Programs and Algorithms of Numerical Mathematics 16

Dolní Maxov, June 3–8, 2012

Proceedings of Seminar

Edited by
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský

Institute of Mathematics
Academy of Sciences of the Czech Republic
Prague 2013

ISBN 978-80-85823-62-2
Matematický ústav AV ČR, v.v.i.
Praha 2013

Contents

Preface 7

Stanislav Bartoň, Tomáš Krumpholc
Driver's influence on kinematics of articulated bus rear axle 9

Daniela Bímová, Dana Černá, Václav Finěk
Wavelet bases for the biharmonic problem 15

Marek Brandner, Jiří Egermaier, Hana Kopincová, Josef Rosenberg
Numerical modelling of flow in lower urinary tract using high-resolution methods 21

Dana Černá, Václav Finěk, Martina Šimůnková
A quadratic spline-wavelet basis on the interval 29

Marta Čertíková, Jakub Šístek, Pavel Burda
On selection of interface weights in domain decomposition methods 35

Jan Chleboun
On the optimal setting of the *hp*-version of the finite element method 45

Pavol Chocholatý
Integro-differential equations with time-varying delay 51

Josef Dalík
Superapproximation of the partial derivatives in the space of linear triangular and bilinear quadrilateral finite elements 57

Ján Eliaš, Jan Zítko
Approximate polynomial GCD 63

Václav Finěk, Martina Šimůnková
Parallel implementation of wavelet-Galerkin method 69

Cyril Fischer
Massive parallel implementation of ODE solvers 75

Milan Hanuš
A new perspective on some approximations used in neutron transport modeling 81

Martin Horák, Milan Jirásek
An extension of small-strain models to the large-strain range based on an additive decomposition of a logarithmic strain 88

Jiří Hozman
Valuing barrier options using the adaptive discontinuous Galerkin method 94

Radek Kaňa, Ctirad Matonoha, Štěpán Papáček, Jindřich Soukup
On estimation of diffusion coefficient based on spatio-temporal FRAP images: An inverse ill-posed problem ... 100

Radka Keslerová, Karel Kozel
Numerical simulation of generalized Newtonian and Oldroyd-B fluids flow ... 112

Jiřina Královcová, Ladislav Lukšan, Jaroslav Mlýnek
Heat exposure optimization applied to moulding process in the automotive industry ... 118

Jiří Krček, Jaroslav Vlček, Arnošt Žídek
Tangential fields in optical diffraction problems ... 124

Lukáš Krupička, Radek Štefan, Michal Beneš
Coupled heat transport and Darcian water flow in freezing soils ... 130

Václav Kučera
Error estimates for nonlinear convective problems in the finite element method 136

Jaroslav Mlýnek, Radek Srb
The optimization of heat radiation intensity ... 142

Vratislava Mošová
Shape functions and wavelets – tools of numerical approximation ... 149

Lukáš Pospíšil
An optimal algorithm with Barzilai-Borwein steplength and superrelaxation for QPQC problem ... 155

Jan Přikryl
Graphics card as a cheap supercomputer ... 162

Ivana Pultarová
Fourier analysis of iterative aggregation-disaggregation methods for nearly circulant stochastic matrices ... 168

Petr Salač
Optimization of plunger cavity ... 174

Karel Segeth
Smooth approximation of data with applications
to interpolating and smoothing ... 181

Jiří Vala
Numerical aspects of the identification of thermal characteristics using the hot-wire method ... 187

Tomáš Vejchodský
Guaranteed and fully computable two-sided bounds of Friedrichs' constant 195

Miloslav Vlasák, Václav Kučera
A priori diffusion-uniform error estimates for singularly perturbed problems: midpoint-DG discretization 202

Jan Vlček, Ladislav Lukšan
Modifications of the limited-memory BFGS method based on the idea of conjugate directions 209

Jan Zítko, Ján Eliaš
Calculation of the greatest common divisor of perturbed polynomials 215

List of participants 223

Preface

This book comprises papers that originated from the invited lectures, survey lectures, short communications, and posters presented at the 16th seminar Programs and Algorithms of Numerical Mathematics (PANM) held in Dolní Maxov, Czech Republic, June 3–8, 2012. All the papers have been peer-reviewed.

The seminar was organized by the Institute of Mathematics of the Academy of Sciences of the Czech Republic. It continued the previous seminars on mathematical software and numerical methods held (biannually, with only one exception) in Alšovice, Bratříkov, Janov nad Nisou, Kořenov, Lázně Libverda, Dolní Maxov, and Prague in the period 1983–2010. The objective of this series of seminars is to provide a forum for presenting and discussing advanced topics in numerical analysis, computer implementation of algorithms, new approaches to mathematical modeling, and single- or multi-processor applications of computational methods.

More than 50 participants from the field took part in the seminar, most of them from Czech universities and from institutes of the Academy of Sciences of the Czech Republic but also from Austria and Slovakia. The participation of a significant number of young scientists, PhD students, and also some undergraduate students is an established tradition of the PANM seminar and it was observed this year, too. We do believe that those, who took part in the PANM seminar for the first time, have found the atmosphere of the seminar friendly and stimulating, and are going to join the PANM community.

The organizing committee consisted of Jan Chleboun, Petr Přikryl, Karel Segeth, Jakub Šístek, and Tomáš Vejchodský. Ms Hana Bílková kindly helped in preparing manuscripts for print.

All papers have been reproduced directly from materials submitted by the authors. In addition, an attempt has been made to unify the layout of the papers.

The editors and organizers wish to thank all the participants for their valuable contributions and, in particular, all the distinguished scientists who took a share in reviewing the submitted manuscripts.

A few days before the final release of these proceedings, we were hit by the sad news that our colleague Josef Dalík suddenly passed away. In our minds, he will be remembered also as a regular participant of PANM seminars. His contributions, not only in the proceedings you are just reading, will be recalling him to our community.

J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

DRIVER'S INFLUENCE ON KINEMATICS OF ARTICULATED BUS REAR AXLE

Stanislav Bartoň, Tomáš Krumpholc

Mendel University in Brno
Zemědělská, 613 00 Brno, Czech Republic
barton@mendelu.cz

Abstract

This paper studies kinematic properties of the rear axle of the particle coach as function of driver's activity. The main goals are the prediction of the trajectory, the computation of the vector of velocity of each wheel of the rear axle as a function of the real velocity vector of the front coach axle and the real curvature of the bus trajectory. The computer algebra system Maple was used for all necessary computations.

1. Introduction

1.1. Classical problem of kinematic

In following computations we should use these main variables: $X(t), Y(t)$ – General coordinates of the moving body, later coordinates of the midpoint of the central axle of the articulated bus, which is equal to the joint point of the rear – towed axle. $x(t), y(t)$ – Coordinates of the midpoint of the towed axle. L – Constatnt distance beween midpoints of the central axle - joint, and rear - towed axle, see Figure 1.

The classical problem of kinematics is the computation of the speed $\vec{V}(t)$ and the acceleration vector $\vec{A}(t)$ of a body as a function of time when the location of the body is given by the functions $\vec{P}(t) = [X(t), Y(t)]$. The next step is the computation of the tangential acceleration $A_t(t)$, which changes the absolute value of the velocity and the normal acceleration $A_n(t)$, which changes the direction of the velocity. And finally, the function of the center of the osculation circle of the trajectory $\vec{C}(t)$ and its radius $R(t)$ are derived. These functions can also be found in [4, 2, 9].

1.2. The influence of the driver

The driver controls the bus using the gas and the brake pedal – he controls the absolute value of the velocity of the bus $|\vec{V}(t)|$. Furthermore – using the steering wheel – he controls the radius of the osculating circle $R(t)$, on which the bus is currently moving. For further calculations it is useful to use the inverse value of the radius of the osculation circle – the curvature of trajectory $k(t) = R(t)^{-1}$. By combining these two controls the bus driver keeps the bus moving smoothly on the road.

2. Inverse problem

Let us assume that we know the temporal behavior of driver's operations. Thus we know the functions of the speed magnitude $|\vec{V}(t)| = v(t)$ and curvature $k(t)$. Then the problem is to compute the trajectory of the bus and the related kinematics variables. For this we need to solve a non-linear system of two ordinary differential equations of second and first order, they are solved in [1, 5]

$$\sqrt{\dot{X}^2+\dot{Y}^2} = v(t),\ \frac{\ddot{Y}\dot{X}-\ddot{X}\dot{Y}}{\left(\dot{X}^2+\dot{Y}^2\right)^{\frac{3}{2}}} = k(t). \tag{1}$$

After some algebraic manipulations the equations (1) are transformed to an explicit system of two differential equations of order two:

$$\ddot{X} = \frac{-\dot{Y}\,k(t)\,v(t)^2 + \frac{d\,v(t)}{dt}\,\dot{X}}{v(t)},\ \ddot{Y} = \frac{\dot{X}\,k(t)\,v(t)^2 + \frac{d\,v(t)}{dt}\,\dot{Y}}{v(t)}. \tag{2}$$

Given an initial velocity $v_0 = |\vec{V}(0)|$ and its initial direction defined by the angle ϕ_0 and the initial position of the bus $[X_0,\ Y_0]$, the solution of (2) can be found to be (see [4,5])

$$X = \int_0^t v(\tau)\,\cos(f)\,d\tau + X_0\ ,\ Y = \int_0^t v(\tau)\,\sin(f)\,d\tau + Y_0\ , \tag{3}$$

where $f = \phi_0 + \int_0^\tau v(\tau)\,k(\tau)\,d\tau$. This is an analytic solution, however, even for simple functions $v(t)$ and $k(t)$ it will not be possible to compute explicit expressions for the integrals. A considerable advantage of this result is that it allows to numerically integrate the position for any given time t. We have not to be concerned with accumulation of rounding errors as e.g by integrating the system (2) with some numerical methods, like Runge-Kutta, see [8].

3. Generalized tractrix as model of the trajectory of the rear axle

Let us assume that the joint of the articulated bus is located in the middle of second axle and that the trajectory of the joint is given by $[X, Y]$. The centre of the rear axle, given by $[x, y]$ - the towed axle - is to be computed. The centres of both axles have to have a constant distance L and the velocity vector of the center of the towed axle has to pass the joint, see Figure 1.

From these conditions (see [3]) we obtain the system of differential equations $[X, Y]$ and $[x, y]$.

$$\dot{x} = \frac{\Delta X\left(\Delta X\,\dot{X} + \Delta Y\,\dot{Y}\right)}{L^2},\ \dot{y} = \frac{\Delta Y\left(\Delta X\,\dot{X} + \Delta Y\,\dot{Y}\right)}{L^2},\ \text{where}\ \begin{array}{l}\Delta X = x - X,\\ \Delta Y = y - Y.\end{array} \tag{4}$$

It is a system of two non-linear differential equations of first order, which for simple functions X and Y is relatively easy to solve.

But if we introduce for X and Y the expressions of Equations (3), we get a very complex system of differential equations, for which it is first necessary to solve for X and Y by the numerical integration. This combination of numerical integration and solving of differential equations is too complex for the computer algebra system MAPLE. It is not possible to use successfully direct numerical solution of equations using the command **dsolve** together with the parameter **numeric**.

3.1. Numerical integration of the equation of motion

It is possible to solve the system (3) together with Equations (4) numerically using Runge-Kuttas method, see [8]. We implemented this in MAPLE as procedure **RK45**. This procedure determines the position and velocity of the towed axles centre at time $t + \Delta t$. The next procedure, named **STEP**, see (5), defines the magnitude of time step Δt using a step size control. For the first iteration a random time step magnitude is chosen, e.g. $\Delta t = 1$ and the position for this time is calculated. Time t and coordinates x, y are saved in the vector $R1$. Similarly the position is calculated in the same procedure, but in two steps with a half time step size $\Delta t/2$ and saved as a vector $R2$. If the difference between these vectors is smaller than the required accuracy, $|R1 - R2| \leq 10^{-6}$, we add the resulting position, saved in the vector $R1$ to vector Λ. Otherwise we reduce the size of time step by half and repeat the process. At the end of the iteration procedure the vector Λ will contain vectors – ordered triplets containing the time and the towed axles position coordinates of the each iteration step.

```
STEP := proc($U$) local $l, R2$; global $R1, \Lambda, \Delta t, t$;
  $l := U[]$;  $R2 := [\mathbf{RK45}(\mathbf{RK45}(l, \frac{\Delta t}{2}), \frac{\Delta t}{2})]$; $R1 := [\mathbf{RK45}(l, \Delta t)]$;
  if sqrt(add($u^2, u = R1 - R2$)) $\leq 10^{-6}$ then $\Lambda := [\Lambda[], R1]$; $t := t + \Delta t$;      (5)
  else $\Delta t := \frac{\Delta t}{2}$ end if
end proc
```

4. Practical application

Let us take as example a passing of a rectangular turn when the bus is breaking. For this case we consider

$$v(t) = V_0 - a\,t,\ k(t) = \frac{4\,t\,(T_f - t)}{T_f^2\,\rho}, \tag{6}$$

where V_0 is the initial velocity, a is deceleration, T_f is the period of turn passing and ρ is the least diameter of a passed turn.

If we choose the direction in time $t = 0$ parallel to x axis, therefore $\phi_0 = 0$, the turn will be finished at the moment, when the vector of immediate velocity $[\dot{X}, \dot{Y}]$ will be parallel to y. Therefore it is stated that $\dot{X} = 0$. From this condition it is obvious, that because of Equation (3) we have (details can be found in [6, 7])

$$T_f = \frac{2\,V_0 - \sqrt{4\,V_0^2 - 6\,a\,\pi\,\rho}}{2\,a}. \tag{7}$$

4.1. Numerical integration

As particular values we take $V_0 = 10\,\mathrm{m\,s^{-1}}$, $a = 0.5\,\mathrm{m\,s^{-2}}$, $\rho = 20\,\mathrm{m}$, $x_0 = 0\,\mathrm{m}$, $y_0 = 0\,\mathrm{m}$ and $L = 4\,\mathrm{m}$. For these values the time necessary to pass the turn is $T_f = 5.45681\,\mathrm{s}$. The initial time is $t = 0\,\mathrm{s}$ and for the initial time step we choose $\Delta t = T_f$. Now we create the Λ list, its first element will be $[t, -L, 0]$, then $\Lambda := [[0, -4, 0]]$. Procedure **STEP** determines the first step size of the time step as $\Delta t = \frac{T_f}{128} = 0.04263\,\mathrm{s}$ and then executes 128 integration steps. For specific integration times it is possible to compute using Equations (3) the position of middle axles centre point. Due to Figure 2 and the following relationship (8) it is hence possible to calculate the position of front, middle and rear – towed axle.

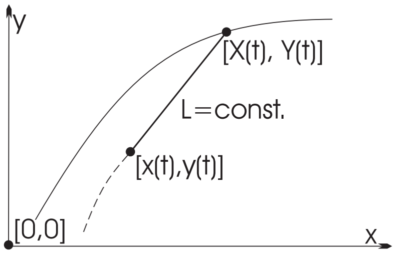


Figure 1: Deriving the motion equation of towed axle.

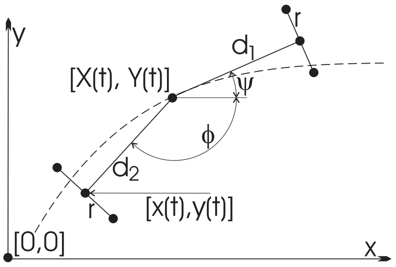


Figure 2: Calculation for single wheels of a bus.

$$\text{Wheel} = [X, Y] + d\,[\cos(\alpha), \sin(\alpha)] + r\,[-\sin(\alpha), \cos(\alpha)], \tag{8}$$

for d we can take d_1 – the distance between centrepoints of middle and front axle, or d_2 – the distance between middle and rear axles centrepoints. For α we can take ψ – the directional vector pointing from the middle axle to the front axles midpoints. This is the directional vector of the velocity $[\dot{X}, \dot{Y}]$ or ϕ – the directional vector pointing from the joint of the middle point of the rear axle, r = wheel spacing of single axles. Angular sizes ϕ and ψ could be easily solved using the vector calculation. The result of the calculation could therefore be a graph on Figure 3., depicting the trajectories of single wheels, or a graph on Figure 4, which represents the angle of cranking of the bus joint.

5. Conclusion and discussion

We have developed a method which allows for any velocity function $v(t)$ and trajectory curvature function $k(t)$ to compute all important kinematic variables of the the articulated bus. This concerns not only the wheels but can be applied for any arbitrary point inside the bus. Just take for that the appropriate sizes of variables d_1, d_2, and r matching the Figure 2 and Equation (8). Furthermore, it is possible to determine the acceleration of any point, including the points which correspond to points of contact between the wheels and the road. This knowledge

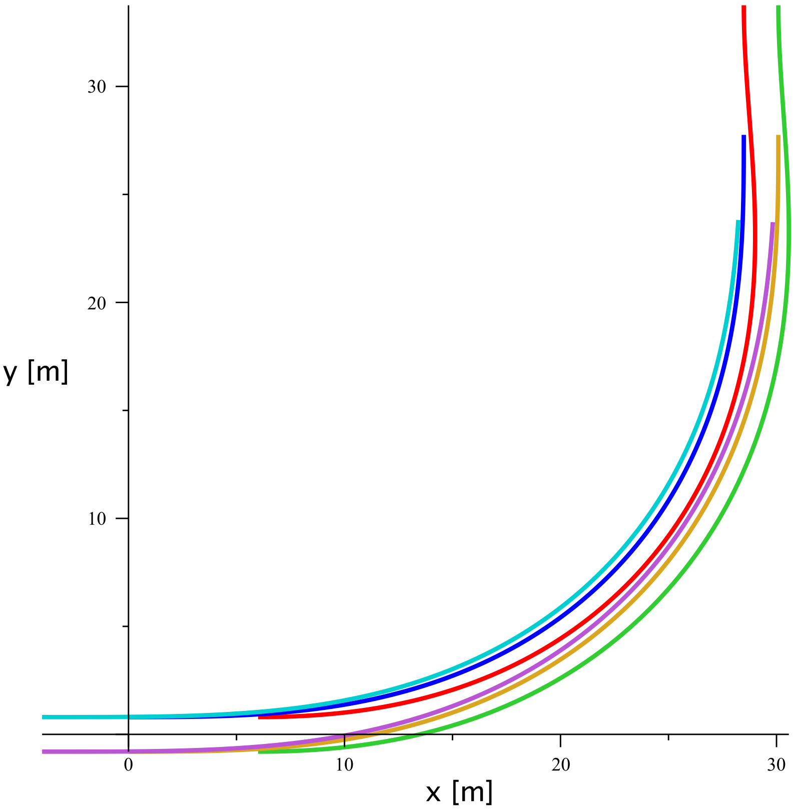


Figure 3: Trajectory of separate wheels of the bus.

of acceleration could be used for the determination of the adhesion threshold limits. This procedure could be also used for the inverse problem. From the moment of the adhesion loss to breakaway it is possible to experimentally find such a velocity and trajectory curvature functions, that caused the skid. Therefore it is possible from the trajectory – a braking track – to estimate the drivers actions, that preceded this event.

From the knowledge of acceleration inside the bus it is possible to perform the calculations of general force, affecting the whole bus as well as individual passengers. Knowledge of this general force is an important factor affecting the stability of the bus. The force affecting the single passenger is a limiting factor for their safety. The method mentioned above allows us to simulate the drivers behavior and the impact on safety of passengers due to their position inside the bus.

Acknowledgments

The work has been supported by research and development projects IP 22/2012, financed by IGA AF MENDELU.

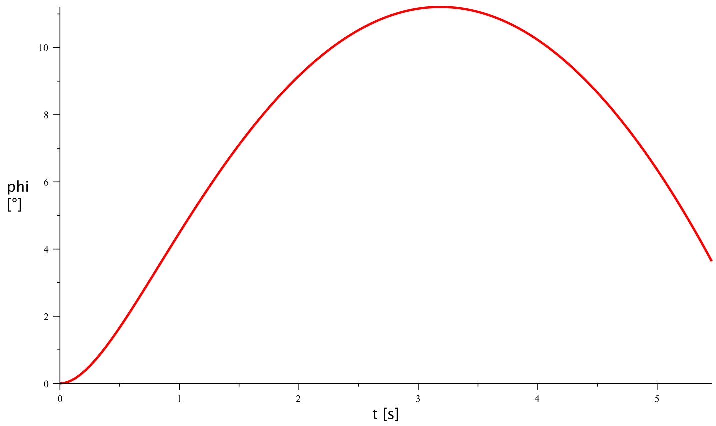


Figure 4: Angle of cranking the joint of the bus.

References

[1] Bartoň, S. and Krumpholc, T.: *Stanovení trajektorie vozidla – inverzní problém kinematiky.* [CD-ROM]. SCO 2011 Workshop Maple, pp. 1–8, 2011.

[2] Brand, L.: *Vector and tensor analysis.* John Wiley, New York, 1947, 121–124.

[3] Gander, W., Bartoň, S., and Hřebíček, J.: The tractrix and simillar curves. In: Gander, W., Hřebíček, J. (Eds), *Solving problems in scientific computing using Maple and Matlab.* 4. ed., pp. 1–26. Heidelberg, Springer, 2004, ISBN 3-540-21127-6.

[4] `http://webfyzika.fsv.cvut.cz/PDF/webFyzika_vztahy_mechanika.pdf`

[5] Krumpholc, T. and Bartoň, S.: *Matematický model řízené zatáčky autobusu.* [CD-ROM]. Recent Advances in Agriculture, Mechanical Engineering and Waste, pp. 120–125. SPU Nitra, 2012, ISBN 978-80-552-0781-0.

[6] Krumpholc, T. and Bartoň, S.: *Stanovení trajektorie vozidla po zásahu řidiče do řízení.* [CD-ROM]. Kvalita a spoĺahlivosť technických systémov – Zborník vedeckých prác, pp. 186–191. SPU Nitra, 2011, ISBN 978-80-552-0595-3.

[7] Krumpholc, T. and Bartoň, S.: *Studie trajektorie autobusu při brzděném průjezdu zatáčkou.* [CD-ROM]. MendelNet 2011 - Proceedings of International Ph.D. Students Conference, pp. 879–904, ISBN 978-80-7375-563-8.

[8] Ralston, A.: *Základy numerické matematiky.* Academia Praha, 1978.

[9] Spallek, K.: *Kurven und Karten.* Bibliographisches Institut Mannheim Wien Zürich, 1980, ISBN 3-411-01593-4.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

WAVELET BASES FOR THE BIHARMONIC PROBLEM

Daniela Bímová, Dana Černá, Václav Finěk
Technical University in Liberec
Studentská 2, 46117 Liberec, Czech Republic
daniela.bimova@tul.cz, dana.cerna@tul.cz, vaclav.finek@tul.cz

Abstract

In our contribution, we study different Riesz wavelet bases in Sobolev spaces based on cubic splines satisfying homogeneous Dirichlet boundary conditions of the second order. These bases are consequently applied to the numerical solution of the biharmonic problem and their quantitative properties are compared.

1. Introduction

Wavelets are an established tool for the numerical solution of operator equations. One of advantages of wavelet methods consists in the existence of a diagonal preconditioner. This preconditioner is optimal in the sense that the condition number of the preconditioned stiffness matrix does not depend on the size of the matrix. Furthermore, a well-known compression property of wavelets enables efficient adaptive solving of operator equations.

In numerical simulations, spline-wavelet bases are of special interest, because they are known in a closed form, they are relatively smooth and they have a small support in comparison with other wavelet bases, e.g. orthonormal wavelet bases. For the numerical treatment of operator equations wavelet bases defined on bounded domain are needed. They are usually derived from wavelet bases on the interval. Recently, several constructions of cubic spline-wavelet bases on the interval adapted to the second order homogeneous Dirichlet boundary conditions were proposed [1, 3, 9, 10]. The bases in [4, 10] have local dual basis functions, which is important in some applications, such as solving nonlinear equations, but for solving partial differential equations the locality of duals is not necessary. Therefore in a construction in [8], the locality of duals is not required. The resulting basis has superb quantitative properties, but wavelets have no vanishing moments. In [5], we also gave up the locality of duals and we designed a cubic spline-wavelet basis with vanishing wavelet moments adapted to homogeneous Dirichlet conditions for the biharmonic problem. In this contribution, we show that our basis have similar excellent quantitative properties as basis from [8] and due to vanishing moments it can be used also in adaptive wavelet methods. In [5], a proof that this basis is a Riesz basis of the space $H_0^s(0,1)$ for $1.5 < s < 2.5$ is presented and properties of the projectors associated with this basis are derived.

2. Construction of wavelet basis

We consider the domain $\Omega \subset \mathbb{R}^d$ and the Sobolev Space $H_0^2(\Omega)$ with the standard $H_0^2(\Omega)$–norm denoted by $\|\cdot\|_{H_0^2(\Omega)}$ and the $H_0^2(\Omega)$–seminorm denoted by $|\cdot|_{H_0^2(\Omega)}$. Let $\mathcal{J}$ be some index set and let each index $\lambda \in \mathcal{J}$ take the form $\lambda = (j,k)$, where $|\lambda| := j \in \mathbb{Z}$ is a *scale* or a *level.* Let

$$l^2(\mathcal{J}) := \left\{ \mathbf{v} : \mathcal{J} \to \mathbb{R}, \sum_{\lambda \in \mathcal{J}} |\mathbf{v}_\lambda|^2 < \infty \right\}, \quad \|\mathbf{v}\|_{l^2(\mathcal{J})} := \left(\sum_{\lambda \in \mathcal{J}} |\mathbf{v}_\lambda|^2 \right)^{1/2}. \tag{1}$$

A family $\Psi := \{\psi_\lambda, \lambda \in \mathcal{J}\}$ is called a *wavelet basis* of $H_0^2(\Omega)$, if

i) Ψ is a *Riesz basis* for $H_0^2(\Omega)$, i.e. the closure of the span of Ψ is $H_0^2(\Omega)$ and there exist constants $c, C \in (0,\infty)$ such that

$$c\,\|\mathrm{b}\|_{l^2(\mathcal{J})} \leq \left\| \sum_{\lambda \in \mathcal{J}} b_\lambda \psi_\lambda \right\|_{H_0^2(\Omega)} \leq C\,\|\mathrm{b}\|_{l^2(\mathcal{J})}, \quad \mathrm{b} := \{b_\lambda\}_{\lambda \in \mathcal{J}} \in l^2(\mathcal{J}). \tag{2}$$

ii) The functions are *local* in the sense that $\mathrm{diam}(\Omega_\lambda) \leq \mathrm{C}2^{-|\lambda|}$ for all $\lambda \in \mathcal{J}$, where Ω_λ is the support of ψ_λ, and at a given level j the supports of only finitely many wavelets overlap at any point $x \in \Omega$.

A wavelet basis is usually formed by two types of functions: scaling functions and wavelets. We focus on a wavelet basis recently constructed in [5] and we briefly review the construction. Let ϕ be a cubic B-spline defined on knots $[0,1,2,3,4]$ and ϕ_b be a cubic B-spline defined on knots $[0,0,1,2,3]$. The graphs of the functions ϕ and ϕ_b are displayed in Figure 1. For $j \in \mathbb{N}$ and $x \in [0,1]$ we set

$$\begin{aligned} \phi_{j,k}(x) &= 2^{j/2}\phi(2^j x - k),\ k = 2, \ldots 2^j - 2, \\ \phi_{j,1}(x) &= 2^{j/2}\phi_b(2^j x), \quad \phi_{j,2^j-1}(x) = 2^{j/2}\phi_b(2^j(1-x)). \end{aligned} \tag{3}$$

We define a wavelet ψ as

$$\psi(x) = -\frac{1}{2}\phi(2x) + \phi(2x-1) - \frac{1}{2}\phi(2x-2). \tag{4}$$

Then ψ has two vanishing wavelet moments, i.e.

$$\int_{-\infty}^{\infty} x^k \psi(x)dx = 0, \quad k = 0, 1. \tag{5}$$

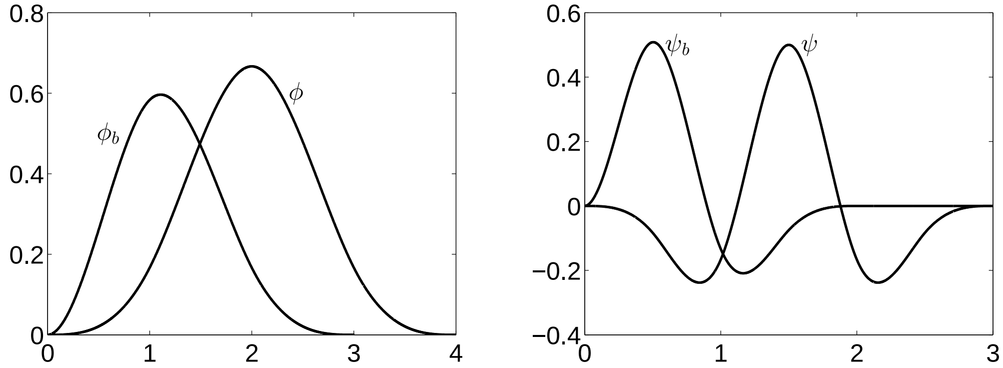


Figure 1: Scaling functions ϕ and ϕ_b and wavelets ψ and ψ_b.

There are several choices for the definition of boundary wavelet. We choose a wavelet with the shortest possible support and the first wavelet moment vanishing:

$$\psi_b(x) = \phi_b(2x) - 0.45\phi(2x). \tag{6}$$

The graphs of the functions ψ and ψ_b are displayed in Figure 1. The inner wavelets correspond to the construction of a wavelet basis for the space $L^2(\mathbb{R})$ in [7].

