab [4], however algorithms could be implemented in other programming languages or platforms. The study developed allows to identify and locate the modified areas in the breakwater and even to make an assessment of the translational movement of Antifer cubes. As far as the author's knowledge, no similar or related study was published until the moment, as can be concluded from the literature review carried out.

2 PROPOSED SOLUTION

2.1 General aspects

The proposed solution should be developed based on the existence of a single pair of photos for each experiment. This was an existing restriction associated to the real procedure carried out during these tests. So the available input information to be worked was condensed in those two digital photographs. Considering other aspects related to an easier and free access to a computation platform, leaded to the use of Scilab [4] and Scilab Image and Video Processing Toolbox (SIVP) [6], which are open source softwares, easy to install, with expeditious programming facilities. As in other platforms, the reading of a color photograph by scilab+SIVP is interpreted as the reading of an hypermatrix with dimension $\mathrm{M} \in \mathbb{R}^{nl,nc,3}$, as:

$$M(i,j,1),\ M(i,j,2) \text{ e } M(i,j,3),\ \ i \in \{1,\cdots,nl\}, j \in \{1,\cdots,nc\}$$

Each entry has a value between 0 and 255 corresponding to the RGB pixel code, where nl an nc are the number of rows an number of columns respectively (Figura 1).

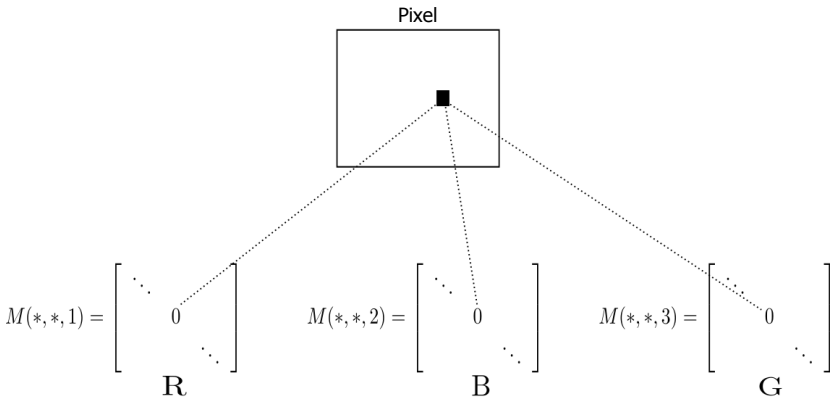


Figure 1: Matricial representation of a digital image

An RGB color space is defined by the encoding of red, blue and green. The RGB encoding of pure red is (255,0,0), pure green (0,255,0), and pure blue (0,0,255), corresponding the total absence of color to the code (0,0,0), i.e. black, and the (255,255,255) code for the white, where all colors are present. The photographs used in this study have a dimension of 3456×2304 pixels, stored in three elements of hypermatrix RBG. Due to the dimension of these matrices, it is particularly evident that the time of image analysis increases if they are based on each element of the hypermatrix. So using logical tests to each entry of the hypermatrix, increases substantially the execution time.

2.2 Proposed methodology

To reduce the time spent in the images analysis, an approach based on the algebraic manipulation and the reduction of the hypermatrix into a single matrix, was adopted. The reduction of the hypermatrix was implemented through the conversion of the color image into a gray scale image. Gray scale images are rendered in all the shades of gray in between black and white. One may select different relations to convert a color image into grayscales such as (1).

$$\mathrm{GRAY} = 0,3 \times \mathrm{R} + 0,59 \times \mathrm{G} + 0,11 \times \mathrm{B} \tag{1}$$

The decision on the relation to use depends on the goal, as it will be referred next.
Physical models of breakwaters are colored, so one should choose an adequate conversion to isolate the primary colors, with a conversion formula of the kind of (2) To this purpose,

one have selected coefficients c_1, c_2 and c_3 in such a way that it would be possible to isolate in gray scale, one of the three primary colors.

$$\text{Primary colour in gray scale} = c_1R + c_2G + c_3B \tag{2}$$

As previously mentioned, one of the main objectives of this study was the detection of modified/damaged zones on the breakwater armour layer, considering the availability of two digital photographs. Therefore for those two images, representing the initial and final moments of a physical experiment, it was required to proceed to their grayscale conversion:

$$BW_i(i.j) \quad \text{initial gray scale matrix} \tag{3}$$

$$BW_f(i.j) \quad \text{final gray scale matrix}, \quad i \in \{1, \cdots, nl\}, j \in \{1, \cdots, nc\} \tag{4}$$

After the conversion of these images, the relevant differences between the two resulting image matrices are analyzed. To this purpose, a position matrix P is considered, with the same size as the BW matrices, where its entries are defined as: if the difference between the initial and final images is relevant, the entry of the P matrix will be 1, otherwise is 0. This relevance is defined as a threshold value.

$$P(i,j) = \begin{cases} 1 & \text{se} \quad |BW_f(i.j) - BW_i(i.j)| > \varepsilon \\ 0 & \text{se} \quad |BW_f(i.j) - BW_i(i.j)| \leq \varepsilon \end{cases} \tag{5}$$

The most significant modified areas will be detected via the calculation of the matrix norm, which in this study was considered to be the unit norm, $\|\cdot\|_1$, defined by:

$$\|A\|_1 = \max_{1 \leq j \leq s} \sum_{i=1}^{r} |a_{ij}| \tag{6}$$

for a matrix $A = [a_{ij}]_{r \times s}$.

After this, the process devoted to identify the location of the most modified zone, begins with the division of the position matrix P into 4 blocks. The matrix block with the most modified zone corresponds to the higher entries equal to one, i.e. with a higher matricial norm. This process is performed iteratively, starting on the entire position matrix, decomposing it into four blocks, after which the one having an higher matricial norm is chosen, and decomposed again into four blocks, and so on, until a pre-defined stopping criteria is met. Figure 2 represents graphically the position matrix P. The entries corresponding to most significant modified areas are equal to 1, and are represented through a darker color.

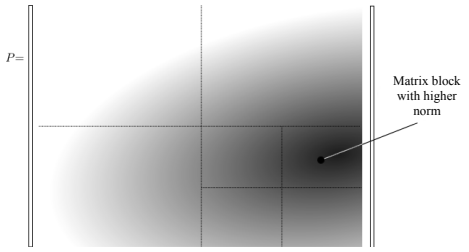
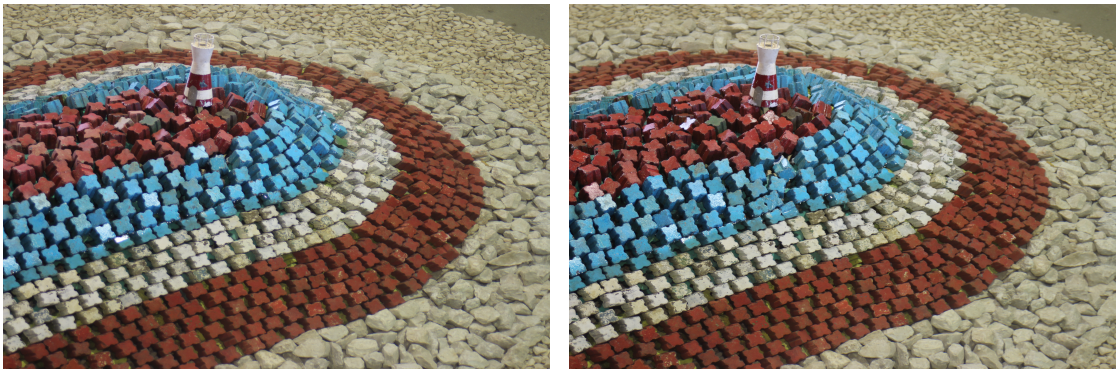


Figure 2: Graphic representation of the matrix P

3 RESULTS

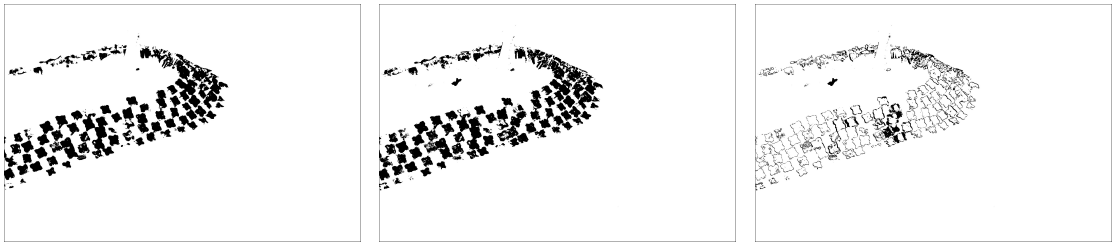
With the conversion of the initial and final digital images to grayscale (2), one has achieved the isolation of the primary colors, represented on those physical model photos (blue and red). Pre and post-experiment photos can be observed in Figure 3.



(a) Inicial state (b) Final state

Figure 3: Physical model

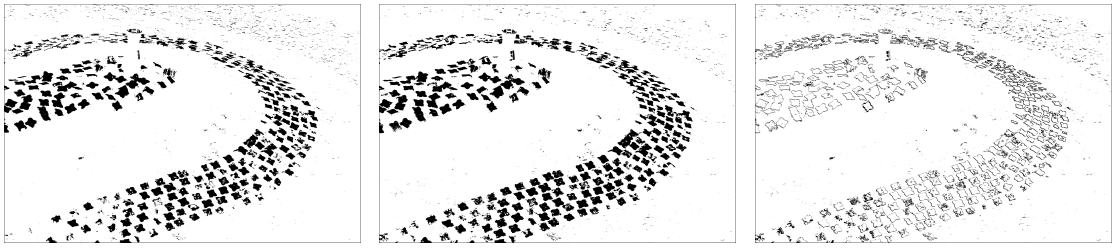
For the blue color isolation, were used the coefficients $c_1 = -0.5, c_2 = -0.5$ and $c_3 = 1.5$:



(a) Inicial state (b) Final state (c) Difference

Figure 4: Isolation test for the blue color

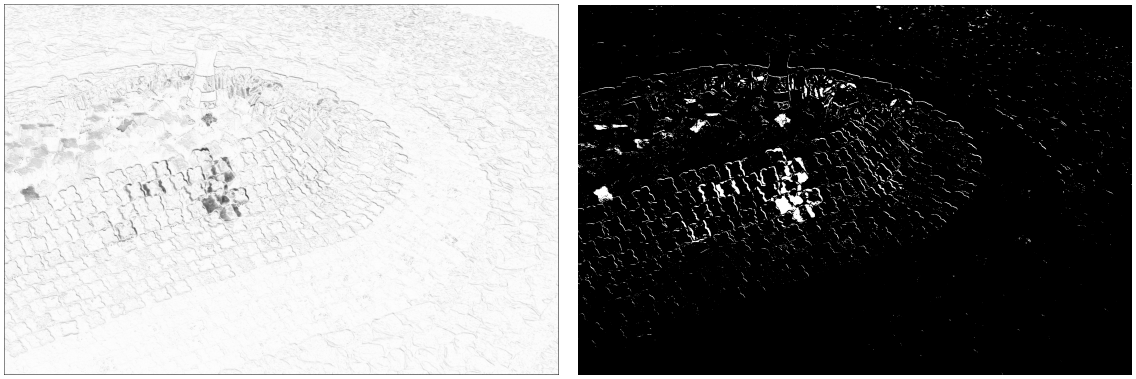
In these images it is visible the effect of isolating the blue primary color, with the inicial (Figure 4(a)) and final (Figure 4(b)) photos and the difference between them (Figure 4(c)).
Considered instead, the isolation of the red color, the coefficients are $c_1 = 1.5, c_2 = -0.5$ and $c_3 = -0.5$. With the red color isolated, another zone of the physical model of the breakwater appear, as can be observed in Figure 5.



(a) Inicial state (b) Final state (c) Difference

Figure 5: Isolation test for the red color

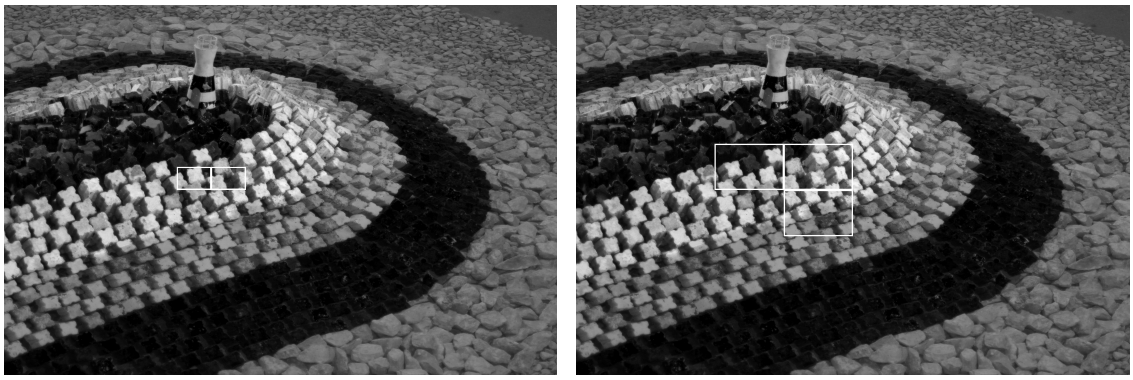
As illustrated for the blue color isolation, the difference between the initial and the final stage of the experiment is also illustrated for the red color (Figure 5). The photographs of the physical model do not contain the green color, however the coefficients for the isolation of such color would be $c_1 = -0.5, c_2 = 1.5$ and $c_3 = -0.5$.
Using the same two photos illustrated in Figure 3, techniques for the localization of the most modified zones were applied. As result of the function created, the grayscale conversion was made both for initial and final photos.The difference between both gray scale conversions is illustrated in (Figure 6(a)). This difference enabled to construct the position matrix P, as mentioned before, illustrated in Figure 6(b).



(a) Differences locations using gray scale conversions (b) Differences locations using position matrix

Figure 6: Gray scale conversion matrix vs. position matrix

Considering the images shown in the present work, the most modified zone can be determined/encountered by using the position matrix P, as shown on Figure 7 Note that it is possible to choose the number of the modified zones and the detection precision. In this example the modified area is $271,46\ cm^2$, corresponding to $2,67\ \%$ of a total modification. The execution time obtained through several tests was approximately 60 seconds, a little bit low or higher, depending on the selected searching and precison levels. The searching number is related with the most modified zone and the precision level with the number of iterations. Figure 7 presents the result of two experiments made, one with 2 searches and 4 iterations (7(a)) and the other with 3 searches and 3 iterations (7(b)).



(a) With 2 searches and a precision level 4 (b) With 3 searches and a precision level 3

Figure 7: Modified zones detection

The number and size of the blocks/submatrices considered will depend on the user, however, the aim is that the number of rectangles matches the movements and displacements of the cubes. Also, the size of the rectangles should be similar to the Antifer cubes.

4 CONCLUSIONS

This study presents an innovative method for the health monitoring and damage detection of physical scale models of breakwaters. To this purpose, one have applied mathematical techniques that allowed the assessment of changes between two photos from experiments taken at different time instants. Contrarily to what is normal in the photogrammetric analysis the used photos were not expressly obtained for this purpose and the breakwater model was not especially prepared either. Thus, no specific cares were taken to obtain these images. This fact increased the complexity of the numerical aspects involved, namely because frequent undesirable reflections due to the presence of water shouldn't affect the interpretation of the image. The algorithms developed during the present work enables to locate and successfully characterize the altered areas in the breakwater armor. No similar published work was found in the literature.

5 ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support given by FCT/MEC through Project PTDC/ATP-AQI/5355/2012 and Project LAETA - UID/EMS/50022/2013.

6 REFERENCES

[1] Robert Y. Hudson. Laboratory investigation of rubble-mound breakwaters. In *Proc. ASCE*, volume 85, pages 93–121, 1959.

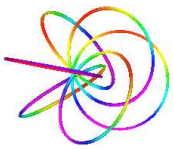
[2] J.W. Van der Meer. *Rock slopes and gravel beaches under wave attack*. PhD thesis, Delft University of Technology, 1988. Also Delft Hydraulics Publication no. 396.

[3] US Army Corps Of Engineers. Coastal engineering manual. *Engineer Manual,Washington, DC.*, 1110:2–1100, 2011. Part VI, Ch.5.

[4] Scilab Enterprises. Scilab: Free and open source software for numerical computation. `http://www.scilab.org`, 2013. Orsay, France.

[5] Steven A. Hughes. *Physical models and laboratory techniques in coastal engineering*. Singapore : World Scientific, 1993.

[6] Shigi Yu. Sivp-scilab image and video processing toolbox. `http://sivp.sourceforge.net`, 2011. accessed in 13 of March 2014.



SYMCOMP 2015
Faro, March 26-27, 2015

NUMERICAL SIMULATION OF ELECTRICAL PROBLEMS IN A VACUUM DISJUNTOR

S. Clain[1] and J. Rodrigues [2]

1: Departamento de Matemática e Aplicações
Campus de Gualtar - 4710-057 Braga
e-mail: clain@math.uminho.pt

2:Área Departamental de Matemática
Instituto Superior de Engenharia de Lisboa
e-mail: jrodri@dec.isel.pt

Keywords: Finite element methods, Domain decomposition methods, Helmotz equations, Biot-Savard

Abstract. *A vacuum circuit breaker is a device that allows the cutting of electrical power. This device consists essentially of two electrodes, one of them being mobile and is subject to a mechanical force produced by a spring, giving rise to the contact between the two electrodes. The current passing between two electrodes is determined by the extension of the contact zone. Moreover, the passage of current generated Laplace forces in areas bordering the contact, but not yet in contact. Due to the curved geometry of the electrodes, these Laplace forces are opposite and therefore cause the repulsion of the electrodes. This means that for a given power we have to evaluate the electric potential, the magnetic field corresponding to the contact zone. bbm*

1 INTRODUCTION

A vacuum circuit breaker is a device that allows the cutting of electrical power. We refer figure 1 to showing the main parts of a typical vacuum interrupter. The apparatus core is essentially constituted of two electrodes, one of them being fixed (1) and the other one is mobile (3) and subject to a mechanical force produced by a spring, maintaining the contact between the two electrodes (2). The current passing between two electrodes is determined by the extension of the contact zone and generated Laplace forces in areas bordering the contact, but not yet in contact. Due to the curved geometry of the electrodes, Laplace forces are opposite and cause the repulsion of the electrodes. When the intensity reach a critical value, the forces separate the two electrodes and the circuit is breaking. For a given intensity and a contact length, we wish evaluate the repulsive Laplace force deriving from the electric and magnetic fields.