For $j \in \mathbb{N}$ and $x \in [0,1]$ we define

$$\begin{aligned} \psi_{j,k}(x) &= 2^{j/2}\psi(2^j x - k + 2), k = 2, ..., 2^j - 1, \\ \psi_{j,1}(x) &= 2^{j/2}\psi_b(2^j x), \quad \psi_{j,2^j}(x) = 2^{j/2}\psi_b(2^j(1-x)). \end{aligned} \tag{7}$$

We denote

$$\begin{aligned} \Phi_j &= \left\{ \phi_{j,k} / \left|\phi_{j,k}\right|_{H_0^2(0,1)}, k = 1, \ldots, 2^j - 1 \right\}, \\ \Psi_j &= \left\{ \psi_{j,k} / \left|\psi_{j,k}\right|_{H_0^2(0,1)}, k = 1, \ldots, 2^j \right\}. \end{aligned} \tag{8}$$

Then the sets

$$\Psi_s = \Phi_2 \cup \bigcup_{j=2}^{1+s} \Psi_j \quad \text{and} \quad \Psi = \Phi_2 \cup \bigcup_{j=2}^{\infty} \Psi_j \tag{9}$$

are a multi-scale wavelet basis and a wavelet basis of the space $H_0^2(0,1)$, respectively. We use $u \otimes v$ to denote the tensor product of functions u and v, i.e. $u \otimes v\,(x_1, x_2) = u\,(x_1)\,v\,(x_2)$. We set

$$\begin{aligned} F_j &= \left\{ \phi_{j,k} \otimes \phi_{j,l} \,/\, \left|\phi_{j,k} \otimes \phi_{j,l}\right|_{H_0^2(\Omega)}, k, l = 1, \ldots, 2^j - 1 \right\} \\ G_j^1 &= \left\{ \phi_{j,k} \otimes \psi_{j,l} \,/\, \left|\phi_{j,k} \otimes \psi_{j,l}\right|_{H_0^2(\Omega)}, k = 1, \ldots, 2^j - 1, l = 1, \ldots 2^j \right\} \\ G_j^2 &= \left\{ \psi_{j,k} \otimes \phi_{j,l} \,/\, \left|\psi_{j,k} \otimes \phi_{j,l}\right|_{H_0^2(\Omega)}, k = 1, \ldots, 2^j, l = 1, \ldots 2^j - 1 \right\} \\ G_j^3 &= \left\{ \psi_{j,k} \otimes \psi_{j,l} \,/\, \left|\psi_{j,k} \otimes \psi_{j,l}\right|_{H_0^2(\Omega)}, k, l = 1, \ldots, 2^j \right\} \end{aligned}$$

where $\Omega = [0,1]^2$. A wavelet basis and a multi-scale wavelet basis of the space $H_0^2(\Omega)$ are defined as

$$\Psi_s^{2D} = F_2 \cup \bigcup_{j=2}^{1+s} \left(G_j^1 \cup G_j^2 \cup G_j^3\right), \quad \Psi^{2D} = F_2 \cup \bigcup_{j=2}^{\infty} \left(G_j^1 \cup G_j^2 \cup G_j^3\right). \tag{10}$$

3. Condition numbers of stiffness matrices

In this section, we compare the condition numbers of the stiffness matrices for the biharmonic problem in two dimensions for different wavelet bases. We consider the biharmonic equation

$$\Delta^2 u = f \quad \text{on} \quad \Omega = (0,1)^d, \quad u = \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega. \tag{11}$$

Let $\langle \cdot, \cdot \rangle$ denote the standard $L^2(\Omega)$–inner product and Ψ^d be a wavelet basis of $H_0^2(\Omega)$. The variational formulation is $\mathbf{Au} = \mathbf{f}$, where $\mathbf{A} = \left\langle \Delta\Psi^d, \Delta\Psi^d \right\rangle$, $u = \mathbf{u}^T \Psi^d$, and $\mathbf{f} = \left\langle f, \Psi^d \right\rangle$. It is known that then $\operatorname{cond} \mathbf{A} \leq C < \infty$. Since $\mathbf{A}$ is symmetric and positive definite, we have also

$$\operatorname{cond} \mathbf{A}_s \leq C, \qquad \text{where} \;\; \mathbf{A}_s = \left\langle \Delta\Psi_s^d, \Delta\Psi_s^d \right\rangle \tag{12}$$

and Ψ_s^d is a multiscale wavelet basis with s levels of wavelets. The condition numbers of the stiffness matrices $\mathbf{A}_s$ are shown in Table 1. A construction by Jia and Zhao from [8] is denoted as JZ11, a construction from [4] is denoted as CF12, a construction of multiwavelet basis from [10] is denoted as S09 and a wavelet basis defined in Section 2 is denoted as new.

s	N	JZ11	N	CF12	N	S09	N	new
				1D				
1	15	45.9	17	61.2	30	472.0	7	3.5
5	255	45.9	257	66.6	510	640.8	127	4.1
9	4095	45.9	4097	66.7	8190	731.4	2047	4.1
				2D				
1	225	34.0	289	128.1	900	484.4	49	8.5
2	961	34.9	1089	141.3	3844	583.4	225	14.3
3	3969	35.1	4225	212.0	15876	626.9	961	17.5
4	16129	35.3	16641	257.6	64516	653.5	3969	18.2
5	65025	35.5	66049	281.2	260100	673.2	16129	18.4
6	261121	35.8	263169	297.2	1044484	689.4	65025	18.6

Table 1: The condition numbers of the stiffness matrices $\mathbf{A}_s$ of the size $N \times N$ corresponding to multi-scale wavelet bases with s levels of wavelets.

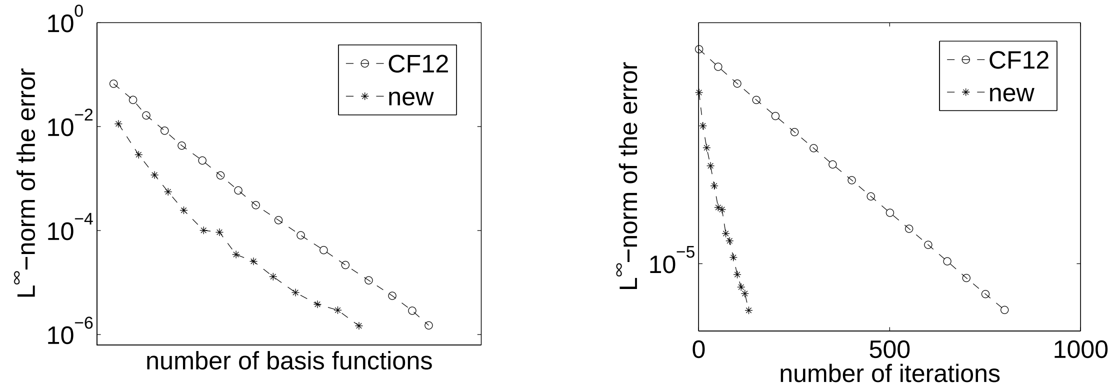


Figure 2: The convergence history for an adaptive wavelet scheme with various wavelet bases.

4. Numerical example

We compare the quantitative behaviour of the adaptive wavelet method with a basis constructed in this paper and a cubic spline-wavelet basis from [4]. In [4] the comparison with other wavelet bases is already done. We consider the equation (11) with a solution u given by

$$u\left(x,y\right)=v\left(x\right)v\left(y\right), \quad v\left(x\right)=x^{2}\left(1-e^{12x-12}\right)^{2}. \tag{13}$$

The solution exhibits a sharp gradient near the point $[1,1]$. We solve the problem by the method designed in [6] with the approximate multiplication of the stiffness matrix with a vector proposed in [2]. The convergence history is shown in Figure 2. In our experiments, the convergence rate, i.e. the slope of the curve, is similar for both bases. However, they significantly differ in the number of basis functions and number of iterations needed to resolve the problem with desired accuracy.

5. Conclusion

We have shown that a wavelet basis from [5] has a short support and the condition number of the corresponding stiffness matrix is smaller than for any other cubic spline wavelet basis adapted to the second-order homogeneous Dirichlet boundary conditions known from literature. It was shown in [8] that Galerkin wavelet method with the wavelet basis from [8] has superb convergence. We have shown that our basis has similar quantitative properties as basis constructed by Jia and Zhao and additionally wavelets have some vanishing wavelet moments. Therefore, unlike basis by Jia and Zhao our basis can be used in adaptive wavelet methods. We implemented adaptive wavelet method with our basis and we have shown that its convergence is improved. However, our basis does not have local duals, therefore in some applications bases from [4, 10] are more appropriate. Furthermore, it should be shown that our basis is indeed a wavelet basis, i.e. that a Riesz basis property (2) is satisfied. The proof and other details can be found in [5].

Acknowledgements

This work was supported by the project ESF "Constitution and improvement of a team for demanding technical computations on parallel computers at TU Liberec" No. CZ.1.07/2.3.00/09.0155 .

References

[1] Černá, D. and Finěk, V.: Cubic spline wavelets satisfying homogeneous boundary conditions for fourth order problems. In: T. E. Simos et al. (Eds.), *Numerical analysis and advanced applications*, AIP Conference Proceedings, vol. 1168, pp. 77–80. American Institute of Physics, New York, 2009.

[2] Černá, D. and Finěk, V.: Approximate multiplication in adaptive wavelet methods. Accepted for publication in Cent. Eur. J. Math., 2012.

[3] Černá, D. and Finěk, V.: Construction of optimally conditioned cubic spline wavelets on the interval. Adv. Comput. Math. **34** (2011), 519–552.

[4] Černá, D. and Finěk, V.: Cubic spline wavelets with complementary boundary conditions. Appl. Math. Comput. **219** (2012), 1853–1865.

[5] Černá, D. and Finěk, V.: Cubic spline wavelets with short support for fourth-order problems. In preparation.

[6] Cohen, A., Dahmen, W. and DeVore, R.: Adaptive wavelet methods II - beyond the elliptic case. Found. Math. **2** (2002), 203–245.

[7] Han, B. and Shen, Z.: Wavelets with short support. SIAM J. Math. Anal. **38** (2006), 530–556.

[8] Jia, R.Q. and Zhao, W.: Riesz bases of wavelets and applications to numerical solution of elliptic equations. Math. Comput. **80** (2011), 1525–1556.

[9] Primbs, M.: New stable biorthogonal spline-wavelets on the interval. Result. Math. **57** (2010), 121–162.

[10] Schneider, A.: Biorthogonal cubic Hermite spline multiwavelets on the interval with complementary boundary conditions. Result. Math. **53** (2009), 407–416.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

NUMERICAL MODELLING OF FLOW IN LOWER URINARY TRACT USING HIGH-RESOLUTION METHODS

Marek Brandner[1], Jiří Egermaier[2], Hana Kopincová[1], Josef Rosenberg[3]

[1] NTIS – New Technologies for Information Society, University of West Bohemia in Pilsen
Univerzitni 8; 306 14, Pilsen; Czech Republic
brandner@kma.zcu.cz,

[2] Department of Mathematics, University of West Bohemia in Pilsen
Univerzitni 8; 306 14, Pilsen; Czech Republic
jirieggy@kma.zcu.cz

[3] Department of Mechanics, University of West Bohemia in Pilsen
Univerzitni 8; 306 14, Pilsen; Czech Republic
rosen@kme.zcu.cz

Abstract

We propose a new numerical scheme based on the finite volumes to simulate the urethra flow based on hyperbolic balance law. Our approach is based on the Riemann solver designed for the augmented quasilinear homogeneous formulation. The scheme has general semidiscrete wave–propagation form and can be extended to arbitrary high order accuracy. The first goal is to construct the scheme, which is well balanced, i.e. maintains not only some special steady states but all steady states which can occur. The second goal is to use this scheme as the component of the complex model of the urinary tract including chemical reactions and contraction of the bladder.

1. Introduction

The voiding is a very complex process. It consists of the transfer of information about the state of the bladder filling in to the spinal cord. Next part is the sending of the action potentials to the smooth muscle cells of the bladder. Even this process is not simple and includes the spreading of the action potential along the nerve axon and the transmission of the mediator (Ach - acetylcholine) in the synapse. The action potential starts the process of the smooth muscle contraction.

The sliding between actin and myosin causing the change of the form (length) of the muscle cell and its stiffness can be observed as a kind of growth and remodeling. This approach described e.g. in [7] is used in this model. To be able to describe the very complex processes in the SMC in the efficient form it is necessary to use the irreversible thermodynamics. This approach was described in [8].

2. Bladder contraction

The whole model of the bladder contraction is described in [6]. It consists of the following parts:

- Model of the time evolution of the Ca^{2+} concentration. The Ca^{2+} intracellular concentration is the main control parameter for the next processes and finally for the smooth muscle contraction. Its increase depends on the flux $J_{agonist}$ of the mediator (in this case acetylcholine) via the nerve synapse.

$$\begin{aligned}
\frac{dc}{dt} &= J_{IP3} - J_{VOCC} + J_{Na/Ca} - J_{SRuptake} + J_{CICR} - J_{extrusion} + J_{leak} \\
&\quad + J_{stretch} \\
\frac{ds}{dt} &= J_{SRuptake} - J_{CICR} - J_{leak} \\
\frac{dv}{dt} &= \gamma(-J_{Na/K} - J_{Cl} - 2J_{VOCC} - J_{Na/Ca} - J_K - J_{stretch}) \qquad (1) \\
\frac{dw}{dt} &= \lambda K_{activate} \\
\frac{dI}{dt} &= J_{agonist} - J_{degrad},
\end{aligned}$$

where the unknown functions represents: $c = c(t)$ calcium concentration in cytoplasm, $s = s(t)$ calcium concentration in ER/SR, $v = v(t)$ membrane tension, $w = w(t)$ probability of opening channels activated by Ca^{2+} and $I = I(t)$ IP3 sensitive reservoirs concentration in cytoplasm. For details and complete description of the functions and parameters see [4].

- Model of the time evolution of the phosphorylation of the light myosin chain. The muscle cell contraction is caused by the relative movement of the myosin and actin filaments. For this it is necessary that the phosphorylation of the mentioned light myosin chain on the heads of the myosin occurs.

$$\begin{aligned}
\frac{dA_M}{dt} &= k_5 A_{M_p} - (k_7 + k_6) A_M, \\
\frac{dA_{M_p}}{dt} &= k_3 M_p + k_6 A_M - (k_4 + k_5) A_{M_p}, \qquad (2) \\
\frac{dM_p}{dt} &= k_1(1 - A_M) + (k_4 - k_1) A_{M_p} - (k_1 + k_2 + k_3) M_p,
\end{aligned}$$

where the unknown functions represent the following: $A_M = A_M(t)$ connected cross-bridges, $A_{M_p} = A_{M_p}(t)$ connected phosphorylated cross-bridges and $M_p = M_p(t)$ unconnected phosphorylated cross-bridges. $k_6 = k_6(c)$, the other terms k_i are constant. For details and complete description of the functions and parameters see [3]. Knowing this process also the time evolution of the ATP consumption (J_{cycl}) can be determined. The ATP (adenosintriphosphate) is the main energy source for the muscle contraction.

$$\frac{dY}{dt} = -Q_Q Y + L J_{cycl}, \tag{3}$$

where $Y = Y(t)$ represents the ATP concentration, Q_Q is the damping parameter and L is the constant.

- Model of the own contraction based on the GRT and the irreversible thermodynamics. The growth and remodelling theory [2] together with the laws of irreversible thermodynamics with internal variables was applied in [8] to describe the mechano-chemical coupling of the smooth muscle cell contraction. The product of the chemical reaction affinity (the ATP hydrolysis) with its rate plays an important role in the discussed model. Further it can be assumed that the rate of the ATP hydrolysis depends on the ATP consumption. The corresponding equations in the non-dimensional form are following:

$$\begin{aligned} \dot{x} &= k_1 \left[\tau - z(x-1)\right], \\ \dot{y} &= \frac{y}{k_2}\left[x\tau - \frac{1}{2}z(x-1)^2 + C'\right], \\ \dot{z} &= \operatorname{sgn}(m)\cdot\left[r - \frac{1}{2}z(x-1)^2\right], \end{aligned} \tag{4}$$

where $x = \frac{l}{l_r}$, $y = \frac{l_r}{l_0}$, l_0 is the initial length of the muscle fibre, l_r its length after stimulation when the fibre is unloaded (s. c. resting length), l the actual length (when the contraction is isometric this is the input value), τ the stress and k is the fibre stiffness, m and r are constants. The non-dimensional values are labeled with the single quote mark. The others symbols are the parameters. The dependence of the single parts of the bladder model is illustrated at the figure 1.

3. Bladder and voiding model

To model the contraction of the bladder during the voiding process we will use the very simple model according [5]. The bladder is modelled as a hollow sphere with the output corresponding to the input into urethra. For the pressure in the bladder the following formula is introduced in [5]

$$p = \frac{V_{sh}}{3V}\cdot\tau, \qquad \tau = \frac{F}{S}, \tag{5}$$

where V_{sh} is the volume of the wall, V the inner volume, S the inner surface, F the force in the muscle cell and τ stress in the muscle fibre, which can be derived as

$$\tau = \frac{\frac{-q}{3\kappa(x\cdot y)^2} + \left[k_1 zy(x-1) + \frac{zyx}{2k_2}(x-1)^2 - \frac{xy}{k_2}C'\right]}{k_1 y + \frac{x^2 y}{k_2}}. \tag{6}$$

This will be putted into the equations for the isotonic contraction.

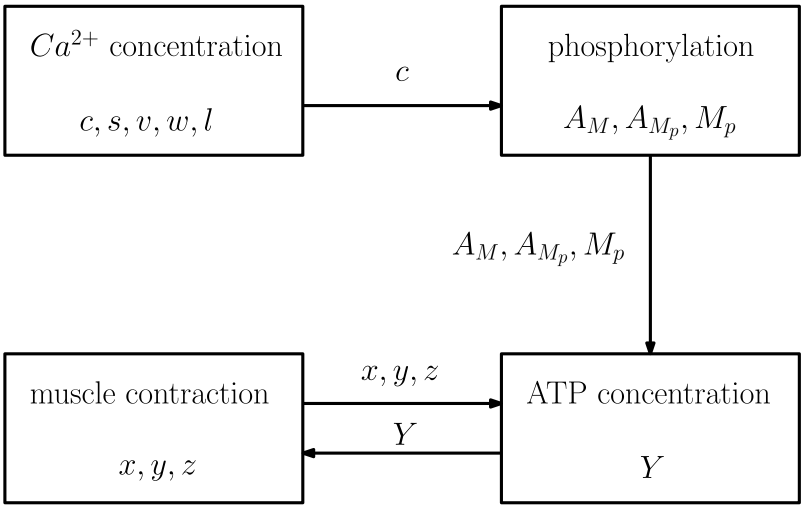


Figure 1: Unknown functions and the dependence of the bladder model parts.

4. Urethra flow

We now briefly introduce a problem describing fluid flow through the elastic tube. In the case of the male urethra, the system based on model in [9] has the following form

$$\begin{aligned} a_t + q_x &= 0, \\ q_t + \left(\tfrac{q^2}{a} + \tfrac{a^2}{2\rho\beta}\right)_x &= \tfrac{a}{\rho}\left(\tfrac{a_0}{\beta}\right)_x + \tfrac{a^2}{2\rho\beta^2}\beta_x - \tfrac{q^2}{4a^2}\sqrt{\tfrac{\pi}{a}}\lambda(Re), \end{aligned} \tag{7}$$

where $a = a(x,t)$ is the unknown cross-section area, $q = q(x,t)$ is the unknown flow rate (we also denote $v = v(x,t)$ as the fluid velocity, $v = \frac{q}{a}$), ρ is the fluid density, $a_0 = a_0(x)$ is the cross-section of the tube under no pressure, $\beta = \beta(x,t)$ is the coefficient describing tube compliance and $\lambda(Re)$ is the Mooney-Darcy friction factor ($\lambda(Re) = 64/Re$ for laminar flow). Re is the Reynolds number. This model contains constitutive relation between the pressure and the cross section of the tube

$$p = \frac{a - a_0}{\beta} + p_e, \tag{8}$$

where p_e is surrounding pressure.

Presented system (7) can be written in the compact matrix form

$$\mathbf{u}_t + [\mathbf{f}(\mathbf{u}, x)]_x = \boldsymbol{\psi}(\mathbf{u}, x), \tag{9}$$

with $\mathbf{u}(x,t)$ being the vector of conserved quantities, $\mathbf{f}(\mathbf{u}, x)$ the flux function and $\boldsymbol{\psi}(\mathbf{u}, x)$ the source term. This relation represents the balance laws. For the following consideration, we reformulate this problem to the nonconservative form.

4.1. Decompositions based on augmented system

The numerical scheme for solving problems (9) can be written in fluctuation form

$$\frac{\partial \mathbf{U_j}}{\partial t} = -\frac{1}{\Delta x}[\mathbf{A}^-(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j+1/2}) + \mathbf{A}(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j-1/2}) + \mathbf{A}^+(\mathbf{U}^-_{j-1/2}, \mathbf{U}^+_{j-1/2})], \quad (10)$$

where $\mathbf{A}^{\pm}(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j+1/2})$ are so called fluctuations. They can be defined by the sum of waves moving to the right or to the left. In what follows, we use the notation $\mathbf{U}^+_{j+1/2}$ and $\mathbf{U}^-_{j+1/2}$ for the approximations of limit values at the points $x_{j+1/2}$. The most common choices are based on the minmod function or ENO and WENO techniques [10].

The our approach is based on the extension of the system (7) by other equations. The advantage of this step is in the conversion of the nonhomogeneous system to the homogeneous quasilinear one. The augmented system can be written in the nonconservative form

$$\begin{bmatrix} a \\ q \\ \phi \\ \frac{a_0}{\beta} \\ \beta \end{bmatrix}_t + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{q^2}{a^2} + \frac{a}{\rho\beta} & 2\frac{q}{a} & 0 & -\frac{a}{\rho} & -\frac{a^2}{\rho\beta^2} \\ 0 & -\frac{q^2}{a^2} + \frac{a}{\rho\beta} & 2\frac{q}{a} & 2\frac{q}{\rho} & -\frac{aq}{\rho\beta^2} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ q \\ \phi \\ \frac{a_0}{\beta} \\ \beta \end{bmatrix}_x = \mathbf{0}, \quad (11)$$

briefly $\mathbf{w}_t + \mathbf{B}(\mathbf{w})\mathbf{w}_x = \mathbf{0}$, where $\phi = av^2 + \frac{a^2}{2\rho\beta}$.

We have five linearly independent eigenvectors. The approximation is chosen to be able to prove the consistency and provide the stability of the algorithm. In some special cases this scheme is conservative and we can guarantee the positive semidefiniteness, but only under the additional assumptions (see [1]).

The fluctuations are then defined by

$$\begin{aligned} \mathbf{A}^-(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j+1/2}) &= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \sum_{p=1, s^{p,n}_{j+1/2}<0}^{m} \gamma^p_{j+1/2}\mathbf{r}^p_{j+1/2}, \\ \mathbf{A}^+(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j+1/2}) &= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \sum_{p=1, s^{p,n}_{j+1/2}>0}^{m} \gamma^p_{j+1/2}\mathbf{r}^p_{j+1/2}, \quad (12) \\ \mathbf{A}(\mathbf{U}^+_{j-1/2}, \mathbf{U}^-_{j+1/2}) &= \mathbf{f}(\mathbf{U}^-_{j+1/2}) - \mathbf{f}(\mathbf{U}^+_{j-1/2}) - \mathbf{\Psi}(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j-1/2}), \end{aligned}$$

where $\mathbf{\Psi}(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j-1/2})$ is a suitable approximation of the source term and $\mathbf{r}^p_{j+1/2}$ are suitable approximations of the eigenvectors of Jacobi matrix $\mathbf{f}'(\mathbf{u})$.

4.2. Steady states

It is very important to choose such approximation which conserves steady states, if these states occur exactly. Steady states mean $\mathbf{u}_t = \mathbf{0}$, therefore $[\mathbf{f}(\mathbf{u})]_x = \boldsymbol{\psi}(\mathbf{u}, x)$.

The steady state for the augmented system means $\mathbf{B}(\mathbf{w})\mathbf{w}_x = \mathbf{0}$, therefore $\mathbf{w}_x$ is a linear combination of the eigenvectors corresponding to the zero eigenvalues. The discrete form of the vector $\Delta\mathbf{w}$ corresponds to the certain approximation of these eigenvectors. It can be shown that

$$\Delta\begin{bmatrix} A \\ Q \\ \Phi \\ \frac{a_0}{\beta} \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{\bar{A}}{\rho}\frac{1}{\overline{\lambda^1\lambda^2}} \\ 0 \\ \frac{\bar{A}}{\rho}\frac{\widetilde{\lambda^1\lambda^2}}{\overline{\lambda^1\lambda^2}} \\ 1 \\ 0 \end{bmatrix} \Delta\left(\frac{a_0}{\beta}\right) + \begin{bmatrix} \frac{\bar{A}^2}{\rho\beta_{j+1}\beta_j}\frac{1}{\overline{\lambda^1\lambda^2}} \\ 0 \\ \frac{\bar{A}^2}{\rho\beta_{j+1}\beta_j}\frac{\widetilde{\lambda^1\lambda^2}}{\overline{\lambda^1\lambda^2}} - \frac{\tilde{A}^2}{2\rho\beta_{j+1}\beta_j} \\ 0 \\ 1 \end{bmatrix} \Delta\beta, \qquad (13)$$

where for j-th cell $\Delta(.) = (.)_{j+1} - (.)_j, \bar{A} = \frac{A_j+A_{j+1}}{2}, \bar{\beta} = \frac{\beta_j+\beta_{j+1}}{2}$, $\tilde{A}^2 = \frac{A_j^2+A_{j+1}^2}{2}$, $\tilde{V}^2 = |V_jV_{j+1}|$, $\bar{V}^2 = \left(\frac{V_j+V_{j+1}}{2}\right)^2$, $\widetilde{\lambda^1\lambda^2} = -\tilde{V}^2 + \frac{\bar{A}\bar{\beta}}{\rho\beta_{j+1}\beta_j}$, and $\overline{\lambda^1\lambda^2} = -\bar{V}^2 + \frac{\bar{A}\bar{\beta}}{\rho\beta_{j+1}\beta_j}$.

We use vectors on the RHS of (13) as consistent approximation of the fourth and fifth eigenvectors of the matrix $\mathbf{B}(\mathbf{w})$. The fluctuations (12) are defined with these vectors and the approximation of the source term is defined by the third line in (13)

$$\mathbf{\Psi}(\mathbf{U}^-_{j+1/2}, \mathbf{U}^+_{j-1/2}) = \frac{\bar{A}}{\rho}\frac{\widetilde{\lambda^1\lambda^2}}{\overline{\lambda^1\lambda^2}}\Delta\left(\frac{a_0}{\beta}\right) + \frac{\bar{A}^2}{\rho\beta_{j+1}\beta_j}\frac{\widetilde{\lambda^1\lambda^2}}{\overline{\lambda^1\lambda^2}} - \frac{\tilde{A}^2}{2\rho\beta_{j+1}\beta_j}\Delta\beta, \qquad (14)$$

where the values $(.)_j$ and $(.)_{j+1}$ should be replaced by their appropriate reconstructed values $(.)^+_{j-1/2}$ and $(.)^-_{j+1/2}$.

5. Complex model of the bladder and the urethra

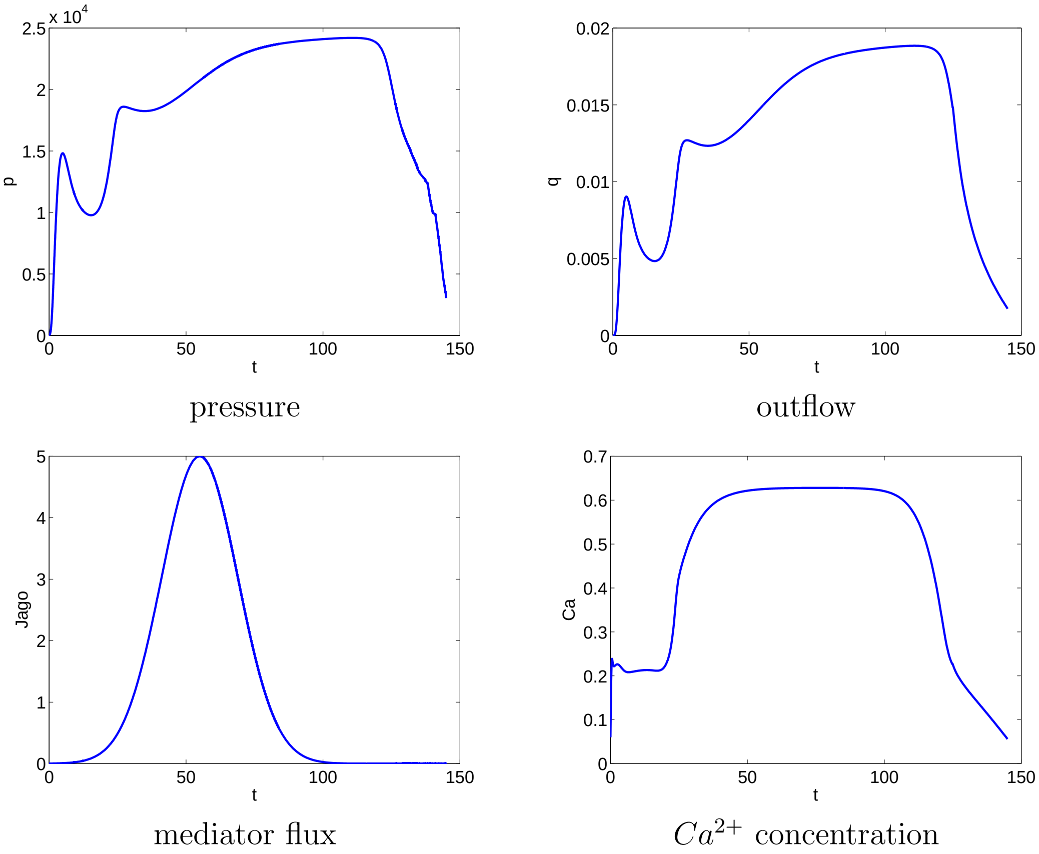
The whole voiding model consists of the detrusor smooth muscle cell model and the model of the urethra flow. It is described by the system of following ordinary differential equations:

- 12 equations describing the bladder model and the detrusor contraction during voiding - the systems (1), (2) and (4).
- $2J$ equations of urethra flow, where J is the number of finite volumes which divide the urethra region

The connection between the detrusor model and urethra flow is implemented by the relation (6) and the constitutive relation (8). The outflow of the bladder is the same as the flow rate in the first finite volume of the urethra region. So the pressure of the bladder is dependent on the flow rate in the tube (6). The cross-section in the first finite volume of the urethra region is then given by the constitutive relation (8). From the view of urethra flow, the inflow boundary condition consists of the given cross-section and extrapolation of the flow rate from the urethra region.

6. Numerical experiment including the complex model of lower urinary tract

Now we present numerical experiment based on the system of differential equations described detrusor smooth muscle cell model (12 equations) and urethral flow



pressure

outflow

mediator flux

Ca^{2+} concentration

Figure 2: Time evolution of the quantities at the bladder neck.

(30 equations). The parameters used in this experiment are the same as in [6]. The figures 2 illustrate time evolution of the quantities at the bladder neck.

For the simplicity the precious modelling of the synapse is neglected and the mediator flux $J_{agonist}$ is chosen - see Fig. 2. The IC units are used although in the medical paper are used for intravesical pressure cm H_2O (1 cm H_2O = 0.1 kPa) and for the outflow ml/s. The concentration is measured in μM where M = mol/l.

7. Conclusion

We presented the complex model of the lower part of the urinary tract. A simple bladder model and the detrusor contraction model were developed during voiding together with the detailed model of urethra flow. The urethra flow was described by the high-resolution positive semidefiniteness method, which preserves general steady states. For the practical application the identification of the parameters is necessary.

Acknowledgements

This work was supported by the European Regional Development Fund (ERDF), project NTIS - New Technologies for Information Society, European Centre of Excellence, CZ.1.05/1.1.00/02.0090 and the project SGS-2010-077 Support of Biomechanics at the Faculty of Applied Sciences, University of West Bohemia in Pilsen.

References

[1] Brandner, M., Egermaier, J., and Kopincová, H.: Augmented Riemann solver for urethra flow modelling. Math. Comput. Simulations **80** (2009), 1222–1231.

[2] Dicarlo, A. and Quiligotti, S.: Growth and balance. Mech. Res. Comm. **29** (2002).

[3] Hai, C. M. and Murphy, R. A.: Adenosine 5'-triphosphate consumption by smooth muscle as predicted by the coupled four-state crossbridge model. Biophysical Journal **61** (1992), 530–541.

[4] Koenigsberger, M., Sauser, R., Seppey, D., Beny, J. L., and Meister, J. J.: Calcium dynamics and vosomotion in arteries subject to isometric, isobaric and isotonic conditions. Biophysical Journal **95** (2008).

[5] Laforet, J. and Guiraud, D.: Smooth Muscle Model For Functional Electric Stimulation Applications. Proceedings of the 29th Annual International Conference of the IEEE EMBS, August 23–26, Lyon, France, 2007.

[6] Rosenberg, J.: Smooth muscle model applied to bladder. Proceeding of the 4th International conference Modelling of mechanical and mechatronic systems MMaMS, Sept. 20-22, Herlany, Slovakia, 2011.

[7] Rosenberg, J. and Hynčík, L.: Modelling of the influance of the stifness evolution on the behaviour of the Muscle fibre. Human Biomechanics, International Conference, 29.9.-1.10.2008 Praha, Czech Republic, 2008.

[8] Rosenberg, J. and Svobodová, M.: Comments on the thermodynamical background to the growth and remodelling theory applied to the model of muscle fibre contraction. Applied and Computational Mechanics **4** (2010), 101–112.

[9] Stergiopulos, N., Tardy, Y., and Meister, J. J.: Nonlinear separation of forward and backward running waves in elastic conduits. Journal of Biomechanics **26** (1993).

[10] Črnjarič Zič, N., Vukovič, S., and Sopta, L.: Balanced finite volume WENO and central WENO schemes for the shallow water and the open-channel flow equations. Journal of Computational Physics **200** (2004), 512–548.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

A QUADRATIC SPLINE-WAVELET BASIS ON THE INTERVAL

Dana Černá, Václav Finěk, Martina Šimůnková
KMD FP TU Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
dana.cerna@tul.cz, vaclav.finek@tul.cz

Abstract

In signal and image processing as well as in numerical solution of differential equations, wavelets with short support and with vanishing moments are important because they have good approximation properties and enable fast algorithms. A B-spline of order m is a spline function that has minimal support among all compactly supported refinable functions with respect to a given smoothness. And recently, B. Han and Z. Shen constructed Riesz wavelet bases of $L_2(\mathbb{R})$ with m vanishing moments based on B-spline of order m. In our contribution, we present an adaptation of their quadratic spline-wavelets to the interval $[0, 1]$ which preserves vanishing moments.

1. Introduction

Wavelets are a widely accepted tool in signal and image processing as well as in numerical solution of operator equations. In this area, methods based on wavelets are successfully used for preconditioning of large systems of linear equations arising from discretization of elliptic partial differential equations, sparse representations of some types of operators and adaptive solving of operator equations. The performance of these methods strongly depends on the choice of a wavelet basis, in particular on its condition number.

Wavelet bases on a general domain are usually constructed in the following way: Wavelets on the real line are adapted to the interval and then by tensor product technique to the n-dimensional cube. Finally by splitting the domain into subdomains which are images of $(0, 1)^n$ under appropriate parametric mappings one can obtain wavelet bases on a fairly general domain. Thus, the properties of wavelet basis on the interval are important for the properties of resulting bases on general domains.

Here, we focus on quadratic spline-wavelets and we construct well-conditioned interval spline-wavelet bases. From the viewpoint of numerical stability, ideal wavelet bases are orthogonal ones. However, they are usually avoided mainly due to the lack of smoothness and their large support. Natural generalization of orthogonal wavelets are biorthogonal wavelets, but their construction and implementation is relatively complicated and wavelets usually have larger support than scaling functions. For more details see for instance [1]. In recent years, there appeared some interesting constructions of biorthogonal wavelets with globally supported dual wavelets [5, 7]. This seems not to cause any problem in numerical solution of linear PDEs because dual functions are not directly used. And recently, B. Han and Z. Shen [6] constructed a Riesz wavelet bases of $L_2(\mathbb{R})$ with m vanishing moments based on B-spline of order m. In our contribution, we present an adaptation of quadratic spline-wavelets proposed in [6] to the interval $[0,1]$ which preserves vanishing moments and compare their properties with quadratic spline wavelets constructed in [1].

2. B-splines

We use a scaling basis based on quadratic B-splines employed for example in [1, 2], because they are well-conditioned and can be easily adapted to the bounded interval by employing multiple knots at the endpoints. Let N be the desired order of polynomial exactness of scaling basis, $j \in \mathbb{N}_0$ and let $\mathbf{t}^j = \left(t_k^j\right)_{k=-N+1}^{2^j+N-1}$ be a Schoenberg sequence of knots defined by

$$\begin{aligned} t_k^j &:= 0, \quad && k = -N+1, \ldots, 0, \\ t_k^j &:= \frac{k}{2^j}, && k = 1, \ldots, 2^j - 1, \\ t_k^j &:= 1, && k = 2^j, \ldots, 2^j + N - 1. \end{aligned}$$

The corresponding B-splines of order N are then defined by

$$B_{k,N}^j\left(x\right) := \left(t_{k+N}^j - t_k^j\right)\left[t_k^j, \ldots, t_{k+N}^j\right]\left(t - x\right)_+^{N-1}, \quad x \in [0,1],$$

where $(x)_+ := \max\{0, x\}$. The symbol $[t_k, \ldots t_{k+N}]\, f(t)$ is the N-th divided difference of f which is recursively defined as

$$\begin{aligned} \left[t_k, \ldots, t_{k+N}\right] f(t) &= \frac{\left[t_{k+1}, \ldots, t_{k+N}\right] f(t) - \left[t_k, \ldots, t_{k+N-1}\right] f(t)}{t_{k+N} - t_k} \quad && \text{if} \quad t_k \neq t_{k+N}, \\ &= \frac{f^{(N)}\left(t_k\right)}{N!} && \text{if} \quad t_k = t_{k+N}, \end{aligned}$$

with $[t_k]\, f(t) = f\left(t_k\right)$. Then, we define the set $\Phi_j = \{\phi_{j,k}, k = -N+1, \ldots, 2^j - 1\}$ of scaling functions where

$$\phi_{j,k} = 2^{j/2} B_{k,N}^j, \quad k = -N+1, \ldots, 2^j - 1, \quad j \geq 0.$$

Thus, there are $2^j - N + 1$ inner scaling functions and $N - 1$ functions at each boundary. The functions $\phi_{j,-N+1}$ and $\phi_{j,2^j-1}$ are the only functions which do not vanish at the boundaries. Therefore, scaling bases satisfying homogeneous Dirichlet boundary conditions are given by

$$\Phi_j^B = \left\{\phi_{j,k}, k = -N+2, \ldots, 2^j - 2\right\}.$$

Inner scaling functions are translations and dilations of one function ϕ corresponding to the primal scaling function constructed by Cohen, Daubechies and Feauveau in [4]. In the case of a quadratic spline-wavelet basis, there is only one boundary scaling function at each boundary. Specifically, the quadratic spline function $\phi(x)$ is defined by

$$\phi(x) = \begin{cases} \frac{x^2}{2} & x \in [0,1], \\ -x^2 + 3x - \frac{3}{2} & x \in [1,2], \\ \frac{x^2}{2} - 3x + \frac{9}{2} & x \in [2,3], \\ 0 & \text{otherwise.} \end{cases}$$

The left boundary function $\phi_B(x)$ is defined by

$$\phi_B(x) = \begin{cases} -\frac{3x^2}{2} + 2x & x \in [0,1], \\ \frac{x^2}{2} - 2x + 2 & x \in [1,2], \\ 0 & \text{otherwise,} \end{cases}$$

and the corresponding right boundary function is symmetrical with respect to the point 3/2. Above scaling functions satisfy the following refinement equations:

$$\phi(x) = \frac{1}{4}\phi(2x) + \frac{3}{4}\phi(2x-1) + \frac{3}{4}\phi(2x-2) + \frac{1}{2}\phi(2x-3),$$

and

$$\phi_B(x) = \frac{1}{2}\phi_B(2x) + \frac{3}{4}\phi(2x) + \frac{1}{4}\phi(2x-1),$$

respectively.

3. Wavelets

In many applications, it is important not only to have wavelets with short support, with vanishing moments but also with a small condition number. Such wavelets should be as close as possible to some orthonormal wavelets or tight frames, for a given order of regularity or vanishing moments. However, construction of optimally conditioned wavelet bases is still an open question. To construct a compactly supported wavelet, one usually starts with a compactly supported refinable function ϕ with stable shifts. Recall that the shifts of a function ϕ are stable if the sequence formed by whole-number shifts of the function ϕ is a Riesz sequence. Then a compactly supported wavelet is obtained by selecting some finite linear combination of these shifts. For further details on this concept, we refer to [3, 8].

While compactly supported refinable functions with stable shifts can be constructed relatively easily, the construction of compactly supported wavelets generated by B-splines is not straightforward. In [6], Riesz wavelet bases of $L_2(\mathbb{R})$ with m vanishing moments based on B-spline of order m have been proposed. Their wavelets are the shortest supported wavelets of regularity $m-1/2$ with m vanishing moments.

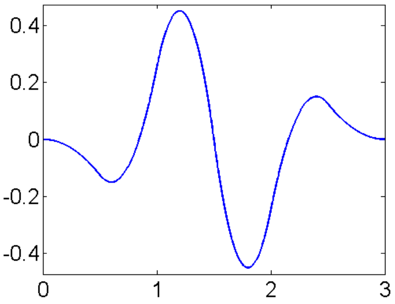


Figure 1: The quadratic wavelet proposed by B. Han and Z. Shen.

The quadratic spline-wavelet constructed by B. Han and Z. Shen is then given by

$$\psi(x) = -\frac{1}{4}\,\phi\,(2x) + \frac{3}{4}\,\phi\,(2x-1) - \frac{3}{4}\,\phi\,(2x-2) + \frac{1}{4}\,\phi\,(2x-3).$$

Its graph is depicted in Figure 1. Now, we would like to adapt it to homogeneous Dirichlet boundary conditions and to keep the number of vanishing moments. First of all, it is not possible to construct boundary wavelets with the same number of vanishing moments as inner wavelets have, and with the same length of support as boundary scaling functions have. They should be supported at least in the interval $[0, 5/2]$. We construct here a boundary wavelet prescribing three vanishing moments, the support in the interval $[0, 5/2]$, homogeneous Dirichlet boundary conditions and finally, it should be from the space spanned by $\{\phi_B(2x),\, \phi(2x-k) \,:\, k \in \mathbb{N}_0\}$. The arising wavelet is then given by these conditions up to multiplication by a constant and is determined by

$$\psi_B(x) = -\frac{5}{2}\,\phi_B\,(2x) + \frac{47}{12}\,\phi\,(2x) - \frac{13}{4}\,\phi\,(2x-1) + \phi\,(2x-2).$$

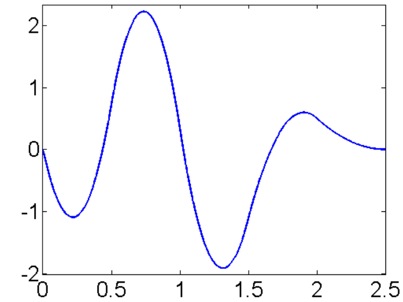


Figure 2: The constructed boundary wavelet.

4. Properties of constructed basis

In this section, we compare selected properties of wavelets introduced in the previous section with wavelets proposed in [1]. We will look at the condition number and the number of nonzero elements for the stiffness matrix corresponding to the equation $u'' = f$ with the Dirichlet boundary conditions $u(0) = u(1) = 0$. We use here also the standard wavelet preconditioning consisting in normalizing all basis function with respect to the arising bilinear form. Further, we solve the above problem corresponding to the exact solution $u = x(1-e^{50x-50})$ which exhibits a steep gradient near the point 1. Results are summarized in Table 1. NZ is the number of nonzero elements in stiffness matrices, COND represents the condition number of diagonally preconditioned stiffness matrices. Achieved approximation error was the same for both bases.

		The proposed basis		CF	
n	$\|\|u_n - u\|\|_{L_2}$	NZ	COND	NZ	COND
8	5.9e-02	58	8.9	-	-
16	1.6e-02	200	10.1	174	12.2
32	2.6e-03	530	10.6	622	12.6
64	3.1e-04	1268	10.8	1822	12.7
128	3.7e-05	2846	10.9	4510	12.8
256	4.5e-06	6128	10.9	10254	12.9
512	5.6e-07	12842	10.9	22190	12.9
1024	7.0e-08	26444	11.0	46590	12.9
2048	8.8e-09	53846	11.0	95998	12.9
4096	1.1e-09	108885	11.0	195502	12.9

Table 1: Obtained numerical results.

5. Conclusion

In this contribution, we proposed new wavelets based on quadratic splines. Due to the shorter support of proposed wavelets, stiffness matrices are sparser than for any known quadratic basis with compactly supported dual wavelets. Moreover, they are slightly better conditioned. Our future aim is to prove that the proposed basis is a Riesz basis and to construct higher order spline-wavelet bases with shorter support than any biorthogonal bases with compactly supported dual wavelets have.

Acknowledgements

This work has been supported by the project ESF No. CZ.1.07/2.3.00/09.0155 “Constitution and improvement of a team for demanding technical computations on parallel computers at TU Liberec”.

References

[1] Černá, D. and Finěk, V.: Construction of optimally conditioned cubic spline wavelets on the interval. Adv. Comput. Math. **34** (2011), 519–552.

[2] Chui, C. K. and Quak, E.: Wavelets on a bounded interval. In: D. Braess, L. L. Schumaker (Eds.), *Numerical Methods of Approximation Theory*, pp. 53–75. Birkhäuser, 1992.

[3] Chui, C. K.: *An introduction to wavelets.* Academic Press, 1992.

[4] Cohen, A., Daubechies, I. and Feauveau, J.-C.: Biorthogonal bases of compactly supported wavelets. Comm. Pure and Appl. Math. **45** (1992), 485–560.

[5] Dijkema, T. J. and Stevenson, R.: A sparse Laplacian in tensor product wavelet coordinates. Numer. Math. **115** (2010), 433–449.

[6] Han, B. and Shen, Z.: Wavelets with short support. SIAM J. Math. Anal. **38** (2003), 530–556.

[7] Jia, R.-Q. and Liu S.-T.: Wavelet bases of Hermite cubic splines on the interval. Adv. Comput. Math. **25** (2006), 23–39.

[8] Keinert, F.: *Wavelets and multiwavelets.* Chapman & Hall/CRC, 2004.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

ON SELECTION OF INTERFACE WEIGHTS IN DOMAIN DECOMPOSITION METHODS

Marta Čertíková[1], Jakub Šístek[2,1], Pavel Burda[1]
[1] Czech Technical University
Technická 4, Prague, Czech Republic
marta.certikova@fs.cvut.cz, pavel.burda@fs.cvut.cz
[2] Institute of Mathematics AS CR
Žitná 25, Prague, Czech Republic
sistek@math.cas.cz

Abstract

Different choices of the averaging operator within the BDDC method are compared on a series of 2D experiments. Subdomains with irregular interface and with jumps in material coefficients are included into the study. Two new approaches are studied along three standard choices. No approach is shown to be universally superior to others, and the resulting recommendation is that an actual method should be chosen based on properties of the problem.

1. Introduction

In many domain decomposition methods, an important role is played by the operator of averaging of a discontinuous function at the interface between adjacent subdomains. Two standard approaches commonly used in literature are: (i) arithmetic average, based simply on counting number of subdomains at an interface unknown, and (ii) weighted average, with weights derived from diagonal stiffness of subdomain Schur complements with respect to the interface. Its simplification presents approximation of the diagonal of the Schur complement by the diagonal of the original matrix, also known as the *stiffness scaling* [3]. The applicability of the so called *ρ-scaling* (see e.g. [3] or [4] for theoretical analysis) is limited to the case of material coefficients constant on each subdomain, which is not preserved in our examples. It also relies on knowledge of coefficients often not available in the solver. In the case of homogeneous material, it simplifies to arithmetic average. Consequently, it is not analyzed separately in this study.

In this paper, we study performance of these standard choices on a series of two-dimensional numerical experiments with the Poisson equation. These were selected to test the performance on regular and irregular subdomains, and in presence of jumps in material coefficients with different alignment with respect to interface. The

Balancing Domain Decomposition by Constrains (BDDC) method [2] is used for this study. In addition to the standard approaches, two new methods are included – averaging based on a unit jump on the interface described in [1], and a new approach based on a unit load applied on boundary of a subdomain. These approaches are shown to be competitive or even preferable in certain situations.

2. Reduction of the problem to the interface

Consider a boundary value problem with a self-adjoint operator defined on domain $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$. If we discretize the problem by means of the standard finite element method (FEM), we arrive at the solution of a system of linear equations in the matrix form

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{K}$ is a large, sparse, symmetric positive definite (SPD) matrix and $\mathbf{f}$ is a vector of the right-hand side.

Let us decompose domain Ω into N non-overlapping subdomains Ω_i, $i = 1, \dots, N$. Unknowns common to at least two subdomains are called *interface unknowns* and the union of all interface unknowns form the *interface.* Remaining unknowns belong to subdomain *interiors.*

The first step used in many domain decomposition methods including BDDC is the reduction of the problem to the interface. Without loss of generality, suppose that unknowns are ordered so that interior unknowns form the first part and the interface unknowns form the second part of the solution vector, i.e. $\mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathrm{o}} & \widehat{\mathbf{u}} \end{bmatrix}^T$, where $\mathbf{u}_{\mathrm{o}}$ stands for all interior unknowns and $\widehat{\mathbf{u}}$ for unknowns at the interface. System (1) can now be formally rewritten to the block form

$$\begin{bmatrix} \mathbf{K}_{\mathrm{oo}} & \mathbf{K}_{\mathrm{or}} \\ \mathbf{K}_{\mathrm{ro}} & \mathbf{K}_{\mathrm{rr}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathrm{o}} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{o}} \\ \widehat{\mathbf{f}} \end{bmatrix}. \tag{2}$$

The hat symbol ($\widehat{\ }$) is used to denote global interface quantities. If we suppose the interior unknowns are ordered subdomain after subdomain, then the submatrix $\mathbf{K}_{\mathrm{oo}}$ is block diagonal with each diagonal block corresponding to one subdomain.

After eliminating all the interior unknowns from (2), we arrive at the *Schur complement problem* for the interface unknowns

$$\widehat{\mathbf{S}}\,\widehat{\mathbf{u}} = \widehat{\mathbf{g}}, \tag{3}$$

where $\widehat{\mathbf{S}} = \mathbf{K}_{\mathrm{rr}} - \mathbf{K}_{\mathrm{ro}}\mathbf{K}_{\mathrm{oo}}^{-1}\mathbf{K}_{\mathrm{or}}$ is the *Schur complement* of (2) with respect to interface and $\widehat{\mathbf{g}} = \widehat{\mathbf{f}} - \mathbf{K}_{\mathrm{ro}}\mathbf{K}_{\mathrm{oo}}^{-1}\mathbf{f}_{\mathrm{o}}$ is sometimes called *condensed right-hand side.* Interior unknowns $\mathbf{u}_{\mathrm{o}}$ are determined by interface unknowns $\widehat{\mathbf{u}}$ via the system of equations $\mathbf{K}_{\mathrm{oo}}\mathbf{u}_{\mathrm{o}} = \mathbf{f}_{\mathrm{o}} - \mathbf{K}_{\mathrm{or}}\widehat{\mathbf{u}}$, which represents N independent subdomain problems with Dirichlet boundary condition prescribed on the interface and can be solved in parallel. The main objective represents the solution of problem (3), which is solved by the preconditioned conjugate gradient method (PCG).

3. Primal DD methods and BDDC

Primal DD methods can be viewed as preconditioners for problem (3), when it is solved by the PCG method. In every iteration of the PCG method, a preconditioned residual $\mathbf{M}\widehat{\mathbf{r}}$ is computed, where $\widehat{\mathbf{r}}$ is the residual. The action of $\mathbf{M}$ is realized by one step of the DD method.

The main idea of the primal DD substructuring methods of Neumann-Neumann type can be expressed as splitting the given residual of the PCG method to subdomains, solving subdomain problems and projecting the result back to the global domain. The primal preconditioner can be written as

$$M = ES^{-1}E^T \,, \tag{4}$$

where operator E^T represents splitting of the residual to subdomains, S^{-1} stands for solution of subdomain problems, and E represents projection of subdomain solutions back to the global problem by some averaging [5]. In the case some subdomains are 'floating', i.e. do not touch a part of boundary with Dirichlet boundary conditions, S is only positive semidefinite, and a generalized inverse S^+ may be needed in (4). The condition number κ of the preconditioned operator $M\widehat{S}$ is bounded by

$$\kappa \leq ||RE||_S^2 \,, \tag{5}$$

where operator R splits the global interface into subdomains and the energetic norm on the right-hand side is defined by the scalar product as $||u||_S^2 = \langle Su, u\rangle$. The relationship (5) was proved in [5] assuming that $ER = I$, which means that if the problem is split into subdomains and then projected back to the whole domain, the original problem is obtained.

If we used independent subdomain problems only (no continuity conditions across the interface), the operator S would be expressed by a block diagonal matrix $\mathbf{S}$ with diagonal blocks representing local Schur complements on subdomains. Relationship between global and local problems can be expressed in matrix form as $\widehat{\mathbf{S}} = \mathbf{R}^{\mathrm{T}}\mathbf{S}\mathbf{R}$.

The main idea of the BDDC method ([2]) is to introduce a global *coarse problem* in order to achieve better preconditioning and to fix 'floating subdomains' by making their local Schur complements invertible. The matrix $\mathbf{S}$ is then positive definite, but it is not block diagonal any more, R now represents splitting of the global interface into subdomains (outside of the coarse unknowns), and E^T distributes residual among neighbouring subdomains only in those interface unknowns which are not coarse. Thus in BDDC, only part of the global residual is split into subdomains; residual at the coarse unknowns is left undivided – it is processed by the global coarse problem.

4. Choice of the averaging operator E

Three standard choices of the averaging operator E recommended already in [2] are (i) the arithmetic average, or weighted average with weights at interface nodes given (ii) by the ratio of the corresponding diagonal entries of the local and global

Schur complement, or (iii) by the ratio of the corresponding diagonal entries of the local and global system matrix $\mathbf{K}$. These choices are denoted here as *aa* (*a*rithmetic *a*verage), *ds* (*d*iagonal of *S*chur complement) and *dk* (*d*iagonal of $\mathbf{K}$), respectively. Method *dk* can be regarded as an approximation of method *ds*, if Schur complements are not computed explicitly.

We try to improve convergence of the BDDC method by choosing some more efficient weights. One of the proposed methods is to choose operator E so that it approximately minimizes the energy norm of the projection RE from estimate (5) for some suitable test vectors representing jumps across the interface. The method, described in more detail in [1], is denoted here as *uj* (*u*nit *j*umps). Here we numerically test just one choice of the test vectors: for every common face of two subdomains, one (local) test vector consisting of ones in the nodes belonging to the face and zeros elsewhere was chosen, corresponding to unit jump. Such choice results in the same weight for every node at the whole face. This, in a sense, makes this method similar to arithmetic average, where also just one weight is used for every node at the face (equal to 0.5).

The second proposed method, denoted as *ul* (*u*nit *l*oads), tries to exploit information of different values of local solution at corresponding interface nodes caused by constant (unit) load at the local interface.

Computation of the weights at interface nodes

For the sake of clarity, formulas are presented for the 2D case, where an interface node is either coarse (so there is no division into subdomains), or it belongs to a face (i.e. to exactly two adjacent subdomains). We also assume one degree of freedom per node, so that numbering of nodes and degrees of freedom coincide. It is straightforward to generalize these methods for 3D cases and more degrees of freedom at a node.

Notation for interface nodes:

j – number of the node in numbering with regard to interface
i – global number of the j-th node on interface
w_j^m – weight at the j-th node at the interface corresponding to the m-th subdomain

Formulas for individual methods:

$$aa: \qquad w_j^m = \frac{1}{2}$$

$$ds: \qquad w_j^m = \frac{s_{pp}^m}{s_{jj}}$$

$$dk: \qquad w_j^m = \frac{k_{qq}^m}{k_{ii}}$$

$$uj: \qquad w_j^m = \frac{\mathbf{d}^T\mathbf{S}^m\mathbf{d}}{\mathbf{d}^T\widehat{\mathbf{S}}\mathbf{d}}$$

$$ul: \qquad w_j^m = \frac{\mathbf{v}^m(j)}{\mathbf{v}^m(j)+\mathbf{v}^n(j)}$$

where:

s_{jj} – diagonal entry of the global Schur complement $\widehat{\mathbf{S}}$

s^m_{pp} – corresponding diagonal entry of the local Schur complement for the m-th subdomain; p is a local number (at the interface of the m-th subdomain) of the j-th node at the (global) interface

k_{ii} – diagonal entry of the (global) system matrix $\mathbf{K}$

k^m_{qq} – corresponding diagonal entry of the local matrix for the m-th subdomain; q is a local number (at the m-th subdomain) of the i-th node (in global numbering)

$\mathbf{d}$ – test vector equal to ones at the face which the j-th node belongs to and zeros otherwise (representing jump at that face)

$\mathbf{S}^m$ – local Schur complement for the m-th subdomain

$\mathbf{v}^m$, $\mathbf{v}^n$ – vectors of solution of the local (subdomain) Schur complement problems with zero values at coarse nodes and the right-hand side equal to one at every interface node that is not coarse, at the m-th and n-th subdomain respectively, where the n-th and m-th subdomain have common face which the j-th node belongs to.

5. Numerical results

The 2D problem of stationary heat conduction (Poisson equation) on a rectangular domain was used for testing. It was discretized by 59 x 59 bilinear finite elements of the same size and shape.

We compared two different divisions into subdomains: rectangular subdomains (Figure 1 left), as the usual choice for rectangular domain, and irregular subdomains (Figure 1 right), typical for domains with irregular shape or when some tool for automatic division into subdomains is used. For the coarse space, just the crosspoints were used. Both homogeneous and nonhomogeneous materials were tested. The nonhomogenity was given by a 1:100 jump in conductivity. Nine different space arrangement of the jump was used, denoted as problems *p1*–*p9* and depicted in Figure 2 (white color represents the conductivity of 1 and black color represents the conductivity of 100).

Five different methods of weights for averaging between the subdomains in the BDDC method were compared, three standard ones (*aa*, *ds* and *dk*) and two new (*uj*, *ul*), all described in Section 4.

Number of PCG iterations for different methods are summarised for rectangular subdomains in Table 1 and for irregular ones in Table 2. The problem *p0* represents problem with constant conductivity on the whole domain, the problems *p1*–*p9* are problems with different locations of jumps in conductivity depicted in Figure 2. As a convergence criterion, norm of the residual less than 10^{-6} was used.

Condition numbers of the preconditioned systems are presented in Tables 3 and 4, where the row k0 is added with the condition number of Schur complement system without preconditioning. Condition numbers were estimated using ratio of the largest and the smallest eigenvalue computed by Matlab function `eig`.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
aa	14	45	14	48	22	22	43	42	46	42
uj	14	6	14	60	21	23	49	37	49	29
ds	14	6	14	28	23	22	30	26	59	16
dk	14	6	14	28	22	22	31	25	59	16
ul	14	6	15	39	23	23	38	35	60	16

Table 1: Number of iterations of PCG, rectangular subdomains.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
aa	13	51	46	65	35	52	58	54	68	83
uj	14	42	41	77	49	72	55	43	70	14
ds	19	23	28	37	37	55	30	33	50	16
dk	20	23	29	37	37	57	32	34	56	16
ul	15	21	24	54	46	64	47	34	64	15

Table 2: Number of iterations of PCG, irregular subdomains.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
k0	5e2	1e3	3e3	2e3	4e4	2e4	2e4	3e4	4e3	3e3
aa	3.71	255	3.65	83	20	33	59	61	69	83
uj	3.72	1.15	3.22	73	18	30	80	50	39	19
ds	3.71	1.15	3.61	104	20	33	50	51	153	7
dk	3.71	1.15	3.65	105	20	33	53	55	160	7
ul	3.94	1.15	3.83	46	21	33	55	58	40	8

Table 3: Condition number of the preconditioned system, rectangular subdomains.

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
k0	6e2	4e3	3e3	2e3	5e4	3e4	3e4	4e4	6e3	4e3
aa	3.26	73	62	72	50	57	82	82	77	136
uj	3.31	91	49	166	147	157	90	131	116	4
ds	8.23	22	148	79	106	172	128	123	94	6
dk	8.79	22	161	80	114	188	141	132	113	7
ul	3.49	19	34	71	109	98	99	135	82	4

Table 4: Condition number of the preconditioned system, irregular subdomains.

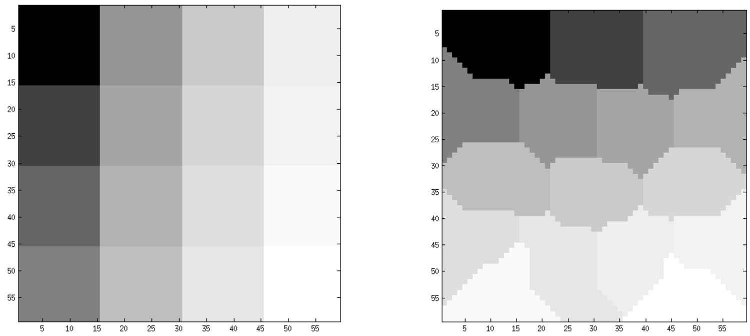


Figure 1: Division into rectangular (left) and irregular (right) subdomains.

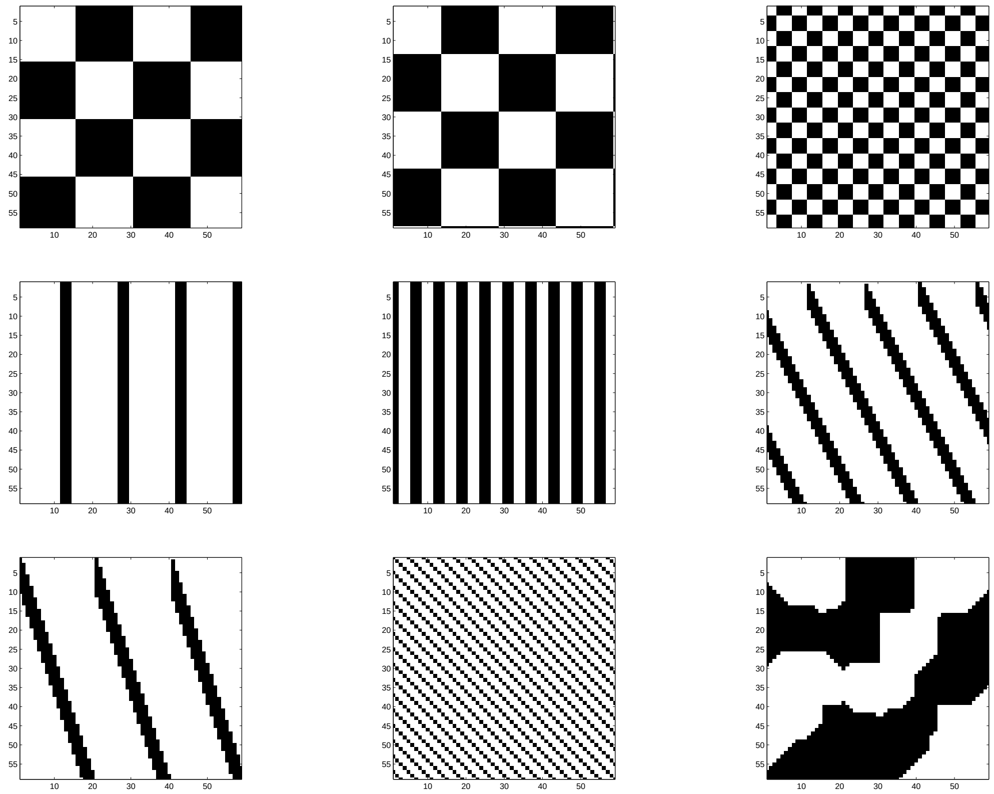


Figure 2: Different nonhomogeneous material properties for problems *p1–p9* (the first row *p1*, *p2*, *p3*, the second row *p4*, *p5*, *p6*, the last row *p7*, *p8*, *p9*).

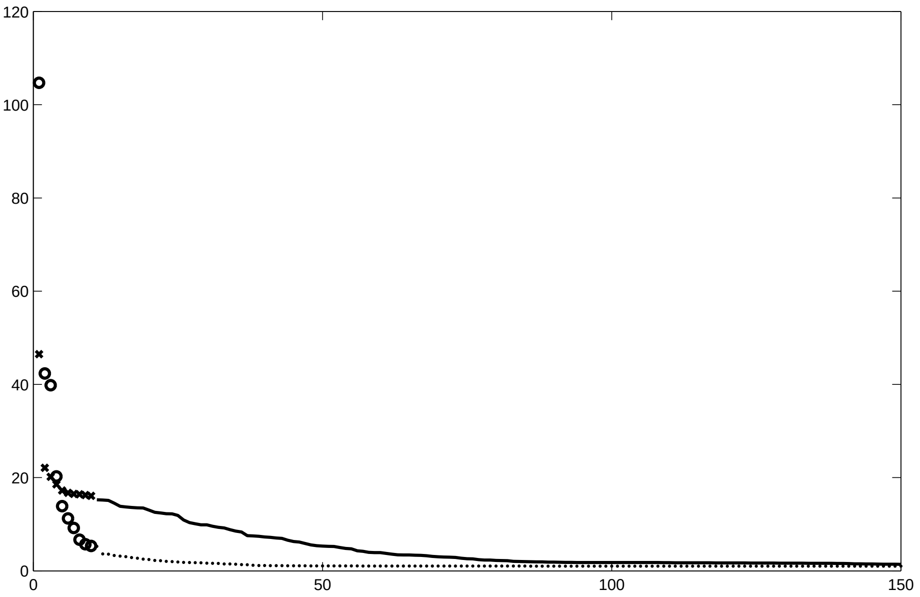


Figure 3: Comparison of the first 150 eigenvalues of $M\widehat{S}$ for methods *dk* ('$\circ$', dotted line), and *ul* ('$\times$', solid line), problem *p3*, regular subdomains.