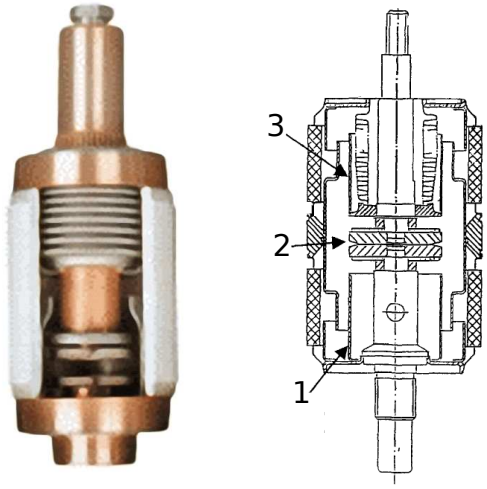


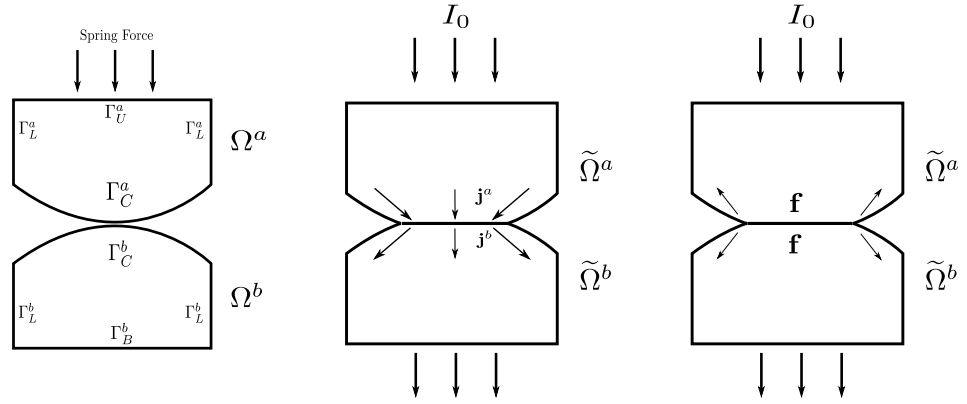
Figure 1: The circuit-breaker [1]

2 THE ELECTRICAL PROBLEM

2.1 Domain definition

At initial stage domain $\Omega = \Omega^a \cup \Omega^b$ corresponds to the initial situation where no force acts. Applying the gravity and the spring we get a displacement $\boldsymbol{u}$ over Ω and a new configuration characterized by an effective contact area $\mathcal{A}$ between the two subdomains. Due to the displacement $\boldsymbol{u}$, domain Ω is mapped into a new domain denoted $\widetilde{\Omega} = \widetilde{\Omega}(\boldsymbol{u}) = \widetilde{\Omega}^a(\boldsymbol{u}) \cup \widetilde{\Omega}^b(\boldsymbol{u})$ depending on the displacement field. In the same way, on has domains $\widetilde{\Omega}^\ell$, $\widetilde{\Gamma}^\ell_L$, $\widetilde{\Gamma}^\ell_C$, $\ell = a, b$ as well as $\widetilde{\Gamma}_B$ and $\widetilde{\Gamma}_U$. For the sake of simplicity, we shall use the

same notations $\boldsymbol{n}$, $\boldsymbol{t}$ to denote the outward normal vector and the tangential vector on the boundary. Notice that $\mathcal{A} = \widetilde{\Gamma}_C^a \cap \widetilde{\Gamma}_C^b$.
When the circuit breaker is close (the electrodes are in contact with a common area $\mathcal{A}$), the current flows across the interface governing by two main principles: conservation of the normal density current and a null potential jump across the interface, as represented at Figure 2 (a). Moreover, the electric current generate a Laplace force (as represented at Figure 2 (b)), repulsive at contact zone, which plays the fundamental role of a circuit breaker mechanism. When current increases, the Laplace force increases and the geometrical design of the apparatus results to a reduction of the contact surface till we reach a complete separation. The present chapter is dedicated to the construction of the electrical model where one computes the electric and the magnetic field in function of the contact surface.



(a) The initial configuration

(b) The normal density current jump across the interface

(c) Global defined Laplace force

Figure 2: Electric-Magnetic problem outline

2.2 Mathematical modelling

A medium voltage circuit breaker is designed to work with continuous or low frequency current (for instance 50 Hz). It results that a common approach use the low frequency approximation (see Rappaz and Touzani [2]) where we neglect the displacement current and the induction effect. Consequently, we use the standard scalar potential formulation and denote by ϕ the scalar electrical potential while $\mathbf{E} = -\nabla\phi$ stands for the electric field and $\mathbf{j} = \sigma E$ represents the current density with $\sigma > 0$ the conductivity we suppose to be constant for the sake of simplicity. When necessary, we shall use the notations ϕ^ℓ, $\mathbf{E}^\ell$, $\mathbf{j}^\ell$, $\ell = a, b$ to characterise the quantities associated to domain $\widetilde{\Omega}^\ell$ respectively. Moreover component of the global vector writes $\mathbf{j} = \left(\mathbf{j}^a, \mathbf{j}^b\right) = \left((j_1^a, j_2^a,), \left(j_1^b, j_2^b, \right)\right)$, $\mathbf{E} = \left(\mathbf{E}^a, \mathbf{E}^b\right)$

and $\phi = \left(\phi^a, \phi^b\right)$.
Assuming that no electrical charge are present in the domain, the density current conservation writes

$$\nabla \cdot \mathbf{j} = 0, \text{ in } \widetilde{\Omega} \tag{1}$$

and deduce the scalar electrical potential formulation

$$-\nabla \cdot (\sigma \nabla \phi) = 0, \text{ in } \widetilde{\Omega} \tag{2}$$

We equipped the equation with the following boundary conditions: $\phi = 0$ on the basement $\widetilde{\Gamma}_B$, a uniform distribution on the upper side

$$\mathbf{j} \cdot \boldsymbol{n} = \frac{I_0}{|\widetilde{\Gamma}_U|} \tag{3}$$

with I_0 the intensity current while we prescribe an homogeneous Neumann condition for the rest of the boundary to model that fact that no current crosses the boundary which are in contact with the vacuum.

2.3 The two domains formulation

From a practical point of view, the electrical problem will be seen as the coupling of two subproblems defined in each subdomain. Here classical Sobolev spaces are used to obtain a variational formulation. We rewrite equation (2) in the following way: find ϕ^a and ϕ^b such that

$$-\nabla \cdot (\sigma \nabla \phi^\ell) = 0, \text{ in } \widetilde{\Omega}^\ell \tag{4}$$

with $\phi^b = 0$ on the basement $\widetilde{\Gamma}_B$, $\quad \mathbf{j}^a \cdot \boldsymbol{n}^a = \dfrac{I_0}{|\widetilde{\Gamma}_U|}$ while we assume homogeneous Neumann condition condition for the vacuum boundary. To complete the new model, we prescribe continuity for the normal current and potential across the contact zone:

$$\mathbf{j}^a \cdot \boldsymbol{n}^a + \mathbf{j}^b \cdot \boldsymbol{n}^b = 0, \qquad \phi^a = \phi^b, \qquad \text{on } \mathcal{A}. \tag{5}$$

Indeed, assume that $\phi \in H^1(\widetilde{\Omega}) \cap C^0(\widetilde{\Omega})$ is a solution of the one domain problem then from the continuity we deduce $\phi^a = \phi^b$ on $\mathcal{A}$. On the other hand, let $\psi \in H_0^1(\widetilde{\Omega})$, integration by part yields

$$0 = \int_{\widetilde{\Omega}} \sigma \nabla \phi \nabla \psi \, dx = \int_{\widetilde{\Omega}^a} \sigma \nabla \phi \nabla \psi \, dx + \int_{\widetilde{\Omega}^b} \sigma \nabla \phi \nabla \psi \, dx$$

Now, integration by parts on each subdomains provide

$$0 = \int_{\widetilde{\Omega}^a} -\nabla \cdot (\sigma \nabla \phi) \psi \, dx + \int_{\widetilde{\Omega}^b} -\nabla \cdot (\sigma \nabla \phi) \psi \, dx + \int_{\mathcal{A}} \sigma (\mathbf{j}^a \cdot \boldsymbol{n}^a + \mathbf{j}^b \cdot \boldsymbol{n}^b) \psi ds.$$

Relation (4) yields that for any $\psi \in H_0^1(\widetilde{\Omega})$ we have

$$\int_{\mathcal{A}} \sigma(\mathbf{j}^a \cdot \boldsymbol{n}^a + \mathbf{j}^b \cdot \boldsymbol{n}^b)\psi ds = 0$$

which implies

$$\mathbf{j}^a \cdot \boldsymbol{n}^a + \mathbf{j}^b \cdot \boldsymbol{n}^b = 0$$

.

3 THE MAGNETICAL FIELD

With $\mathbf{E}$ in hand, we deduce the current density $\mathbf{j}$ and we aim to compute the associated electrical field to at last deduce the Laplace force. To this end, let us by $\mathbf{B}$ the magnetic induction field. For three-dimensional configuration, the Ampère-Maxwell law writes

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j}, \quad \text{in } \mathbb{R}^3 \tag{6}$$

with μ_0 the magnetic permeability in the vacuum or non-ferromagnetic material.
Assuming invariance following the z direction and that the magnetic field only depend on x and y, we deduce that the only non-vanishing component is $B(x_1.x_2) = \mathbf{B}_z(x_1, x_2)$ and the Ampère-Maxwell equation writes

$$\partial_2 B = \mu_0 \mathbf{j}_1, \qquad -\partial_1 B = \mu_0 \mathbf{j}_2, \quad \text{in } \mathbb{R}^2$$

where $\mathbf{j}$ is a given function on $\mathbb{R}^2$ with compact support.
Dealing with the rotational operator $\nabla\times$ in $\mathbb{R}^2$, we deduce that B is also the solution of problem $\mu_0 \nabla \times \mathbf{j} = \nabla \times \nabla \times B = -\Delta B$ (see [2]) with the asymptotic behaviour $B(\boldsymbol{x}) = \mathcal{O}(|\boldsymbol{x}|^{-1})$ when $|\boldsymbol{x}| \to \infty$.
Another alternative is to introduce the potential magnetic vector $\mathbf{A}$ such that $B = \nabla \times \mathbf{A}$ where $\mathbf{A}$ is the solution of problem

$$\nabla \times (\nabla \times \mathbf{A}) = -\Delta \mathbf{A} = \mu_0 \mathbf{j}, \quad \text{in } \mathbb{R}^2$$

with the asymptotic behaviour $|\mathbf{A}(\boldsymbol{x})| = \mathcal{O}\big(\ln(|\boldsymbol{x}|)\big)$.
However this relationship is not useful for magnetic field computation since the function is defined in the whole domain $\mathbb{R}^2$ while we just need to determine B on domain $\widetilde{\Omega}$. An alternative approach consists to use the integral representation, namely the Biot-Savart formula. For two-dimensional geometries $\widetilde{\Omega} \subset \mathbb{R}^2$ (see [3]). The vector potential magnetic field and the magnetic field at a point $\boldsymbol{x}$ are given by

$$\mathbf{A}(\boldsymbol{x}) = -\frac{\mu_0}{2\pi} \int_{\widetilde{\Omega}} \mathbf{j}(\boldsymbol{y}) \ln(|\boldsymbol{x} - \boldsymbol{y}|) d\boldsymbol{y} \tag{7}$$

$$B(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \times \mathbf{A} = -\frac{\mu_0}{2\pi} \int_{\widetilde{\Omega}} \mathbf{j}(\boldsymbol{y}) \times \nabla_{\boldsymbol{x}} \ln(|\boldsymbol{x} - \boldsymbol{y}|) d\boldsymbol{y}, \tag{8}$$

for a current $\mathbf{j}$ flowing in the direction of $\mathbf{e}_1$ and $\mathbf{e}_2$ where $\times$ represents the external product between two vectors. After some algebraic manipulation, equation (8) writes

$$B(\boldsymbol{x}) = \frac{\mu_0}{2\pi}\int_{\widetilde{\Omega}} \frac{\det[\mathbf{j}(\boldsymbol{y}),(\boldsymbol{x}-\boldsymbol{y})]}{|\boldsymbol{x}-\boldsymbol{y}|^2} d\boldsymbol{y}. \tag{9}$$

At least, the Laplace force is given by

$$\mathbf{f} = \mathbf{j} \times \mathbf{B},$$

and in our specific case with $\mathbf{B} = B\mathbf{e}_3$, the force writes

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = B \begin{pmatrix} j_2 \\ -j_1 \end{pmatrix}. \tag{10}$$

4 MATRICIAL REPRESENTATION

4.1 Representation for $A^a_{e_h}$

Using finite element methods we will introduce the matricial representation for this problem. We identify the new meshes of $\widetilde{\Omega}^\ell$ by $\widetilde{\mathcal{T}}^\ell_h$, $\ell = a, b$, respectively, and To enforce the Dirichlet condition we use a penalisation method which seems more adapted. Indeed, the contact zone may change with respect to the elasticity problem hence to avoid a new codification of the boundary and to reshape the matrix, we always use the same stiff matrix and introduce the Dirichlet condition by multiplying the entries corresponding to the nodes of $\mathcal{A}^a_{\eta,h}$.

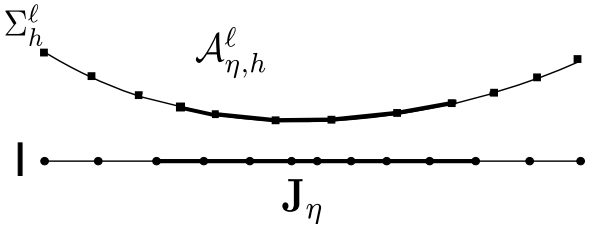


Figure 3: Discrete active zone definition

The rigid matrix writes

$$[A^a_e] = \left[A^a_{e_h}\left(\phi^a_i, \phi^a_j\right)\right]_{i,j=1,\ldots,n^a}$$

while the associated write-hand side is given by

$$[\Theta^a]_i = \begin{cases} 0 & \Leftarrow \quad i \in \{1,\ldots,\tilde{n}^a\} \\ \int_{\widetilde{\Gamma}_{U_h}} \frac{I_0}{|\widetilde{\Gamma}_{U_h}|}\varphi^a_i \, ds, & \Leftarrow \quad i \in \{\tilde{n}^a+1,\ldots,n^a\} \end{cases}$$

Let denote by $\mathcal{E}^a_{\eta,h}$ the nodes which correspond to domain $\mathcal{A}^a_{\eta_h}$. Once we compute the system matrices at each step, the Dirichlet condition correspondent to $\mathcal{A}^a_{\eta_h}$ can be imposed by substitution method.
Resolution of the elliptic problem turns to solve the simple matricial problem

$$[A^a_e]\,[\Phi^a] = [\Theta^a]$$

where $[\Phi^a]$ are the unknowns on the nodes.

4.2 Representation for $A^b_{e_h}$

Since the Dirichlet condition does not change with the iteration, we do not use a penalization method for operator $A^b_{e_h}$ and recall that P_i, $i = 1, \cdots, \tilde{n}^b$, correspond to the nodes of $\widetilde{\Omega}^b$ except the node of boundary $\widetilde{\Gamma}_B$. The rigid matrix then writes

$$\left[A^b_e\right] \;=\; \left[A^b_{e_h}\left(\phi^b_i, \phi^b_j\right)\right]_{i,j=1,\ldots,\tilde{n}^b}$$

Let ρ^b_h be a given constant piecewise function on $\widetilde{\Gamma}^a_{C_h}$ characterized by vector $[\rho] = \left[\rho_1 \cdots \omega_{s^b_C}\right]^T$. We introduce the Neumann conditions with vector

$$\left[\Theta^b\left([\rho]\right)\right]_i = \sum_{T \subset \widetilde{\Gamma}^b_{C_h}} \int_T \rho^b_h \phi^b_i \, ds, \Leftarrow i \in \left\{1, \ldots, n^b_C\right\}.$$

The elliptic problem consist in solving the matricial problem

$$\left[A^b_e\right]\left[\widetilde{\Phi}^b\right] = \left[\Theta^b\right]$$

where $\left[\widetilde{\Phi}^b\right]$ are the unknowns on the nodes. We complete the vector setting $\left[\Phi^b\right] = \left[\left[\widetilde{\Phi}^b\right], 0\right]$ taking into account the homogeneous boundary condition.

4.3 Current density and normal projection

Setting $\mathbf{j}^\ell_h = \nabla\phi^\ell_h \in X^\ell_h$, we obtain a constant piecewise vector over $\widetilde{\Omega}^\ell_h$ which represents the current density field. We shall represent the vector in two vectors depending on the coordinates,

$$\left[\mathbf{j}^\ell_1\right] \;=\; \left[\;\mathbf{j}^\ell_{1,1} \cdots \mathbf{j}^\ell_{n^\ell_K,1}\;\right]^T \;\text{ and }\; \left[\mathbf{j}^\ell_2\right] \;=\; \left[\;\mathbf{j}^a_{1,2} \cdots \mathbf{j}^a_{n^\ell_K,2}\;\right]^T.$$

that we gather in the matrix form

$$\left[\;\mathbf{j}^\ell\;\right] \;=\; \left[\;\left[\mathbf{j}^\ell_1\right] \quad \left[\mathbf{j}^\ell_2\right]\;\right]^T.$$

We report here the definition of the matricial expression of the projection following the normal direction. We denote by $\left[\overline{N}^{\ell}\right] \in \mathbb{R}^{n_C^{\ell} \times 2n_C^{\ell}}$ the matrix of the outwards normals over Γ_C^{ℓ} and set

$$\left[\overline{N}^{\ell}\right] = \begin{bmatrix} \boldsymbol{n}_{h\,1,1}^{\ell} & \cdots & 0 & \boldsymbol{n}_{h\,1,2}^{\ell} & \cdots & 0 \\ & \ddots & & & \ddots & \\ 0 & \cdots & \boldsymbol{n}_{h\;n_C^{\ell},1}^{\ell} & 0 & \cdots & \boldsymbol{n}_{h\;n_C^{\ell},2}^{\ell} \end{bmatrix}$$

We introduce the global matricial representation of $\left[\overline{N}^{\ell}\right]$ on Ω_h^{ℓ}:

$$\left[N^{\ell}\right] = \left[\begin{array}{cc} \left[\overline{N}^{\ell}\right] & \mathbf{0} \end{array}\right]$$

with $\left[N^a\right] \in \mathbb{R}^{n_C^{\ell} \times (\tilde{n}^a + n^a)}$ and $\left[N^b\right] \in \mathbb{R}^{n_C^{\ell} \times 2\tilde{n}^b}$.