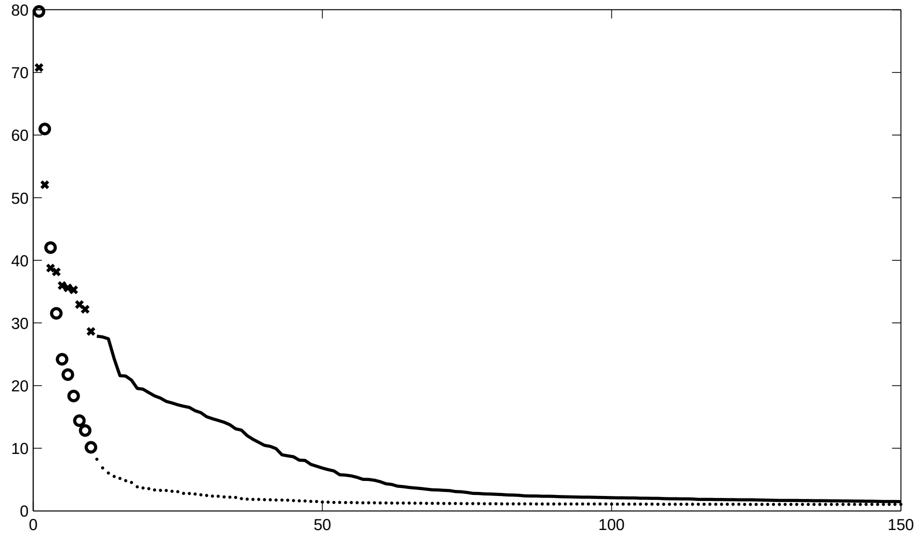


Figure 4: Comparison of the first 150 eigenvalues of $M\widehat{S}$ for methods *dk* ('$\circ$', dotted line), and *ul* ('$\times$', solid line), problem *p3*, irregular subdomains.

6. Conclusions

Our numerical results lead to several observations:

- Arithmetic average (method *aa*) is surprisingly robust even if jumps in coefficients of the equation occur, as long as the jumps do not exactly coincide with the interface (for instance see problem *p2*, where the jumps are shifted only one row of elements from the interface).
- Weights computed as the ratio of the corresponding diagonal entries of local and global Schur complements can be very successfully approximated using the original system matrix K instead of the Schur complements.
- For irregular shape of interface without jumps in coefficients (problem *p0*), using either *ds* or *dk* method instead of arithmetic averages (*aa*) can lead to worse convergence.
- Method *ul* seems to give promising results: it is usually better than arithmetic average, often it is comparable or better than *ds* or *dk*, and it does not seem to have difficulties with irregular shape of interface. However in some cases it leads to worse convergence than all of the standard methods.
- Both proposed methods, *uj* and *ul*, lead very often to lower condition number of the preconditioned system than all standard methods, *aa*, *ds* and *dk*. However, they often give worse convergence results. The reason for this seems to be the distribution of eigenvalues, as illustrated for problem *p3* with rectangular and irregular subdomains in Figures 3 and 4, respectively. For both cases, the first 150 eigenvalues for methods *dk* (circles) and *ul* (crosslines) are compared. For the first few largest eigenvalues, the values for the *dk* method are larger than the values for the *ul* method, which leads to larger condition number (the smallest eigenvalue is allways equal to one). However, following values for the *dk* method quickly drop down and cluster around 1, and they are much lower than the values for the *ul* method. As is well known, clustering of eigenvalues is another important aspect influencing the rate of convergence of PCG.

For equation without jumps in coefficients, the method of choice seems to be the arithmetic averaging. It can lead to very good convergence even if there are jumps in coefficients, except the case where jumps exactly coincide with the interface or some part of it.

If there are jumps in coefficients, the best choice is usually choosing weights as the ratio of corresponding diagonal entries of local and global Schur complements (method *ds*). As these numbers typically are not in hand, a very good substitute is using diagonal entries of local and global original system matrices (method *dk*).

Interesting results are obtained by the method *ul*, which deserves further investigation. Method *uj* does not lead to better convergence than the standard methods.

Acknowledgements

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic under research project LH11004, by the Czech Science Foundation under project 106/08/0403, and by the Academy of Sciences of the Czech Republic through RVO: 67985840.

References

[1] Čertíková, M., Burda, P., Novotný, J., and Šístek, J.: Some remarks on averaging in the BDDC method. In: T. Vejchodský, J. Chleboun, P. Přikryl, K. Segeth, J. Šístek (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 15*, pp. 28–34. IM ASCR, Prague, 2010.

[2] Dohrmann, C. R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25** (2003), 246–258.

[3] Klawonn, A., Rheinbach, O., and Widlund, O. B.: An analysis of a FETI-DP algorithm on irregular subdomains in the plane. SIAM J. Numer. Anal. **46** (2008), 2484–2504.

[4] Mandel, J. and Brezina, M.: Balancing domain decomposition for problems with large jumps in coefficients. Math. Comp. **65** (1996), 1387–1401.

[5] Mandel, J. and Sousedík, B.: BDDC and FETI-DP under minimalist assumptions. Computing **81** (2007), 269–280.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

ON THE OPTIMAL SETTING OF THE *hp*-VERSION OF THE FINITE ELEMENT METHOD

Jan Chleboun

Faculty of Civil Engineering, Czech Technical University
Thákurova 7, 166 29 Prague 6, Czech Republic
chleboun@mat.fsv.cvut.cz

Abstract

The goal of this contribution is to find the optimal finite element space for solving a particular boundary value problem in one spatial dimension. In other words, the optimal use of available degrees of freedom is sought after. This is done through optimizing both the mesh and the polynomial degree of the basis functions. The resulting combinatorial optimization problem is solved in parallel by a Matlab program running on a cluster of multi-core personal computers.

1. Introduction

A finite element mesh is among principal factors that affect the performance of the *h*-version of the finite element method (FEM). An appropriately defined mesh or, to be more correct, a sequence of appropriately defined meshes can accelerate the convergence of the method. Since the FEM projects the exact solution to the mesh-dependent finite element space, the distance between the exact solution and the finite element (FE) space determines the error, that is, the distance between the exact solution and its FE approximation. Various techniques have been proposed to adaptively modify FE meshes and, consequently, FE spaces in order to minimize the error [2, 3, 10].

In the *h*-version of the FEM, however, the polynomials forming the basis of the FE space either remain unchanged during the mesh modification process or only limited increase/decrease of the polynomial degree is allowed. Typically, piecewise linear and quadratic or even cubic functions are considered.

In the *hp*-version of the FEM, both mesh and polynomial degree modifications are supported and low as well as higher order polynomials can be found together in FE spaces, see [5, 6, 8, 11, 12]. Nevertheless, this freedom has its dark side. Unlike the *h*-version of the FEM, where the FE space improvement is mediated solely by adaptive mesh optimization, the mesh as well as the polynomial degree can be adaptively changed in the *hp*-FEM and it is difficult to determine which of the two approaches is more efficient or how to combine them to get best results. We refer to [1, 7, 9, 13] for various algorithms and analyses focusing on one-dimensional boundary value problems (BVPs).

This contribution presents computational results of the optimization of FE spaces that have a fixed dimension. The goal of the optimization is to minimize the difference between a FE solution and the exact solution of a BVP. The difference is measured in the H^1-norm. The results can (a) serve as benchmarks for the performance of adaptive algorithms, and (b) help to evaluate the efficiency of polynomial degree optimization and mesh optimization.

2. Optimization problem

Let $u(x) = 1/(1.25 - x)$ and let f, a, and b be inferred to comply with the following BVP on the interval $[-1, 1]$

$$-u'' + u = f, \tag{1}$$

$$u'(-1) = a, \quad u'(1) = b. \tag{2}$$

Omitting the knowledge of u, we solve (1)–(2) by the FEM: Find $u_{\mathcal{T}_h,p} \in V^{\mathcal{T}_h,p}$ such that

$$\int_{-1}^{1} \left(u'_{\mathcal{T}_h,p} v'_{\mathcal{T}_h,p} + u_{\mathcal{T}_h,p} v_{\mathcal{T}_h,p} \right) \mathrm{d}x = \int_{-1}^{1} f v_{\mathcal{T}_h,p} \, \mathrm{d}x + b v_{\mathcal{T}_h,p}(1) - a v_{\mathcal{T}_h,p}(-1) \tag{3}$$

holds for any $v_{\mathcal{T}_h,p} \in V^{\mathcal{T}_h,p}$. The finite element space $V^{\mathcal{T}_h,p}$ is defined on the mesh $\mathcal{T}_h$ determined by points $-1 = x_0 < x_1 < \cdots < x_m = 1$. If $C([-1, 1])$ denotes the space of continuous functions on $[-1, 1]$ and $P_{d_k}([x_{k-1}, x_k])$ is the space of polynomials on $[x_{k-1}, x_k]$ of degree d_k or less, we have

$$V^{\mathcal{T}_h,p} = \left\{ v_{\mathcal{T}_h,p} \in C([-1, 1]) : \; v_{\mathcal{T}_h,p}|_{[x_{k-1},x_k]} \in P_{d_k}([x_{k-1}, x_k]), \; k = 1, \ldots, m \right\}.$$

The basis functions of $V^{\mathcal{T}_h,p}$ are defined via the Lobatto shape functions (LSFs; see [12]) with their polynomial degree limited to at most 10. Let us note that each LSF of order two and higher is a bubble function because its support comprises only one mesh subinterval.

Various FE spaces can be designed with the same dimension N. To this end, we introduce $p = (d_1, \ldots, d_m)$, m-tuples that describe the polynomial degree distribution over the mesh intervals. By counting the LSFs inclusive of piecewise linear basis functions, we arrive at $N = d_1 + \cdots + d_m + 1$.

Next, let $\mathcal{P}_N$ be the set of all polynomial degree distributions that correspond to N-dimensional FE spaces. As an example, take $N = 5$ and

$$\mathcal{P}_5 = \{(1, 1, 1, 1), (2, 1, 1), (1, 2, 1), (1, 1, 2), (2, 2), (3, 1), (1, 3), (4)\},$$

where $(1, 1, 1, 1)$ represents a FE space with five piecewise linear functions and three unspecified mesh nodes between -1 and 1 (inner nodes), whereas (4) represents the unique FE space formed by quartic polynomials on $[-1, 1]$.

Each $p \in \mathcal{P}_N$ determines a family $\mathcal{M}_p$ of meshes $\mathcal{T}_h$ that, if combined with the polynomial degree distribution p, lead to FE spaces with the dimension N.

As already indicated, we are interested in the minimization of

$$\Phi(p, \mathcal{T}_h) = \|u - u_{\mathcal{T}_h,p}\|_{H^1(-1,1)}$$

where $u_{\mathcal{T}_h,p} \in V^{\mathcal{T}_h,p}$ solves (3). More precisely, if a fixed dimension N is given, we search for $p^0 \in \mathcal{P}_N$ and $\mathcal{T}_h^0$ such that

$$\Phi(p^0, \mathcal{T}_h^0) = \min_{p\in\mathcal{P}_N} \min_{\mathcal{T}_h\in\mathcal{M}_p} \Phi(p, \mathcal{T}_h). \tag{4}$$

Problem (4) was solved in the MATLAB® environment. To avoid mesh degeneration, a minimum distance of mesh nodes was bounded from below by a small positive constant.

The position of mesh nodes was optimized by the MATLAB® Optimization Toolbox™ `fmincon` function designed to search for local minima. Since the goal of the inner minimization in (4) is to find a global minimum, multiple runs of `fmincon` were performed on an initial uniform mesh as well as on a number of initial random meshes.

The computational complexity of problem (4) is rapidly increasing with N. Indeed, $|\mathcal{P}_N|$, the cardinality of $\mathcal{P}_N$, is equal to 2^{N-2} if $N = 3, 4, \ldots, 11$. For $N > 11$, the constraint put on the maximum polynomial degree inhibits the exponential growth of $|\mathcal{P}_N|$, but not strongly. It is $|\mathcal{P}_{14}| = 4088$, for instance.

The inner minimizations are mutually independent for different $p \in \mathcal{P}_N$ and were solved in parallel on a cluster of personal computers with (up to) 200 cores.

3. Results

Let $N = 14$. Figure 1 (left) shows the values $\Phi(p, \mathcal{T}_h)$ where $\mathcal{T}_h$ are uniform (non-optimized) meshes. The numbers on the horizontal axis correspond to the position of a particular p in the sequence of all $p \in \mathcal{P}_{14}$. The dependence of p on its ordinal number cannot be given by a simple formula. Let us only say that, very roughly, the higher the ordinal number, the higher the polynomial degrees in p.

We observe that $\Phi(p, \mathcal{T}_h)$ is rather sensitive to p because the values span from 0.0062 (minimum, $p = (3, 10)$) to 2.895 (maximum, $p = (6, 6, 1)$ or $p = (10, 2, 1)$, for example).

The right part of Figure 1 depicts the histogram of $\Phi(p, \mathcal{T}_h)$ on uniform meshes.

Figure 2 is an analogy to Figure 1; it presents the same type of results for optimized meshes. The dependence on p is clearly visible. The S-shaped patterns correspond to the structure of the ordering of $\mathcal{P}_{14}$. In each pattern, $\Phi(p, \mathcal{T}_h)$ decreases if the higher order polynomials move towards the right-end of the mesh. The first pattern from the left begins with $p = (1, 1, \ldots, 1)$ giving the maximum $\Phi(p, \mathcal{T}_h) = 0.298$ and ends with $p = (1, 2, 2, \ldots, 2)$ and $\Phi(p, \mathcal{T}_h) = 0.075$; ordinal

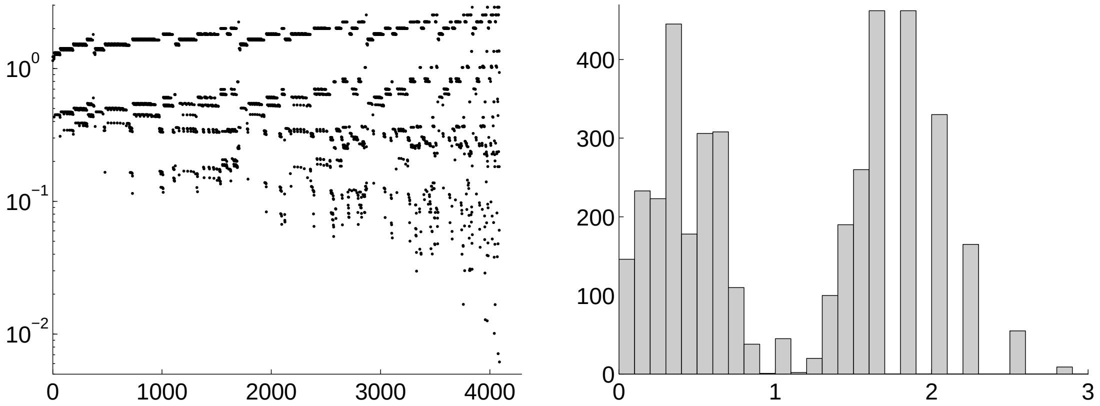


Figure 1: $N = 14$, uniform meshes. Values $\Phi(p, \mathcal{T}_h)$ (left) and the histogram (right).

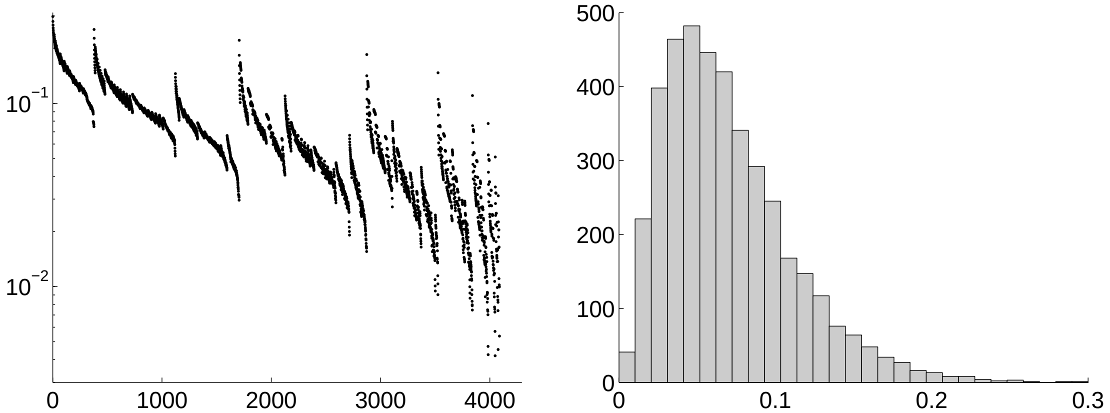


Figure 2: $N = 14$, optimal meshes. Values $\Phi(p, \mathcal{T}_h)$ (left) and the histogram (right).

number 377. The next pattern begins with $p = (3, 1, 1, \ldots, 1)$ and $\Phi(p, \mathcal{T}_h) = 0.253$; ordinal number 378.

By comparing the cluster of minimum and near-to-minimum values in Figure 1 and Figure 2, we also infer that though the exact solution u is not a polynomial, it is sufficiently well approximated by a few higher order polynomials. The minimum value of $\Phi(p, \mathcal{T}_h)$ attained on the optimized meshes is equal to 0.0042 if $p = (5, 8)$. This is not a significant improvement over the uniform meshes.

Although the sensitivity to p is strong in the optimal mesh results, we should not overlook the decrease in Φ. Even for the worst-case p, the error is one order lower if the mesh is optimal. This is not the only evidence that mesh optimization

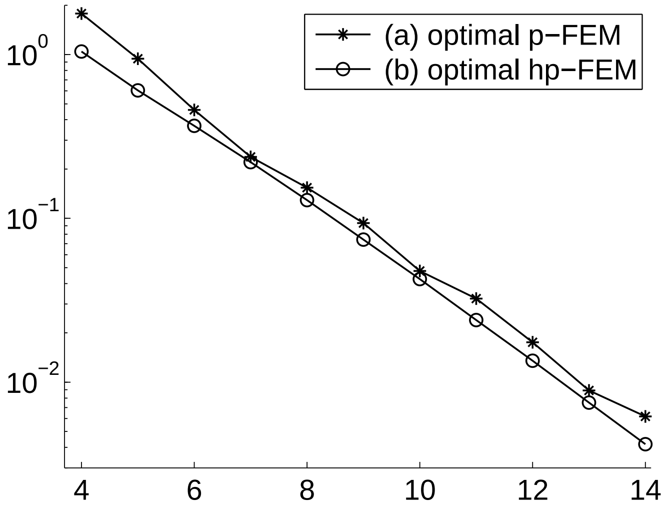


Figure 3: Convergence of the minimum values of Φ if (a) p is optimal and $\mathcal{T}_h$ is uniform; (b) both p and $\mathcal{T}_h$ are optimal. The horizontal axis shows N and the vertical axis shows Φ, the error.

pays off. Let us compare the histograms. Among uniform meshes, only 146 degree distributions guarantee the error less than 0.1; see the first bar in Figure 1 (right). For the optimized meshes, we obtain more than 3300 such degree distributions.

Figure 3 shows the rate of convergence of both optimal p-FEM and optimal hp-FEM. If evaluated through the minimum values of Φ, the difference between the two methods applied to (3) is small. However, one should take into account that there are only a few optimal and almost optimal p distributions on uniform meshes, but significantly more p-$\mathcal{T}_h$ couples can guarantee good performance if the mesh is optimized; consider $0 < \Phi(p, \mathcal{T}_h) \leq 0.05$ and compare Figure 1 and Figure 2. As a consequence, although we strive to optimize both the mesh and p in the hp-FEM, it seems to be advisable to pay somewhat more attention to the former than to the latter. This conclusion agrees with that of [4] where a more detailed analysis of a different BVP is presented.

Acknowledgements

This research was supported by grant No. P105/10/1682 of the Czech Science Foundation. The author thanks both Dr. Milan Pultar for his invaluable help with cluster computing and Dr. Richard (Dick) Haas for hours of fruitful discussions.

References

[1] Babuška, I., Strouboulis, T., and Copps, K.: *hp* optimization of finite element approximations: Analysis of the optimal mesh sequences in one dimension. Comput. Methods Appl. Mech. Eng. **150** (1997), 89–108.

[2] Babuška, I. and Strouboulis, T.: *The finite element methods and its reliability.* Clarendon Press, Oxford, 2001.

[3] Bois, R., Fortin, M., and Fortin, A.: A fully optimal anisotropic mesh adaptation method based on a hierarchical error estimator. Comput. Methods Appl. Mech. Eng. (2012), 12–27.

[4] Chleboun, J. and Solin, P.: On optimal node and polynomial degree distribution in one-dimensional *hp*-FEM. Computing, DOI 10.1007/s00607-012-0232-x, available online.

[5] Demkowicz, L. F.: *Computing with hp-adaptive finite elements. Vol. 1: One- and two-dimensional elliptic and Maxwell problems. With CD-ROM.* Applied Mathematics and Nonlinear Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2007.

[6] Demkowicz, L.F. et al.: *Computing with hp-adaptive finite elements. Vol. II: Frontiers: Three-dimensional elliptic and Maxwell problems with applications.* Applied Mathematics and Nonlinear Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2008.

[7] Dörfler, W. and Heuveline, V.: Convergence of an adaptive *hp* finite element strategy in one space dimension. Appl. Numer. Math. **57** (2007), 1108–1124.

[8] Eibner, T. and Melenk, J.: An adaptive strategy for *hp*-FEM based on testing for analyticity. Comput. Mech. **39** (2007), 575–595.

[9] Gui, W. and Babuška, I.: The *h*, *p* and *h*-*p* versions of the finite element method in 1 dimension. III. The adaptive *h*-*p* version. Numer. Math. **49** (1986), 659–683.

[10] Marcuzzi, F., Cecchi, M., and Venturin, M.: An anisotropic unstructured triangular adaptive mesh algorithm based on error and error gradient information. Math. Comput. Simul. **78** (2008), 645–652.

[11] Schwab, C.: *p- and hp-finite element methods: Theory and applications in solid and fluid mechanics.* Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1998.

[12] Šolín, P., Segeth, K., and Doležel, I.: *Higher-order finite element methods.* Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[13] Wihler, T. P.: An *hp*-adaptive strategy based on continuous Sobolev embeddings. J. Comput. Appl. Math. **235** (2011), 2731–2739.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

INTEGRO-DIFFERENTIAL EQUATIONS WITH TIME-VARYING DELAY

Pavol Chocholatý

Faculty of Mathematics, Physics and Informatics
Mlynská dolina, Bratislava, Slovakia
pavol.chocholaty@fmph.uniba.sk

Abstract

Integro-differential equations with time-varying delay can provide us with realistic models of many real world phenomena. Delayed Lotka-Volterra predator-prey systems arise in ecology. We investigate the numerical solution of a system of two integro-differential equations with time-varying delay and the given initial function. We will present an approach based on q-step methods using quadrature formulas.

1. Introduction

Integro-differential equations (IEs) are one of the most important mathematical tools used in modelling problems of many real world phenomena. Here, we consider the Lotka-Volterra like predator-prey model [1]. This system of two IEs is frequently used to describe the dynamics of biological systems in which two species interact. One is the population of predators of the size $x_1(t)$ and the other is that of preys of the size $x_2(t)$

$$x_1'(t) = \left[c - k_1 x_2(t) - \int_{-\tau}^{0} \alpha_1(x_2(t+s))ds\right] x_1(t)$$

$$x_2'(t) = \left[-c + k_2 x_1(t) - \int_{-\tau}^{0} \alpha_2(x_1(t+s))ds\right] x_2(t)$$

where $x_1'(t)$ and $x_2'(t)$ represent the growth of the two populations with time, c, k_i, α_i are parameters representing the interaction of the two species.

Also, one of the models for human immunodeficiency virus (HIV) in a homogeneously mixed single-gender group with distributed waiting times can be described using IEs, see [3].

So elaboration of numerical methods for IEs is a very important problem. Presently, various specific numerical methods are constructed for solving specific IEs. Most investigations are devoted to numerical methods for systems with discrete delays, see e.g. [2].

The approach described in this article has been applied to numerical solution of integro-differential equations with time-varying delay (IDETVD) .

2. Equations with time-varying delay

Delay differential equations (DDEs) represent the principal form of mathematical models occuring in Ecology. In DDEs, also called functional differential equations or time-delay systems, dependent variables are simultaneously evaluated at more than one value of the independent variable.

The considered DDE Cauchy problem is

$$\begin{aligned} x' &= f(t, x(t+\tau_1), \cdots, x(t+\tau_k)), \quad t \geq t_0, \\ x(t) &= \Psi(t), \quad t \leq t_0 \end{aligned}$$

f is a function with the independent variable t representing time, dependent variable $x(t)$ is a phase vector and $x(t+\tau_j)$, $\tau_j \in < -r_j, 0 >$, $\quad j = 1, 2, \cdots, k$ are the functions characterizing the influence of the pre-history of the phase vector on the dynamics of the system. A class of DDE with constant delay $\tau_j, j = 1, 2, \cdots, k$ is called DDEs with discrete delay. Supposed that delay $\tau_j = \tau_j(t)$ we speak about differential equations with time-varying delay.

Let us consider some of them. The delay logistic equation

$$x'(t) = r(t)x(t)\left(1 - \frac{x(\tau(t))}{K}\right), \quad \tau(t) \leq t$$

describes a delay population model and is known as Hutchinson's equation [2]. One can see that it is insufficient to know the initial value only to define the phase vector $x(t)$. It is also necessary to know an inital function (initial pre-history) $\Psi(t)$. Hence the DDEs are generalizations of the ODEs such that the velocity $x'(t)$ of a process depends also on the pre-history $x(\tau(t))$, $\tau(t) \leq t$.

Delay can also be distributed as in the equation

$$x'(t) = f\left(t, x(t), \int_{\tau(t)}^{0} \alpha(t, s, x(t+s))ds\right).$$

So, the Volterra integro-differential equations

$$x'(t) = f\left(t, x(t), \int_{0}^{t} \beta(t, s, x(s))ds\right)$$

represent a special class of DDEs with distributed delays.

The purpose of this article is to derive a numerical method for the approximate solution of delay differential systems with time-varying delay of the form

$$x'(t) = f\left(t, x(t), x(\tau_1(t)), \int_{\tau(t)}^{0} \chi(t, s, x(t+s))ds\right).$$

In [3], Kim and Pimenov proposed an exact solution to a system of IDETVD

$$x_1'(t) = -\sin(t)x_1(t) + x_1(t - \frac{t}{2}) - \int_{-t/2}^{0} \sin(t+s)x_1(t+s)ds - \mathrm{e}^{\cos(t)} \tag{1}$$

$$x_2'(t) = -\cos(t)x_2(t) + x_2(t - \frac{t}{2}) - \int_{-t/2}^{0} \cos(t+s)x_2(t+s)ds - \mathrm{e}^{\sin(t)} \tag{2}$$

corresponding to an inital function

$$\begin{array}{rcl}\Psi_1(s) & = & \mathrm{e}^{\cos(s)} \\ \Psi_2(s) & = & \mathrm{e}^{\sin(s)}\end{array}, \quad s \leq 0.$$

The solution $(x_1(t), x_2(t))^T, t \in [0, \infty)$ of (1), (2) has the form

$$\begin{array}{rcl}x_1(t) & = & \mathrm{e}^{\cos(t)}, \\ x_2(t) & = & \mathrm{e}^{\sin(t)}.\end{array}$$

Then by considering the maximum absolute errors in the solution at grid points for different choices of step size, we can conclude how further presented approaches produce accurate results in comparison with those exact ones.

3. A numerical approach

The most popular numerical approaches for solving Cauchy problem of ODEs are called finite difference methods. Approximate values are obtained for the solution at a set of grid points $\{t_n : n = 1, 2, \cdots, N\}$ and the approximate value at each point t_{n+1} is obtained by using some of values obtained in previous steps. The best known methods are Euler's methods (explicit, implicit), trapezoidal method, Milne's methods, Adams methods.

Most integrals cannot be evaluated explicitly and with many others it is often faster to integrate them numerically rather than evaluating them exactly. Formulas using such interpolation with evenly spaced grid points are the composite trapezoidal rule and the composite Simpson's rule. These Newton-Cotes formulas can be used to construct a composite method with mentioned methods.

The simplest way how to solve our problem is the combination of the explicit Euler's method with the trapezoidal rule, outlined in the following procedure solving the problem (1) on an equidistant mesh $t_{n+1} - t_n = h$, where we abbreviate $x_1(t)$ by $x(t)$.

First, the trapezoidal rule is defined by applying

$$x(t_{n+1}) = x(t_n) + \frac{h}{2}\Bigg[- \sin(t_n)x(t_n) + x(t_n/2) - \int\limits_{-t_n/2}^{0} \sin(t_n + s)x(t_n + s)ds - \mathrm{e}^{\cos(t_n)}$$

$$- \sin(t_{n+1})x(t_{n+1}) + x(t_{n+1}/2) - \int\limits_{-t_{n+1}/2}^{0} \sin(t_{n+1} + s)x(t_{n+1} + s)ds - \mathrm{e}^{\cos(t_{n+1})}\Bigg]$$

to successive subintervals $[t_n, t_{n+1}]$, where

$$h = 2H,\ t_n = 2kH,\ t_{n+1} = 2(k + 1)H,\ k = 0, 1, 2, \ldots.$$

Hence,

$$x(2(k+1)H) = x(2kH) + H\Bigg[-\sin(2kH)x(2kH) + x(kH)$$
$$-\int_{-kH}^{0} \sin(2kH+s)x(2kH+s)ds - \mathrm{e}^{\cos(2kH)} - \sin(2(k+1)H)x(2(k+1)H)$$
$$+ x((k+1)H) - \int_{-(k+1)H}^{0} \sin(2(k+1)H+s)x(2(k+1)H+s)ds - \mathrm{e}^{\cos(2(k+1)H)}\Bigg]$$

Since

$$\int_{-(k+1)H}^{0} A = \int_{-(k+1)H}^{-kH} A + \int_{-kH}^{0} A$$

and letting

$$I(k) = \int_{-kH}^{0} \sin(2kH+s)x(2kH+s)ds$$
$$I(k+1) = \int_{-(k+1)H}^{0} \sin(2(k+1)H+s)x(2(k+1)H+s)ds$$

we have

$$x(2(k+1)H) = x(2kH) + H\left[-\sin(2kH)x(2kH) + x(kH) - I(k) - \mathrm{e}^{\cos(2kH)} - \right.$$
$$\left. -\sin(2(k+1)H)x(2(k+1)H) + x((k+1)H) - I(k+1) - \mathrm{e}^{\cos(2(k+1)H)}\right]$$

Now, we shall confine our discussion to evaluating $I(k)$ and $I(k+1)$ approximately. For a sufficiently small mesh size H the composite trapezoidal rule gives a good approximation to the integrals

$$I(k) = \sum_{p=0}^{k-1} \frac{H}{2}\big[\sin((k+p)H)x((k+p)H) + \sin((k+p+1)H)x((k+p+1)H)\big]$$
$$I(k+1) = \sum_{p=0}^{k} \frac{H}{2}\big[\sin((k+p+1)H)x((k+p+1)H)$$
$$+ \sin((k+p+2)H)x((k+p+2)H)\big]$$

However, it is possible to obtain finite sums which give better approximations by the same amount of computation. One sees, immediately, that $x(t_{n+1})$ can be computed when t_{n+1} is the even multiple of H. If t_{n+1} is the odd multiple of H then we apply explicit Euler method to the model equation on an equidistant mesh $t_{n+1} - t_n = h$.