4.4 Representation for the mappings on contact zone

We recall the matricial representation $\left[C^{\ell,I}\right] \in \mathbb{R}^{n^I \times n_C^{\ell}}$ and $\left[C^{I,\ell}\right] = \left[C^{\ell,I}\right]^T \in \mathbb{R}^{n_C^{\ell} \times n^I}$ for operators $C_{h,\eta}^{\ell,I}$ and $C_{\eta,h}^{I,\ell}$ respectively

$$\left[C^{\ell,I}\right]_{ki} = \int_I \mu_k(\xi)\phi_i^{\ell}(\xi)d\xi, \quad k = 1, \cdots, n^I,\ i = 1, \cdots, n_C^{\ell}.$$

In the same way, we represent operators $D_{h,\eta}^{\ell,I}$ and $D_{\eta,h}^{I,\ell}$ with $\left[D^{\ell,I}\right] \in \mathbb{R}^{s^I \times s_C^{\ell}}$ and $\left[D^{I,\ell}\right] = \left[D^{\ell,I}\right]^T \in \mathbb{R}^{s_C^{\ell} \times s^I}$ with

$$\left[D^{\ell,I}\right]_{ki} = \int_I \lambda_k(\xi)\vartheta_i^{\ell}(\xi)d\xi, \quad k = 1, \cdots, s^I,\ i = 1, \cdots, s_C^{\ell}.$$

4.5 The iterative problem within the matricial form

We now give the iterative procedure at the matricial level. Notice that the procedure corresponds to the one one really implemented on computer thus the importance to define completely all the step. The iterator index is r and we shall compute a sequence of vectors $[\nu]^r$ which shall converge.
Assume that vector $[\nu]^r$ is known such that $[\nu]_k^r = 0$ for the nodes N_k outside of $\mathbf{J}_{\eta}$. the procedure is given by the following substeps (we omit subscript k for the sake of simplicity):

1. Compute vector $[w^a] = \left[D^{I,\ell}\right][\nu]^r$
2. Compute $[\Phi^a]$ solving problem

$$[A_e^a][\Phi^a] = [\Theta^a]$$

with penalization with respect to $[\omega^a]$.

3. Compute $[\mathbf{j}^a]$ with $\nabla\phi_h^a$ and compute $[\rho^a] = [N^a][\mathbf{j}^a]$.
4. Compute $[\tau] = \left[D^{\ell,I}\right][\rho^a]$ and $[\rho^b] = -\left[D^{\ell,I}\right][\tau]$
5. Compute $\left[\Phi^b\right] = \left[\left[\widetilde{\Phi}^b\right], 0\right]$ solving

$$\left[A_e^b\right]\left[\widetilde{\Phi}^b\right] = \left[\Theta^b\right]$$

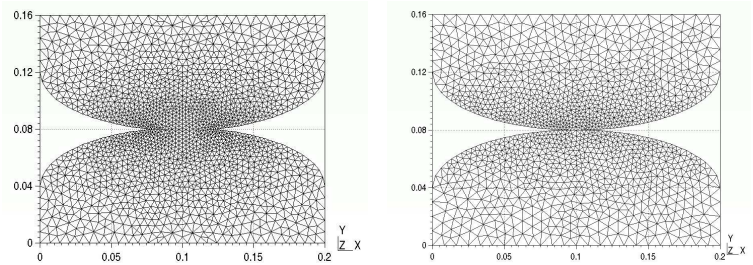
6. Extract $[\omega^b]$ from $\left[\Phi^b\right]$ and compute $[\tilde{\nu}] = \left[C^{\ell,I}\right][\omega^b]$ where we cancel the entries k which correspond to the nodes N_k outside of $\mathbf{J}_\eta$.
7. Compute the new vector $[\nu]^{r+1} = \theta[\nu]^r + (1-\theta)[\tilde{\nu}]$

We repeat the algorithm till we satisfy the convergence criterion.

5 NUMERICAL SIMULATION

5.1 One domain case

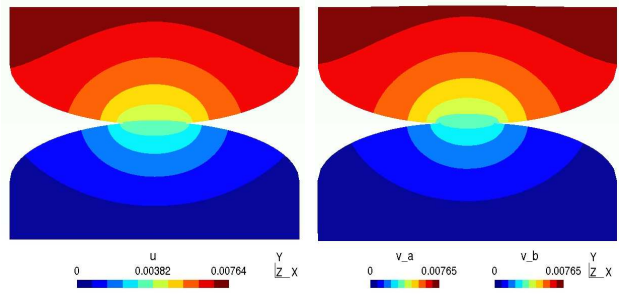
We begin by considering the case with one domain. Once the length of $\mathbf{J}_\eta$ is determined we solve the discrete electric problem on the domain have the configuration of the two electrodes in contact, Figure 4. We this domain we do not have to worry about the potential continuity at the contact zone. More, these results will give us a benchmark for the later results obtained with the domain decomposition technique and two domains.
For this case we consider $m(\mathbf{J}_\eta) = 0.0468$ and $I_0 = 10kA$.



(a) One domain mesh (b) Two domain meshes

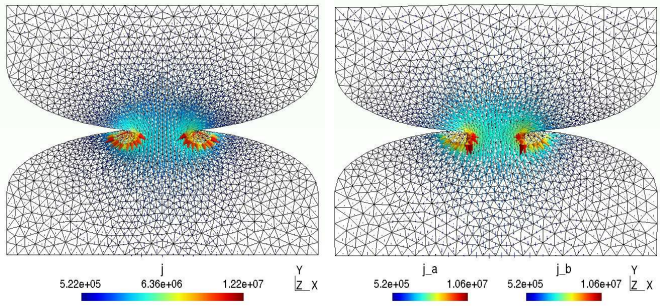
Figure 4: Domain mesh

Figure 4 shows that with both strategies we obtain very similar results. This validates the results and in particular our technique for passing information between the domains at domain decomposition method.
At Figures 6 and 7 although the range of values are similar, the observed difference is justified by the mesh difference, since the density is approximated numerically in the



(a) One domain (b) Two domains

Figure 5: Potential scalar field



(a) One domain (b) Two domains

Figure 6: Current density field

barycentric coordinates in the element from the given potential at the nodes. Also in these figures we can see the continuity of density at the the contact zone.
The magnetic induction (component following $\mathbf{e}_3$) is exactly the same Figure 8. About Laplace force there is a reduction though the magnitude is equal and we have almost the symmetry between the two domains Figure 9, where we recall again that the meshes are not symmetrical and that the Laplace force are calculated is approximated numerically in the barycentric coordinates in the element.
At this section we will analyse the numerical solutions produced at electro-magnetic state when the electrodes are in contact through a determined active zone.
For each numerical problem we determine the volume repulsive force generated for a given $m(\mathbf{J})$ and intensity current I_0, for two profiles cases: elliptic and circular.

5.2 Two domains case

We are interested to analysing the profiles of the potential contact zone: elliptic and circular. From [4] we use two intensity values $I = 10$ kA , $I = 20$ kA, $I = 40$ kA and

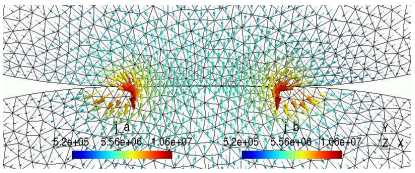
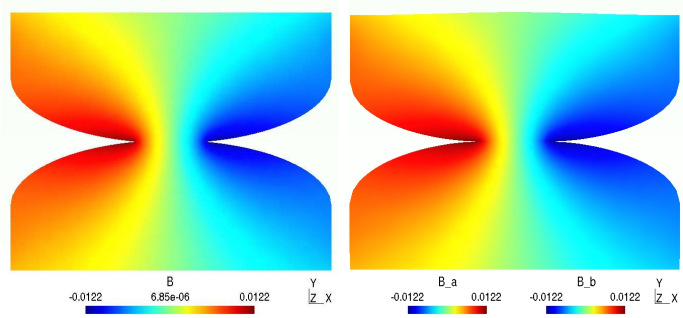
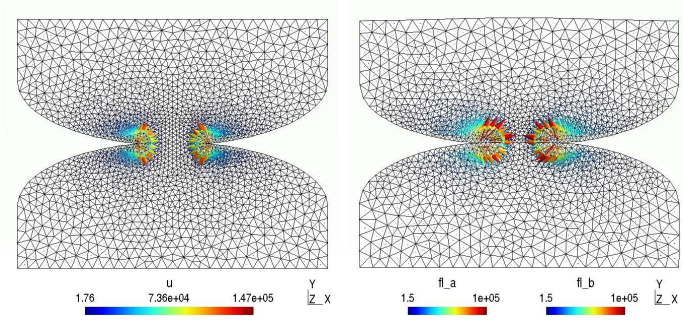


Figure 7: Zoom of current density field at contact zone



(a) One domain (b) Two domains

Figure 8: Magnetic induction (component following $\mathbf{e}_3$)



(a) One domain (b) Two domains

Figure 9: Laplace force

$I = 60$ kA. We consider that a inicial spring force has applied, characterized by

$$F = \kappa \times \alpha \ (N) \tag{11}$$

with

$$\kappa = \frac{E \times A}{L} \tag{12}$$

named the axial stiffness, where
$E = 115$ GPa $= 115 \times 10^6 \mathrm{N/m}^2$, the copper elasticity modulus, [5],

A, the contact zone area (in m^2) and
L the circuit breaker height (in m).
The value α represents the vertical displacement, so here we will suppose $\alpha \leq 0.2$

5.3 Elliptic profile

Here we consider a contact with a elliptic profile with a contact zone length $m(\mathbf{J}_\eta) = 0.032$, corresponding to $\alpha = 0.1$.

Table 1: Repulsive force generated with an elliptic profile and large active zone and test for several values of θ

intensity current (A)	θ	repulsive force generated (N)
1×10^4	.125	44.53
1×10^4	.25	44.53
1×10^4	.5	44.53
2×10^4	.125	178.13
2×10^4	.25	178.13
2×10^4	.5	178.13
4×10^4	.125	712.53
4×10^4	.25	712.53
4×10^4	.5	712.53
6×10^4	.125	1603.2
6×10^4	.25	1603.2
6×10^4	.5	1603.2

From the results explained at Table 1, we can expect a function of the kind

$$f_R(I_0) = a_1 I_0^2. \tag{13}$$

Using least squares method we obtain

$$a_1 = 44.53.$$

represented at Figure 10.

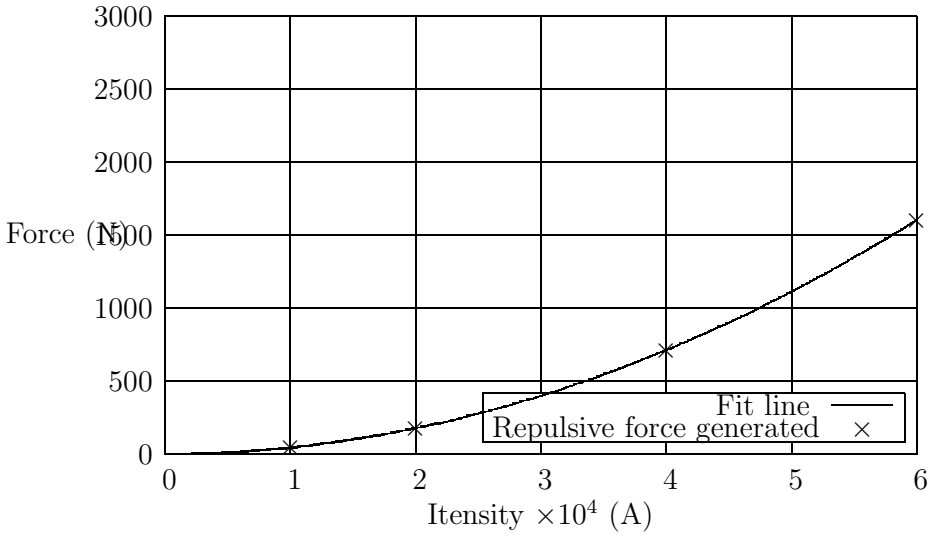


Figure 10: Elliptic profile and large active zone: Repulsive force generated "×" and least squares approximation "line"

Now we test a reduction of the contact zone, $m(\mathbf{J}_\eta) = 0.012$, corresponding to $\alpha = 0.01$.

Table 2: Repulsive force generated with an elliptic profile and small active zone

intensity current (A)	repulsive force generated (N)
1×10^4	51.1
2×10^4	204.42
4×10^4	817.69
6×10^4	1839.82

From the results explained at Table 2, we can expect a function of the kind

$$f_R(I_0) = a_2 I_0^2. \tag{14}$$

Using least squares method we obtain

$$a_2 = 51.1.$$

represented at Figure 11.

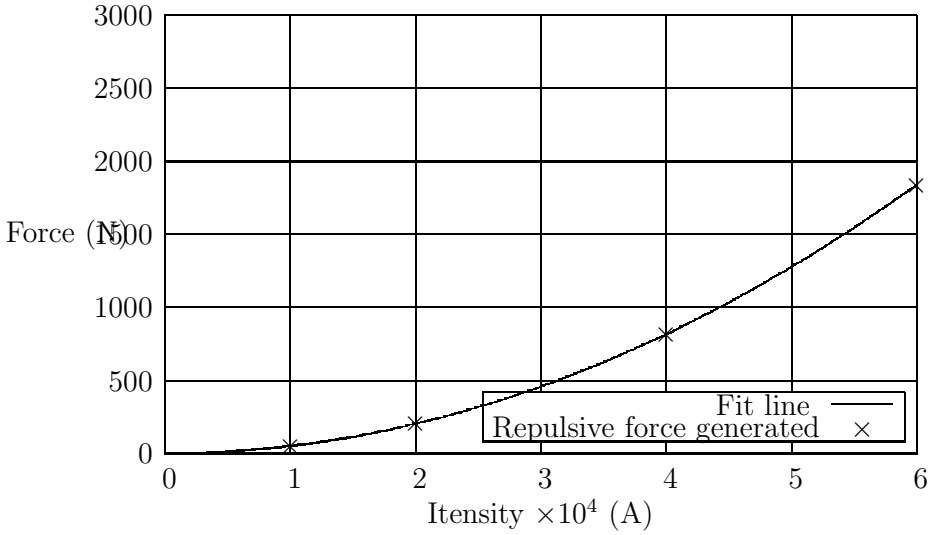


Figure 11: Elliptic profile and small active zone: Repulsive force generated "×" and least squares approximation "line"

6 Circular profile

Here we begin by consider a circular contact profile after deformation with$m(\mathbf{J}_\eta) = 0.045$, corresponding to $\alpha = 0.1$.
From the results explained at Table 3, we can expect a function of the kind

$$f_R(I_0) = a_3 I_0^2. \tag{15}$$

Using least squares method we obtain

$$a_3 = 186.71.$$

represented at Figure 12.
Like as above Consider now a small contact zone, $m(\mathbf{J}_\eta) = 0.016$, corresponding to $\alpha = 0.01$.
From the results explained at Table 4, we can expect a function of the kind

$$f_R(I_0) = a_4 I_0^2. \tag{16}$$

Table 3: Repulsive force generated with an circular profile and large active zone

intensity current (A)	repulsive force generated (N)
1×10^4	186.71
2×10^4	746.85
4×10^4	2987.42
6×10^4	6721.69

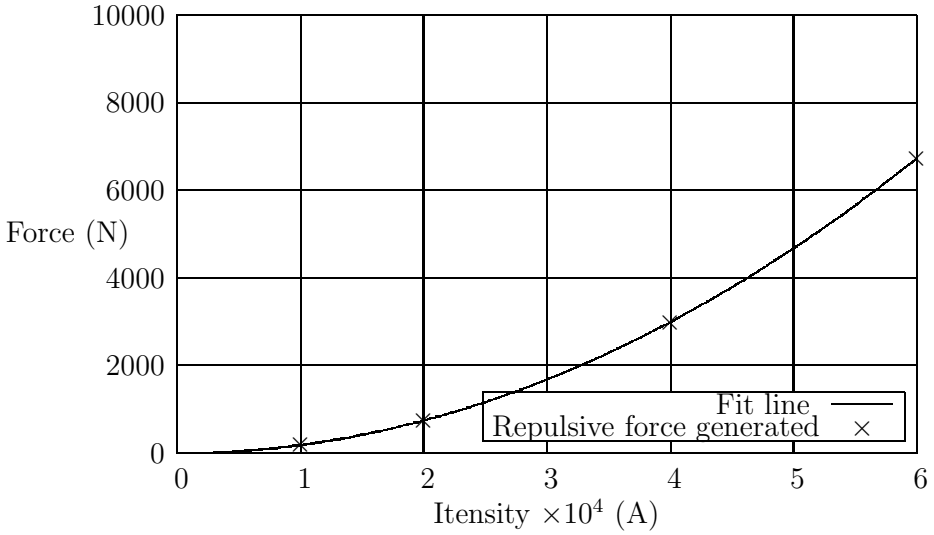


Figure 12: Circular profile and large active zone: Repulsive force generated "×" and least squares approximation "line"

Using least squares method we obtain

$$a_4 = 200.35.$$

represented at Figure 13.
Observing figures we note that the repulsive force depends on the electrode profile being more important in the case of a circular profile. It also appears that the repulsive force increases as the extent of the active area decreases.

Table 4: Repulsive force generated with an circular profile and small active zone

intensity current (A)	repulsive force generated (N)
1×10^4	200.34
2×10^4	801.39
4×10^4	3205.59
6×10^4	7212.59

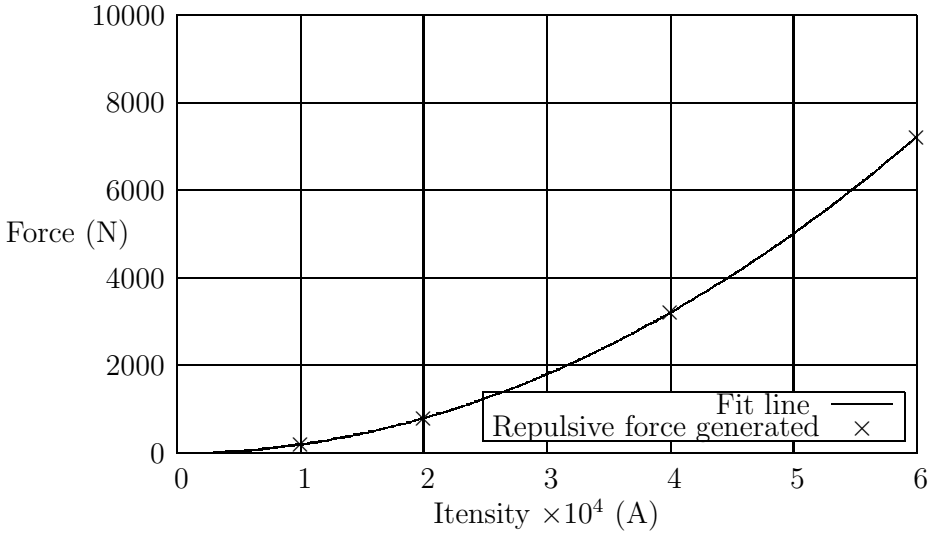


Figure 13: Circular profile and small active zone: Repulsive force generated "×" and least squares approximation "line"

7 CONCLUSION

From the results we have obtained can conclude that the proposed domain decomposition algorithm converges to a continuous solution of the scalar electrical potential field, independently of parameter θ.
Comparing figures 10 - 12 and 11 - 13 we note that the repulsive force depends on the electrode profile and being more important in the case of a circular profile.
With this tests, we can also observe that the repulsive force is inversely proportional to the length of the active zone.
Only two-dimensional configurations have been considered in the present work but the

real problem is three-dimensional and brings more difficulties. The contact zone is more complex and the computational effort is larger.

8 REFERENCES

[1] R. Garzon. *High Voltage Circuit Breakers: Design and Applications*. Pitman Advanced Publishing Program, Boston, 1985.

[2] J. Rappaz R. Touzani. *Mathematical Models for Eddy Currents and Magnetostatics With Selected Applications*. Springer - Scientific Computation, New-York, 2014.

[3] J.-C. Suh. The evaluation of the biot-savart integral. *Journal of Engineering Mathematics*, pages 375–395, 2000.

[4] A. Slama. *Modlisation des sources de courant en mouvement et des efforts lectrodynamiques dans les appareils de coupure*. PhD gnie electrique, Institut National Polytechnique de Grenoble - France, 2001.

[5] American Society for Metals Handbooks. Metals handbook, January 2012.

SYMCOMP 2015
Faro, March 26-27, 2015

SIZING OPTIMIZATION OF CONCRETE CABLE-STAYED BRIDGES

Alberto M. B. Martins, Luís M. C. Simões and João H. J. O. Negrão

Department of Civil Engineering
Faculty of Sciences and Technology
University of Coimbra – Pólo II
3030-788 Coimbra, Portugal
e-mail: alberto@dec.uc.pt, web: http://www.uc.pt/fctuc/dec
e-mail: {lcsimoes, jhnegrao}@dec.uc.pt

Keywords: Concrete cable-stayed bridges, cross-sectional design variables, time-dependent effects, construction stages

Abstract *Cable-stayed bridges are highly redundant structures in which the deck behaves like a continuous beam elastically supported by the inclined stays. They represent an aesthetically appealing and efficient structural solution for medium-to-long spans and are widely used all over the world. Their behaviour is governed by the stiffness of the load-bearing elements (towers, deck and cable stays) and the cable force distribution. The design of cable-stayed bridges involves a significant amount of design variables and design objectives which can be treated efficiently by using optimization algorithms.*

In concrete bridges the stresses and deformations are significantly influenced by the construction sequence and by the concrete time-dependent effects. Furthermore, the geometrical nonlinear behaviour that arises when dealing with cables, large and flexible structures should also be considered in the analysis.

In this paper the concrete cable-stayed bridge design is formulated as a multi-objective optimization problem with objectives of minimum cost, minimum deflections and stresses. The cable-stays areas and prestressing forces, the deck and towers cross-sections are considered as design variables. A numerical method is developed to obtain the optimum design of such structures. This numerical method includes a structural analysis module and a sensitivity analysis and optimization module. The structural analysis accounts for all the relevant effects (concrete time-dependent effects, construction stages and geometrical nonlinear effects). The structural response to changes in the design variables is done by a discrete direct sensitivity analysis procedure and an entropy-based approach was used for structural optimization. The features and applicability of the proposed method are demonstrated by a numerical example concerning the optimization of a real sized concrete cable-stayed bridge.

1. INTRODUCTION

Cable-stayed bridges are highly redundant structures in which the deck behaves like a continuous beam elastically supported by the inclined stays. They represent an aesthetically appealing and efficient structural solution for medium-to-long spans and are widely used all over the world. Their behaviour is governed by the stiffness of the load-bearing elements (towers, deck and cable stays) and the cable force distribution [1].
Cable-stayed bridge optimum design is an iterative procedure defining overall geometry, calculation of the cable forces distribution and finding of the members cross-sections to satisfy displacement and stress criteria during the erection stages and for the complete bridge. The design of a cable-stayed bridge is a laborious task due to the high redundancy, the large amount of design variables and design objectives that must be dealt with. Optimization techniques are not commonly used in civil engineering practice. However, in the design of large and complex structures like cable-stayed bridges, the use of optimization techniques naturally arises as an efficient way to deal with the large amount of information in view of reduction of material cost and thus obtaining economical and structurally efficient solutions.
The previous research works concerning the use of optimization algorithms in the design of cable-stayed bridges dealt mainly with the problem of cable tensioning in steel bridges [2], [3], [4], [5]. The cable forces optimization on concrete cable stayed-bridges was also reported [6], [7], [8], [9].
The optimization of steel and composite bridges using geometric and cross-sectional design variables and aiming the cost minimization was carried out in previous research works [10], [11], [12], [13]. The optimum design of steel bridges considering three-dimensional modelling, the seismic action and including box-girder deck sections was also reported [14], [15], [16].
Recently it has been reported the results concerning the optimization of cable forces [17], cable forces and cross-sectional areas of stay cables [18] and the optimum design of composite steel-concrete cable-stayed bridges using a genetic-algorithm based optimization [19].
Concerning the optimization of concrete cable-stayed bridges, the cable forces were computed considering the time-dependent effects but not the construction stages [6], [7], [8]. The unit load method proposed by Janjic et al. [9] accounts for the construction stages, the time-dependent and geometrical nonlinear effects, but allows the calculation of the cable forces only, not considering cross-sectional or geometrical design variables. Furthermore, the structural response to changes in the cable forces is evaluated by using the concept of influence matrix which requires a number of structural analyses equal to the number of cable stays.
The optimum design of a concrete cable-stayed bridge using a genetic-algorithm based optimization was presented in [20]. Cross-sectional and geometrical design variables were used, however the cable forces, which play a fundamental role in the behaviour of this type of structure, were not considered. Moreover, the construction sequence and the time-dependent

effects were neglected.
From our knowledge, the optimum design aiming the cost minimization of concrete cable-stayed bridges using the cable forces and sizing design variables has not been yet reported. Therefore, it is worth developing a numerical method for the optimum design of these structures that takes into account all the relevant effects, namely, the construction stages, time-dependent effects and geometrical nonlinear effects. The current work on the optimization of concrete cable-stayed bridges started with cable force calculation [21].
In this paper the optimum design of a concrete cable-stayed bridge is formulated as a multi-objective optimization problem with objectives of minimum cost, minimum deflections and stresses and a Pareto solution is sought. An entropy-based approach is used to find the minimax solution through the minimization of a convex scalar function. The cost minimization is formulated as the minimization of material volume. The design variables considered are the cable-stays areas and prestressing forces, deck and towers cross-sections. The structural response to changes in the design variables is assessed by a discrete direct sensitivity analysis procedure that requires one structural analysis in each iteration.
The innovation of the current study is the consideration of the construction stages, the time-dependent and the geometrical nonlinear effects in the optimum design of concrete cable-stayed bridges. The developed numerical method allows the calculation of the cable installation forces, cable adjustment forces and the members cross-sectional dimensions that satisfy the displacements and stresses throughout the structure during erection and for the complete bridge.
A computer program developed in MATLAB environment was used to perform the structural analysis, sensitivity analysis and optimization. The structural analysis was done by means of a finite element model that includes the load history and geometry changes due to the construction sequence, the geometrical nonlinear effects and the time-dependent effects of concrete. Numerical examples are presented to illustrate the features of the described procedure.

2. MATERIAL MODELLING AND STRUCTURAL ANALYSIS

The stresses and deformations in concrete cable-stayed bridges are significantly influenced by the construction process and the concrete rheological behaviour [22], [23], [24], [25]. Moreover, the geometrical nonlinear behaviour that arises when dealing with cables, large and flexible structures should also be considered in the analysis [26], [27], [28], [29].

2.1. Modelling of the time-dependent effects

In this study, the time-dependent effects of creep, shrinkage and aging of concrete are evaluated according with Eurocode 2 formulations [30]. The creep model is based on linear viscoelasticity and takes into account ageing effects. Shrinkage strains are time-dependent but stress independent. Detailed considerations concerning the modelling of the time-dependent effects are presented in a recent research work by the authors [21].
Concrete strength and modulus of elasticity increase with time as result of curing. At an early age, the strength and modulus of elasticity increase quickly and the increase then gradually

stagnates but does not stop completely. According to Eurocode 2 [30] the concrete modulus of elasticity at an age, *t*, in days is given by

$$E_{cm}(t)=\left(\exp\left\{s\left[1-\left(\frac{28}{t}\right)^{1/2}\right]\right\}\right)^{0.3} E_{cm} \tag{1}$$

where E_{cm} is the mean modulus of elasticity of concrete at an age of 28 days, t is the age of the concrete in days and s is a coefficient depending on the cement type.
The total strain at time t of a concrete specimen, uniaxially loaded with a stress σ_c at time t_0, can be written as the sum of the stress dependent, $\varepsilon_{c\sigma}(t,t_0)$, and stress independent, $\varepsilon_{cn}(t)$, strains:

$$\varepsilon_c(t)=\varepsilon_{c\sigma}(t,t_0)+\varepsilon_{cn}(t)=J(t,t_0)\cdot\sigma_c(t_0)+\varepsilon_{cn}(t) \tag{2}$$

where $J(t,t_0)$ is the creep function and if the stresses are less than 45% of the characteristic value of concrete compressive strength (f_{ck}), the principle of superposition is valid and the creep strain varies linearly with the applied stress.
In a cable-stayed bridge the stresses continually change during both the construction phase and the service life of the structure. Under variable stresses and using the principle of superposition, Equation 2 can be rewritten as:

$$\varepsilon_c(t)=J(t,t_0)\cdot\sigma_c(t_0)+\int_{t_0}^{t} J(t,\tau)\frac{\partial\sigma_c(\tau)}{\partial\tau}d\tau+\varepsilon_{cn}(t) \tag{3}$$

Several approaches have been proposed to solve this equation, simplified methods, step-by-step numerical integration and approximation of the creep function. In this paper the creep function is approximated by a Dirichlet series [31] leading to:

$$J(t,t_0)\cong\frac{1}{E_c(t_0)}+\frac{1}{E_{cm}}\sum_{i=1}^{n}a_j(t_0)\left(1-e^{-\alpha_j(t-t_0)}\right) \tag{4}$$

where n is the number of terms of the Dirichlet series and the coefficients a_j are obtained from a curve fitting using the least squares method. The coefficients $1/\alpha_j$ are called retardation times and are chosen to cover the range of time values for the creep coefficients calculation.
According to Eurocode 2 [30] the total shrinkage strain at an age t, $\varepsilon_{cs}(t)$, is the sum of the autogenous (ε_{ca}) and the drying shrinkage (ε_{cd}). The drying shrinkage at an age t is defined as

$$\varepsilon_{cd}(t)=\beta_{ds}(t,t_s)\cdot k_h\cdot\varepsilon_{cd,0} \tag{5}$$

where $\beta_{ds}(t,t_s)$ and k_h are coefficients depending on the member notional size and the age of concrete at the beginning of the drying shrinkage. $\varepsilon_{cd}(t)$ is a parameter that depends upon the environmental relative humidity, the cement type and the concrete compressive strength.

The autogenous shrinkage develops due to chemical reactions during hardening in the early age of concrete and it can be expressed at an age *t* by

$$\varepsilon_{ca}(t) = \beta_{as}(t) \cdot \varepsilon_{ca}(\infty) \tag{6}$$

where $\varepsilon_{ca}(\infty)$ is the long-term value of the autogenous shrinkage strain and $\beta_{as}(t)$ is a function that gives the evolution of the autogenous shrinkage with time.

In the structural analysis the time-dependent effects were simulated by equivalent nodal forces that produce the same displacement field as the time-dependent effects and from which is calculated the actual deformation state. These forces are calculated, for each time interval, as initial deformations using the finite element formulation and the corresponding values of the creep and shrinkage strains computed according to the formulation previously presented. The stresses are then computed using only the elastic constitutive relationship between stresses and mechanical origin deformations.

2.2. Geometrical nonlinear effects

There are three main sources of geometrical nonlinearity in cable-stayed bridges: the nonlinear axial force-elongation relationship for the inclined cable stays due to the sag caused by their own weight; the nonlinear axial force and bending moment-deformation relationships for the towers and the deck under combined bending and axial forces; and the geometry change caused by large displacements [26], [27], [28], [29].

The geometrical nonlinear effects were considered by means of a second-order elastic analysis. The cable stays were modelled as truss elements with stiffness matrix computed using the equivalent modulus of elasticity, i.e., Ernst formula [32] which can describe the catenary action of a cable. The value of the cable equivalent modulus of elasticity is given by

$$E_{eq} = \frac{E}{1 + \dfrac{(\gamma \cdot L \cos\alpha)^2 E}{12\sigma^3}} \tag{7}$$

where E_{eq} is the equivalent cable modulus of elasticity, E is the effective cable material modulus of elasticity, γ is the specific weight of the cable material, L is the length of the chord, α is the angle between the cable chord and the horizontal direction and σ is the tension stress in the cable.

The second-order effects due to changes in structure geometry were considered by using the equivalent lateral force method, usually used for the second-order analysis of buildings [33]. This method deals only with the global structural instability or P-Δ effect. Member instability or the P-δ effect is ignored. It is an iterative method that uses a set of transversal loads to simulate the effect of structure instability.

2.3. Structural analysis including erection stages

The modelling and analysis of the erection stages was performed using a forward analysis procedure following the erection sequence. The forward analysis allows the knowledge of the

stresses and displacements throughout the construction, enabling the direct consideration of the time dependent effects. It is suitable for the purpose of structural optimization, because all the information concerning stresses, displacements and their sensitivities is simultaneously available at the end of each stage and prior to calling the optimization module.
The erection is performed considering the balanced cantilever method, the most common and widely used method of cable-stayed bridge construction. The construction starts by building the towers and the start of the cantilevers at each side of the towers. In subsequent stages the other deck segments and stay cables are erected until the deck's closure. In each stage, one deck segment and a stay cable are installed. Each deck segment is installed tangentially to the existing one at which it connects. At the installation time the cable contributes only with the prestressing force at the anchorages. In following stages it is considered like a new member of the structure contributing to the overall stiffness matrix. The cable stays installation forces are computed using the optimization algorithm described in section 3.
The structural analysis was performed using a finite element-based computer program and the bridge was modelled as a two-dimensional framed structure. The tower and deck were modelled as 2-node/6 degrees of freedom Euler-Bernoulli beam elements and the stays were modelled using two-node truss elements with an equivalent modulus of elasticity, according to Ernst formulation. To account for the geometrically nonlinear effects the structural analysis was carried out in an iterative manner. Figure 1 presents a flow chart of the computer program developed.

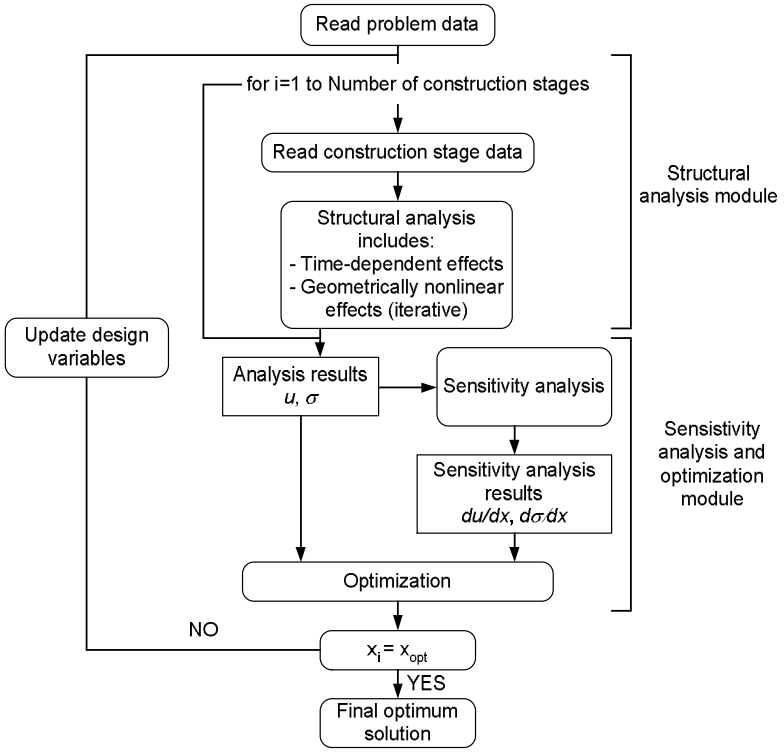


Figure 1. Flow chart of the developed computer program

3. OPTIMUM DESIGN FORMULATION

As previously referred, in this paper, the optimum design of concrete cable-stayed bridges is posed as a multi-objective optimization problem. This involves the definition of the design variables, the design objectives and the objective function.

3.1. Design variables

The design variables considered were the cable-stays areas, prestressing forces and the cross-sectional dimensions of the deck and towers. The design variables are represented by x_i, and the global design variable vector is

$$x = \{x_1, x_2, x_3, \ldots, x_N\}^T \tag{8}$$

The stay cables were made of 0.6 inch diameter strands (15.7 mm nominal diameter and 1.5cm^2 of cross-sectional area). Rectangular hollow sections and single cell box cross-section were considered, respectively, for the towers and deck cross-sections. The parameters that define the geometry of the referred cross-sections and that will be considered as sizing design variables (Table 2) are presented in Figure 2.