Then, the explicit Euler method is defined by applying

$$x(t_{n+1}) = x(t_n) + h\left[-\sin(t_n)x(t_n) + x(t_n/2) - \int_{-t_n/2}^{0} \sin(t_n+s)x(t_n+s)ds - \mathrm{e}^{\cos(t_n)}\right]$$

to successive subintervals $[t_n, t_{n+1}]$, where $h = H, \quad t_n = 2kH, \quad t_{n+1} = (2k+1)H$, $k = 0, 1, 2, \cdots$. This yields

$$x((2k+1)H) = x(2kH) + H\Bigg[-\sin(2kH)x(2kH) + x(kH) \\ - \int_{-kH}^{0} \sin(2kH+s)x(2kH+s)ds - \mathrm{e}^{\cos(2kH)}\Bigg]$$

It can be seen that this formula contains the integral $I(k)$.

Also,

$$x((2k+1)H) = x(2kH) + H\left[-\sin(2kH)x(2kH) + x(kH) - I(k) - \mathrm{e}^{\cos(2kH)}\right].$$

4. Numerical experiments

In order to test the viability of the proposed composite methods and to demonstrate its convergence computationally we have considered several tests with some steps, to assess the convergence property and efficiency of methods mentioned in Section 3.

We divide the time interval $t \in [0, 6.3]$ into N subintervals in order to obtain the approximate values for the solution at the grid points t_n. Here we are only interested in showing the errors of the solution at some grid points. The idea was to calculate the numerical solution by Milne-Simpson's method of 5-th order with the Simpson's rule on an equidistant mesh $t_{n+1} - t_n = h = 0.003$. Table 1 contains the errors in this numerical solution in selected gridpoints.

Numerical and exact results are illustrated in Figure 1 in the time varying plane and in Figure 2 in the phase plane also.

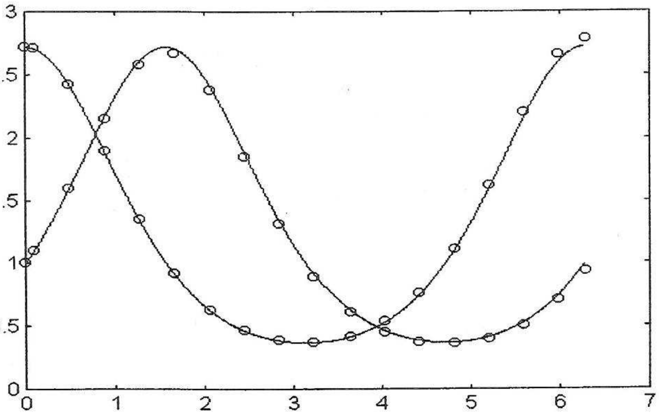


Figure 1: Graph of $x_1(t)$ and $x_2(t)$ versus time.

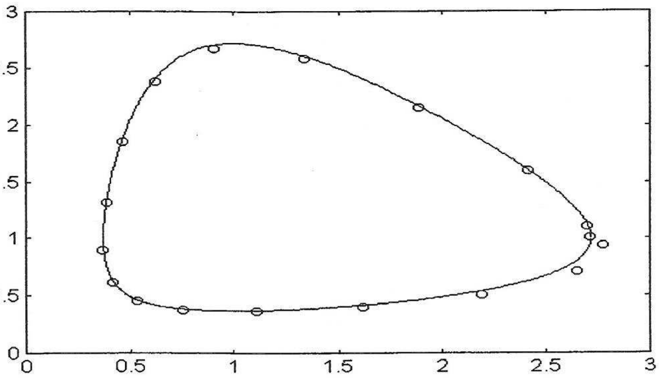


Figure 2: Graph of $x_1(t)$ versus $x_2(t)$.

t	$x_1(t)$	error of $x_1(t)$	$x_2(t)$	error of $x_1(t)$
2.1	0.6035988	0.0052716	2.3707579	0.0185469
4.2	0.6124669	0.0043827	0.4182918	0.0038792
6.3	2.7178976	0.0118452	1.0169558	0.0298354

Table 1: Errors in the numerical solution.

The solid lines indicate the graphs of exact solution $(x_1(t), x_2(t))^T$ with $x_1(0) = \mathrm{e}$, $x_2(0) = 1$, $t \in [0, 6.3]$. Our program begins with the second order trapezoidal formula and the explicit Euler's formula, the accuracy then increases as extra starting values become available.

References

[1] Gopalsamy, K.: *Stability and oscillation in delay differential equations of population dynamics.* Kluwer Academic Publishers, Dordrecht, Boston, London, 1992.

[2] Hairer, E., Norsett, S. and Wanner, G.: *Solving ordinary differential equations. Nonstiff problems.* Springer, Berlin, 1987.

[3] Kim, A. V. and Pimenov, V. G.: Numerical methods for delay differential equations – application of i-smooth calculus. *Lecture Notes Series*, vol. 44 Seoul National University, Korea, 1999.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

SUPERAPPROXIMATION OF THE PARTIAL DERIVATIVES IN THE SPACE OF LINEAR TRIANGULAR AND BILINEAR QUADRILATERAL FINITE ELEMENTS

Josef Dalík

Brno University of Technology
Žižkova 17, 602 00 Brno, Czech Republic
dalik.j@fce.vutbr.cz

Abstract

A method for the second-order approximation of the values of partial derivatives of an arbitrary smooth function $u = u(x_1, x_2)$ in the vertices of a conformal and nonobtuse regular triangulation $\mathcal{T}_h$ consisting of triangles and convex quadrilaterals is described and its accuracy is illustrated numerically. The method assumes that the interpolant $\Pi_h(u)$ in the finite element space of the linear triangular and bilinear quadrilateral finite elements from $\mathcal{T}_h$ is known only.

1. Introduction

The problem to find second-order approximations of the first partial derivatives of smooth functions u in the vertices of triangulations by means of the interpolant $\Pi_h(u)$ only is actual since its formulation in [6] in the year 1967. Besides the widely acknowledged method [7] there exist successful methods like [5] and [3]. In this paper, we generalize the method of averaging from [2] to nonobtuse regular triangulations consisting of triangles as well as convex quadrilaterals in general. Numerical experiments indicate the second-order accuracy of this procedure. These high-order approximations of the partial derivatives have many applications. See [1] for some of them.

We denote $[a_1, a_2]$ the Cartesian coordinates of a point a and $|ab|$ the length of the segment $\overline{ab}$. For arbitrary points $a^1, \ldots, a^m$, operations ,,+“ and ,,−“ mean addition and subtraction modulo m on the set $\{1, \ldots, m\}$.

2. Bilinear quadrilateral finite elements

Besides the linear triangular finite elements, we work with the following bilinear quadrilateral ones.

Definition 1. A *reference bilinear finite element* consists of

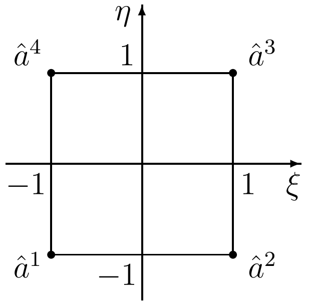


Figure 1: The reference square.

a) the *reference square* $\hat{K} = \overline{\hat{a}^1\hat{a}^2\hat{a}^3\hat{a}^4}$ from Fig. 1,

b) the *local space* $\mathbb{Q}^{(1)} = \{a + b\xi + c\eta + d\xi\eta \,|\, a, b, c, d \in \mathbb{R}\}$ and of

c) the *parameters* $\hat{p}(\hat{a}^1), \ldots, \hat{p}(\hat{a}^4)$ related to every function $\hat{p} \in \mathbb{Q}^{(1)}$. The parameters determine the function $\hat{p}$ uniquely.

Definition 2. A *bilinear quadrilateral finite element* consists of

a) an image $K = \overline{a^1a^2a^3a^4}$ of $\hat{K}$ by the injective bilinear mapping

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = F_K(\xi, \eta) \equiv \sum_{i=1}^{4} \hat{N}^i(\xi, \eta) \begin{bmatrix} a_1^i \\ a_2^i \end{bmatrix} \tag{1}$$

with the *Lagrange base functions*

$$\begin{array}{ll} \hat{N}^1(\xi,\eta) = (1-\xi)(1-\eta)/4, & \hat{N}^2(\xi,\eta) = (1+\xi)(1-\eta)/4, \\ \hat{N}^3(\xi,\eta) = (1+\xi)(1+\eta)/4, & \hat{N}^4(\xi,\eta) = (1-\xi)(1+\eta)/4 \end{array}$$

in the space $\mathbb{Q}^{(1)}$ related to the nodes $\hat{a}^1, \ldots, \hat{a}^4$ consecutively. Then $F_K(\hat{a}^i) = a^i$ for $i = 1, \ldots, 4$ obviously and F_K is an injection if and only if K is a *convex quadrilateral*, i.e. the inner angle $\angle a^{i-1}a^ia^{i+1}$ of K is less than π for $i = 1, \ldots, 4$ due to [4], Section 3.3,

b) the *local space* $\mathbb{Q}_K^{(1)} = \{q \,|\, q = \hat{q} \circ F_K^{-1}$ for some $\hat{q} \in \mathbb{Q}^{(1)}\}$ and of

c) the *parameters* $q(a^1), \ldots, q(a^4)$ related to every $q \in \mathbb{Q}_K^{(1)}$. The parameters determine the function q uniquely.

Lemma 1. *The functions* $1, x_1, x_2$ *belong to* $\mathbb{Q}_K^{(1)}$ *for every convex quadrilateral* K.

Proof. If $K = \overline{a^1a^2a^3a^4}$ is a convex quadrilateral then $\mathbb{Q}_K^{(1)} = \{q \,|\, q \circ F_K \in \mathbb{Q}^{(1)}\}$ is a direct consequence of Definition 2. This and

$$\begin{aligned} 1 \circ F_K &= 1 \in \mathbb{Q}^{(1)} \\ x_1 \circ F_K &= \hat{N}^1(\xi,\eta)a_1^1 + \ldots + \hat{N}^4(\xi,\eta)a_1^4 \in \mathbb{Q}^{(1)} \\ x_2 \circ F_K &= \hat{N}^1(\xi,\eta)a_2^1 + \ldots + \hat{N}^4(\xi,\eta)a_2^4 \in \mathbb{Q}^{(1)} \end{aligned}$$

give us the statement.

Definition 3. If K is a triangle and convex quadrilateral then we denote by $\Pi_K(u)$ the linear and bilinear interpolant of a function $u \in C(K)$ in the vertices of K, respectively.

Lemma 2. *Let us consider a bilinear quadrilateral finite element* $K = \overline{a^1a^2a^3a^4}$, $l = 1, 2$ *and a linear triangular finite element* $T_j = \overline{a^{j-1}a^ja^{j+1}}$. *Then the graph of* $\Pi_{T_j}(u)$ *is the tangent plane to that of* $\Pi_K(u)$ *at the point* a^j, *so that*

$$\frac{\partial \Pi_K(u)}{\partial x_l}(a^j) = \frac{\partial \Pi_{T_j}(u)}{\partial x_l} \quad \forall\ u \in C(K)$$

for $j = 1, \ldots, 4$.

Proof. As the functions from $\mathbb{Q}_K^{(1)}$ are linear on every side of K, $\Pi_K(u)$ is linear on the segments $\overline{a^{j-1}a^j}$ and $\overline{a^ja^{j+1}}$. Hence the segments $\overline{p^{j-1}p^j}$ and $\overline{p^jp^{j+1}}$ for $p^i = [a_1^i, a_2^i, u(a^i)]$, $i = j-1, j, j+1$, are subsets of graph$(\Pi_K(u))$. These segments belong to a unique plane. This one is the tangent plane of graph$(\Pi_K(u))$ at a^j and it contains graph$\big(\Pi_{T_j}(u)\big)$ as well. Lemma 2 follows immediately.

3. Nonobtuse regular triangulations

The symbols $\mathbb{P}^{(1)}$ and $\mathbb{P}^{(2)}$ are reserved for the spaces of real linear and quadratic polynomials in two variables and Ω for a non-empty bounded connected polygonal domain in the plane. We say that K is an *element* when K is a triangle or a convex quadrilateral, denote $|K|$ the area of K, h_K the diameter of K and ϱ_K the maximal diameter of the circles inside of K.

A system $\mathcal{T}_h$ of elements is said to be a *triangulation* of Ω when $\cup_{K \in \mathcal{T}_h} K = \overline{\Omega}$, any two different elements have disjoint interiors and any side of an element is either a side of another element or a subset of the boundary $\partial\Omega$. Let us consider a *vertex a* of (an element from) a triangulation $\mathcal{T}_h$. We call b a *neighbour* of a (in $\mathcal{T}_h$) when the segment $\overline{ab}$ is a side of an element from $\mathcal{T}_h$ and denote $\mathcal{N}_h(a)$ the set of neighbours of a in $\mathcal{T}_h$. We say that a is an *inner* and *boundary* vertex when $a \in \Omega$ and $a \in \partial\Omega$, respectively.

Definition 4. A system $\mathbf{T}$ of triangulations of Ω is said to be

a) a *family* when for every $\varepsilon > 0$ there exists $\mathcal{T}_h \in \mathbf{T}$ satisfying $h_K < \varepsilon$ for all $K \in \mathcal{T}_h$.

b) *shape-regular* when there is $\sigma > 0$ such that $\varrho_K / h_K > \sigma$ for all elements K of any triangulation from $\mathbf{T}$.

We work with a shape-regular family $\mathbf{T}$ of triangulations of Ω such that all inner angles of the triangles from any triangulation in $\mathbf{T}$ are less than or equal to the right angle. We call these triangulations *nonobtuse regular.*

4. The method of averaging

It is well-known that $\partial u/\partial x_l(a) = \partial\Pi_K(u)/\partial x_l(a) + O(h_K)$ for a vertex a of an element K from a nonobtuse regular triangulation, function $u \in C^2(K)$ and for $l = 1, 2$. We construct a weight vector such that the corresponding weighted average of the values of $\partial\Pi_K(u)/\partial x_l$ in various vertices of the elements K with vertex a approximates $\partial u/\partial x_l(a)$ with an error of the second order. A special case of this construction has been analysed in [2] for the nonobtuse regular triangulations consisting of triangles only.

Calculating the approximations of $\partial u/\partial x_l(a)$, we use local Cartesian coordinates with origin a.

Defrinition 5. Let $\mathcal{T}_h$ be a nonobtuse regular triangulation. We say that $r = (b^1, \dots, b^n)$ is a *ring* around

a) an inner vertex a of $\mathcal{T}_h$ when

a1) $\{b^1, \dots, b^n\} \supseteq \mathcal{N}_h(a)$ and

$$b^i \notin \mathcal{N}_h(a) \implies K = \overline{ab^{i-1}b^ib^{i+1}} \in \mathcal{T}_h \text{ and } \angle b^{i-1}ab^{i+1} > \pi/2,$$

a2) $\angle b^n ab^1, \dots, \angle b^{n-1}ab^n$ have the same orientation and

a3) $\angle b^n ab^1 + \cdots + \angle b^{n-1}ab^n = 2\pi$.

b) a boundary vertex a of $\mathcal{T}_h$ when there is an inner vertex b^j such that

b1) $(b^1, \dots, b^{j-1}, a, b^{j+1}, \dots, b^n)$ is a ring around b^j with $n \geq 5$ or

b2) $\overline{ab^{j+1}b^jb^{j-1}} \in \mathcal{T}_h$ and $(b^1, \dots, b^{j-1}, b^{j+1}, \dots, b^n)$ is a ring around b^j.

We say that the triangles $U_1 = \overline{b^nab^1}, \dots, U_n = \overline{b^{n-1}ab^n}$ are *related* to r and set $H(a) = \max_{1\leq i\leq n} |ab^i|$.

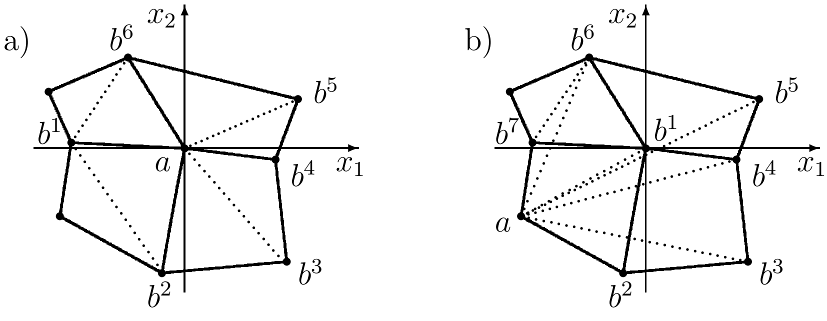


Figure 2: A ring around a) an inner vertex a and b) a boundary one.

In Fig. 2, the thick lines denote the quadrilaterals from the given triangulation and the dotted lines indicate triangles $U_1, \dots, U_6$ in the case a) and $U_1, \dots, U_7$ in b).

Definition 6. Let $l = 1, 2$, $r = (b^1, \ldots, b^n)$ be a ring around a vertex a of a nonobtuse regular triangulation and let $u \in C(\overline{\Omega})$. Then we set

$$\mathrm{B}_l[u](a) = f_1 \frac{\partial \Pi_1(u)}{\partial x_l} + \cdots + f_n \frac{\partial \Pi_n(u)}{\partial x_l}. \tag{2}$$

Here $\Pi_1(u), \ldots, \Pi_n(u)$ are the linear interpolants of u in the vertices of the triangles $U_1, \ldots, U_n$ related to r and the *weight vector* $f = [f_1, \ldots, f_n]^\top$ is the minimal 2-norm vector such that $\mathrm{B}_l[u](a)$ is *consistent*, i.e. $\mathrm{B}_l[u](a) = \partial u/\partial x_l(a)$ for all $u \in \mathbb{P}^{(2)}$. Due to [2], f is the minimal 2-norm solution of the equations $M(r)f = d$ with

$$M(r) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \frac{x_n^2 y_1 - x_1^2 y_n}{D_1} & \frac{x_1^2 y_2 - x_2^2 y_1}{D_2} & \cdots & \frac{x_{n-1}^2 y_n - x_n^2 y_{n-1}}{D_n} \\ \frac{y_n y_1 (x_n - x_1)}{D_1} & \frac{y_1 y_2 (x_1 - x_2)}{D_2} & \cdots & \frac{y_{n-1} y_n (x_{n-1} - x_n)}{D_n} \\ \frac{y_n y_1 (y_n - y_1)}{D_1} & \frac{y_1 y_2 (y_1 - y_2)}{D_2} & \cdots & \frac{y_{n-1} y_n (y_{n-1} - y_n)}{D_n} \end{bmatrix}, \qquad d = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$[x_i, y_i] = b^i$ and $D_i = D(a, b^{i-1}, b^i)$ for $i = 1, \ldots, n$.

Definition 5 is in agreement with Lemma 2 and with the following statement:

Lemma 3. *The system of equations $M(r)f = d$ related to the ring $r = (b^1, \ldots, b^4)$ around a vertex a is*

a) unsolvable if a is a boundary vertex and

b) solvable if and only if the vertices b^1, a, b^3 as well as b^2, a, b^4 are situated on one straight-line if a is an inner vertex.

We omit the proof of Lemma 3.

Example. For $a = [0, 0]$, we approximate the partial derivative $\partial u/\partial x_1(a) = -0.5403023$ of $u(x_1, x_2) = \sin(1 + 2x_1 + x_2)/(x_2 - 2)$ by $B_1[u](a)$. In Table 1, we use the ring from Fig. 2 a) with $H(a) = 1.3453624/2^i$ for $i = 1, \ldots, 8$.

i	$H(a)$	$B_1[u](a)$	$\partial u/\partial x_1(a) - B_1[u](a)$
1	6.72681 e-1	-0.460947	-7.93549 e-2
2	3.36341 e-1	-0.519906	-2.03960 e-2
3	1.68170 e-1	-0.535183	-5.11974 e-3
4	8.40852 e-2	-0.539023	-1.27939 e-3
5	4.20426 e-2	-0.539983	-3.19584 e-4
6	2.10213 e-2	-0.540222	-7.98508 e-5
7	1.05106 e-2	-0.540282	-1.99563 e-5
8	5.25532 e-3	-0.540297	-4.98822 e-6

Table 1

i	$H(a)$	$B_1[u](a)$	$\partial u/\partial x_1(a) - B_1[u](a)$
1	1.15244	-0.	-0.104569 e-1
2	5.76222 e-1	-0.577975	3.76723 e-2
3	2.88111 e-1	-0.556928	1.66261 e-2
4	1.44055 e-1	-0.545228	4.92589 e-3
5	7.20277 e-2	-0.541620	1.31737 e-3
6	3.60138 e-2	-0.540642	3.39385 e-4
7	1.80069 e-2	-0.540388	8.60568 e-5
8	9.00346 e-3	-0.540324	2.16627 e-5

Table 2

In Table 2, we use the ring from Fig. 2 b) with $H(a) = 2.3048861/2^i$ for $i = 1, \dots, 8$.

This example indicates the second order of error of the approximations $\mathrm{B}_l[u](a)$ both for the inner and the boundary vertices a, but an analysis of the accuracy of this averaging operator is necessary.

Acknowledgements

This outcome has been achieved with the financial support of the Ministry of Education, Youth and Sports, project No. 1M0579, within the activities of the CIDEAS Research Centre.

References

[1] Ainsworth, M. and Oden, J.: *A posteriori error estimation in finite element analysis.* Wiley, New York, 2000.

[2] Dalík, J.: Averaging of directional derivatives in vertices of nonobtuse regular triangulations. Numer. Math. **116** (2010), 619–644.

[3] Hlaváček, I., Křížek, M., and Pištora, V.: How to recover the gradient of linear elements on nonuniform triangulations. Appl. Math. **41** (1996), 241–267.

[4] Strang, G. and Fix, G. J.: *An analysis of the finite element method.* Prentice-Hall, Inc. Englewood Cliffs, N. J., 1973.

[5] Zhang, Z. and Naga, A.: A new finite element gradient recovery method: superconvergence property. SIAM J. Sci. Comput. **26** (2005), 1192–1213.

[6] Zienkiewicz, O. C. and Cheung Y. K.: *The finite element method in structural and continuum mechanics.* McGraw Hill, London, 1967.

[7] Zienkiewicz, O. C. and Zhu, J. Z.: The superconvergence patch recovery and *a posteriori* error estimates. Part 1: The recovery technique. Internat. J. Numer. Methods Engrg. **33** (1992), 1331–1364.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

APPROXIMATE POLYNOMIAL GCD

Ján Eliaš, Jan Zítko

Department of Numerical Mathematics,
Faculty of Mathematics and Physics, Charles University in Prague
Sokolovská, Prague, Czech Republic
janelias@ymail.com, zitko@karlin.mff.cuni.cz

Abstract

The computation of polynomial greatest common divisor (GCD) ranks among basic algebraic problems with many applications, for example, in image processing and control theory. The problem of the GCD computing of two exact polynomials is well defined and can be solved symbolically, for example, by the oldest and commonly used Euclid's algorithm. However, this is an ill-posed problem, particularly when some unknown noise is applied to the polynomial coefficients. Hence, new methods for the GCD computation have been extensively studied in recent years.

The aim is to overcome the ill-posed sensitivity of the GCD computation in the presence of noise. We show that this can be successively done through a TLS formulation of the solved problem, [1, 5, 7].

1. Approximate greatest common divisor

Suppose a pair of two polynomials f and g of degrees m and n,

$$f(x) = \sum_{i=0}^{m} a_i x^{m-i} \quad (a_0 a_m \neq 0) \quad \text{and} \quad g(x) = \sum_{j=0}^{n} b_j x^{n-j} \quad (b_0 b_n \neq 0) \tag{1}$$

with a non-trivial GCD h of degree d is given, $1 \leq d \leq n \leq m$. Vectors of polynomial coefficients are denoted by bold lower-case Latin letters, e.g. $\mathbf{f} = [a_0, a_1, \ldots, a_m]^T$ represents the vector of coefficients of f. Similarly, $\mathbf{g}$, $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{h}$ will denote the vectors of coefficients of involved polynomials g, u, v and h.

Then there exist polynomials u and v of degrees $m - d$ and $n - d$, respectively, so that

$$uh = f \quad \text{and} \quad vh = g. \tag{2}$$

Equations in (2) can be rewritten to the matrix-vector notation as

$$S_d(f, g) \begin{bmatrix} \mathbf{v} \\ -\mathbf{u} \end{bmatrix} = \mathbf{0}, \tag{3}$$

where

$$S_d(f,g) \quad = \quad \left[\underbrace{\begin{matrix} a_0 & & & \\ a_1 & a_0 & & \\ \vdots & a_1 & \ddots & \\ a_m & \vdots & \ddots & a_0 \\ & a_m & & a_1 \\ & & \ddots & \vdots \\ & & & a_m \end{matrix}}_{n-d+1 \text{ col.}} \underbrace{\begin{matrix} b_0 & & & \\ b_1 & b_0 & & \\ \vdots & b_1 & \ddots & \\ b_n & \vdots & \ddots & b_0 \\ & b_n & & b_1 \\ & & \ddots & \vdots \\ & & & b_n \end{matrix}}_{m-d+1 \text{ col.}}\right] \in \mathbb{R}^{(m+n-d+1)\times(m+n-2d+2)}$$

is, except the case $d = 1$, rectangular $m+n-d+1$ by $m+n-2d+2$ matrix called the dth Sylvester subresultant matrix. The coefficients $\{a_i\}$ of f occupy the first $n-d+1$ columns and the coefficients $\{b_j\}$ of g occupy the last $m-d+1$ columns. Hence, $S_d(f,g)$ is the block matrix consisting of the two Cauchy matrices, $S_d(f,g) = [C_{n-d+1}(f),\, C_{m-d+1}(g)]$.[1] The Sylvester matrix is then the matrix $S(f,g) = S_1(f,g)$ $= [C_n(f),\, C_m(g)] \in \mathbb{R}^{(m+n)\times(m+n)}$.

The most important relations between the GCD and the Sylvester matrices are summarised in the following theorem.[2]

Theorem 1. *Suppose that f and g are polynomials of degrees m and n, $m \geq n$, and $h = GCD(f,g)$. Then*

i) *rank*$(S(f,g)) = m+n-d \iff \deg h = d,$

ii) *rank*$(S_d(f,g)) = m+n-2d+1 \iff \deg h = d,$

iii) *the coefficient vector* $\mathbf{h}$ *is a solution of the linear system*

$$\begin{bmatrix} C_{d+1}(u) \\ C_{d+1}(v) \end{bmatrix} \mathbf{h} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{4}$$

Moreover, if $\deg h = d$, *then*

iv) *rank*$(S_j(f,g)) < m+n-2j+2, \quad j = 1,\ldots,d,$

v) *rank*$(S_j(f,g)) = m+n-2j+2, \quad j = d+1,\ldots,n.$ □

Hence, if $\deg h = d$, then $S_d = S_d(f,g)$ is rank deficient by 1 since S_d has $m+n-2d+2$ columns and rank $m+n-2d+1$ by recalling the property *ii)* from the theorem. Therefore,

$$S_d \begin{bmatrix} \mathbf{v} \\ -\mathbf{u} \end{bmatrix} = \mathbf{0} \Longrightarrow \exists\, s \in \mathbb{R} : \begin{bmatrix} \mathbf{v} \\ -\mathbf{u} \end{bmatrix} = s\, \mathbf{v}_{\min}(S_d),$$

where $\mathbf{v}_{\min}(S_d)$ is the right singular vector associated with $\sigma_{\min}(S_d) = 0$.

[1] The subscripts $n-d+1$ and $m-d+1$ in $C_{n-d+1}(f)$ and $C_{m-d+1}(g)$ represent the number of columns filled with the coefficients of f and g, respectively.

[2] A proof is outlined in the second authors' paper of these proceedings.

The coefficients of h can be now easily computed. For this purpose we have to calculate the smallest singular pair $\{\sigma_{\min}, \mathbf{v}_{\min}\}$ of every matrix in the sequence $S_n, S_{n-1}, \ldots, S_1$ until the first rank deficient matrix is found.[3] Once, the rank deficient matrix S_d is revealed, $\mathbf{v}$ and $\mathbf{u}$ can be extracted from the singular vector $\mathbf{v}_{\min}(S_d)$. The coefficients of h are then computed from (4).

The smallest singular value and its corresponding right singular vector of S_d can be computed by the Gauss-Newton method, [4], i.e. by the iteration process

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \begin{bmatrix} 2\tau\mathbf{x}_i^T \\ S_d \end{bmatrix}^\dagger \begin{bmatrix} \tau\mathbf{x}_i^T\mathbf{x}_i - \tau \\ S_d\mathbf{x}_i \end{bmatrix} \quad \text{and} \quad \zeta_{i+1} = \frac{\|S_d\mathbf{x}_{j+1}\|_2}{\|\mathbf{x}_{j+1}\|_2}$$

for τ sufficiently large.[4] Then

$$\mathbf{x}_i \xrightarrow[i\to\infty]{} \mathbf{v}_{\min}(S_d) \quad \text{and} \quad \zeta_i \xrightarrow[i\to\infty]{} \sigma_{\min}(S_d).$$

The GCD solver in [4, 6] is based on this iteration process. However, note that in real computations some threshold θ must be applied to ζ_i to reveal the rank deficiency. Assuming that the level of noise is not known, the solver in [4, 6] cannot be used, since the numerical rank cannot be computed reliably, [1, 5].

Whether the level of imposed noise is known or not, $\mathbf{v}_{\min}(S_d)$, $\mathbf{u}$ and $\mathbf{v}$ are computed approximately and so the coefficients of h are not calculated exactly. Hence, an approximate greatest common divisor (AGCD) is only computed.

2. Impact of noise

Numerically, S_d is considered to be rank deficient whenever $\sigma_{\min}(S_d) \leq \theta$ for a prescribed threshold θ. If rounding errors are only assumed, then $\theta = \varepsilon\|S_d\|_2$ with a machine precision ε is usually used, [2] p. 261. However, if some additional noise of unknown level is considered, then computations with all similar choices of θ usually fail. In this case a different approach has to be developed.

Dependence of the GCD computation on noise can be seen from the following example. Consider two polynomials f and g of degree 32,

$$\begin{aligned} f(x) &= \prod_{i=1}^{8} \left[(x - r_1\alpha_i)^2 + r_1^2\beta_i^2\right] \prod_{i=9}^{16} \left[(x - r_2\alpha_i)^2 + r_2^2\beta_i^2\right], \\ g(x) &= \prod_{i=1}^{16} \left[(x - r_1\alpha_i)^2 + r_1^2\beta_i^2\right], \end{aligned} \tag{5}$$

where $\alpha_i = \cos\left(\frac{\pi i}{m}\right), \beta_i = \sin\left(\frac{\pi i}{m}\right), i = 1, \ldots, n, r_1 = 0.5$ and $r_2 = 1.5$. These polynomials have the exact GCD of degree 16. So the rank of the Sylvester matrix $S(f, g)$ is 48 by recalling Theorem 1 *i)*.

[3] Note that if S_d is the first rank deficient matrix and $d < n \leq m$, then every S_j in $S_n, \ldots, S_{d+1}$ has full column rank using Theorem 1 *v)*.

[4] The symbol $(\cdot)^\dagger$ denotes the Moore-Penrose inverse of $(\cdot)$.