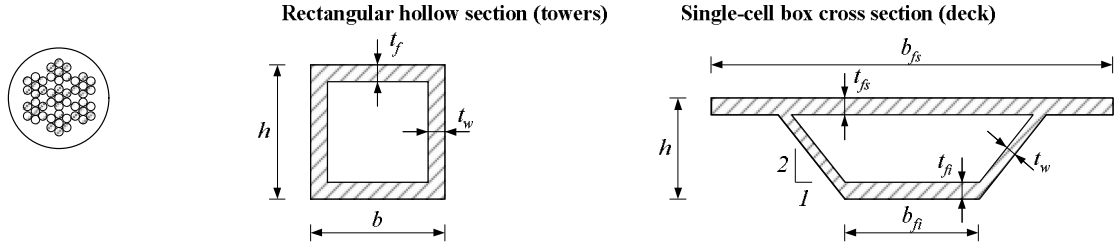


Figure 2. Cable-stays, tower and deck cross-sections

The cross-sectional design variables of deck and towers have direct impact in weight (or cost) reduction. Cable areas and cable forces play an essential role on the stress distribution throughout the structure because they define the extent of the beam-like behaviour of the deck. Furthermore, they are fundamental for adjusting bridge geometry and deflection control, which otherwise could only be achieved by a severe stiffening of the deck, in opposition to the expected reduction of material. Therefore, its use as design variables is essential in any optimization procedure.

3.2. Design objectives

The design of cable-stayed bridges involves achieving some design objectives in order to check the service and strength criteria. The objectives should be cast in a normalized form. They arise from imposing limits in the displacements and stresses during the construction stages and for the complete bridge under permanent and live loads. Moreover, the design should seek to minimize the cost of the structure. Considering this, the first goal can be expressed as

$$g_1(x) = \frac{C}{C_0} - 1 \leq 0 \tag{9}$$

where C is the current cost of the structure and C_0 is a reference cost, which corresponds to the initial cost of each analysis and optimization cycle. This ensures that in each cycle the cost is always one of the leading objectives for the optimization algorithm. The cost of the structure was formulated as the cost of the materials (concrete, reinforcing and prestressing steel). The materials unit prices were obtained from consulting Portuguese supplier companies.

The second set of objectives comes up from limiting the deck vertical displacements and the towers horizontal displacements to achieve the desired final deck profile and to minimize the tower bending deflections

$$g_2(x) = \frac{|\delta|}{\delta_0} - 1 \leq 0 \tag{10}$$

where δ and δ_0 are, respectively, the displacement value and the limit value for the displacement under control.

A third set of goals arises from imposing limits for the deck and towers stresses during the construction stages and for the complete bridge under permanent load. These goals are related to the service conditions. According to Eurocode 2 [30] the concrete compressive stresses were limited to 45% of the characteristic value of the concrete compressive strength (f_{ck}), so the concrete remains within the range of linear creep and also the longitudinal cracking is prevented. The concrete tensile stresses were limited to the 5% fractile of the characteristic axial tensile strength of concrete ($f_{ctk,0.05}$) to avoid cracking and, thus, ensuring durability.

$$g_4(x) = \frac{\sigma_c}{0.45 f_{ck}} - 1 \leq 0 \tag{11}$$

$$g_3(x) = \frac{\sigma_c}{f_{ctk,0.05}} - 1 \leq 0 \tag{12}$$

where σ_c is the acting stress in the concrete members. The tensile and compressive stresses and the correspondent allowable stresses are used considering the respective signals.

The concrete members should also be checked for the maximum stresses. This goal can be expressed as

$$g_5(x) = \frac{\sigma_c}{\sigma_{allow}} - 1 \leq 0 \tag{13}$$

where σ_c is the acting stress in the concrete members and σ_{allow} is the corresponding allowable stress in tension or compression. The acting stresses are calculated from the acting axial force (N_{Ed}) and bending moment (M_{Ed}). The allowable value is computed as a stress equivalent to the combined axial force-bending moment design resistance (N_{Rd}; M_{Rd}) of the

cross-section. This value is obtained from the non-dimensional interaction diagram generated for the respective cross-section of the concrete member. For these calculations the reinforcing steel area was not a design variable and was considered 2% of the concrete cross-sectional area. This was adopted as mean value of reinforcing steel area for the concrete members and represents a common practical value.
Figure 3 presents a scheme of the non-dimensional interaction diagrams used.

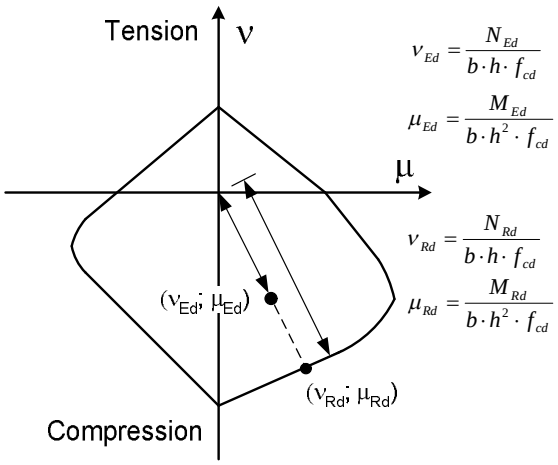


Figure 3. Non-dimensional interaction diagram for a concrete member

The remaining objectives concerning the stresses in the stays are

$$g_6(x) = \frac{\sigma}{k \cdot f_{pk}} - 1 \leq 0 \tag{14}$$

$$g_7(x) = 1 - \frac{\sigma}{0.1 f_{pk}} \leq 0 \tag{15}$$

where σ and f_{pk} are the acting stress in the stays and the characteristic value of the prestressing steel tensile strength, respectively. If the acting stress is greater than $0.1 f_{pk}$, the Equation 14 applies, if the acting stress is less than or equal to $0.1 f_{pk}$, the Equation 15 applies. The lower limit of $0.1 f_{pk}$ for the tensile stress in the stays was considered to ensure their structural efficiency. According to Eurocode 3-1-11 [34] the value of *k* in Equation 14 is equal to 0.55 during erection, 0.50 for service conditions and 0.74 for strength verification.

3.3. Objective function

The design of concrete cable-stayed bridges is formulated as a multi-objective optimization problem from which a Pareto optimal solution vector is obtained. This means that no other feasible vector exists that could decrease one objective function without increasing at least another one. The optimum vector usually exists in practical problems and is not unique. The

objective of the multi-objective optimization problem is to minimize the set of all objectives over the design variables. This is achieved by the minimax optimization problem

$$Min_x Max_j \left(g_1, g_2, ..., g_j \right) \tag{16}$$

This problem is discontinuous and non-differentiable and thus difficult to solve. However, it may be shown [35] that using the Shannon/Jaynes maximum entropy principle and Cauchy's arithmetic-geometric mean inequality, the solution of the minimax optimization problem with objectives defined by Equations 9 to 15 can be found indirectly by minimizing the unconstrained convex scalar function, that turns out to be the Kreisselmeier-Steinhauser function

$$\min F(x) = \min \frac{1}{\rho} \ln \left[\sum_{j=1}^{M} e^{\rho \left(g_j(x) \right)} \right] \tag{17}$$

which is both continuous and differentiable and thus considerably easy to solve. This function depends only on one control parameter, ρ, which must not be decreased to ensure that a multiobjective solution is found. In this work a constant value for ρ of 10 was used. The optimization of the convex scalar function (Equation 17) may be solved by conventional quasi-Newton methods, with which an optimal solution (in the Pareto sense) is achieved for each starting trial design. The goal functions $g_j(x)$ do not have an explicit algebraic form and are only obtained numerically from the structural analysis results of a particular design variable vector. The strategy adopted was to solve Equation 17 by means of an iterative sequence of explicit approximation models. An explicit approximation can be formulated by taking Taylor series expansions of all the goal functions $g_j(x)$ truncated after the linear term. This gives:

$$\min F(x) = \min \frac{1}{\rho} \ln \left[\sum_{j=1}^{M} e^{\rho \left(g_{0j}(x) + \sum_{i=1}^{N} \frac{dg_{0j}(x)}{dx_i} \Delta x_i \right)} \right] \tag{18}$$

where N and M are, respectively, the number of design variables and the number of objectives. $g_{0j}(x)$ and $dg_{0j}(x)/dx_i$ are the objectives and their sensitivities evaluated for the current design variable vector (x_0), at which the Taylor series expansion is made. Solving Equation 18 for particular numerical values of $g_{0j}(x)$ forms only one iteration of the problem's complete solution. The solution vector (x_1) of such iteration represents a new design that must be analysed and gives new values for $g_{1j}(x)$, $dg_{1j}(x)/dx_i$ and (x_1), to replace those corresponding to (x_0) in Equation 18. Iterations continue until changes in the objective function become small. Upper and lower bound constraints of 2% of the current values of the design variables were imposed as move limits to ensure the accuracy of the explicit approximation. The optimization process was carried out using the MATLAB function *fmincon*, which minimizes a scalar function with various variables subjected to

bound constraints using a sequence of quadratic problems.

3.4. Sensitivity analysis

The sensitivity analysis module allows knowing the way a variation in each design variable will affect the design objectives and the objective function of the optimization problem. Given the availability of the source code, the discrete nature of cable-stayed bridge structures and the large number of objectives (stresses and displacements) under control, the analytical discrete direct method was used for the sake of sensitivity analysis. The sensitivities of displacements are obtained by differentiating the equilibrium equations

$$K \cdot u = F \tag{19}$$

from which the following expression is obtained:

$$K \frac{du}{dx_i} = \frac{dF}{dx_i} - \frac{dK}{dx_i} u = Q_{vi} \tag{20}$$

where Q_{vi} is the virtual pseudo-load vector of the system with respect to the *ith* design variable. The displacement sensitivities can be expressed as:

$$\frac{du}{dx_i} = K^{-1} \cdot Q_{vi} \tag{21}$$

which requires storing the stiffness matrix, pre-programming stiffness matrix and right-hand side derivatives so the displacement derivatives may be computed by the solution of *N* pseudo-load right hand sides.
The stress sensitivities are determined from the chain derivation of the finite element stress-displacement relation

$$\sigma = D \cdot B^e \cdot u^e \tag{22}$$

$$\frac{d\sigma}{dx_i} = \frac{d\left(D \cdot B^e\right)}{dx_i} \cdot u^e + D \cdot B^e \cdot \frac{du^e}{dx_i} \tag{23}$$

The first term of the right-hand side may be directly computed during the computation of element contribution for the global system, on the condition that derivative expressions are pre-programmed and called on that stage. Since the displacement derivatives are known, the second term on the right-hand side is easily computed.

4. NUMERICAL EXAMPLE

4.1. Description of the numerical model

To illustrate the features of the proposed numerical method, it was applied to analyze a cable-stayed bridge structure. The numerical model comprises a symmetrical concrete cable-stayed bridge with a total span of 320 m and a central span of 166 m. The towers total height

is of 52 m with the deck placed 15 m above the foundation. Figure 4 shows the geometry of the bridge example.

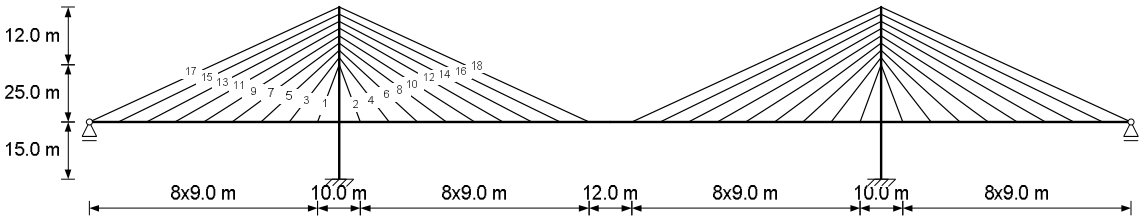


Figure 4. Bridge geometry and cable stay numbering

As previously referred, the bridge's deck and towers were modelled with 2 node Euler-Bernoulli beam elements. The stays were modelled with 2 node truss elements with an equivalent modulus of elasticity given by Ernst formulae. By using structure symmetry only half of the structure was modelled. Table 1 presents the properties of the materials used in the numerical models.

Concrete C35/45 (deck and towers)	**Prestressing steel – Cable-stays**
E = 34 GPa; γ = 25 kN/m^3 f_{ck} = 35 MPa; $f_{ct0.05k}$ = 2.2 MPa f_{cd} = 23.3 MPa Mean Relative Humidity = 80% Type N cement Cost: 135 €/m^3	E = 195 GPa; γ = 77 kN/m^3 f_{pk} = 1860 MPa; $f_{p0,1k}$ = 1770 MPa Cost: 1200 €/ton
	Reinforcing steel – A500 NR
	E = 200 GPa; f_{yk} = 500 MPa; f_{yd} = 435 MPa $\varepsilon_{yk} = 2.5\times10^{-3}$; $\varepsilon_{yd} = 2.174\times10^{-3}$ Cost: 900 €/ton

Table 1. Material properties

The construction stages using the balanced cantilever method are represented in Figure 5. For the complete bridge, the deck-to-pylon connection is only vertical but, to ensure structure stability, in the erection stages this connection was considered fixed.

The actions considered during the construction stages were the self-weight of the deck and the construction loads of 1.0 kN/m^2 due to personnel and hand tool, 0.5kN/m^2 due to non-permanent equipment and a concentrated load of 400 kN on the cantilevers edge [36]. In each construction stage, one deck segment and a stay cable are added, symmetrically in each side of the towers. In Stage 10 a simple support is added at the end of the lateral span and in Stage 11 a connection restraining the bending moment and the axial force is added at mid-span to simulate the bridge closure.

The cable forces were computed in two steps (installation forces and adjusting forces) using the optimization method described in section 3. The installation forces ensure that in Stage 10 the deck presents the desired geometry (for adding the support in the lateral span and to perform the bridge closure at the main span) and the stresses throughout the structure during the construction stages remain within the allowable limits. In Stage 11 the cable forces were adjusted to achieve the desired deck profile at the end of construction for the bridge under self-weight and an additional permanent load of 2.5 kN/m^2 (flooring, walkways, safety

barriers and guardrails).

After stage 11, the bridge in the permanent state and without time-dependent effects is analysed for three load cases and design objectives of maximum stresses are considered. The load cases consist of the permanent load plus a live load of 4 kN/m^2 (road traffic) being applied on the whole deck or only on central or side spans.

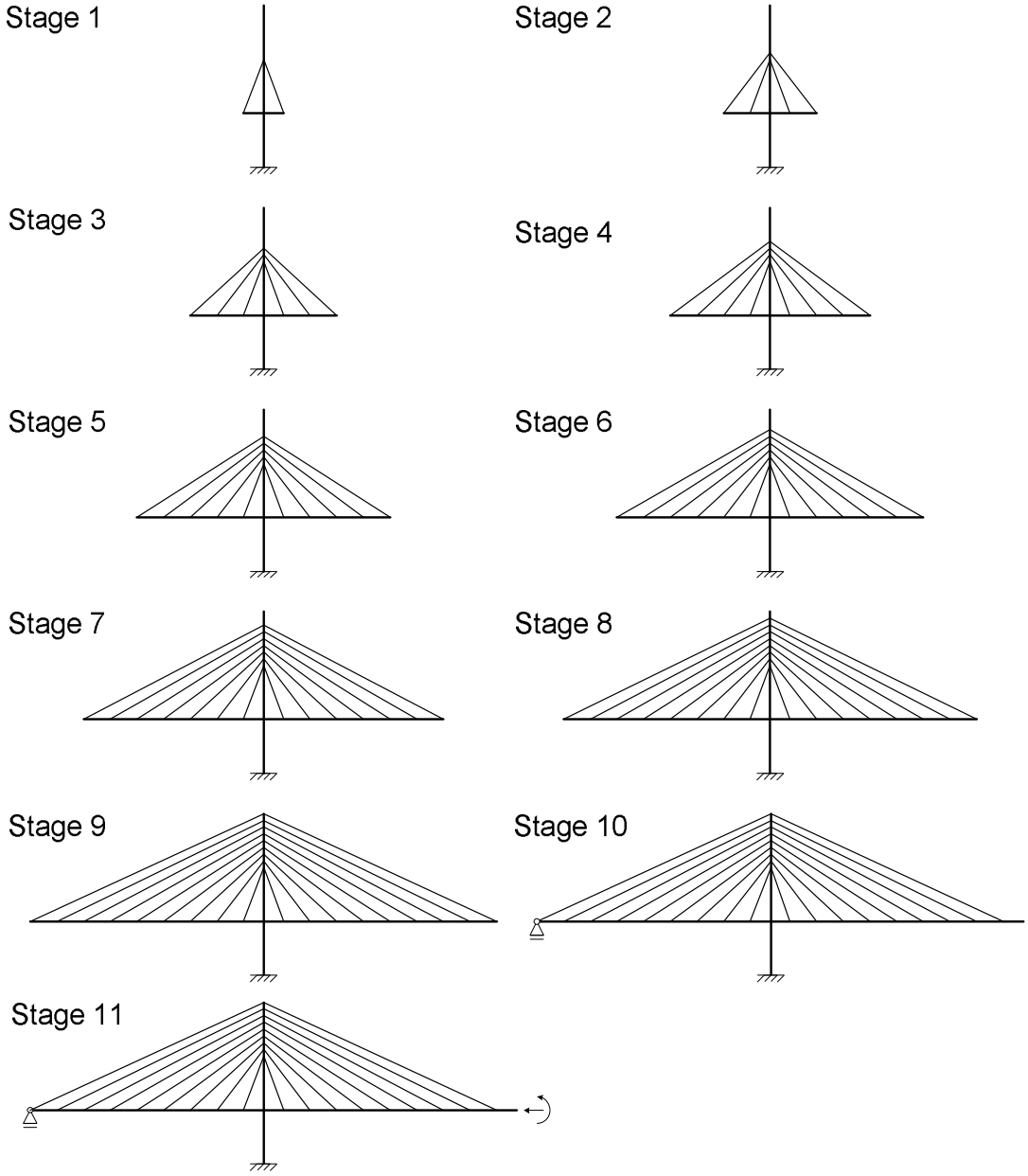


Figure 5. Construction stages

4.2. Design variables

For the numerical example analysed the overall geometry of the bridge and the deck width (19.0 m) were pre-assigned constant design parameters. Figure 6 presents the zones considered for the definition of the sizing design variables. A total of 74 design variables were considered and are described in Table 2.

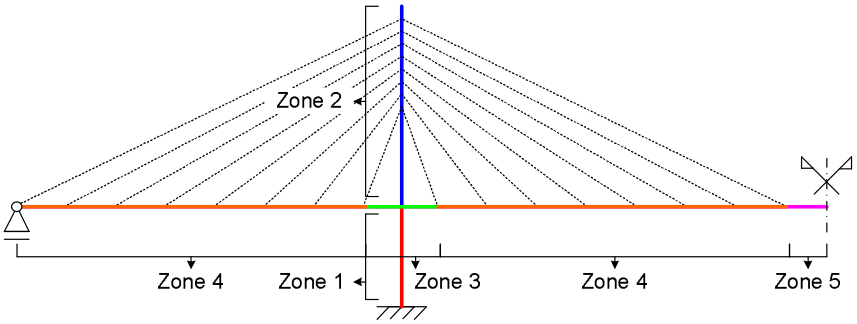


Figure 6. Zones for the cross-sectional design variables

Number	Design variable	Number	Design variable
1 & 2	Installation force – stays 1 & 2	33 & 34	Installation force – stays 13 & 14
3 & 4	Cable area – stays 1 & 2	35 & 36	Cable area – stays 13 & 14
5	h deck cross-section Zone 3	37 & 38	Installation force – stays 15 & 16
6	t_w deck cross-section Zone 3	39 & 40	Cable area – stays 15 & 16
7	t_{fs} deck cross-section Zone 3	41 & 42	Installation force – stays 17 & 18
8	t_{fi} deck cross-section Zone 3	43 & 44	Cable area – stays 17 & 18
9 & 10	Installation force – stays 3 & 4	45	h deck cross-section Zone 5
11 & 12	Cable area – stays 3 & 4	46	t_w deck cross-section Zone 5
13	h deck cross-section Zone 4	47	t_{fs} deck cross-section Zone 5
14	t_w deck cross-section Zone 4	48	t_{fi} deck cross-section Zone 5
15	t_{fs} deck cross-section Zone 4	49 to 66	Adjustment forces – stays 1 to 18
16	t_{fi} deck cross-section Zone 4	67	h towers cross-sections Zone 1
17 & 18	Installation force – stays 5 & 6	68	h towers cross-sections Zone 2
19 & 20	Cable area – stays 5 & 6	69	b towers cross-sections Zone 1
21 & 22	Installation force – stays 7 & 8	70	b towers cross-sections Zone 2
23 & 24	Cable area – stays 7 & 8	71	t_w towers cross-sections Zone 1
25 & 26	Installation force – stays 9 & 10	72	t_w towers cross-sections Zone 2
27 & 28	Cable area – stays 9 & 10	73	t_f towers cross-sections Zone 1
29 & 30	Installation force – stays 11 & 12	74	t_f towers cross-sections Zone 2
31 & 32	Cable area – stays 11 & 12		

Table 2. Design variables

4.3. Optimum design results

The initial values and the results obtained after optimization are presented in Tables 3 and 4, for the cable forces and the sizing design variables, respectively.

Cable-stay	Cable installation forces [kN]		Cable adjustment forces [kN]	
	Initial	Final	Initial	Final
1	1000	1013	1400	1397
2	1000	976	1400	1401
3	1500	1551	2200	2216
4	1500	1485	2200	2217
5	1500	1563	2400	2417
6	1500	1565	2400	2451
7	1500	1623	2600	2659
8	1500	1715	2600	2699
9	2000	2240	2900	3042
10	2000	2394	2900	3048
11	2000	2408	3000	3261
12	2000	2464	3000	3181
13	2000	2596	3100	3507
14	2000	2490	3100	3282
15	2000	2663	3300	3851
16	2000	2462	3300	3475
17	2000	2706	3800	4504
18	2000	2446	3500	3660

Table 3. Cable forces (initial and final values)

The results presented in Table 3 show that the cable forces increase from the tower to the mid-span. The forces are similar for symmetrically disposed stays. As desired for an adequate structural behaviour the highest forces occur in the backstays. These forces counterbalance the largest load in the central span when compared with the side span and control the tower bending deflections and stresses.

For the example analysed the optimization leads to a cost reduction of 0.93%. Despite this small cost reduction, the structural behaviour is significantly improved, as can be stated by the results presented in Figures 8 and 9.

In this example, the deck represents the largest contribution to the bridge cost with a value of 36% of the total cost. The cable stays represent 31% and the towers represent 34% of the bridge total cost.

Figure 7 shows the evolution of the objective function and of the bridge cost during the analysis-and-optimization process. It can be noticed the efficiency of the numerical method seeking the minimization of the objective function. The solution is obtained in a small number of iterations.

Design variable	Initial	Final	Design variable	Initial	Final
3	1.050×10^{-2}	7.938×10^{-3}	32	1.050×10^{-2}	1.191×10^{-2}
4	1.050×10^{-2}	1.058×10^{-2}	35	1.050×10^{-2}	1.191×10^{-2}
5	3.000	3.553	36	1.050×10^{-2}	1.191×10^{-2}
6	0.200	0.229	39	1.050×10^{-2}	1.192×10^{-2}
7	0.200	0.229	40	1.050×10^{-2}	1.250×10^{-2}
8	0.200	0.229	43	1.050×10^{-2}	1.132×10^{-2}
11	1.050×10^{-2}	1.088×10^{-2}	44	1.050×10^{-2}	1.088×10^{-2}
12	1.050×10^{-2}	9.114×10^{-3}	45	2.000	1.530
13	2.500	3.102	46	0.200	0.153
14	0.200	0.150	47	0.200	0.153
15	0.200	0.151	48	0.200	0.153
16	0.200	0.168	67	7.000	8.023
19	1.050×10^{-2}	1.132×10^{-2}	68	7.000	7.481
20	1.050×10^{-2}	8.820×10^{-3}	69	7.000	8.289
23	1.050×10^{-2}	1.132×10^{-2}	70	7.000	7.352
24	1.050×10^{-2}	1.058×10^{-2}	71	0.500	0.546
27	1.050×10^{-2}	1.132×10^{-2}	72	0.500	0.531
28	1.050×10^{-2}	1.088×10^{-2}	73	0.500	0.577
31	1.050×10^{-2}	1.191×10^{-2}	74	0.500	0.529

Table 4. Sizing design variables (initial and final values)

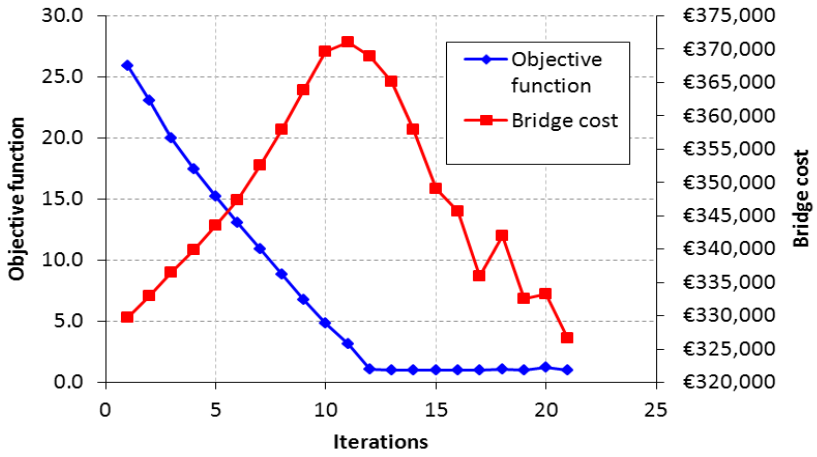


Figure 7. Objective function and bridge cost *vs* number of iterations

Some results obtained for the complete bridge under dead load (Stage 11) are presented in Figures 8 and 9. From Figure 8 it can seen the the deck vertical displacements are small (less than 5cm) when compared with the deck length, which ensures an almost horizontal deck profile for the complete bridge under permanent load. As stated before, the optimization leads to a significant improvement in the structural behaviour. The initial solution leads to

excessive values of the deck vertical displacements, which are modified during the optimization process to ensure the desired deck profile for the complete bridge under permanent load.

The results presented in Figure 9 show that the deck normal stresses at the end of construction remain within the allowable limits imposed for service conditions. It is also worth noting that the deck is almost completely under compression and the stresses are within a narrow range.

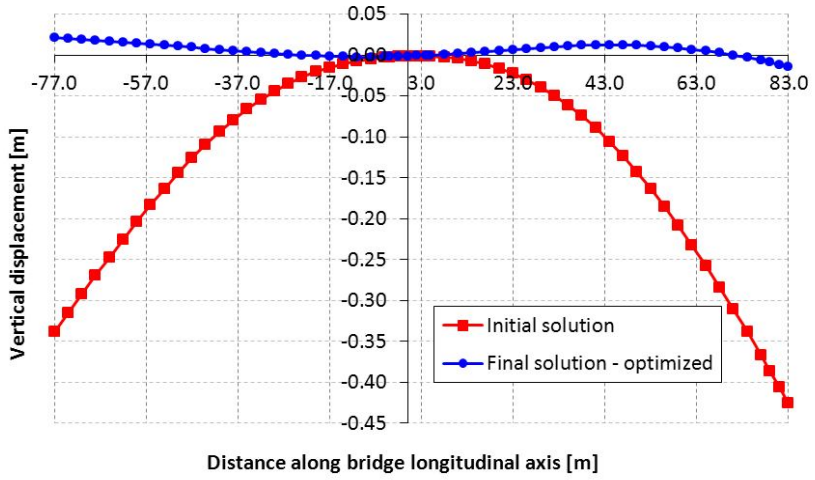


Figure 8. Deck vertical displacements (initial and final solution)

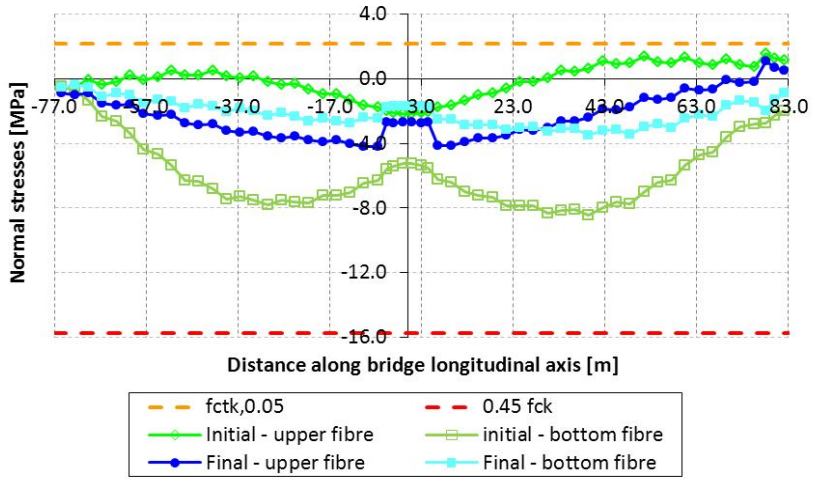


Figure 9. Deck normal stresses (initial and final solution)

An optimized solution is obtained by rearranging the cable forces distribution and the cross-sectional dimensions of the load-bearing elements. Besides the cost reduction, this solution ensures the desired deck profile for the complete bridge under dead load and that the stresses

during erection and for the complete bridge (under dead and traffic live loads) are within the allowable limits.

5. CONCLUSIONS

In this paper, the optimum design of concrete cable-stayed bridges is formulated as a multi-objective optimization problem with objectives of minimum cost, minimum deflections and stresses. According to the results obtained the following conclusions can be drawn:

- It is possible to formulate and solve the design problem of a concrete cable-stayed bridge as a multi-objective optimization problem considering the cable-prestressing forces and the cross-sectional dimensions of the cables, towers and deck as design variables and including the influence of the construction stages, the time-dependent effects and the geometrical nonlinearities.
- Cable forces, cable areas, deck and tower sizing used as design variables allows the cost minimization and finding structural efficient solutions giving adequate values of displacements and stresses throughout the structure.
- Although requiring some programming investment, the discrete direct method of sensitivity analysis, which only requires one structural analysis in each iteration, proved to be efficient in predicting the bridge structural behaviour.
- Even dealing with a large number of objectives and design variables, the optimization algorithm used in this work is efficient since the solution is obtained in a small number of iterations.
- The design algorithm is also robust because it takes into account all relevant actions (loads, time-dependent effects and geometrical nonlinearities) and allows checking the service and strength criteria for the erection stages and for the complete bridge.
- The solutions obtained are Pareto optima and the results depend on the starting point. Several starting points are used to achieve satisfactory solutions from the engineering point of view. A good starting point is the solution considering the complete bridge under permanent load.
- The numerical method enables the calculation of the cable installation forces as well as the adjustment forces to achieve the desired deck profile at closure.
- For the bridge analysed, it can be stated that the deck represents 36% of the bridge cost. The cable stays represent 31% and the towers represent 34% of the bridge total cost.

REFERENCES

[1] Walther, R., Houriet, B., Isler, W., Moia, P. and Klein, J. F., *Cable-stayed bridges*, Second edition, Thomas Telford Publishing, 1999.

[2] Chen Qin, "Optimization of cable-stretching planning in the construction of cable-stayed bridges", *Engineering Optimization* 19: 1-20, 1992.

[3] Negrão, J. H. J. O., and Simões, L. M. C., "Cable stretching force optimization in cable-stayed bridges", Paper presented at the *WCSMO-2 The Second World Congress of Structural and Multidisciplinary Optimization*, IFTR: 983-987, 1997.

[4] Sung Y. C., Chang, D. W. and Teo, E. H., "Optimum post-tensioning cable forces of Mau-Lo Hsi cable-stayed bridge", *Engineering Structures* 28: 1407–1417, 2006.
[5] Baldomir, A., Hernandez, S., Nieto, F. and Jurado, J. A, "Cable optimization of a long span cable stayed bridge in La Coruña", *Advances in Engineering Software* 41: 931–938, 2010.
[6] Furukawa, K., Sugimoto, H., Egusa, T., Inoue, K., and Yamada, Y., "Studies on the optimization of cable prestressing for cable-stayed bridges." Paper presented at the *International Conference on Cable-Stayed Bridges*, Bangkok, Thailand, 1987.
[7] Furukawa, K., Sakai, I., Kumagai, S., Arai, H. and Kasuga, A., "Optimization of cable forces in cable-stayed prestressed concrete bridges based on minimum strain energy criterion." Paper presented at the *International Conference on Cable-Stayed Bridges*, Bangkok, Thailand, 1987.
[8] Kasuga A, Arai, H., Breen, J. E. and Furukawa, K., "Optimum cable-force adjustments in concrete cable-stayed bridges", *Journal Structural Engineering ASCE* 121(4): 685-694, 1995.
[9] Janjic, D., Pircher, M. and Pircher, H., "Optimization of cable tensioning in cable-stayed bridges", *Journal of Bridge Engineering* 8 (3): 131-127, 2003.
[10] Torii, K. and Ikeda, K., "A study of the optimum design method for cable-stayed bridges." Paper presented at the *International Conference on Cable-Stayed Bridges, Bangkok*, Thailand, 1987.
[11] Ohkubo, S. and Taniwaki, K., "Shape and sizing optimization of steel cable-stayed bridges." In Proceedings of *OPTI 91 – Optimization of Structural Systems and Industrial Applications*, edited by S. Hernandez and C. A. Brebbia, Elsevier Applied Sciences, Cambridge, MA, USA, 1991.
[12] Simões, L. M. C. and Negrão J. H. O., "Sizing and geometry optimization of cable-stayed bridges", *Computers & Structures* 52 (2): 309-321, 1994.
[13] Long, W., Troitsky, M. S. and Zielinski, Z. A., "Optimum design of cable stayed bridges", *Journal of Structural Engineering and Mechanics* 7 (3): 241-257, 1999.
[14] Negrão, J. H. O. and Simões, L. M. C., "Optimization of cable-stayed bridges with three-dimensional modelling", *Computers & Structures* 64 (1-4): 741-758, 1997.
[15] Simões L. M. C. and Negrão, J. H. J. O., "Optimization of cable-stayed bridges subjected to earthquakes with non-linear behaviour", *Engineering Optimization* 31: 457-478, 1999.
[16] Simões, L. M. C., and J. H. J. O. Negrão. 2000. "Optimization of cable-stayed bridges with box-girder decks", *Advances in Engineering Software* 31: 417-423.
[17] Hassan, M. M., Nassef, A. O. and El Damatty, A. A., "Determination of optimum post-tensioning cable forces of cable-stayed bridges", *Engineering Structures* 44: 248–259, 2012.
[18] Hassan, M. M., "Optimization of stay cables in cable-stayed bridges using finite element, genetic algorithm, and B-spline combined technique", *Engineering Structures* 49: 643–654, 2013.
[19] Hassan M. M., Nassef, A. O. and El Damatty, A. A., "Optimal design of semi-fan cable-stayed bridges", *Canadian Journal of Civil Engineering* 40: 285–297, 2013.

[20] Lute, V., Upadhyay, A. and Singh, K. K., "Computationally efficient analysis of cable-stayed bridge for GA-based optimization", *Engineering Applications of Artificial Intelligence* 22: 750-758, 2009.
[21] Martins, A. M. B., Simões, L. M. C. and Negrão, J. H. J. O., "Cable stretching force optimization of concrete cable-stayed bridges including construction stages and time-dependent effects", *Structural and Multidisciplinary Optimization*, 2014 (accepted for publication) doi: 10.1007/s00158-014-1153-4.
[22] Khalil, M. S., Dilger, W. H. and Ghali, A., "Time-dependent analysis of PC cable-stayed bridges", *Journal of Structural Engineering ASCE* 109 (8): 1980-1996, 1983.
[23] Cluley, N. C. and Shepherd, R., "Analysis of concrete cable-stayed bridges for creep, shrinkage and relaxation effects", *Computers & Structures* 58 (2): 337-350, 1996.
[24] Cruz, P. J. S., Marí, A. R., and Roca, P., "Non-linear time-dependent analysis of segmentally constructed structures", *Journal of Structural Engineering ASCE* 124 (3): 278-287, 1998.
[25] Somja, H. and de Ville de Goyet, V., "A new strategy for analysis of erection stages including an efficient method for creep analysis", *Engineering Structures* 30: 2871–2883, 2008.
[26] Nazmy, A. S. and Abdel-Ghaffar, A. M., "Three-dimensional nonlinear static analysis of cable-stayed bridges", *Computers & Structures* 34 (2): 257-271, 1990.
[27] Wang, Pao-Hsii and Yang, Chiung-Guei, "Parametric studies on cable-stayed bridges", *Computers & Structures* 60 (2): 243-260, 1996.
[28] Karoumi, Raid, "Some modelling aspects in the nonlinear finite element analysis of cable supported bridges", *Computers & Structures* 71: 397-412, 1999.
[29] Freire, A. M. S., Negrão, J. H. O. and Lopes, A. V., "Geometrical nonlinearities on the static analysis of highly flexible steel cable-stayed bridges", Computers & Structures 84: 2128-2140, 2006.
[30] EN 1992-1-1. 2010. *NP EN 1992-1-1 Eurocódigo 2 – Projecto de estruturas de betão, Parte 1-1: Regras gerais e regras para edifícios*. IPQ – Instituto Português da Qualidade.
[31] Bazant Z. P., "Material models for structural creep analysis." *In Mathematical modelling of creep and shrinkage of concrete*, edited by Z. P. Bazant, 99-215. John Wiley and Sons, Ltd, 1988.
[32] Ernst, J. H., "Der E-Modul von Seilen unter berucksichtigung des Durchhanges", Der Bauingenieur 40 (2): 52-55, 1965 (in German).
[33] Chen, W. F. and Lui, E. M., *Stability design of steel frames*. CRC Press, 1991.
[34] EN 1993-1-11. *EN 1993-1-11 Eurocode 3 – Design of steel structures, Part 1-11: Design of structures with tension components*. CEN – Comité Européen de Normalisation, 2006.
[35] Simões, L. M. C. and Templeman, A. B., "Entropy-based synthesis of pretensioned cable net structures", Engineering Optimization 15: 121-140, 1989.
[36] prEN 1991-2. *prEN 1991-2 Eurocode 1 – Actions on structures, Part 2: Traffic loads on bridges*. CEN – Comité Européen de Normalisation, 2002.