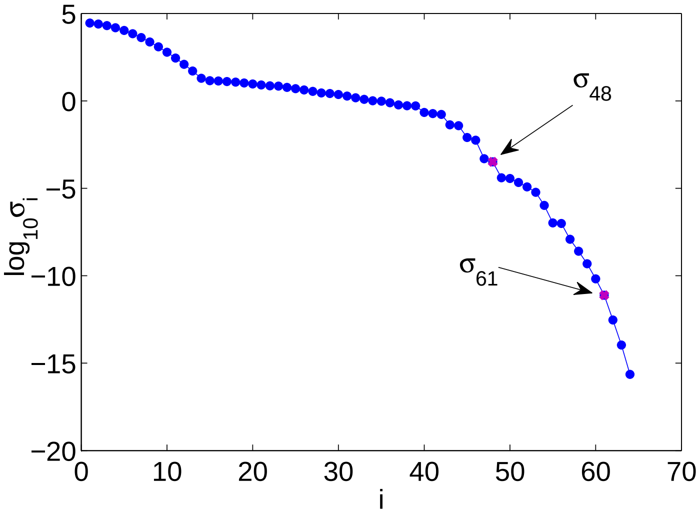


Figure 1: Singular values of the Sylvester matrix $S(f, g)$ for f and g perturbed componentwisely by the noise of the SNR $= 10^8$.

The numerical rank of $S(f, g)$ is well defined and can be revealed by using Gauss-Newton iteration for the choice $\theta = \varepsilon \|S(f, g)\|_2 \approx 10^{-12}$ in case when only rounding errors are considered.

Suppose now, that a noise of the signal-to-noise ratio SNR $= 10^8$ is componentwisely imposed to the coefficients of f and g. Figure 1 shows the singular values of the Sylvester matrix of perturbed polynomials. For the choice $\theta = 10^{-12}$ the numerical rank is 61 that is incorrect. The correct numerical rank 48 can be revealed with $\theta = 10^{-4}$. The question, however, is how to estimate this θ only from the involved data.

3. TLS formulation, methods for AGCD

For the exact polynomials the system of equations (3) can be transformed to the system

$$A_d\mathbf{x} = \mathbf{c_d}, \tag{6}$$

where $\mathbf{c_d}$ is the first column of S_d and A_d is formed from the remaining $m+n-2d+1$ columns of S_d, $S_d = [\mathbf{c_d}, A_d]$.

While the system (6) possesses exactly one solution $\mathbf{x}$ for the exact polynomials, it does not possess any solution for the inexact polynomials, since the perturbed polynomials are coprime with probability almost one, i.e. $\mathbf{c_d} \notin \text{Range}(A_d)$ for the inexact polynomials. However, if the polynomials f and g are coprime, we can demand to compute the minimal corrections of their coefficients, i.e. polynomials δf and δg so that $f + \delta f$ and $g + \delta g$ have a non-trivial GCD with the highest possible degree. Then, $\text{AGCD}(f, g) = \text{GCD}(f + \delta f, g + \delta g)$.

Let us denote the Sylvester matrix of δf and δg by $\delta S_d = \delta S_d(\delta f, \delta g)$, $\delta S_d =$

$[\mathbf{h_d}, E_d]$, and let $\mathbf{z} = [\delta \mathbf{f}^T, \delta \mathbf{g}^T]^T$ be the vector of the coefficients of δf and δg. Then δf and δg can be computed so that

$$(A_d + E_d)\mathbf{x} = \mathbf{c_d} + \mathbf{h_d}$$

has exactly one solution $\mathbf{x}$ and $\|\mathbf{z}\|_2$ is minimal. Hence, the problem, that is finally solved, is the structured TLS problem:

$$\begin{aligned} &\min_{\mathbf{z},\mathbf{x}} \|\mathbf{z}\|_2 \\ &\text{subject to } (A_d + E_d)\mathbf{x} = \mathbf{c_d} + \mathbf{h_d} \\ &\text{and } [\mathbf{h_d}, E_d] \text{ is of the same structure as } [\mathbf{c_d}, A_d]. \end{aligned} \tag{7}$$

Two methods for solving (7) are presented in [3]. These methods are modified and customised for the AGCD computation in [1, 5].

Methods for the AGCD computation are not discussed in this paper, however note, that Sylvester matrices are badly conditioned, for example, if considered polynomials have coefficients that differ by several orders in magnitude. It is therefore necessary to apply some preprocessing operations on polynomials before a method is used. Particularly, these operations include

- normalisation of the coefficients by the geometric mean that preserves the propagation of noise,
- variable substitution $x = \gamma w$ for minimising the ratio of the maximum to the minimum coefficient of both polynomials f and g,
- considering a parameter α in $S(f, \alpha g)$ for weighting the coefficients of one polynomial with respect to the coefficients of the second polynomial,
- column pivoting, i.e. a column of S_d for which the residual $\|A_d\mathbf{y} - \mathbf{c_d}\|_2$ is minimal replaces $\mathbf{c_d}$ in (6).

There are several possible ways how to compute α and γ, for example, they can be computed as values that minimise the ratio

$$\frac{\max\left\{\max_{i=0,\dots,m} |a_i\gamma^{m-i}|, \max_{j=0,\dots,n} |\alpha b_j \gamma^{n-j}|\right\}}{\min\left\{\min_{i=0,\dots,m} |a_i\gamma^{m-i}|, \min_{j=0,\dots,n} |\alpha b_j \gamma^{n-j}|\right\}}.$$

More information on the preprocessing operations is provided in [5].

Finally, Figure 2 shows the singular values of $S(f, g) + \delta S(\delta f, \delta g)$ for the polynomials f and g in (5) perturbed componentwisely by the noise of the SNR $= 10^8$. The polynomials δf and δg are obtained by solving (7). We can see that the numerical rank is now perfectly defined and so further computation of the GCD by the procedure discussed in Section 1 can be processed.

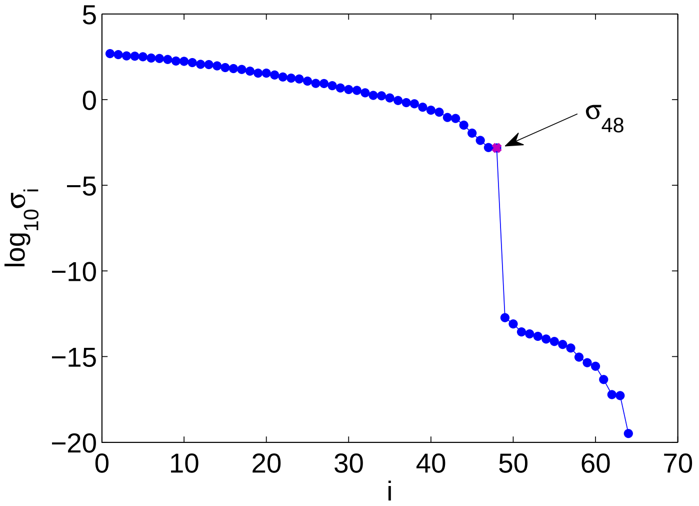


Figure 2: Singular values of $S(f,g)+\delta S(\delta f,\delta g)$ where f and g in (5) are perturbed componentwisely by the noise of the SNR $= 10^8$, and δf and δg are obtained by solving (7).

Acknowledgements

This work was supported by the Department of Numerical Mathematics, Charles University in Prague. The authors thank for this support.

References

[1] Eliaš, J.: *Approximate Polynomial Greatest Common Divisor*. Master Thesis, Charles University in Prague, 2012.

[2] Golub, G. H. and Van Loan, C. F.: *Matrix Computations*. 3rd Ed. The John Hopkins University Press, Baltimore, USA, 1996.

[3] Lemmerling, P., Mastronardi, N., and Van Huffel, S.: Fast algorithm for solving the Hankel/Toeplitz Structured Total Least Squares Problem. Numer. Algorithms **23** (2000), 371–392.

[4] Li, T. Y. and Zeng, Z.: A rank-revealing method with updating, downdating and applications. SIAM J. Matrix Anal. Appl. **26** (2005), 918–946.

[5] Winkler, J. R. and Hasan, M.: A non-linear structure preserving matrix method for the low rank approximation of the Sylvester resultant matrix. J. Comput. Appl. Math. **234** (2010), 3226-3242.

[6] Zeng, Z.: The approximate GCD of inexact polynomials, Part I: univariate algorithm. Preprint (2004).

[7] Zítko, J. and Eliaš, J.: Application of the rank revealing algorithm for the calculation of the GCD. In: *Winter School and SNA'12*, pp. 175–180. Technická Univerzita v Liberci, Liberec, 2012.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

PARALLEL IMPLEMENTATION OF WAVELET-GALERKIN METHOD

Václav Finěk, Martina Šimůnková

KAP and KMD FP TU Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
vaclav.finek@tul.cz, martina.simunkova@tul.cz

Abstract

We present here some details of our implementation of Wavelet-Galerkin method for Poisson equation in C language parallelized by POSIX threads library and show its performance in dimensions $d \in \{3, 4, 5\}$.

1. Introduction

Due to storage requirements and computational complexity, the approximate solution of PDEs computed by standard numerical methods is usually limited to problems with up to three or fourth dimensions. However in mathematical modeling, there is a lot of problems which involve more than three or four dimensions. For example, the pricing of financial derivatives, problems in quantum mechanics and particle physics. Here, the dimension grows with the number of considered derivatives, electrons or nuclei. An important issue for numerical methods for higher-dimensional PDEs is that typical domains are usually hypercubes. And it is well-known, that the curse of dimensionality can be broken on tensor product domain $(0,1)^d$ by using sparse grids [1] or by wavelets [5].

To use wavelets efficiently to solve PDEs, it is necessary to have very efficient matrix-vector multiplication for vectors and matrices in wavelet coordinates and to have at one's disposal suitable wavelet bases. Wavelets should have short supports and vanishing moments, be smooth and known in closed form, and a corresponding wavelet basis should be well-conditioned.

In [5], authors were able to solve Poisson equation up to 10 dimensions by applying an adaptive wavelet scheme with orthonormal continuous piecewise linear multiwavelets proposed in [6]. They exploited the fact that the corresponding stiffness matrices are in tensor product wavelet coordinates well-conditioned independently on the dimension. Their approximations converged in energy norm with the same rate as the best N-term approximations independent of d with the cost of producing these approximations proportional to their length up to a constant factor growing potentially with the dimension, but only linearly.

We try to improve results obtained in [5] by applying higher order wavelet basis. In recent years, several promising constructions of wavelets were proposed. We mention, for example, a construction of spline-wavelet bases on the interval proposed in [2]. Their bases are compactly supported and generate multiresolution analyses on the unit interval with the desired numbers of vanishing wavelet moments for primal and dual wavelets. Moreover, the condition number of interval spline-wavelet bases is close to the condition number of the spline wavelet bases on the real line for bases up to order 4. In our contribution, we use recently proposed wavelets based on quadratic splines [3] which have shorter supports and are better conditioned. It is a modification of basis proposed in [4] with an improved condition number. Some preliminary results were already presented in [7]. There, a sequential algorithm was used to solve Poisson equation for $d \in \{2, 3\}$.

2. Problem formulation

We solve Dirichlet problem

$$\begin{aligned} -\sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2} &= f \qquad x \in \Omega = (0,1)^d \\ u &= 0 \qquad x \in \partial\Omega \end{aligned}$$

by Galerkin method. Basis functions are wavelets based on quadratic splines proposed in [3] extended to higher dimensions by tensor product. Stiffness matrices are computed exactly. Used quadratic splines have points of discontinuity at $\frac{1}{2^L}, \frac{2}{2^L}, \ldots, \frac{2^L-1}{2^L}$ where L denotes the number of decomposition levels. Right-hand side integrals are calculated by adaptive Simpson rule. We split integration to hypercubes of size $(2^{-L})^d$ which enables efficient parallelization. We solve the arising system of linear algebraic equations originated from discretization by the conjugate gradient method with standard wavelet preconditioning consisting in normalizing all basis functions with respect to a bilinear form corresponding to stiffness matrix. It practically means that the stiffness matrix is multiplied from both sides by a diagonal matrix which has at its diagonal square root of diagonal elements of the original stiffness matrix.

We aim at an efficient implementation of adaptive wavelet methods for higher dimensional problems. For this purpose it is necessary to implement an efficient storage of sparse vectors and sparse matrices in wavelet coordinates and their efficient multiplication. We have so far implemented an efficient algorithm for matrix-vector multiplication in the case $d = 1$ and because stiffness matrices for Poisson equation in higher dimensions are computed from the stiffness matrices for Poisson equation in one dimension and from matrices of scalar products of basis functions in one dimension, we apply it here also for $d \in \{3, 4, 5\}$. Here, we present a non-adaptive implementation. It means that we choose a number of levels L and a dimension d which leads to 2^{Ld} basis functions.

3. Implementation and parallelization

In next subsections, we shortly describe some implementation details – a computation of right-hand side integrals and a multiplication of vector by stiffness matrix.

3.1. Computation of right-hand side integrals

Right-hand side integrals are in the form

$$\int_{(0,1)^d} \psi_{i_1}(x_1)\dots\psi_{i_d}(x_d) f(x_1\dots x_d)\,\mathrm{d}x_1\dots\mathrm{d}x_d, \tag{1}$$

where functions ψ_{i_j} are piecewise quadratic. Therefore we can split hypercube $(0,1)^d$ to hypercubes of size $(2^{-L})^d$ and compute integrals

$$\int x_1^{i_1}\dots x_d^{i_d} f(x_1\dots x_d)\,\mathrm{d}x_1\dots\mathrm{d}x_d \tag{2}$$

at each small hypercube for $i_1,\dots,i_d \in \{0,1,2\}$. Consequently, we compute (1) as a linear combination of integrals (2). To calculate (2) we use Fubini's theorem and a recursion. Let us denote $x_i = \frac{i}{2^L}$. We designed an implementation of Simpson rule for a computation of iterated integrals $I = \int_{x_i}^{x_{i+1}} F(x)\,\mathrm{d}x$ described below in 1.-5. Main goal of our design is to omit evaluation of the same value of function F twice because it is again an integral and its evaluation is computationally expensive.

1. Compute recursively $F(x_i)$ and $F(x_{i+1})$ and evaluate
$$I_0 = \tfrac{x_{i+1}-x_i}{2}\left(F(x_i)+F(x_{i+1})\right).$$
2. Set $j=0$.
3. Compute recursively $F(\xi_{i,k})$ with
$$\xi_{i,k} = x_i + \tfrac{2k-1}{2^{L+j+1}} \text{ for } k = 1,2,3,\dots,2^j$$
and evaluate
$$I' = \tfrac{x_{i+1}-x_i}{2^j}\sum_{k=1}^{2^j} F(\xi_{i,k}).$$
4. If $|I_j - I'| > \varepsilon$, set $j = j+1$, compute $I_j = \frac{1}{2}(I_{j-1}+I')$ and go to step 3.
5. Compute $I \approx \frac{1}{3}(I_j + 2I')$.

As mentioned above, we compute right-hand side integrals separately on hypercubes $(2^{-L})^d$. These integrals can be computed independently which enables simple parallelization. Our implementation is in C language and for parallelization we use a POSIX threads library. Every thread takes an index of a hypercube from a global variable in a loop, then increases the index and computes integrals. Taking and increasing global variable is a critical section. Therefore we use mutex (mutual exclusion) to synchronize threads there.

3.2. Multiplication of vector by stiffness matrix

We have implemented a very efficient algorithm of matrix multiplication in case $d = 1$. It stores stiffness matrix with entries

$$d_{ij} = \int_0^1 \psi_i'(x)\psi_j'(x)\,\mathrm{d}x \tag{3}$$

in a constant space with respect to number of levels L and run in a linear time with respect to a matrix order. You can find a description of this algorithm in [8]. We use a tensor product of 1D bases as a multi-dimensional basis

$$\psi_{i_1,\dots,i_d}(x_1,\dots,x_d) = \psi_{i_1}(x_1)\cdots\psi_{i_d}(x_d)$$

and entries of the corresponding stiffness matrix

$$a_{i_1,\dots,i_d,i_1',\dots,i_d'} = \int_{[0,1]^d} \nabla\psi_{i_1,\dots,i_d}\nabla\psi_{i_1',\dots,i_d'}. \tag{4}$$

We derive how to express the matrix **a** through matrices **d** and **g**

$$g_{ij} = \int_0^1 \psi_i(x)\psi_j(x)\,\mathrm{d}x.$$

Note that used spline-wavelet basis is not orthonormal and so **g** is not identity matrix. We put $d = 3$ for the sake of simplicity. Matrix (4) is then given by

$$\begin{aligned} a_{i,j,k,i',j',k'} \;=\; \int_{[0,1]^3} & \psi_i'(x_1)\psi_j(x_2)\psi_k(x_3)\psi_{i'}'(x_1)\psi_{j'}(x_2)\psi_{k'}(x_3) + \\ & +\psi_i(x_1)\psi_j'(x_2)\psi_k(x_3)\psi_{i'}(x_1)\psi_{j'}'(x_2)\psi_{k'}(x_3) + \\ & +\psi_i(x_1)\psi_j(x_2)\psi_k'(x_3)\psi_{i'}(x_1)\psi_{j'}(x_2)\psi_{k'}'(x_3) \end{aligned}$$

and can be expressed as

$$a_{i_1,i_2,i_3,i_1',i_2',i_3'} = d_{ii'}g_{jj'}g_{kk'} + g_{ii'}d_{jj'}g_{kk'} + g_{ii'}g_{jj'}d_{kk'}$$

and multiplication of right-hand side **r** with **a** as

$$\sum_{i',j',k'} \left(d_{ii'}g_{jj'}g_{kk'} + g_{ii'}d_{jj'}g_{kk'} + g_{ii'}g_{jj'}d_{kk'}\right) r_{i'j'k'}. \tag{5}$$

To compute (5) we use the following algorithm

1. Compute $r^0_{ij'k'} = \sum_{i'} g_{ii'}r_{i'j'k'}$ and $r^1_{ij'k'} = \sum_{i'} d_{ii'}r_{i'j'k'}$ as a one-dimensional multiplication for all j', k'.

2. $r^0_{ijk'} = \sum_{j'} g_{jj'}r^0_{ij'k'}$,
   $r^1_{ijk'} = \sum_{j'} g_{jj'}r^1_{ij'k'}$,
   $r^2_{ijk'} = \sum_{j'} d_{jj'}r^0_{ij'k'}$.

3. $r^1_{ijk} = \sum_{k'} g_{kk'} r^1_{ijk'}$,
   $r^2_{ijk} = \sum_{k'} g_{kk'} r^2_{ijk'}$,
   $r^3_{ijk} = \sum_{k'} d_{kk'} r^0_{ijk'}$.

4. $r_{ijk} = r^1_{ijk} + r^2_{ijk} + r^3_{ijk}$.

Then, we have 8 matrix-vector multiplication in steps 1.–3. In each step, all multiplications are independent and are computed in parallel. In the case $d = 4$, we have 13 multiplications in 4 groups and for $d = 5$, we have 19 multiplications in 5 groups.

4. Numerical experiments

We run our code for Poisson equation in dimensions $d \in \{3, 4, 5\}$ with the solution

$$u(x_1, x_2, \ldots, x_d) = (1 - x_1)(1 - x_2) \ldots (1 - x_d) \left(1 - e^{(-10x_1x_2\ldots x_d)}\right).$$

In Table 1, d denotes dimension, L denotes the decomposition level of wavelet basis, N is the matrix size, $RHS16$ (m) and $RHS8$ (m), respectively denotes time of computation of right-hand side integrals in minutes in 16 and 8 threads, respectively, $\#CG$ denotes the number of iterations of the conjugate gradient method and $CG(m)$ denotes time of computation of the conjugate gradient method in minutes. We used for our computation a processor with frequency 2.3 GHz and with 16 cores.

d	L	N	$RHS16(m)$	$RHS8(m)$	#CG	CG (m)	L_2 norm of error
3	8	2^{24}	10	21	177	100	$1.6 \cdot 10^{-11}$
3	9	2^{27}	136	260	199	920	$1.1 \cdot 10^{-12}$
4	5	2^{20}	9	18	161	5	$5.9 \cdot 10^{-9}$
4	6	2^{24}	49	92	203	120	$4.5 \cdot 10^{-10}$
5	4	2^{20}	250	480	128	3	$1.5 \cdot 10^{-8}$
5	5	2^{25}	520	-	176	200	$1.5 \cdot 10^{-9}$

Table 1: Results of numerical experiments.

5. Conclusion

We have presented here some details of our implementation of Wavelet-Galerkin method for Poisson equation in dimension $d \in \{3, 4, 5\}$ in C language parallelized by POSIX threads library. Parallelization of evaluation of right-hand side integrals is efficient – enables concurrent evaluation by as many threads as the number of available computational cores. The ratio of total CPU time and real time is in the case of 16 threads around 15.8. This is not the case for the conjugate gradient method and our future goal is to improve it. Another goal is to design and implement appropriate data structures for adaptive methods.

Acknowledgements

This work has been supported by the project ESF No. CZ.1.07/2.3.00/09.0155 ”Constitution and improvement of a team for demanding technical computations on parallel computers at TU Liberec”

References

[1] Bungartz, H. J. and Griebel, M.: Sparse grids. Acta Numer. **13** (2004), 147–269.

[2] Černá; D. and Finěk, V.: Construction of optimally conditioned cubic spline wavelets on the interval. Adv. Comput. Math. **34** (2011), 519–552, 2011.

[3] Černá; D. and Finěk, V.: The construction of well-conditioned wavelet basis based on quadratic B-splines. To appear In: Simos, T. E. (Ed.) *ICNAAM – Numerical Analysis and Applied Mathematics*, American Institute of Physics, New York, 2012.

[4] Černá, D., Finěk, V., and Šimůnková, M.: A quadratic spline-wavelet basis on the interval. In: Chleboun, J., Segeth, K., Šístek, J., Vejchodský, T. (Eds.), *Programs and Algorithms of Numerical Matematics 16*, pp. 29–34. Institute of Mathematics AS CR, Prague, 2013.

[5] Dijkema, T. J., Schwab, Ch., and Stevenson, R.: An adaptive wavelet method for solving high-dimensional elliptic PDEs. Constr. Approx. **30** (3) (2009), 423–455.

[6] Donovan, G. C., Geronimo, J. S., and Hardin, D. P.: Intertwining multiresolution analyses and the construction of piecewise-polynomial wavelets. SIAM J. Math. Anal. **27** (6) (1996), 1791–1815.

[7] Finěk, V. and Šimůnková, M.: Effective implementation of wavelet Galerkin method. To appear. In: Venkov, G., Kovacheva, R., Pasheva, V. (Eds.), *AMEE – Applications of Mathematics in Engineering and Economics*, American Institute of Physics, New York, 2012.

[8] Šimůnková, M.: *Multiplication by wavelet matrix – efficient implementation.* Submitted to ACC Journal.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

MASSIVE PARALLEL IMPLEMENTATION OF ODE SOLVERS

Cyril Fischer

[1] Institute of Theoretical and Applied Mechanics AS CR, v.v.i.
Prosecká 76, Prague 9, Czech Republic
fischerc@itam.cas.cz

Abstract

The presented contribution maps the possibilities of exploitation of the massive parallel computational hardware (namely GPU) for solution of the initial value problems of ordinary differential equations. Two cases are discussed: parallel solution of a single ODE and parallel execution of scalar ODE solvers. Whereas the advantages of the special architecture in the case of a single ODE are problematic, repeated solution of a single ODE for different data can profit from the parallel architecture. However, special algorithms have to be used even in the latter case to avoid code divergence between individual computational threads. The topic is illustrated on several examples.

1. Introduction

The modern Graphical Processor Unit (GPU) serves as a powerful graphics engine thanks to its highly parallel programmable processor. As a parallel device it features peak arithmetic and memory bandwidth that substantially outpaces its CPU counterpart. Contemporary graphics processors thus operate as co-processors within the host computer.

There are several peculiarities in the GPU architecture from the point of view of a regular PC user, namely rather complicated memory access and parallel architecture of the graphics multiprocessor. Whereas a high level programming interface can unify the memory access up to certain limit, parallel algorithms for GPUs have to be treated in a special way, see e.g. [6].

Graphics processors are built as multithreaded SIMD (single instruction, multiple data) devices. It means that each instruction of the code is preformed on a set of data at once. Any exception in data treatment (like data dependent branches) results in splitting of the program flow and leads to a significant degradation of performance.

2. Parallel solution of an ordinary differential equation

Solution of an ordinary differential equation is a sequential problem in its nature. There are only several parallelisable points in a general procedure of an ODE solver, usually regarding evaluation of the integrated functions.

In his exhaustive review paper [2] Burrage follows the classification of the seminal Gear's work [4]. He divides the techniques into two different categories: parallelism across the method and parallelism across the system. The first group comprises methods which exploit concurrent function evaluations within one ore more steps. The second group includes methods based on waveform relaxation. These methods decouple the original system into a set of small and independent subsystems which can then be solved in parallel.

According to literature survey, both categories are still being developed, but not in conjunction with GPU computation. The high cost of communication between CPU and GPU and demand of synchronous operation on the whole set of data make this computational environment specific. To achieve improvement if a GPU is used, any of the methods mentioned above ought to request thousands of function evaluation for each step. On the other hand, if the dimension of the coupled ODE system is large, a single RHS evaluation can exploit GPU accelerated matrix/vector multiplication, matrix inverse evaluation etc.

Only a few papers dealing with ODEs and GPU are availalabe up to now. They mostly reflect problems originating from biomechanics or chemistry, see e.g. [10] or are dealing with evolutionary differential equations. Only three projects can be found on internet which aim at implementation ODE solvers to the GPU hardware:

CULSODA [3] is an adaptation of the highly sophisticated algorithm LSODA (adaptive steplength, automatic stiff/non-stiff method switching etc, see [5]) for NVIDIA's CUDA compiler. It does not contain any parallel code in itself. The procedure can be used to perform a parameter study of a single ODE system. As it will be shown in the next section, usability of the CULSODA code is rather limited. It seems that the project has been abandoned since 2009.

ODEINT_V2 [1] is a general C++ library for numerical solution of ODE. With most integration methods it offers exploitation of CUDA capable hardware. The authors claim that the GPU is worth to employ in the following applications: parameter studies, large systems like ensembles of lattices, discretizations of PDEs, etc.

CUDA-sim [9, 10] is a Python package providing CUDA GPU-accelerated biochemical network simulation. The package offers ODE solver based on CULSODA implementation and stochastic differential equation solver according to the Euler-Maruyama algorithm. It's usage is limited to the case when a large number of independent ODEs are to be solved en bloc.

It seems that none of the sophisticated methods mentioned in [2] or newer papers is implemented in the available libraries. On the other hand, there are certain tasks which are based on solution of a large number of independent ODEs or systems. Even if the beautiful theory of parallel ODE solvers cannot be employed in this cases, existing cheap GPUs could significantly speed up the computation in such a situation as it will be shown in the next section.

3. Example

Two tasks routinely performed by engineers are good candidates for a parallel processing example: computation of *response spectra* and *resonance curve*.

Response spectra $\rho(\omega)$ of recorded time history is a plot of the peak or steady-state response of a series of oscillators of varying natural frequency ω^2 that are forced into motion by the recorded signal $a(t)$, cf. (1). The natural frequency of the oscillators is taken as the independent variable, coefficient of damping β is predefined as a parameter, see e.g. [8]. Resonance curve $R(\omega)$ is a similar plot of the peak or steady-state response of a structure described by a system of differential equations $f(t)$, that is forced into motion by a harmonic function $a_0 \sin(\omega t)$. In this case the independent variable is the frequency ω of the harmonic input motion (2) .

$$\rho(\omega) = \max_{0<t<T} |y(t)|\,, \qquad \text{where } \ddot{y} + 2\beta\dot{y} + \omega^2 y = -\ddot{a} \tag{1}$$

$$R(\omega) = \max_{0<t<T} |y(t)|\,, \qquad \text{where } f(y, \dot{y}, \ddot{y}, \dots, t) = a_0 \sin(\omega t) \tag{2}$$

The both problems are similar and lead to solution of a large number of independent initial value problems. In this case, the parallel computation is an obvious choice.

Evaluation of the resonance spectra will be illustrated using the following example. The equation (3) describes movement of the mathematical pendulum with an external excitation in the suspension point (see the detailed derivation in [7]):

$$\left.\begin{array}{lcll}\ddot{\xi} + \dfrac{1}{2r^2}\xi\dfrac{\mathsf{d}^2}{\mathsf{d}t^2}(\xi^2+\zeta^2) + 2\beta_\xi\dot{\xi} + \omega_0^2\left(\xi + \dfrac{1}{2r^2}\xi(\xi^2+\zeta^2)\right) & = & -\ddot{a} & (a) \\ \ddot{\zeta} + \dfrac{1}{2r^2}\zeta\dfrac{\mathsf{d}^2}{\mathsf{d}t^2}(\xi^2+\zeta^2) + 2\beta_\zeta\dot{\zeta} + \omega_0^2\left(\zeta + \dfrac{1}{2r^2}\zeta(\xi^2+\zeta^2)\right) & = & 0 & (b)\end{array}\right\} \tag{3}$$

where ξ, ζ are components of the projection of the pendulum's bob to the (xy) plane, r is the length of the pendulum, $\omega_0^2 = g/r$ is the natural frequency of the corresponding linear pendulum and g is the gravitational acceleration. The viscous damping is denoted as β_ξ, β_ζ in respective directions. As the harmonic excitation $a(t) = a_0 \sin(\omega t)$ acts in the ξ direction only, the basic type of motion takes course in the vertical (xz) plane if the time history starts under homogeneous initial conditions. With the increasing amplitude of the excitation $a(t)$, the in-plane movement can lose its stability and movement in the transversal direction ζ can occur.

If the resonance curve is to be computed, it is necessary to perform numerical solution of the system (3) for a large number of excitation frequencies ω. Two numerical methods enter into comparison: the CULSODA solver and in-house implemented simple backward Euler solver.

Table 1 shows the timings for CULSODA solver and various numbers of ODE solution threads. Two experiments are shown: the real case, when each thread solves ODE for its own excitation frequency, and the unrealistic one when all threads do exactly the same work (the frequency ω is the same for all the threads). The first case shows the devastative effect of the thread divergence on the overall performance

# values	1024	8192	16 384	32 768	1024	8192	16 384	32 768
	thread divergence ($\omega = 1, ..., 10$)				no thread divergence ($\omega = 1$)			
GPU (sec)	19.2	40.7	70.7	127.3	0.4	0.9	1.7	3.4
CPU (sec)	0.54	4.21	8.54	16.4	0.4	2.8	6.0	11.1

Table 1: Timing of the resonance curve enumeration using CULSODA in single prec. CPU: 2× Intel Xeon X5560 (16 threads), GPU: NVIDIA Quadro 4000 (256 cores).

# values	64	1024	8192	16 384	32 768
GPU (sec)	1.75	1.93	2.97	5.84	11.06
CPU (sec)	0.74	8.89	72.49	143.21	295.1

Table 2: Timing of the resonance curve enumeration using simple backward Euler method with constant step length. CPU: 2× Intel Xeon X5560 (16 threads, single precision), GPU: NVIDIA Quadro 4000 (256 cores, single precision).

of the GPU. The second case shows the theoretical potential of the GPU. Starting from certain problem size (1000 threads in this example) GPU is processing faster than CPU. The thread divergence in the first case is caused by two reasons: (a) the equation significantly changes its properties for growing ω and (b) the LSODA code changes its course accordingly.

The adaptivity of the LSODA algorithm is a great disadvantage when used in the GPU code. To eliminate the thread divergence a simple backward Euler ODE solver has been implemented. To avoid any unnecessary jumps and conditions, the linear equation solver procedure used in the Newton method has been hard-coded for a predetermined dimension using the Crammer rule. Number of iterations of the Newton method has been fixed. The method exhibits reasonable accuracy for a sufficiently small step. Table 2 lists the corresponding timings for the backward Euler solver. There is no doubt about the winner in this case: starting with 256 computational threads the GPU becomes significantly faster.

The both results are not intended to compare the individual methods. In a scalar case the LSODE algorithm is faster. In order to keep the accuracy reasonable the backward Euler code uses much smaller step size than the LSODE solver. Moreover, whereas the LSODE requires about 2 function evaluation for each step and one Jacobian for (about) 10 steps, the backward Euler evaluates $f(y, t)$ and the Jacobian in each step and for each iteration of the Newton method.