SYMCOMP 2015
Faro, March 26-27, 2015

A SIMULATION TOOL FOR THE DESIGN OF A RAILWAY ELECTRIC TRACTION SYSTEM

Nuno Dias[1], Nuno Henriques[1], João Calado[1 2], Maria Calado[3] and Silvio Mariano[3]

1: ISEL/IPL - Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1
Lisboa, Portugal

2: IDMEC/IST - Instituto Superior Técnico
Av. Rovisco Pais, 1
Lisboa, Portugal

3: UBI – Universidade da Beira Interior
Faculdade de Engenharia
Calçada Fonte do Lameiro
Covilhã, Portugal

e-mail: nmdias@dem.isel.pt, nhenriques@dem.isel.pt, jcalado@dem.isel.pt, rc@ubi.pt, sm@ubi.pt

Keywords: simulation, railways, traffic, train driving, power supply, traction substation

Abstract *An electrified railway system includes interconnections and complex interactions of various sub-systems and for that reason the computer simulation is the only viable means for evaluation and analysis of such systems and to estimate the power consumption and the optimized behaviours for energy savings.*

This paper discusses the difficulties and requirements of effective simulation models for this kind of application and presents the theoretical basis and procedures used to design an electrical supply system of railway lines by the use of a simulation tool developed in Matlab. Three algorithms used in the simulation tool are described: train movement simulator, rail traffic simulator and electrical system simulator. The train movement simulator determines the position and power consumption of each train at each time step provided the line operation policy. The rail traffic simulator resolves the interaction between several trains participating in the transportation process and leads to the formation of an electric power network, defining at each time step the position of each train and calculating its distance to the substation and other elements of electric infrastructure. The electrical system simulator solves the AC network power flow at each time step and provides the catenary and train voltages, the conductor currents and the substation power flow.

Although this simulation tool consists in a general-purpose multi-train simulator, the analysis and simulation results given are applied to electrified mass rapid transit systems.

1. INTRODUCTION

Electrified railways systems are one of the most attractive transportation technologies in terms of costs, pollution and energy efficiency. The railway network expansion and the increase of both the power requested by rolling stock and train frequency requires complex studies concerned with the electrification project of new lines as well as the changes to be made to optimize the electrical infrastructure of existing lines.

At first glance, an electrified railway line resembles a typical power transmission and distribution system. The main difference is that the loads, represented by trains, move and often changes operation modes can be observed. Consequently, the power system configuration is inevitably changing while the trains move. In addition, power demand varies over a wide range and loads may become sources when regenerative breaking is allowed. Other uncertainties are resulted from a number of factors, such as service scheduling, train speed, traffic demands, traction equipment control and driver´s behaviour, among others.

The complexity of electrified railway systems imposes the development of methodologies for their dimensioning based on numerical simulation models. These solutions allow obtaining quick and more reliable results than those obtained by empirical methods. Simulation is commonly adopted as the most cost-effective tool for electric traction power studies. With the help of simulation tools, the electrical system could be designed so the train voltages are in the admissible range and the conductor currents and transformer powers are smaller than their designed rated values.

This paper presents a simulation tool of an electrical supply system of railway lines oriented to its design. The paper is organized as follows: Section 2 describes the modelling of train motion; Section 3 describes the typical configurations of the electrical supply system of railway lines; Section 4 overviews the simulation tool and describe respectively the three main components of the tool: the train movement simulator, the rail traffic simulator and the electrical simulator; Section 5 illustrates the capabilities of the tool; Section 6 provides few conclusion remarks.

2. MODELLING OF TRAIN MOTION

2.1. Motion Equation

The motion of a train along a track can be predicted using a mathematical model and is determined by the forces acting on the train. The main forces acting on a moving train may include the traction effort, running resistance, alignment resistance and braking force. Tractive effort provides the propulsion to overcome resistances and to accelerate the train.

Running resistance is the force opposing the movement of the train. Alignment resistance is composed of grade resistance and curve resistance. Both are due to the geometry of railway alignment. Braking force is used to decelerate the train and to bring it to full stop.

In the computation of a train movement, the train can be reduced to a point, usually its gravity centre, which moves along a trajectory, defined by the axis of the railroad, by effect of external actions that are exerted on it. Thus, the main movement of a train can be described by Newton's second law, which can be expressed as Equation (1):

$$kM\frac{dv}{dt} = F_t\ v\ - R_r\ v\ - R_c\ x, v\ - R_g(x) \quad (1)$$

On the left side of the equation, *M* is the mass of the train, *v* is the speed of the train, *t* is time, and *dv/dt* is the acceleration of the train. *k* is a factor that is commonly used to consider the rotating inertia and is usually assumed to be less than 1,2 [1].
The right-hand side of the equation is the sum of the forces acting on the train, tangent to its trajectory:

- $F_t\ v$ is the tractive effort or braking force produced in response to control setting when the train is travelling at speed *v*;
- $R_r\ v$ is the resistive force at speed *v* due mainly to the rolling resistance of the wheels, bearing resistance and aerodynamic drag – running resistance;
- $R_c\ x, v$ is the resistive force due to the curvature of the track when the train is at position *x* and the train is travelling at speed *v* - curve resistance;
- $R_g(x)$ is the resistive force due to slope of the track when the train is at position *x* - grade resistance.

Tractive effort and braking force can be calculated from the characteristic curves as a function of speed and is decided for each time step in order to fulfil the speed limits (rolling stock limits and infrastructure limits), maximum acceleration/deceleration, wheel/rail adhesion or other constraints defining the driving mode. Braking force is the sum of regenerative and the mechanical braking force. Regenerative braking force means the resistance force from motor to generate electricity by using the inertia force of train.

2.2. Running Resistance

Running resistance is the result of a set of resistances of different nature that, in straight and levelled track, oppose the train movement. Usually, these resistances are separated into two main categories: basic resistance and air resistance. The first one is purely mechanical and can be further decomposed into rolling resistance and way resistance. However, they are rarely analyzed separately since individual resistances usually cannot be measured precisely. Experimental studies indicate that some portion of basic resistance is constant at all speeds, while the other is proportional to train speed [5]. On the other hand, air resistance depends on the square of the relative speed of train to air, which is equal to train speed in still air [4]. Thus, the train running resistance $R_r\ v$ is approximated by a quadratic function that is variously known as the "Davis equation" [2]:

$$R_r\ v\ = A + Bv + Cv^2 \quad (2)$$

where *A*, *B* and *C* are positive constants for a specific train, usually obtained from field tests or provided by train manufactures.
The coefficients *A* and *B* include the mechanical resistances and depend on the train mass, so that at lower speed (less than 30 m/s) the running resistance R_r is mainly dependent on the train mass. At higher speed, the Cv^2 term related to the aerodynamic resistance becomes dominant. The values of these coefficients are usually set for open air conditions and require modification for the tunnel environment, where the C term, in particular, is larger. In recent

years different authors [2-5] have described the approaches of various national railway network undertaking to the calculation of train running resistance.
Since most of train resistances vary with trainload, it is convenient to express them through resistance coefficients, which are defined as the resistance per weight of the train. Thus, equation (2) can be further rewritten as

$$R_r\ v\ = r_r W = \ a + bv + cv^2\ W \tag{3}$$

where *W* is the weight of the train and *a*, *b* and *c* are positive constants that result from the division of *A*, *B* and *C* constants of equation (2) by the weight of the train.

2.3. Curve Resistance

When a train runs in a curve, its instantaneous moving direction is tangent to the curve. Since rail tracks force the train to travel along the curve, the centrifugal force acting on the train and the friction between wheels and tracks produce extra resistance to train movement. This additional resistance is called curve resistance, which can be expressed as

$$R_c\ x, v\ = r_c W \tag{4}$$

where $R_c\ x, v\ $ is the curve resistance, *W* the weight of the train and $r_c\ is\ the$ coefficient of curve resistance (‰).
The coefficient of curve resistance r_c depends on the friction coefficient, the gauge, the distance between axles in vehicles, and the radius of the curve. It usually increases as the increases of gauge, axle distance, and friction coefficient, but decreases as radius increases. For a specific rail system, the above factors are considered to be known data except radius, which varies with alignment geometry. Usually is not considered the influence of speed by assuming in all cases that speed at every curve is compatible with pre-set limits concerning safety reasons and the wear and tear of the rails. Hence, in practice, it is assumed for all types of rolling stock that the coefficient of curve resistance r_c can be expressed as a constant *K* multiplied by the reciprocal of the radius γ, as shown in equation (5):

$$r_c = \frac{K}{\gamma} \tag{5}$$

The constant *K* depends on the rolling stock and track gauge The typical value of *K* in equation (5) ranges from 500 to 800, depending on rail systems. Note that different *K* values do not make big differences in computing the coefficient curve resistance r_c. For example, when the radius of a curve is 300 m (which is very small for rail systems), the difference between the resulting coefficient r_c for K = 500 and K = 800 is only 1‰. When the radius is 3000 m, the difference is even as small as 0.1‰. Given the similarity of conditions of insertion in the curve of modern rolling stock material, it is usual to give to all vehicles the same value for constant *K*. It is assumed by many railway operators, for the european standard gauge (1.435 m), a value of *K* = 800.

2.4. Grade Resistance

The gradient force is mass-related and can be added as an equivalent linear force. Figure 1 shows that the gravity force of a train on a slope can be resolved into two components: the force along the slope and the force normal to the slope. The former creates a resistance to the movement of the train and is defined as grade resistance. From simple trigonometry, it can be shown that grade resistance is

$$R_g\ x\ = W sen\theta = W\frac{H}{L} \tag{6}$$

where $R_g\ x\ $ is the grade resistance, θ the angle to the horizontal (^{o}), W the weight of the train, L the slope length and H the altitude. Since S is the projection of that slope in the horizontal plane, the slope of the ramp (in ‰) is given by:

$$i = tg\ \theta = \frac{H}{S} \tag{7}$$

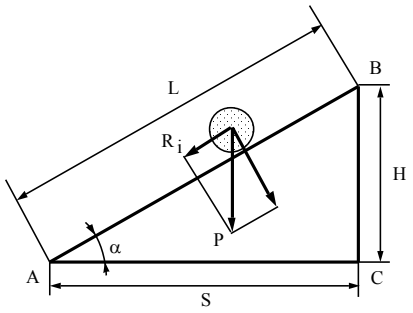


Figure 1. The illustration of grade resistance.

In simple traction grip the ramp slope in general does not exceed more than 30 ‰. For that reason, the angle θ values usually are less than 2˚ and so it can be considered that $sen\ \theta \approx tg\ \theta$. Let r_g be the coefficient of grade resistance (‰). Then equation (6) may be rewritten as

$$R_g\ x\ = r_g W = iW \tag{8}$$

In railways lines, the grade resistance is significant in comparison to the resistance of a levelled track, limiting to very low values the maximum allowed slope, so the trains don't have to be equipped with very high power traction motors or their speeds dropping to unacceptable values. The maximum slope of the ramps in Portugal is around 18 to 20 ‰ for the broad gauge lines and 25 ‰ for the narrow gauge lines [6].

Note that grade resistance is only a general term. It is not necessarily a resistance to train motion. The actual effect depends on the sign of θ. For positive values of, which indicates an incline, grade resistance is in the opposite direction of train movement. On the other hand, for negative values of θ, which represents a decline, grade resistance is in the same direction of train movement and can contribute to train acceleration. If, however, θ is null (level

alignment), then grade resistance is neither an accelerating force nor a decelerating force for the train.

2.5. Equivalent Grade

Since r_c and r_g coefficients are independent of train speed and both are in ‰, it is convenient to add them together. The resulting sum of these coefficients, r_e, represents the coefficient of total alignment resistance and is called equivalent grade [4, 6]:

$$r_e = r_c + r_g \tag{10}$$

A train travels on a grade of r_g with a curve resistance coefficient r_c is equivalent to travel on an equivalent grade r_e.

2.6. Total Resistance

The total resistance to the motion of a train, denoted by $R_t\ x, v$, is the sum of the running and alignment resistances:

$$R_t\ x, v\ = R_r\ v\ + R_c\ x, v\ + R_g\ x\ =\ r_r + r_g + r_c\ W \tag{11}$$

whereby the coefficient of total resistance r_r is given by

$$r_t = r_r + r_g + r_c = a + bv + cv^2 + r_e \tag{12}$$

Thus, equation (1) may be rewritten as

$$kM\frac{dv}{dt} = F_t\ v\ - R_t\ x, v\ = F_t\ v\ - (a + bv + cv^2 + r_e)W \tag{13}$$

3. TRACTION POWER SUPPLY SYSTEM

3.1. General Considerations

The traction power system is, in its simplest form, an electrical circuit with enormous dimensions. The power supply may have single or multiple sources and the feeding arrangements vary in different systems. Figure 2 is intended to represent the different types of traction power supply currently used. The electric power supply to the trains can be made both in DC and in AC.

The loads in the electrical circuits are the trains which are moving and demanding different levels of power according to their operational modes and speeds. A train may become a source if regenerative braking is allowed. Contrary to what happens in normal applications the electrical circuit is deformable, since its dimensions change with the movement of trains.

In order to attain steady-state solution of the electrical circuit, the sources, feeding networks and loads must be properly modelled, which is not a simple task, given the varieties of supply systems and operation conditions.

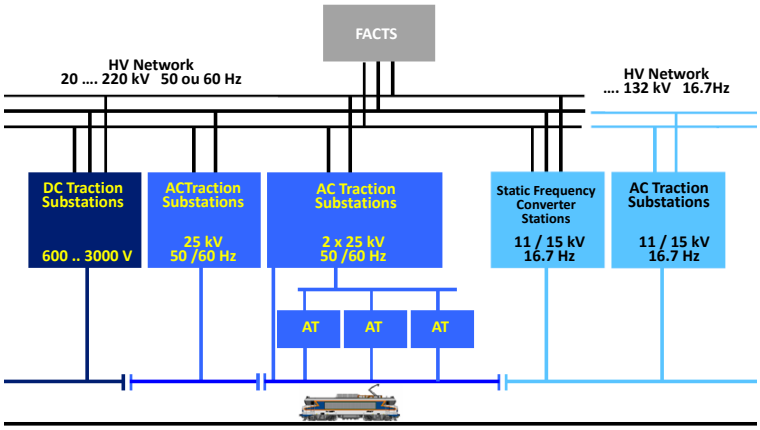


Figure 2. Different types of traction power supply.

3.2. Power-supply systems of AC electrified railways

Power for AC railway traction is usually obtained from three-phase utility supply system, at transmission or sub-transmission voltage level, through traction feeding substations. 25kV traction network at 50 (or 60Hz) is the most commonly adopted system [7]. The general structure of power-supply systems of AC electrified railways is described in Figure 3.

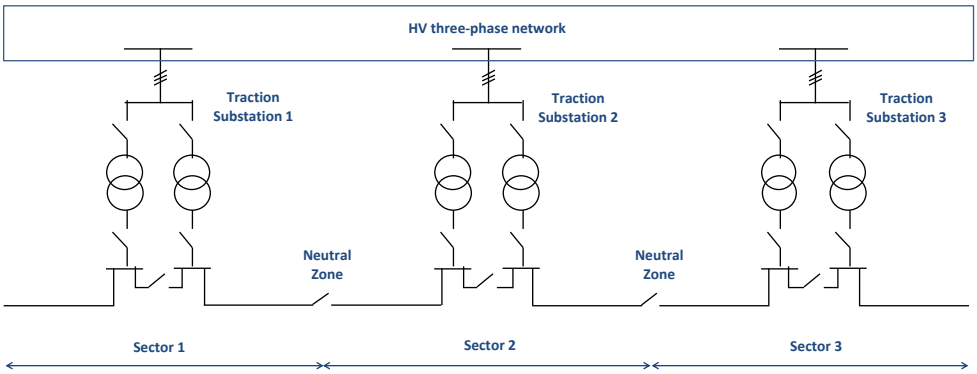


Figure 3. Structure of the power supply system.

The rail line is usually divided into a number of electrically-isolated feeding sections separated by track neutral zones. Traction substations are connected between two of the three phases of the high voltage network. Usually each substation has two transformers, each one connected between two of the three phases and feeds one sector, which means that each section is fed by a single-phase supply. As a result, adjacent sections are fed by different phases of a three-phase network. Power is carried to the trains through overhead catenary and current takes the rails as return paths.

It should be noted that topology can be modified in case of failures to guarantee the operation. Track neutral zones and also parallel sections are equipped with isolators and switchgear to

enable continual feeding in cases of failures and outages by isolating certain faulty equipment or section or even allowing the reconfiguration to a different feeding network. In case that a substation is out of service, the two adjacent substations respectively feed each adjacent section.

The catenary can be equipped with either one or two active conductors resulting respectively in a mono-voltage system (1x25kV) or bi-voltage system (2x25kV). Figures 4 and 5 show the basic topology of these systems. In mono-voltage system (Fig. 4), the catenary is simply fed by a single-phase two winding transformer and is set to the specified voltage level.

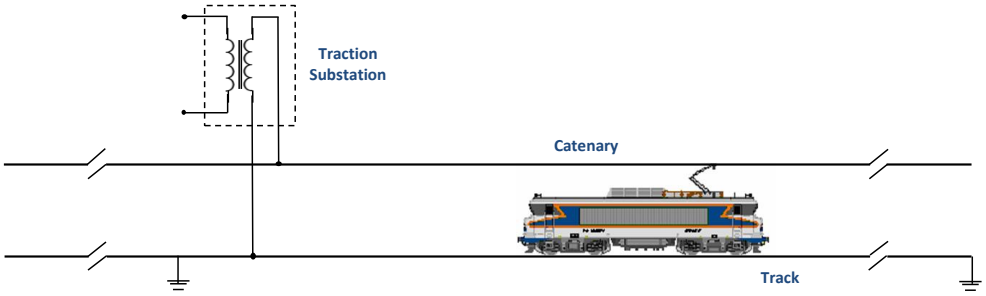


Figure 4. Mono-voltage system configuration.

In bi-voltage systems, the catenary is fed by a single-phase three winding transformer and autotransfomers connect the positive and the negative phases [8]. The autotransformers distributed along the catenary allow to transmit power at 50 kV whereas the trains are fed at 25 kV (Fig. 5).

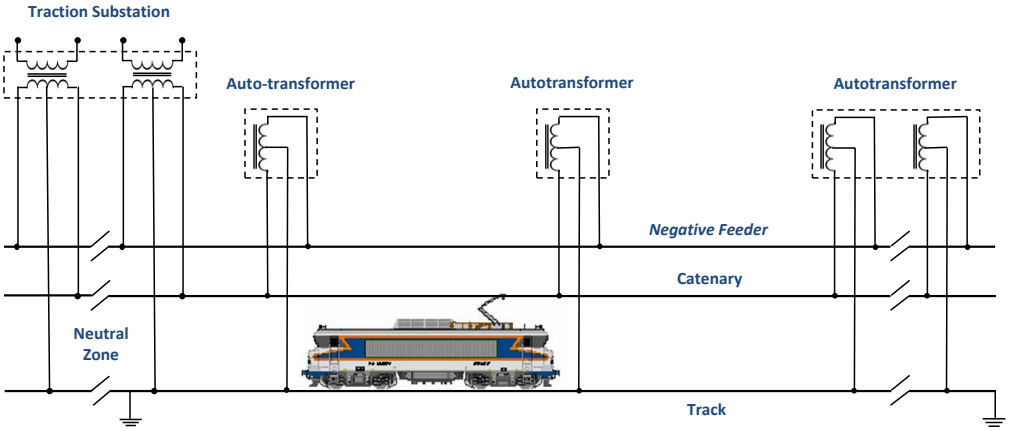


Figure 5. Bi-voltage system configuration.

With these bi-voltage systems, which maintain the use of rolling stock commonly used in 25 kV, the overall voltage drops are two to three times lower than those seen with the mono-voltage systems, allowing almost double the distance between substations and lower sections of the catenary conductors.

3.3. Electrical dimensioning criteria

The problem of designing the electrical supply system consists of determining the values of the design variables so the required technical constraints are fulfilled. Railways power supply systems are dimensioned to be able to supply the power required by the trains, usually assuming a long-term estimation off the traffic needs. This will be achieved if the following restrictions are fulfilled:

- Voltage in catenary has to be within the range specified by UIC-600 standard, which specifies the upper and lower limits depending of its duration;
- Currents circulating along catenary and through transformers and other electrical equipment have to be lower than the rated values, in order to ensure no overheating will occur.
- Power supplied by the three-phase network through the traction substations have to be limited if the network is too weak, in order to ensure its proper operation;
- The unbalanced voltages introduced to the three-phase high-voltage network by the bi-phase railways traction supply has to be lower than a certain imposed limit.

4. SIMULATION PROCEDURE

4.1. Overview

The purpose of the analysis is to calculate voltages, currents and power flows in the whole distribution system in order to evaluate the equipment ratings, estimate power requirements and to verify that the system will maintain a voltage level at the contact point with electrical vehicles sufficient to meet vehicle and system performance criteria.

Basic calculations are train movement simulation and load flow analysis (see figure 6). The whole calculation runs in time steps for the desired time period. The train movement simulator gives the exact locations, active and reactive power demand of each train in every time step provided the line operation policy. This enables the rail traffic simulator build-up the traffic mesh resolving the interaction of the several trains and leads to the formation of electric power network, defining at each time step the position of each train and calculating its distance to the substation and other elements of electric infrastructure.

The electric network is formed from data obtained by the rail traffic simulator. It is possible to form electrical network and simulate the traffic in the selected time period by step by step calculation. The forming of electrical network in railway system is specific because the scheme of powering is continuously changing; i.e. some trains enter and some others leave the powering area. A customary manner of operation for this system is radial operation with a single substation.

The electrical system simulator solves the AC network power flow at each time step and determines the electrical conditions in the power utility distribution system, providing the catenary and train voltages, the conductor currents and the substation power flow.

To determine the power consumption of a traction unit, the resistive effort must be known. Therefore, the input data necessary for the calculation are the railway track profile parameters, planned velocities for each railway track section, as well as characteristics of trains, rolling

stock and signalization.

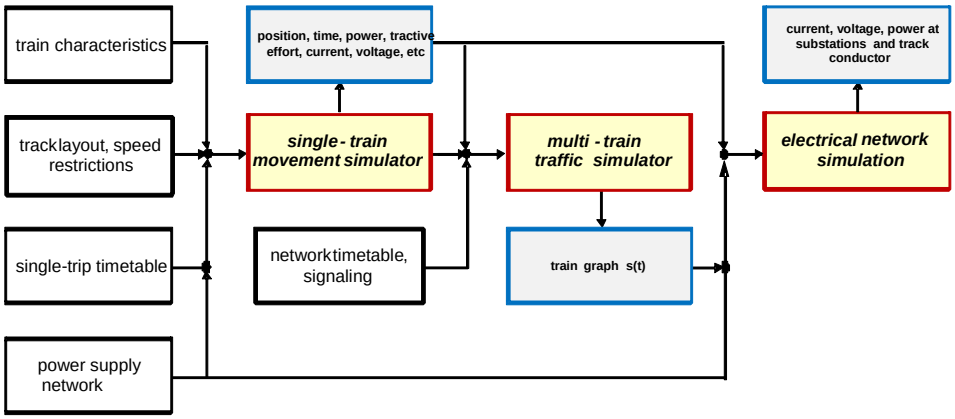


Figure 6. Solving procedure.

4.2. Train Movement Simulator

The movement of a train between two consecutive stations can be described by Newton's second law, which, as seen above, can be expressed as equation (13). The solution of this equation provides the position, time, speed and acceleration of the train, taking into account the speed and accelerations constraints.
Once the movement of the train has been solved, the electrical consumed power P_c and regenerated power P_r is determined using equations (14) and (15) respectively:

$$P_c = {}^{F_t}\ \eta_g + P_{anc} \tag{14}$$

$$P_r = F_t . v . \eta_g + P_{anc} \tag{15}$$

where η_m is the efficiency of the electrical to mechanical power conversion, η_g is the efficiency of the mechanical to electrical power conversion and P_{anc} is the power consumed by ancillary services (air conditioning, pressure equipment, lights, etc.). The reactive power Q is determined from the train power factor.
The power consumed by one train depends on the speed and acceleration that it has at each instant of time. Its computation is based on the traction effort characteristic (supplied by the manufacturer of the motors), the weight of the train and the distances between the stations.

4.3. Rail Traffic Simulator

The simulation of each train movement between two consecutive stops is done assuming that starting time and starting station are the temporal and spatial origins. For that reason, once all the train movements have been calculated, the traffic mesh has to be built-up by shifting the individual movements to their space and time origin. To assign time origins, train frequencies

and stop times in the stations have to be specified. Once the traffic mesh has been constructed, for each time step all the active trains are identified and the traffic scenario can be constructed.

As a result of this procedure, is obtained a schedule table where are represented all the paths of the circulating trains, the start and arrival times, and other significant points of the railways network. Usually, the schedule table for dimensioning the electric traction system refers to the situation in which the maximum number of trains is running.

Thus, rail traffic simulator gives a schedule diagram of the system with the representation of all the circulating trains. This diagram gives, at each simulation moment, the position and the active and reactive power supplied to each motor coach. In this way, for each simulation moment, the position and electric characteristics of the electric systems (nodes) are obtained. These allow to obtain a system "picture", with a specific configuration with the position of the mobile nodes (electric vehicles) related to the fixed nodes (substations and catenary).

It should be noted that in the adopted model, trains do not interact with another trains, contrary to what happens in reality. For this reason, the timetable must reflect overtakes, waiting states or stop times in stations, and all necessary behaviors to allow an efficient exploitation of the railway line, ensuring that there are not any collisions of trains. However, in rail traffic simulator the distance between trains is checked assuming that exists a fixed block signaling system.

A fixed block signaling solution allows a single train to occupy a specific block or area on the guide way. As a train moves along the track, it will occupy blocks which prevent another train from entering that area; mechanical or colorful light signals provide information to the driver on available blocks and routes. Block lengths are calculated using velocities, grades, stopping areas, operational properties of the train, and location of physical elements such as switches and stations.

Figure 7. Example of fixed-block signalisation system.

A major issue with fixed block signaling revolves around the length of the block. As the speed of a area increases, the size of the block increases to account for a larger stopping distance; established lines with faster velocities will have longer block lengths, which can have adverse effects on the inter-station headway.

4.4. Electric Simulator

Once the rail traffic simulator builds the scenarios of electrical system, making the movement interaction of all trains, the location and magnitude of the train loads are available to the electrical simulator. Thus, this simulator determines magnitudes of power system for each scenario, computing voltages, currents and power flows. Note that the two simulators are uncoupled [7], which means that the movement of the trains is not affected by the steady state

of electrical system.
The load flow solution is obtained using an equivalent conductor model of the electrical system. Once the bus voltages of such system have been determinate, the currents of conductors can be computed. The load flow is solved using the Newton-Raphson method.
Note that only was developed the model applied to the mono-voltage system configuration.

5. ILLUSTRATIVE EXAMPLE

In order to illustrate the potencial of the present simulation tool, a 28 km line of the REFER (Portuguese Railways Network) , between Lisboa and Sintra, has been used. This line carries approximately 90 million passengers per year and is most of the stations are less than 2km from each other. A traffic scenario corresponding to a Monday, between 5:30 and 10:00, had been considered. Several electric train UQE Series 2300 used by the Portuguese Operator CP (278 kN maximum traction effort, 120 km/h maximum speed, without regeneration) are also considered, with the regular timetable.

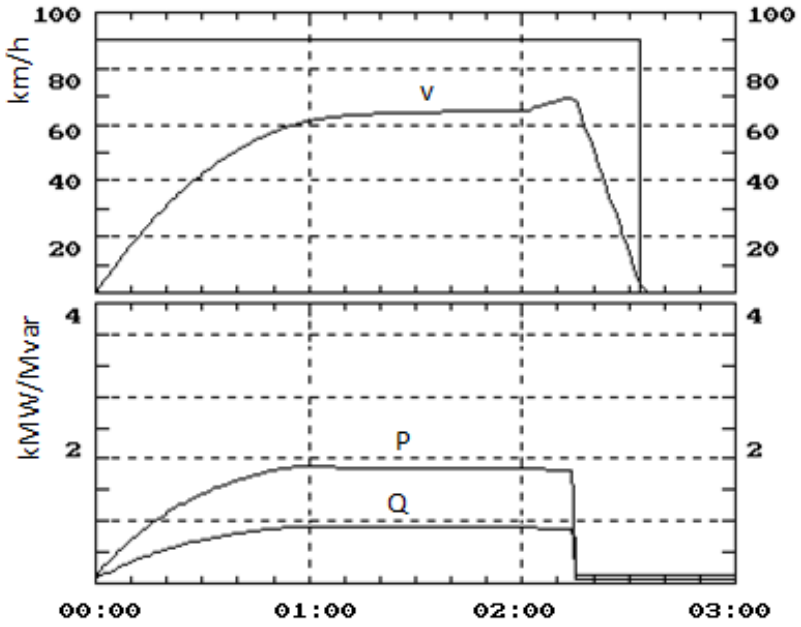


Figure 8. Speed and power of a train.

Figure 8 displays the train instantaneous speed between two stations, as well as active and reactive power required.
Figure 9 shows the evolution of the voltage and current in traction substation over a time period of 30 min.

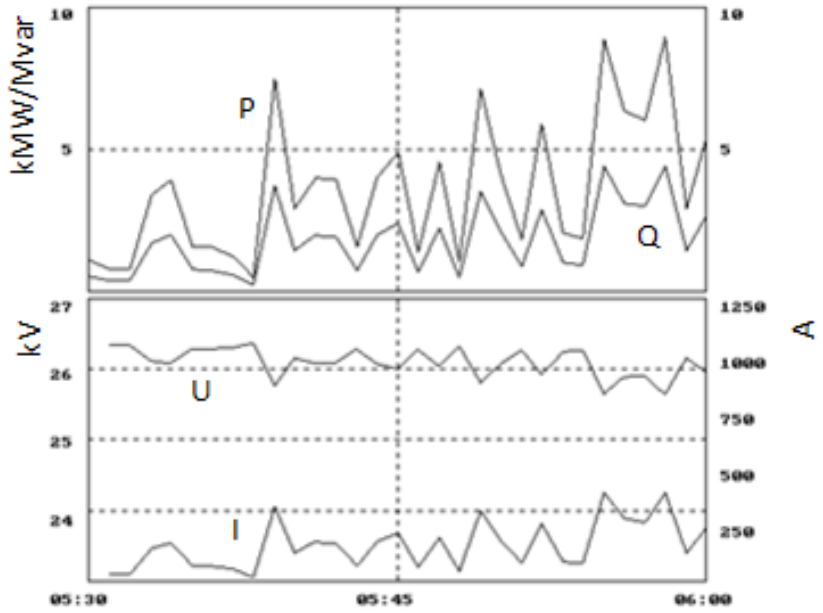


Figure 9. Voltage and current in traction substation.

Figure 10 shows an example of a schedule table for a given time period under study.

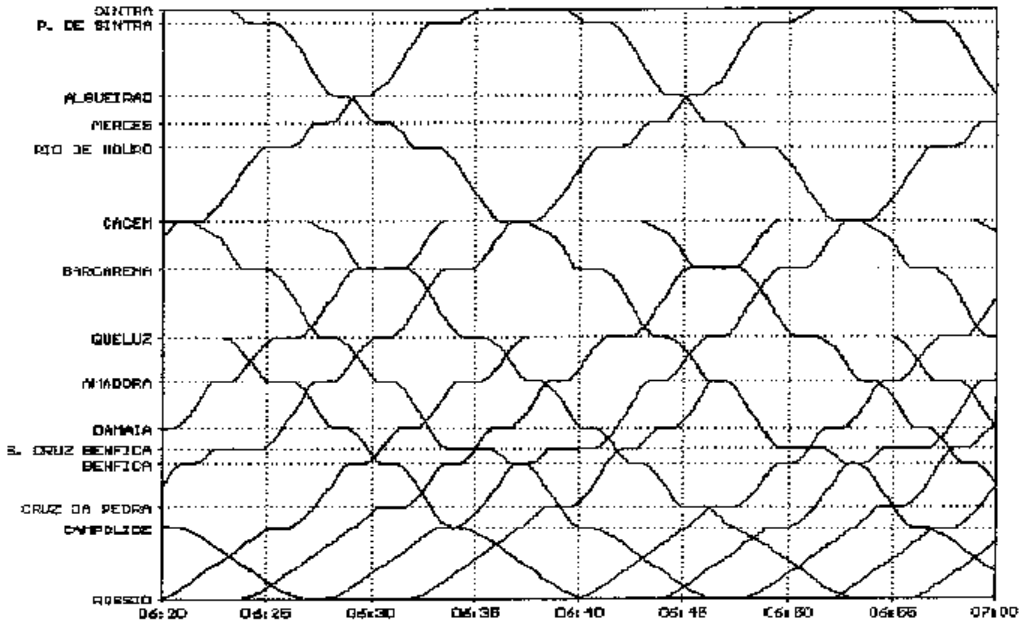


Figure 10. Schedule table.

The comparison of the simulated values with the values provided by the rail operator seems to demonstrate the validity of the simulator tool.

6. CONCLUSIONS

This paper has presented a tool to help in the design of electrical supply system railway lines. The main components of the simulation tool are train movement simulator, rail traffic simulator and electrical simulator. the tool was used to carry out various studies related to the electric traction system of an urban railway line - Sintra Line (Lisbon), in order to verify its validity.
In the future, optimization of the design procedure will be explored, including extending the application of electrical simulator to by-voltage system configuration.

REFERENCES

[1] B. P. Rochard and F. Schmid, "A review of methods to measure and calculate train resistances", *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 214, no. 4, pp. 185–199, 2000
[2] D. S. Armstrong and P. H. Swift, "Lower energy technology. Part A, identification of energy use in multiple units". Report MR VS 077, British Rail Research, Derby, 20 July 1990.
[3] R. G. Gawthorpe, "Train drag reduction from simple design changes", *International Journal of Vehicle Design*, vol. 3, 1983.
[4] P. Lukaszewicz, "Running resistance - results and analysis of full-scale tests with passenger and freight trains in Sweden", *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 221, no. 2, pp. 183–193, 2007.
[5] Sh. Lu, "Optimising Power Management Strategies for Railway Traction Systems". Phd Thesis, University of Birmingham, 2011.
[6] Dias, N., Henriques, N., Calado, J., Calado, M. and Mariano, S., "Topographic Modeling of a Railway Track using a Global Navigation Satellite System", *Procedings of the First International Conference on Railway Technology: Research, Development and Maintenance*, Civil-Comp Press, Stirling, Scotland, 2012, paper 145.
[7] Goodman, C.J., Siu, L.K. and Ho, T.K., 'A Review of Simulation Models for Railway Systems', *International Conf. On Development in Mass Transit Systems*, pp. 80-85, 1998.
[8] E. Pilo, L. Rouco, A. Fernández and A. Hernández-Velilla, "A simulation tool for the design of the electrical supply system of high-speed railway lines," in *IEEE PES Summer Meeting 2000*, Seattle (USA)
[9] P. H. Hsi, S. L. Chen and R. J. Li, "Simulating on-line dynamic voltages of multiple trains under real operating conditions for AC railways," *IEEE Transactions on Power Systems*, vol. 14, pp. 452-459, 1999
[10] R. J. Hill and I. H. Cevik, "On-line simulation of voltage regulation in autotransformer-fed AC electric railroad traction networks," *IEEE Transactions on*

Vehicular Technology, vol. 42, pp. 365-372, 1993

[11] E. Pilo, L. Rouco and A. Fernández, "A reduced representation of 2/spl times/25 kV electrical systems for high-speed railways," in IEEE/ASME Joint Rail Conference, Chicago, 2003, pp. 199-205

[12] Hill, R.J. 'Electric Railway Traction - Part 3 Traction Power Supplies', *Power Engineering Journal*, 1994, 275-286.

Author Index