The response spectrum is usually computed using the Newmark method. The Newmark method is an explicit second order method whose recurrence is tailored for the second order equation of motion, see (1). It is especially convenient in the (usual) case when some measured record enters the computation as a set of regularly sampled discrete values. It supposes a fixed length of the integration step. Figure 1 shows timing of the implementation for different numbers of frequencies. As the simple

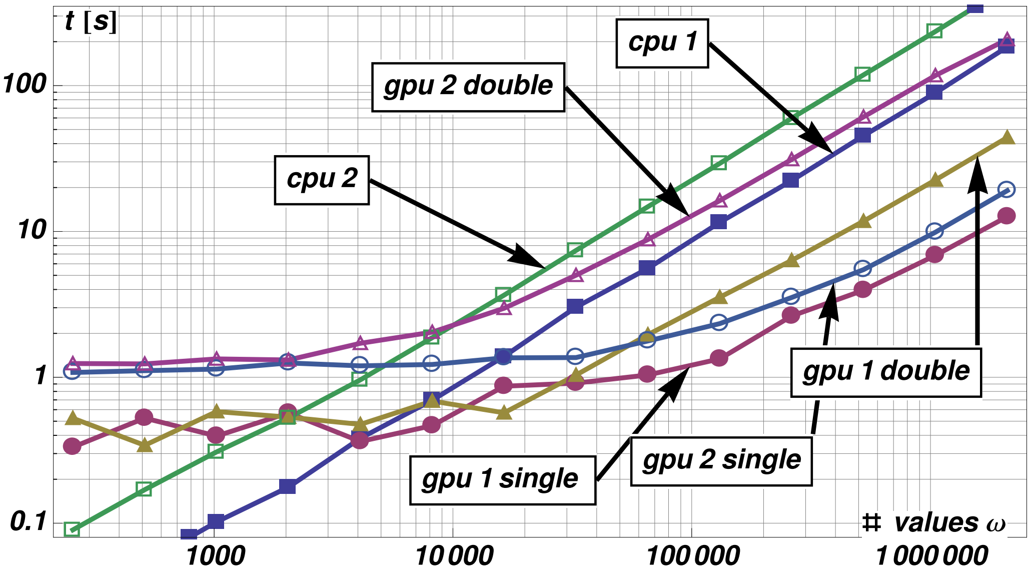


Figure 1: Timing of response spectra enumeration (El Centro earthquake record, 31 000 samples, $\Delta t = 0.001$) using the Newmark recurrence formula. The CPU timing for single and double precision does not differ.
CPU 1: 2× Intel Xeon X5560 (16 threads), GPU 1: NVIDIA Quadro 4000 (256 cores). CPU 2: Intel i7 950 (8 threads), GPU 2 : NVIDIA GT430 (64 cores)

Newmark recurrence does not contain any branches, thread divergence cannot occur in this case.

The speed difference between single and double precision arithmetic is shown in Figure 1. The performance penalty for double precision is 1/2 for the advanced NVIDIA Quadro 4000 GPU and 1/8 for the entry level GT430. The ratios correspond well to the architectures of the both cards (one double precision floating point unit is common for 2 cores in the high end NVIDIA GPUs or for 8 cores in GT430). However, even the slower card can compete well with the dual Xeon workstation for well chosen problems.

4. Conclusions

The great progress of the computer technology enables the scientific community to solve fairly complex problems. Current GPUs are cheap devices with power of a supercomputer. This demands the scientists to adopt higher level of knowledge of programming techniques.

We have shown two examples of GPU utilization for numerical solution of initial value problems. Three numerical methods were presented: i) The adaptive solver LSODA provided to be the least advantageous for GPU. ii) As an alternative, the backward Euler method is reasonably accurate and stable and can be programmed in a manner which meet requirements of a fast GPU code. iii) Simple recurrence formula, simple arithmetic, no special functions, no jumps or branches and no inter-thread communications make the Newmark procedure an ideal candidate for employment of GPU.

It seems that the advanced methods for solution of a single large ODE system are not very convenient when used for machines with SIMD architecture. However, the presented examples exhibited possibility of fruitful utilization GPUs in practice. It has been shown that in certain cases even an entry-level GPU can work faster than a high-end CPU.

Acknowledgements

The kind support of the Czech Scientific Foundation No. 103/09/0094, Grant Agency of the ASCR No. A200710902 and RVO 68378297 institutional support are gratefully acknowledged.

References

[1] Ahnert, K. and Mulanski, M.: ODEINT_v2, `http://www.odent.com/`, 2012

[2] Burrage, K.: Parallel methods for initial value problems. Applied Numerical Mathematics. **11** (1993), 5–25.

[3] CULSODA, `http://code.google.com/p/culsoda/`, 2009

[4] Gear, C. W.: Parallel methods for ordinary differential equations. Calcolo **25** (1988), 1–20.

[5] Hindmarsh, A. C.: ODEPACK, A Systematized Collection of ODE Solvers, Scientific Computing. In Stepleman,R. S. et al. (Eds.) *IMACS Transactions on Scientific Computation*, vol. 1. , pp. 55–64. Amsterdam, North-Holland, 1983.

[6] Kirk, D. B. and Hwu, W. W.: *Programming Massively Parallel Processors: A Hands-on Approach.* Morgan Kaufmann Publishers, Burlington, 2010.

[7] Náprstek, J. and Fischer, C.: Auto-parametric semi-trivial and post-critical response of a spherical pendulum damper. Comp. Struct. **87** (2009). 1204–1215.

[8] Newmark, N. M. and Hall, W. J.: Earthquake Spectra and Design, *Engineering Monographs on Earthquake Criteria, Str uctural Design, and Strong Motion Records*, vol. 3. Earthquake Eng. Res. Inst., Oakland, California, 1982.

[9] Zhou, Y. and Barnes, C.: CUDA-sim, `http://www.theosysbio.bio.ic.ac.uk/ /resources/cuda-sim/`, Version 0.07, 21/12/2011

[10] Zhou, Y., Liepe, J., Sheng, X., Stumpf, M. P. H. and Barnes, C.: GPU accelerated biochemical network simulation. Bioinformatics **27** (2011), 874–876.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

A NEW PERSPECTIVE ON SOME APPROXIMATIONS USED IN NEUTRON TRANSPORT MODELING

Milan Hanuš

University of West Bohemia
Univerzitni 22, Pilsen, Czech Republic
mhanus@kma.zcu.cz

Abstract

In this contribution, we will use the *Maxwell-Cartesian spherical harmonics* (introduced in [1, 2]) to derive a system of partial differential equations governing transport of neutrons within an interacting medium. This system forms an alternative to the well known P_N approximation, which is based on the expansion into tesseral spherical harmonics ([3, p. 197]). In comparison with this latter set of equations, the Maxwell-Cartesian system posesses a much more regular structure, which may be used for various simplifications that could be advantageous from computational point of view.

1. Introduction

Consider the monoenergetic, steady-state neutron transport problem with fixed volumetric sources[1] in a domain $V \subset \mathbb{R}^3$ filled with isotropic medium interacting with the neutrons. Solution of this problem describes the stationary distribution of neutrons within V together with their motion directions and is called *angular neutron flux density* (or shortly *angular flux*). In standard notation, the angular flux is expressed by function $\psi(\mathbf{r}, \mathbf{\Omega})$ where $\mathbf{r} = [x, y, z]^T \in V$ and components of the direction vector are given in spherical coordinates as

$$\mathbf{\Omega} = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{bmatrix} = \mathbf{\Omega}(\vartheta, \varphi), \quad \vartheta \in [0, \pi],\ \varphi \in [0, 2\pi).$$

Angular neutron flux is therefore a function defined in a five-dimensional domain $V \times S_2$ (S_2 denoting the unit 2-sphere). Numerical methods that can be used to determine the solution of practically significant neutron transport problems are usually constructed by first semi-discretizing the governing equation with respect to the angular variable $\mathbf{\Omega}$, yielding a system of PDE's in space, and using standard numerical methods like the finite volume or finite element methods to solve this system.

[1] The extension to energy- or time-dependent problems would be straightforward.

In this paper, we will be interested only in the angular semi-discretization of the neutron transport equation. One of the most popular method to accomplish this task is the *method of spherical harmonics*. In this method, the angular flux is expanded into a series of tesseral spherical harmonics ([3, p. 197]) which form a complete orthonormal basis of the space $L^2(S_2)$ of functions square-integrable with respect to $\mathbf{\Omega}$. By considering only spherical harmonics of degrees $n \leq N$ [2] and performing the Galerkin projection (with respect to $\mathbf{\Omega}$) of the neutron transport equation onto the subspace spanned by those functions, a system of first-order hyperbolic PDE's in space, conventionally referred to as the P_N system, is obtained[3]. However, the resulting system is quite complicated, strongly coupled through differential terms and lacks the invariance with respect to change of coordinate axes.

We therefore propose to use a different set of expansion functions to perform the angular semi-discretization. In [7], the author arrived at a "far more symmetric and compact" ([7, p. 1455]) form of the angularly semi-discretized system of equations governing the distribution of plasma by using an expansion of the solution in terms of special spherical harmonic tensors rather than the usual sets of tesseral spherical harmonics. Although the derivation and properties of the spherical harmonic tensors were not described in much detail in the paper, they are actually equal (up to a normalization) to the *Maxwell-Cartesian spherical harmonic tensors*, rigorously developed in [1]. Use of these tensors have so far proven to be advantageous in solving various electro-magnetics and quantum-mechanical problems ([1, 2]). Nevertheless, they have not been used for neutron transport problems until very recently ([5]).

In section 2, we will introduce the neutron transport equation and basic notation. Section 3 provides a brief review of the Maxwell-Cartesian spherical harmonic tensors, leaving the details to the original papers [1, 2]. The alternative to the P_N system is derived in section 4; the derivation is different from that of [5] and, in our opinion, provides more insight into the structure of the equations. How this insight can be used to obtain in a new way some known (but not yet completely understood) approximations used in neutron transport methods is suggested in the conclusion.

2. Mathematical model

The equation governing transport of neutrons, also known as the linear Boltzmann's transport equation (shortly BTE), reads (in standard notation, as e.g. in [11, Sec. 9.7]):

$$\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega})+\sigma_t(\mathbf{r})\psi(\mathbf{r},\mathbf{\Omega})-\int_{S_2}\left[\sigma_s(\mathbf{r},\mathbf{\Omega}\cdot\mathbf{\Omega}')+\frac{\nu(\mathbf{r})\sigma_f(\mathbf{r})}{4\pi}\right]\psi(\mathbf{r},\mathbf{\Omega}')\,\mathrm{d}\mathbf{\Omega}' = q(\mathbf{r},\mathbf{\Omega}), \tag{1}$$

where $\mathbf{r} \in V$, $\mathbf{\Omega} \in S_2$. The σ functions are called *macroscopic cross-sections* for a particular interaction, distinguished by the subscripts (t for the total probability

[2] Note that there are $2n+1$ tesseral harmonics of given degree $n \geq 0$.

[3] Detailed description is given in many classical books on nuclear reactor physics, like [11, 4].

of collision of any type with nuclei at $\mathbf{r}$, s for scattering from direction $\mathbf{\Omega}'$ to $\mathbf{\Omega}$ upon the collision and finally f for the collision which results in fissioning the target nucleus and releasing an average number of ν neutrons in the process). We will assume that all the macroscopic cross-sections are given bounded measurable functions and look for a non-negative $\psi \in L^2(V \times S_2)$ [4] given the volumetric distribution of neutron sources $q \in L^2(V \times S_2)$. Boundary conditions will not be considered here – we may remark, however, that the developments in this work make the incorporation of the well-known *Marshak approximation* of the boundary conditions (e.g., [11, p. 340]) straightforward.

3. Maxwell-Cartesian spherical harmonics

A general linear combination of tesseral harmonics of given degree n is called *surface spherical harmonic* of degree n. As shown in [1], a surface spherical harmonic of degree n can also be uniquely represented by a totally symmetric traceless Cartesian tensor of rank n [5] . Moreover, that paper presents a systematic way of obtaining a TST tensor of any rank n whose components are surface spherical harmonics of degree n in Cartesian frame of reference as defined by Maxwell in [8, p. 160]. Specifically, Maxwell's spherical harmonics based on Cartesian axes can be obtained (up to a normalization constant) as components of $\mathbb{P}^{(n)}(\mathbf{\Omega}) = \mathscr{D}_n\mathbf{\Omega}^n$ where $\mathscr{D}_n$ is the so-called *detracer operator* which projects a general totally symmetric tensor of rank n into the space of TST tensors of rank n (definition and various properties of this operator are given in [1, Sec. 5]; we use here the "projection version" of the operator, as discussed in the note in [1, p. 4311]).

We note that projection of $\mathbb{P}^{(n)}$ along, say, z-axis (or any other because of the symmetry) yields (up to a normalization factor) the Legendre polynomials P_n (cf. Tab. 1):

$$\mathbb{P}^{(n)}(\mathbf{\Omega}) \cdot \mathbf{e}_z^n = P^{(n)}_{\alpha_1 \dots \alpha_n}(\mathbf{\Omega})\delta_{3\alpha_1} \cdots \delta_{3\alpha_n} = P^{(n)}_{33\dots 3}(\mathbf{\Omega}) = \frac{n!}{(2n-1)!!} P_n(\Omega_z) \equiv C_n P_n(\Omega_z) \tag{2}$$

[4] We denote by $L^2(V \times S_2)$ the space of square-integrable functions with respect to the measure $\mathrm{d}\mu(V \times S_2) = \mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{\Omega} = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,\sin\vartheta\mathrm{d}\vartheta\,\mathrm{d}\varphi$.

[5] We will indicate a Cartesian tensor of rank n by superscribed (n) and index its 3^n components by a sequence of n Greek letters in subscript (each attaining value 1, 2, or 3, corresponding to Cartesian axes x, y, z, respectively); for vectors (rank-1 tensors), we will keep using conventional bold-face letters. Einstein's summation convention will be used whenever same indices appear in a tensor expression written in component notation. When a tensor is invariant under any permutation of its indices, it is called *totally symmetric*; contraction of two tensors of same rank is a number $\mathbb{A}^{(n)} \cdot \mathbb{B}^{(n)} := A^{(n)}_{\gamma_1\dots\gamma_n} B^{(n)}_{\gamma_n\dots\gamma_1}$, contraction of a tensor $\mathbb{P}^{(n)}$ in first index pair is defined as $P^{(n)}_{\alpha\alpha\gamma_3\dots\gamma_n}$ and called *trace* in that index pair. Totally symmetric tensor whose trace in any index pair vanishes is called *totally symmetric traceless* (abbr. TST). The tensor product $\mathbb{C}^{(n+m)} = \mathbb{A}^{(n)} \otimes \mathbb{B}^{(m)}$ has components $C^{(n+m)}_{\alpha_1\dots\alpha_n\beta_1\dots\beta_m} = A^{(n)}_{\alpha_1\dots\alpha_n} B^{(m)}_{\beta_1\dots\beta_m}$ and the m-th power of a rank-n tensor is defined as $\mathbb{A}^{(n)} \otimes \mathbb{A}^{(n)} \otimes \cdots \otimes \mathbb{A}^{(n)}$ (m-times). Finally, we will consider the differential operator ∇ as a vector $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$, but keep writing $\nabla\mathbb{A}^{(n)}$ for $\nabla \otimes \mathbb{A}^{(n)}$.

($\mathbf{e}_z = [0, 0, 1]^T$, δ_{ij} is the Kronecker delta); also, the well-known formulas for Legendre polynomials could be extended within the tensorial framework to obtain explicit formulas for $\mathbb{P}^{(n)}$([3, Chap. VI]). This feature of Maxwell-Cartesian tensors makes the resulting angular discretization a natural multidimensional extension of the 1D P_N system (which is actually based on a Legendre expansion of angular flux), having an analogous form as the latter.

n	$\mathbb{Y}^n(\boldsymbol{\Omega}) \propto$	$\mathbb{P}^{(n)}(\boldsymbol{\Omega})$	$C_n P_n(\Omega_z)$
0	1	1	1
1	$\Omega_x,\ \Omega_y, \Omega_z$	$\Omega_x,\ \Omega_y,\ \Omega_z$	Ω_z
2	$-\Omega_x^2 - \Omega_y^2 + 2\Omega_z^2,\ \Omega_y\Omega_z,$	$\Omega_x^2 - \frac{1}{3},\ \Omega_x\Omega_y,\ \Omega_x\Omega_z,$	$\Omega_z^2 - \frac{1}{3}$
	$\Omega_z\Omega_x,\ \Omega_x\Omega_y,\ \Omega_x^2 - \Omega_y^2$	$\Omega_y^2 - \frac{1}{3},\ \Omega_y\Omega_z,\ \Omega_z^2 - \frac{1}{3}$	

Table 1: Tesseral and Maxwell-Cartesian sph. harmonics and Legendre polynomials.

4. The MCP$_N$ approximation

Using the results of [3, Art. 114], the expansion of angular neutron flux in terms of surface spherical harmonics could be written as

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi} \int_{S_2} \psi(\mathbf{r}, \boldsymbol{\Omega}') P_n(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \, \mathrm{d}\boldsymbol{\Omega}' \,. \tag{3}$$

As shown in [1, Sec. 7, Corollary II], Maxwell-Cartesian tensors appear in the following form of “addition theorem” for Legendre polynomials P_n:

$$\mathbb{P}^{(n)}(\boldsymbol{\Omega}) \cdot \mathbb{P}^{(n)}(\boldsymbol{\Omega}') = C_n P_n(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \tag{4}$$

(C_n defined in (2)). Combining the two results, we obtain

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{n=0}^{\infty} \psi^{(n)}(\mathbf{r}) \cdot \mathbb{P}^{(n)}(\boldsymbol{\Omega}), \tag{5}$$

where we defined

$$\psi^{(n)}(\mathbf{r}) := \frac{2n+1}{4\pi C_n} \int_{S_2} \psi(\mathbf{r}, \boldsymbol{\Omega}) \mathbb{P}^{(n)}(\boldsymbol{\Omega}) \, \mathrm{d}\boldsymbol{\Omega} \,. \tag{6}$$

To find the relations that must be satisfied by the angular expansion moments $\psi^{(n)}(\mathbf{r})$ in order for (5) (or equivalently (3)) to be the solution of the BTE, we insert the expansion (5) into (1) (with source term represented in terms of angular expansion moments analogously to (6)). Applying eq. (4) to the generalized Fourier-series expansion of the scattering term in terms of the Legendre polynomials, we obtain:

$$\begin{aligned}\int_{S_2} \sigma_s(\mathbf{r}, \boldsymbol{\Omega}\cdot\boldsymbol{\Omega}')\psi(\mathbf{r},\boldsymbol{\Omega}')\,\mathrm{d}\boldsymbol{\Omega}' &= \sum_{n=0}^{\infty} \frac{2n+1}{4\pi}\sigma_{sn}(\mathbf{r}) \int_{S_2} P_n(\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi(\mathbf{r},\boldsymbol{\Omega}')\,\mathrm{d}\boldsymbol{\Omega}' \\ &= \sum_{n=0}^{\infty} \sigma_{sn}(\mathbf{r})\psi^{(n)}(\mathbf{r})\cdot\mathbb{P}^{(n)}(\boldsymbol{\Omega}).\end{aligned}$$

Because of the orthogonality of $\mathbb{P}^{(n)}$ of different ranks ([1, Sec. 8.7]), the fission part will have the following form:

$$\int_{S_2} \frac{\nu(\mathbf{r})\sigma_f(\mathbf{r})}{4\pi}\psi(\mathbf{r},\boldsymbol{\Omega}')\,\mathrm{d}\boldsymbol{\Omega}' = \sum_{n=0}^{\infty} \frac{\nu(\mathbf{r})\sigma_f(\mathbf{r})}{4\pi}\psi^{(n)}(\mathbf{r})\cdot \int_{S_2} \mathbb{P}^{(n)}(\boldsymbol{\Omega}')\,\mathrm{d}\boldsymbol{\Omega}' = \nu(\mathbf{r})\sigma_f(\mathbf{r})\phi(\mathbf{r}), \tag{7}$$

where $\phi \equiv \psi^{(0)}$ is the *scalar flux*. Therefore, by inserting (5) into (1) and using these results we obtain

$$\sum_{n=0}^{\infty} \left[\boldsymbol{\Omega}\cdot\nabla\psi^{(n)} + \sigma_t\psi^{(n)} - \sigma_{sn}\psi^{(n)} - \delta_{n0}\nu\sigma_f\phi - q^{(n)}\right]\cdot\mathbb{P}^{(n)}(\boldsymbol{\Omega}) = 0 \tag{8}$$

where each term in the square brackets is dependent only on $\mathbf{r}$ (omitted for brevity).

Because of the linear dependence among certain functions in each $\mathbb{P}^{(n)}(\boldsymbol{\Omega})$ (owing to the requirement of vanishing trace), we cannot deduce from (8) that for each n, all components of the tensor in brackets must vanish. The angular discretization is further hampered by the advection term which still contains the angular variable $\boldsymbol{\Omega}$. However, using the *detracer exchange theorem* ([2, Sec. 5.2]) and total symmetry of $\psi^{(n)}$ (by definition (6)), we note that the expansion (5) is actually equivalent to a power series in $\boldsymbol{\Omega}$:

$$\psi(\mathbf{r},\boldsymbol{\Omega}) = \sum_{n=0}^{\infty} \psi^{(n)}(\mathbf{r})\cdot\boldsymbol{\Omega}^n. \tag{9}$$

Using this fact to simplify the advection term $(\boldsymbol{\Omega}\cdot\nabla)\psi^{(n)}\cdot\boldsymbol{\Omega}^n$, we may rewrite equation (8) as

$$\sum_{n=0}^{\infty} \left[\nabla\psi^{(n-1)} + \sigma_t\psi^{(n)} - \sigma_{sn}\psi^{(n)} - \delta_{n0}\nu\sigma_f\phi - q^{(n)}\right]\cdot\boldsymbol{\Omega}^n = 0 \tag{10}$$

with the term with $n < 0$ discarded.

Equation (10) expresses a vanishing linear combination of monomials restricted to the unit sphere. Even though the monomials of all degrees are completely linearly independent, once restricted to the unit sphere there exist nontrivial linear combinations, such as $\Omega_x^2+\Omega_y^2+\Omega_z^2-1=0$. Hence we still cannot deduce that the expression in square brackets in (10) must be a zero tensor. However, in view of the theorem in [1, Sec. 4.2], it is possible to eliminate these nontrivial linear combinations by requiring the coefficients of the combination to form TST tensors for each n. Since $\psi^{(n)}$ and $q^{(n)}$ are TST by definition, we only have to symmetrize and detrace the advection terms. We need to be careful, however, not to change the original equation. This can be done by a clever rearranging of the terms in the sum. After using the definition of the detracer operator, symmetrization by

$$\left[\mathbb{A}^{(n)}\right]_{\text{sym}} = \widetilde{\mathbb{A}}^{(n)} \quad \text{with components} \quad \widetilde{A}^{(n)}_{\alpha_1 \ldots \alpha_n} = \frac{1}{n!} \sum_{\pi(\alpha_1 \ldots \alpha_n)} A^{(n)}_{\alpha_1 \ldots \alpha_n}$$

(where the sum is over all permutations of the tensor indices) and regrouping the sum by $\mathbf{\Omega}^n$, we arrive at the final equation (with $\Sigma_n := \sigma_t \psi^{(n)} - \sigma_{sn}\psi^{(n)} - \delta_{n0}\nu\sigma_f$)

$$\sum_{n=0}^{\infty} \left\{ \left[\nabla\psi^{(n-1)} - \frac{n-1}{2n-1} \mathbb{I} \otimes \nabla \cdot \psi^{(n-1)} \right]_{\text{sym}} + \frac{n+1}{2n+3} \nabla \cdot \psi^{(n+1)} + \Sigma_n \psi^{(n)} - q^{(n)} \right\} \cdot \mathbf{\Omega}^n = 0, \tag{11}$$

which implies that each component of the TST coefficient tensor of rank n in curly brackets must vanish.

5. Conclusion and outlook

By truncating the expansion (9) (or (5)) at $n = N$ for some $N \geq 0$, we obtain from eq. (11) an alternative set to the ordinary P_N equations which could be called an *MCP$_N$ approximation* (because its solution represents the expansion of angular flux into Maxwell-Cartesian surface spherical harmonics of degrees up to N). The symmetric and traceless structure of the MCP_N equations could be used to provide new perspectives on some other widely used approximations or to create new ones. For instance, by projecting each tensor along any chosen axis, we obtain one dimensional equations equivalent with the 1D P_N equations (after suitable normalization of $\psi_z^{(n)}$). This indicates the possibility to investigate the original ad-hoc derivation of the popular SP_N approximation (by formal extension of the 1D P_N equations into 3D, [6]) in the current tensorial framework. Similarly, a different normalization of $\psi^{(n)}$ leads to the system derived in [5]. However, the derivation in [5] is partially formal and does not take into account all the important properties of spherical harmonic tensors (in particular their tracelessness and linear dependence for given degree).

There is also an interesting link to an old article of Selengut ([10]) in which the full multidimensional P_3 solution is obtained by solving a set of two coupled diffusion equations[6] with special interface conditions in presence of multiple heterogeneous regions. Selengut's derivation of the set is however quite puzzling (see also commentary

[6] much like in the SP_3 approximation, but apparently without restrictions on dimensionality or cross-sections other than the usual isotropic scattering and volumetric source assumptions

in [9, Sec. 5.2]) and his equations have never been either analyzed or at least numerically tested. On the other hand, we have been able to derive Selengut's equation for scalar flux (even with anisotropic scattering) by combining the MCP$_3$ equations into an equation for $\psi^{(2)}$ and adding a *compatibility condition* of vanishing trace of $\psi^{(2)}$. Further work in this direction is currently under way.

Acknowledgements

This work relates to Department of the Navy Grant N62909-11-1-7032 issued by Office of Naval Research Global.

References

[1] Applequist, J.: Traceless cartesian tensor forms for spherical harmonic functions: new theorems and applications to electrostatics of dielectric media. J. Phys. A **22** (1989), 4303–4330.

[2] Applequist, J.: Maxwell-Cartesian spherical harmonics in multipole potentials and atomic orbitals. Theoretical Chemistry Accounts **107** (2002), 103–115.

[3] Byerly, W. E.: *An elementary treatise on Fourier's series and spherical, cylindrical and ellipsoidal harmonics, with applications to problems in mathematical physics.* Ginn and Co., Boston, USA, 1893, 2nd edn.

[4] Cacuci, D. G.: *Handbook of nuclear engineering, Volume I: Nuclear engineering fundamentals.* Springer Science + Business Media LLC, 2010.

[5] Coppa, G. G. M.: Deduction of a symmetric tensor formulation of the P_n method for the linear transport equation. Progress in Nuclear Energy **52** (2010), 747–752.

[6] Gelbard, E. M.: Application of spherical harmonics methods to reactor problems. Tech. Rep. WAPD-BT-20, Bettis Atomic Power Laboratory, 1960.

[7] Johnston, T. W.: General spherical harmonic tensors in the Boltzmann equation. J. Math. Phys. **7** (1966), 1453–1458.

[8] Maxwell, J. C.: *A treatise on electricity and magnetism*, vol. I. Clarendon Press, Oxford, UK, 1873.

[9] McClarren, R. G.: Theoretical aspects of the simplified P_n equations. Transport Theory Statist. Phys. **39** (2011), 73–109.

[10] Selengut, D. S.: A new form of the P3 approximation. Transactions of American Nuclear Society **13** (1970), 625–626.

[11] Stacey, W. M.: *Nuclear reactor physics.* John Wiley & Sons, Inc., New York, 2007, 2nd edn.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

AN EXTENSION OF SMALL-STRAIN MODELS TO THE LARGE-STRAIN RANGE BASED ON AN ADDITIVE DECOMPOSITION OF A LOGARITHMIC STRAIN

Martin Horák, Milan Jirásek

Faculty of Civil Engineering, CTU in Prague
Thákurova 7/2077, 166 29 Praha 6 Dejvice, Czech Republic
Martin.Horak@fsv.cvut.cz, Milan.Jirasek@fsv.cvut.cz

Abstract

This paper describes model combining elasticity and plasticity coupled to isotropic damage. However, the the conventional theory fails after the loss of ellipticity of the governing differential equation. From the numerical point of view, loss of ellipticity is manifested by the pathological dependence of the results on the size and orientation of the finite elements. To avoid this undesired behavior, the model is regularized by an implicit gradient formulation. Finally, the constitutive model is extended to the large-strain regime. The large strain model is based on the additive decomposition of the logarithmic strain and preserves the structure of the small-strain theory.

1. Introduction

In this proceedings we will explore an extension of the small strain models combining elasticty and plasticity with isotropic damage, to the large strain regime. The extension to the large strain regime is based on the additive decomposition of the logarithmic strain into elastic and plastic part. The main acctractivness of this approach is in the modular framework consisting from three steps:

1. Definition of the elastic and plastic part of the logarithmic strain.

2. Computation of the generalized stress tensor, energy conjugated to the logarithmic strain and appropriate generalized stiffness via an algorithm that preserves structure of the small strain theory.

3. Transformation of the generalized tensors to the second Piola Kirchhoff stress and appropriate stiffness.

2. Constitutive model

In this section a model combining elasto-plasticity coupled with isotropic damage is described. The main feature of plasticity models is irreversibility of plastic strain while irreversible processes related to damage lead to degradation of stiffness. The basic equations include an additive decomposition of total strain into elastic (reversible) part and plastic (irreversible) part,

$$\varepsilon_{ij} = \varepsilon^e_{ij} + \varepsilon^p_{ij}, \tag{1}$$

the stress strain law,

$$\sigma_{ij} = (1 - \omega\left(\kappa\right))\,\bar{\sigma}_{ij} = (1 - \omega\left(\kappa\right))\,D^e_{ijkl}\varepsilon^e_{kl}, \tag{2}$$

loading-unloading conditions in Kuhn-Tucker form,

$$f(\bar{\sigma}_{ij},\kappa) \leq 0 \qquad \dot{\lambda} \geq 0 \qquad \dot{\lambda} f(\bar{\sigma}_{ij},\kappa) = 0, \tag{3}$$

evolution laws for plastic strain,

$$\dot{\varepsilon}^p_{ij} = \dot{\lambda}\frac{\partial f}{\partial \bar{\sigma}_{ij}}, \tag{4}$$

and for cumulated plastic strain,

$$\dot{\kappa} = \sqrt{\dot{\varepsilon}^p_{ij}\dot{\varepsilon}^p_{ij}}, \tag{5}$$

the law governing the evolution of the damage variable,

$$\omega(\kappa) = \omega_c(1 - \mathrm{e}^{-a\kappa}), \tag{6}$$

and the hardening law,

$$\sigma_Y(\kappa) = 1 + \sigma_H(1 - \mathrm{e}^{-s\kappa}). \tag{7}$$

In the equations above, $\bar{\sigma}_{ij}$ is the effective stress tensor, D^e_{ijkl} is the elastic stiffness tensor, f is the yield function, λ is the plastic multiplier, ω is the damage variable, κ is the cumulated plastic strain, σ_Y is the yield stress and s, a, σ_H and ω_c are positive material parameters, to be identified from experiments. Superior dot marks the derivative with respect to time. To describe specific material, suitable yield function needs to be introduced.

2.1. Regularization

Standard damage-plasticity models with softening may lead to localization of inelastic strain into narrow process zones. For traditional models formulated within the classical framework of continuum mechanics, such zones have an arbitrarily small thickness, and failure can occur at extremely low energy dissipation, which is not realistic. The mathematical model becomes ill-posed due to the loss of ellipticity of the governing differential equation and results obtained numerically are not objective with respect to the discretization. A general way to overcome pathological sensitivity of the numerical results to the finite element mesh is to adopt nonlocal continuum formulations. We focus our attention to the implicit gradient formulation, which requires only C^0 continuous finite element approximation. The nonlocal cumulated plastic strain is computed from a Helmholtz-type differential equation

$$\bar{\kappa} - l^2 \nabla^2 \bar{\kappa} = \kappa \tag{8}$$

with homogeneous Neumann boundary condition

$$\frac{\partial \bar{\kappa}}{\partial \mathbf{n}} = \mathbf{0}. \tag{9}$$

In (8), l is the length scale parameter and ∇ is the Laplace operator. Note that for present formulations, the nonlocal cumulated plastic strain affects only damage evolution while the yield condition remains local.

However, it can be shown that the implicit gradient formulation does not provide full regularization of the present model, thus the so-called over-nonlocal formulation has to be introduced. In this formulation, the damage variable is computed from over-nonlocal cumulated plastic strain, which is obtained as a combination of local cumulated plastic strain κ and nonlocal cumulated plastic strain $\bar{\kappa}$.

$$\hat{\kappa} = (1 - n)\kappa + n\bar{\kappa} \tag{10}$$

Full regularization can be achieved only if the parameter n is greater than 1.

3. Large-strain material models

Two sources of nonlinearities exist in the modeling of material. The first one is the material nonlinearity. A suitable material model at small strain has been presented in previous chapters. The second source of nonlinearity is related to the geometry. At first, we introduce strain measures. Next, extension of the constitutive model into the large-strain range based on the additive decomposition of the logarithmic strain is presented.

3.1. Generalized strain measures

A family of strain measures derived from the right Cauchy-Green deformation tensor was introduced by Seth and Hill [5, 6]. These generalized strain measures are defined as

$$\mathbf{E}^{(m)} = \frac{1}{2m} \left(\mathbf{C}^m - \mathbf{I} \right), \qquad m \neq 0 \tag{11}$$

$$\mathbf{E}^{(m)} = \frac{1}{2} \ln \mathbf{C}, \qquad m = 0 \tag{12}$$

where $\mathbf{I}$ is the second-order unit tensor. In the special cases when $m = 0$ and $m = 0.5$ we obtain the so-called Hencky (logarithmic) and Biot tensor, while for

$m = 1$ we obtain the right Green-Lagrange strain tensor. Recall that the Cauchy-Green deformation tensor is defined as

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} \tag{13}$$

where $\mathbf{F}$ is deformation gradient. The spectral decomposition of $\mathbf{C}$ is

$$\mathbf{C} = \sum_{a=1}^{3} \lambda_a \mathbf{N}^a \otimes \mathbf{N}^a \tag{14}$$

where λ_a are the eigenvalues of the right Cauchy-Green deformation tensor and $\mathbf{N}^a$ are the corresponding eigenvectors. Equations (11) and (12) can be rewritten as

$$\mathbf{E}^{(m)} = \frac{1}{2m}\left(\sum_{a=1}^{3}(\lambda_a^m - 1)\mathbf{N}^a \otimes \mathbf{N}^a\right), \qquad m \neq 0 \tag{15}$$

$$\mathbf{E}^{(m)} = \frac{1}{2}\sum_{a=1}^{3} \ln \lambda_a \mathbf{N}^a \otimes \mathbf{N}^a, \qquad m = 0 \tag{16}$$

For a hyperelastic material, the generalized stress tensors work-conjugate to the Seth-Hill strain measures and the corresponding generalized stiffness tensors can be derived from the Helmholtz free-energy density function ψ and expressed as

$$\mathbf{S}^{(m)} = \frac{\partial\psi}{\partial\mathbf{E}^{(m)}} \tag{17}$$

$$\mathbf{D}^{(m)} = \frac{\partial^2\psi}{\partial\mathbf{E}^{(m)}\partial\mathbf{E}^{(m)}} \tag{18}$$

Application of the chain rule leads to the transformation formulas between generalized stress tensor and tangent moduli and Lagrangean objects: the second Piola-Kirchhoff stress

$$\mathbf{S} = 2\,\frac{\partial\psi}{\partial\mathbf{C}} = \mathbf{S}^{(m)} : \mathbf{P}^{(m)} \tag{19}$$

and the stiffness tensor

$$\mathbf{D} = \mathbf{P}^{(m)} : \mathbf{D}^{(m)} : \mathbf{P}^{(m)} + \mathbf{S}^{(m)} : \mathbf{L} \tag{20}$$

The transformation formulas exploit projection tensors

$$\mathbf{P}^{(m)} = 2\,\frac{\partial\mathbf{E}^{(m)}}{\partial\mathbf{C}} \tag{21}$$

$$\mathbf{L} = 4\,\frac{\partial^2\mathbf{E}^{(m)}}{\partial\mathbf{C}\partial\mathbf{C}} \tag{22}$$

To differentiate the generalized strain measure with respect to the Cauchy-Green deformation tensor, we exploit the following formulas, which are valid if the eigenvalues are mutually different. The result for multiple eigenvalues is obtained by the rule of l'Hospital, see [2] for more details.

$$\frac{\partial \lambda_a}{\partial \mathbf{C}} = \mathbf{N}^a \otimes \mathbf{N}^a \tag{23}$$

$$\frac{\partial \mathbf{N}_a}{\partial \mathbf{C}} = \sum_{b \neq a}^{3} \frac{1}{\lambda_b - \lambda_a} \mathbf{N}^b \left(\mathbf{N}^a \otimes \mathbf{N}^b + \mathbf{N}^b \otimes \mathbf{N}^a \right) \tag{24}$$

Using equations (23) and (24) leads to the expression for the fourth-order projection tensor

$$\mathbf{P}^{(m)} = \sum_{a=1}^{3} \sum_{b=1}^{3} P_{aabb} \mathbf{N}^a \otimes \mathbf{N}^a \otimes \mathbf{N}^b \otimes \mathbf{N}^b + \sum_{a=1}^{3} \sum_{b \neq a}^{3} P_{abab} \left(\mathbf{N}^a \otimes \mathbf{N}^b \right) \otimes \left(\mathbf{N}^a \otimes \mathbf{N}^b + \mathbf{N}^b \otimes \mathbf{N}^a \right) \tag{25}$$

The components of this tensor are

$$P_{aabb} = \lambda_a^{m-1} \delta_{ab} \tag{26}$$

$$P_{abab} = \frac{\lambda_a^m - \lambda_b^m}{2m(\lambda_a - \lambda_b)} \tag{27}$$

where δ_{ab} is the Kronecker delta. The term $\mathbf{S}^{(m)} : \mathbf{L}$ is not described here; its detailed derivation can be found in [3] or [4].

3.2. Large-strain plasticity based on the logarithmic strain

The main attractiveness of the large-strain plasticity theory based on the logarithmic strain is in the modular framework consisting of three steps. In the first step, a logarithmic strain measure is computed from equation (12). In the second step, this strain measure enters a constitutive law, which may have an identical structure as in the small-strain theory. In the third step, the generalized stress tensor is transformed into the second Piola-Kirchhoff stress using expression (19) and the appropriate stiffness tensor is obtained merely by replacing the generalized stiffness tensor in equation (20) by the generalized algorithmic elasto-plastic stiffness tensor. We can define the elastic part of the logarithmic strain as

$$\mathbf{E}_e^{(0)} = \mathbf{E}^{(0)} - \mathbf{E}_p^{(0)} \tag{28}$$

with $\mathbf{E}^{(0)} = \frac{1}{2} \ln \mathbf{C}$, $\mathbf{E}_p^{(0)} = \frac{1}{2} \ln \mathbf{G}_p$. $\mathbf{G}_p$ is a Lagrangian object often called plastic metric. However, free energy function defined in terms of the logarithmic strain is not polyconvex. Polyconvexity of the free-energy function is a very important mathematical condition, which guarantee existence of the solution, see [11] for more details. Nevertheless, the model is suitable for description of materials for which the yield limit is reached at small strains.

Acknowledgements

This work has been supported by the Grant Agency of the Czech Republic, projects no. GAP108/11/1243.

References

[1] Charlebois, M., Jirásek, M., and Zysset, P.: A nonlocal constitutive model for trabecular bone softening in compression. Biomechanics and Modeling in Mechanobiology **9** (2010), 597–611.

[2] Miehe, C. and Lambrecht, M.: Algorithms for computation of stresses and elasticity moduli in terms of Seth-Hill's family of generalized strain tensors. Communications in Numerical Methods in Engineering **17** (2001), 337–353.

[3] Miehe, C., Apel, N., and Lambrecht, M.: Anisotropic additive plasticity in the logarithmic strain space. Modular kinematic formulation and implementation based on incremental minimization principles for standard materials. Computer Methods in Applied Mechanics and Engineering **191** (2002), 5383–5425.

[4] Schröder, J., Gruttmann, F., and Löblein, J.: A simple orthotropic finite elasto-plasticity model based on generalized stress-strain measures. Computational Materials Science **28** (2003), 696–703.

[5] Patzák, B. and Bittnar, Z.: Design of object-oriented finite element code. Advances in Engineering Software **32** (2001), 759–767.

[6] Bazant, Z. P. and Jirásek, M.: Nonlocal integral formulations of plasticity and damage: Survey of progress. Journal of Engineering Mechanics **128** (2002), 21–52.

[7] de Borst, R., Pamin, J.: Some novel developments in finite element procedures for gradient-dependent plasticity. Internat. J. Numer. Methods Engrg. **39** (1996), 2477–2505.

[8] Peerlings, R. H. J., de Borst, R., Brekelmans, W. A. M., de Vree, J. H. P.: Gradient-enhanced damage for quasi-brittle materials. Internat. J. Numer. Methods Engrg. **39** (1996), 3391–3403.

[9] Seth, B. R.: Generalized strain measure with application to physical problems. Second-Order Effects in Elasticity, Plasticity and Fluid Dynamics, 1964.

[10] Hill, R. : On constitutive inequalities for simple materials. J. Mech. Phys. Solids **16** (1968), 229–242.

[11] Ball, J. M.: Convexity conditions and existence theorems in nonlinear elasticity. Arch. Ration. Mech. Anal. **63** (1977), 337–403.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

VALUING BARRIER OPTIONS USING THE ADAPTIVE DISCONTINUOUS GALERKIN METHOD

Jiří Hozman

Technical University of Liberec, Faculty of Science, Humanities and Education
Studentská 2, 461 17, Liberec, Czech Republic
jiri.hozman@tul.cz

Abstract

This paper is devoted to barrier options and the main objective is to develop a sufficiently robust, accurate and efficient method for computation of their values driven according to the well-known Black-Scholes equation. The main idea is based on the discontinuous Galerkin method together with a spatial adaptive approach. This combination seems to be a promising technique for the solving of such problems with discontinuous solutions as well as for consequent optimization of the number of degrees of freedom and computational cost. The appended numerical experiment illustrates the potency of the proposed numerical scheme.

1. Introduction

During the last decade, financial models have acquired increasing popularity in option pricing. The valuation of different types of option contracts is very important in modern financial theory and practice – especially exotic options such as discrete barrier options. Most of the analytical formulas for these options is limited by strong assumptions, which led to the application of numerical methods instead.

Therefore, the main goal of this paper is to develop an efficient, robust and accurate numerical method for the barrier option pricing problem, which arises from the concept of the *discontinuous Galerkin* (DG) approach for the space semi-discretization, for more details see [5], and the *backward Euler* scheme for the discretization of the resulting ODE systems. In order to increase the efficiency of the proposed method additionally, this approach is equipped with an *h-adaptivity* technique based on regularity and residual indicators, cf. [1, 2]. The resulting numerical scheme is applied to a standard problem of discrete double barrier option pricing.

2. Barrier option pricing model

In what follows, we consider the double time-independent discrete barrier knock-out option, i.e. option that expires worthless if one of the two barriers has been hit at a monitoring date, see e.g. [1] and [6]. We denote by x the price of an underlying asset (e.g. stock) and by t the time to expiry of the option and let $M := \{0 = t_0^M < t_1^M < \ldots < t_{l-1}^M < t_l^M = T\}$ be the set of monitoring dates and B_- be the lower barrier and B_+ the upper barrier active only at discrete instances $t_l^M \in M$.

Let $\Omega \equiv (0, S_{max})$, $0 < B_- < B_+ < S_{max}$, be a bounded open interval and T stands for maturity. The price $u : Q_T = \Omega \times (0,T) \to I\!R$ of the discrete barrier option satisfies the *Black-Scholes* partial differential equation with initial and boundary conditions:

$$\frac{\partial}{\partial t}u(x,t) - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2}{\partial x^2}u(x,t) - rx\frac{\partial}{\partial x}u(x,t) + ru(x,t) = 0 \quad \text{in } Q_T, \tag{1}$$

$$u(0,t) = 0 \quad \text{and} \quad u(S_{max},t) = 0, \tag{2}$$

$$u(x,0) = \begin{cases} \max(x-K,0)\cdot\chi_{[B_-,B_+]}, & \text{(call)} \\ \max(K-x,0)\cdot\chi_{[B_-,B_+]}, & \text{(put)} \end{cases}, \quad x \in \Omega, \tag{3}$$

where $\sigma > 0$ and $r > 0$ are constant model parameters denoting the volatility of stock price and the risk-free interest rate, respectively.

From the mathematical point of view the problem (1)–(3) represents a *convection-diffusion-reaction* equation equipped with a set of two homogeneous Dirichlet boundary conditions (2) prescribed at the endpoints of interval $(0, S_{max})$ and with the initial condition (3), where symbol K stands for the strike price and $\chi_{[B_-,B_+]}$ denotes the characteristic function of the barrier interval.

Moreover the discrete monitoring of the contract introduces an updating of the solution $u(x,t)$ at the monitoring dates $t_l^M \in M$:

$$u(x, t_l^M) = \lim_{\varepsilon \to 0+} u(x, t_l^M - \varepsilon) \cdot \chi_{[B_-,B_+]}. \tag{4}$$

3. Discontinuous Galerkin discretization

Let $\mathcal{T}_h$ $(h > 0)$ be a family of *partitions* of the closure $\overline{\Omega} = [0, S_{max}]$ of the domain Ω into N closed mutually disjoint subintervals $I_k = [x_{k-1}, x_k]$ with length $h_k := x_k - x_{k-1}$. Then we set $\mathcal{T}_h = \{I_k, 1 \le k \le N\}$ with spatial step $h := \max_{1\le k\le N} h_k$ and call interval I_k an *element*. We additionally assume that the following conditions are satisfied:

$$\exists\, C_q \ge 1 : h_k \le C_q h_{k'} \quad \forall\, I_k, I_{k'} \in \mathcal{T}_h \text{ sharing a node} \quad \text{(local quasi-uniformity)} \tag{5}$$

$$\exists\, k_1, k_2 \in I\!N \quad \text{such that} \quad x_{k_1} = B_- \quad \text{and} \quad x_{k_2} = B_+ \qquad \text{(barrier consistency)} \tag{6}$$

The DG method can handle different polynomial degrees over elements. Therefore, we assign a positive integer p_k as a *local polynomial degree* to each $I_k \in \mathcal{T}_h$. Then we set the vector $\mathsf{p} = \{p_k, I_k \in \mathcal{T}_h\}$. Over the triangulation $\mathcal{T}_h$ we define the finite dimensional space of discontinuous piecewise polynomial functions:

$$S_{h\mathsf{p}} \equiv S_{h\mathsf{p}}(\Omega, \mathcal{T}_h) = \{v; v|_{I_k} \in P_{p_k}(I_k)\ \forall\, I_k \in \mathcal{T}_h\}, \tag{7}$$

where $P_{p_k}(I_k)$ denotes the space of all polynomials of degree $\le p_k$ on I_k, $I_k \in \mathcal{T}_h$. Consequently, the approximate solution of the continuous problem (1)–(4) is sought in the space $S_{h\mathsf{p}}$.

Let us denote $v(x_k^\pm) = \lim_{\varepsilon\to 0+} v(x_k \pm \varepsilon)$. Then we define the *jump* and *average* of v at inner points x_k of Ω by $[v(x_k)] = v(x_k^-) - v(x_k^+)$ and $\langle v(x_k)\rangle = \frac{1}{2}\left(v(x_k^-) + v(x_k^+)\right)$, respectively. We also extend the definition of jump and mean value for endpoints of Ω, i.e. $[v(x_0)] = -v(x_0^+)$, $\langle v(x_0)\rangle = v(x_0^+)$, $[v(x_N)] = v(x_N^-)$ and $\langle v(x_N)\rangle = v(x_N^-)$.

Firstly, we recall the space semi-discrete DG scheme presented in [4] and [5]. To this end we introduce the following bilinear forms:

$$
\begin{aligned}
a_h^\Theta(u,v) &= \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} \frac{1}{2}\sigma^2 x^2 \frac{\partial u(x,t)}{\partial x} v'(x)\,\mathrm{d}x - \sum_{k=0}^{N} \left\langle \frac{1}{2}\sigma^2 x_k^2 \frac{\partial u(x_k,t)}{\partial x} \right\rangle [v(x_k)] \\
&\quad + \Theta \sum_{k=0}^{N} \left\langle \frac{1}{2}\sigma^2 x_k^2\, v'(x_k) \right\rangle [u(x_k,t)], \qquad (8) \\
b_h(u,v) &= -\sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} (\sigma^2 - r)x\, u(x,t)\, v'(x)\,\mathrm{d}x + \sum_{k=0}^{N} H\left(u(x_k^-,t), u(x_k^+,t)\right) [v(x_k)], \qquad (9) \\
J_h^\omega(u,v) &= \sum_{k=0}^{N} \omega_k [u(x_k,t)]\,[v(x_k)]. \qquad (10)
\end{aligned}
$$

The crucial item of the DG formulation is the treatment of the linear convection and diffusion terms. For the convection form b_h we treat its terms with the aid of a *numerical flux* H, see [3]. The diffusion form a_h^Θ includes *stabilization* terms which are added to the formulation of the problem in order to guarantee the stability of the numerical scheme. According to the value of parameter Θ, we speak of *symmetric* ($\Theta = -1$), *incomplete* ($\Theta = 0$) or *nonsymmetric* ($\Theta = 1$) variants. Furthermore, in order to replace the inter-element discontinuities, the semi-discrete scheme is completed with the *penalty* J_h^ω weighted by the penalty parameter function ω_k defined in the spirit of [4]. Let us note that the right-hand side term vanishes due to the prescribed homogeneous Dirichlet boundary conditions in (2).

In order to simplify the notation we define the bilinear form:

$$
\mathcal{B}_h^\Theta(u,v) := a_h^\Theta(u,v) + b_h(u,v) + \alpha J_h^\omega(u,v) + (2r - \sigma^2)(u,v), \quad \alpha > 0, \qquad (11)
$$

where $(\cdot,\cdot)$ denotes inner product and the forms $a_h^\Theta(\cdot,\cdot)$, $b_h(\cdot,\cdot)$ and $J_h^\omega(\cdot,\cdot)$ are given by (8), (9) and (10), respectively. The value of multiplicative constant α before the penalty form J_h^ω depends on the properties of diffusion term, see [4]. Finally, we end up with the following DG formulation for the *semi-discrete solution* $u_h(t) \in S_{h\mathsf{p}}$:

$$
\frac{d}{dt}(u_h(t), v_h) + \mathcal{B}_h^\Theta(u_h(t), v_h) = 0 \quad \forall\, v_h \in S_{h\mathsf{p}}, \forall\, t \in (0,T), \qquad (12)
$$

which represents an ODE system and due to bilinearity of form (11) we can easily discretize (12) by the implicit Euler method. Let $0 = t_0 < t_1 < \ldots < t_r = T$ be a partition of $[0,T]$ with time steps $\tau_l \equiv t_{l+1} - t_l$, $l = 0, 1, \ldots, r-1$. We define the *approximate solution* of problem (1)–(4) as functions $u_h^l \approx u_h(t_l)$, $t_l \in [0,T]$, $l = 0, \ldots, r-1$, satisfying the following numerical scheme:

$$
\begin{aligned}
\left(u_h^{l+1}, v_h\right) + \tau_l \mathcal{B}_h^\Theta\left(u_h^{l+1}, v_h\right) &= \left(u_h^l, v_h\right) \qquad \forall\, v_h \in S_{h\mathsf{p}}, \qquad (13) \\
u_h^{l+1} &:= u_h^{l+1} \cdot \chi_{[B_-,B_+]} \quad \forall\, t_{l+1} \in M, \qquad (14)
\end{aligned}
$$

where u_h^0 is $S_{h\mathsf{p}}$-approximation of u^0. The discrete problem (13) is equivalent to a system of linear algebraic equations at each time level $t_{l+1} \in [0, T]$.

4. Mesh adaptation

In this section, we introduce an h-adaptive DG technique for the solution of problem (1)–(4). Since we deal with nonstationary problems, it is suitable to use adaptive mesh refinement during the computation in order to improve the numerical solution and to optimize the number of degrees of freedom and computational cost, consequently.

We start from a uniform coarse grid $\mathcal{T}_{0,h} := \mathcal{T}_h$ and construct at each time instance $t_l \in [0, T]$ a new mesh $\mathcal{T}_{l,h}$ depending on the previous grid $\mathcal{T}_{l-1,h}$ through the following h-adaptation operations: cutting (C) one element I_k into I_{k_1} and I_{k_2} and gluing (G) two elements I_{k_1} and I_{k_2} together into I_k. The described adaptation process has to comply with restrictions on a minimal admissible size of mesh step h_{min}, a maximal admissible size of mesh step h_{max}, a maximal number of elements N_{max} and keeping of local quasi-uniformity (5) and barrier consistency (6), respectively.

The main idea of the proposed h-adaptive strategy is based on

- *mesh refinement* in domains with irregular solution (low regularity) or with high value of residual estimate,
- *mesh coarsening* in domains with solution of high regularity and low value of residual estimate.

The estimation of the regularity of the solution is essential for mesh refinement. The presented approach is based on a measure of inter-element jumps arising from the shock capturing techniques in hyperbolic problems, for a survey see [2].

We have employed the following element-wise *regularity indicator*:

$$g_{I_k}(u_h) := \frac{1}{h_k^{2p_k+1}} \left(\sum_{i=k-1}^{k} [u_h(x_i)]^2 \right), \quad k = 1, \dots N, \tag{15}$$

which recognizes the subdomains of Ω where the solution is smooth ($g_{I_k} \approx 0$) from the areas with discontinuities or with a very steep gradient ($g_{I_k} \gg 1$).

The second key ingredient of the mesh refinement is the *residual estimator* which is chosen proportionally to the strong formulation of the local residue from [1] as

$$r_{I_k}(u_h) := \frac{\partial u_h}{\partial t} - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 u_h}{\partial x^2} - rx\frac{\partial u_h}{\partial x} + ru_h, \quad I_k \in \mathcal{T}_h. \tag{16}$$

Then the *local* and *global* residual estimators of the approximate solution u_h are defined by $res_{I_k}(u_h) := \|r_{I_k}\|_{L^2(I_k)}$ and $res_G(u_h) := \sqrt{\sum_{I_k \in \mathcal{T}_h} res_{I_k}^2}$, respectively.

Our interest is to find a solution $\widetilde{u_h} \in S_{h\mathsf{p}}$ such that $res_G(\widetilde{u_h}) \leq TOL$, where $TOL > 0$ is a given tolerance. In order to satisfy this condition we prescribe the following stopping criterion for the h-adaptivity: $res_{I_k} \leq \frac{TOL}{N}, \forall\ I_k \in \mathcal{T}_h$, which guarantees the uniform distribution of the global residue.

The whole h-adaptation DG algorithm can be schematically written as

```
1. let $TOL > 0$, $0 < h_{min} \leq h_{max}$ and $N_{max}$ be given,
2. let $B_-, B_+ \longleftrightarrow \mathcal{T}_{0h}$ and $S_{h\mathsf{p}}$ be set up, let $u^0 \longleftrightarrow u_h^0$ be given,
3. repeat time loop (until $t_l > T$) ($l = 1, \dots, r$)
   (a) solve problem (13)–(14) on $\mathcal{T}_{l-1,h} \Longrightarrow u_h^l$,
   (b) evaluate indicators $g_{I_k}(u_h^l)$, $res_{I_k}(u_h^l)$, $\forall\, I_k \in \mathcal{T}_{l-1,h} \Longrightarrow res_G(u_h^l)$,
   (c) if $res_G(u_h^l) > TOL \Rightarrow h$-refinement,
       (C) $h$-refine elements with $res_{I_k} > \frac{TOL}{N}$,
       (G) $h$-derefine elements with $res_{I_k} < \delta\frac{TOL}{N} \wedge g_{I_k}(u_h^l) \approx 0$,
       (•) construct new mesh $\mathcal{T}_h^{new} \longrightarrow \mathcal{T}_{l-1,h}$ and space $S_{h\mathsf{p}}$, go to (a),
   (d) if $res_G(u_h^l) \leq \frac{TOL}{\beta} \Rightarrow h$-coarsening,
       (G) $h$-derefine elements with $res_{I_k} < \delta\frac{TOL}{N} \wedge g_{I_k}(u_h^l) \approx 0$,
       (•) construct new mesh $\mathcal{T}_h^{new} \longrightarrow \mathcal{T}_{l-1,h}$ and space $S_{h\mathsf{p}}$, go to (a),
```

where $\beta > 1$ and $\delta \in (0, 1)$ are user-defined parameters, in our computations they are typically chosen as $\beta = 3.0$ and $\delta = 0.1$.

5. Numerical example

The presented numerical example represents the case of a discrete double barrier call option with the expiration date $T = \frac{8}{12}$ (e.g. 8 months) and the strike price $K = 6.0$. The prescribed barriers are $B_- = 4.0$, $B_+ = 8.0$ and computational domain was set as $\Omega = [0, 9]$. The Black-Scholes model parameters were the risk-free interest rate $r = 1.0y^{-1}$ and volatility $\sigma^2 = 0.01y^{-1}$. The initial uniform mesh with spatial step $h = 0.25$ was adaptively refined according to h-adaptation parameters $h_{min} = 10^{-3}$ and $h_{max} = 0.5$. The time step is $\tau = \frac{1}{120}$ and we consider monthly monitoring. We carried out computations by piecewise quadratic approximations, set $\Theta = 0$ and used the restarted GMRES for the solving of linear systems (13).

Table 1 illustrates the development of the global residue and the number of elements during the computation in comparison with an adapted and uniform mesh. One can easily observe that for approximately the same values of the global residue, it is sufficient to use less elements in the adapted case than for the uniform one. Figure 1 shows the corresponding isolines of option price and global residue in space-time plot with well-resolved monthly monitoring.

6. Conclusion

We have dealt with the numerical solution of the discrete barrier option pricing models, represented by the linear convection-diffusion-reaction equation. We have presented DG approach together with simple h-adaptivity technique. Presented numerical example illustrated the potency of the resulting scheme.

Acknowledgements

This work was supported by the ESF Project No. CZ.1.07/2.3.00/09.0155.

time (bimonthly)	res_G (adapted)	$\#\mathcal{T}_{lh}^*$	res_G (uniform)	$\#\mathcal{T}_{lh}$
0.000000	19.960869	36	10.888578	120
0.166667	1.498133	178	1.459563	120
0.333333	0.615153	58	0.476494	120
0.500000	0.572154	44	0.475957	120
0.666667	0.119596	58	0.124287	120

Table 1: Comparison of h-adaptive and uniform approach w.r.t. res_G; $\mathcal{T}_{lh}^*$ denotes input meshes after monitoring without the updated h-refinement or h-coarsening.

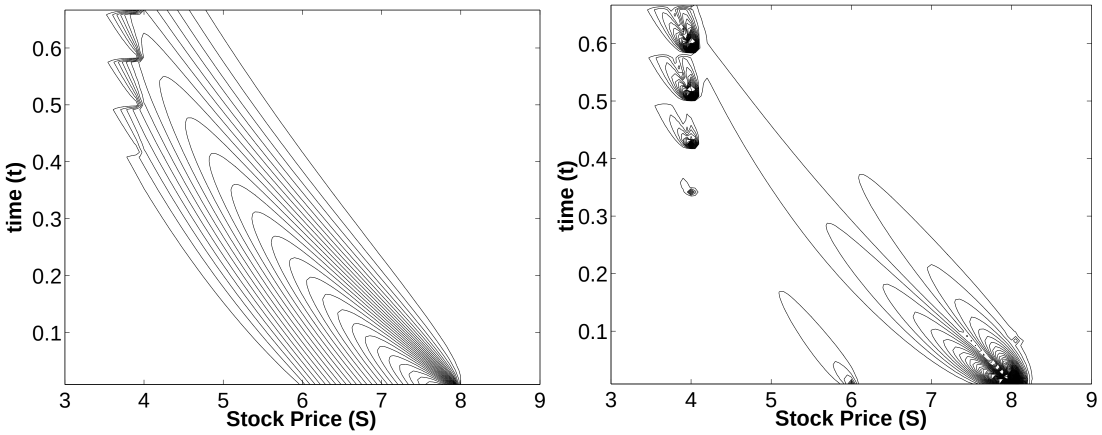


Figure 1: The isolines of price u (left) and corresponding global residue res_G (right).

References

[1] Achdou Y., and Pironneau, O.: *Computational methods for option pricing.* Society for Industrial and Applied Mathematics, Philadelphia, 2005.

[2] Dolejší, V., Feistauer, M., and Schwab C.: On some aspects of the discontinuous Galerkin finite element method for conservation laws. Math. Comput. Simulation **61** (2003), 333–346.

[3] Feistauer, M., Felcman, J., and Straškraba, I.: *Mathematical and computational methods for compressible flow.* Oxford University Press, Oxford, 2003.

[4] Hozman, J.: *Discontinuous Galerkin method for convection-diffusion problems.* Ph.D. thesis, Charles University Prague, 2009.

[5] Riviére, B.: *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation.* Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 2008.

[6] Seydel, R.: *Tools for computational finance:* 4th *edition.* Springer, Berlin, 2008.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

ON ESTIMATION OF DIFFUSION COEFFICIENT BASED ON SPATIO-TEMPORAL FRAP IMAGES: AN INVERSE ILL-POSED PROBLEM

Radek Kaňa[1], Ctirad Matonoha[2], Štěpán Papáček[3], Jindřich Soukup[3]

[1] Institute of Microbiology,
Academy of Sciences of the Czech Republic
Opatovický mlýn, 379 81 Třeboň, Czech Republic
kana@alga.cz

[2] Institute of Computer Science,
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
matonoha@cs.cas.cz

[3] University of South Bohemia in České Budějovice, Faculty of Fisheries and Protection of Waters, CENAKVA, School of Complex Systems, Zámek 136, 373 33 Nové Hrady,
Czech Republic
spapacek@frov.jcu.cz, jindra@matfyz.cz

Abstract

We present the method for determination of phycobilisomes diffusivity (diffusion coefficient D) on thylakoid membrane from fluorescence recovery after photobleaching (FRAP) experiments. This was usually done by analytical models consisting mainly of a simple curve fitting procedure. However, analytical models need some unrealistic conditions to be supposed. Our method, based on finite difference approximation of the process governed by the Fickian diffusion equation and on the minimization of an objective function representing the disparity between the measured and simulated time-varying fluorescent particles concentration profiles, naturally accounts for experimentally measured time-varying Dirichlet boundary conditions and can include a reaction term as well. The result we get is the overall (time averaged) diffusion coefficient D and the sequence of diffusivities D_j based on two successive fluorescence profiles in *j-th* time interval. Due to the ill-posedness of our inverse problem, regularization algorithms are implemented. On the synthetic example, we illustrate the behaviour of solution depending on regularization parameter for different signal to noise ratio.

1. Introduction

Fluorescence Recovery After Photobleaching (FRAP) measuring technique is widely used since 1970s to study the organization and dynamics of many photosynthetic pigment-protein complexes in the photosynthetic membrane [16]. Later on, FRAP has been extended to the investigation of protein dynamics within the living cells [14]. Using fluorescence confocal microscopy we get the spatio-temporal FRAP images, and consequently the mobility of photosynthetic complexes in a native intact membrane, i.e. the diffusivity or diffusion coefficient D,[1] is reconstructed using either a *closed form model* or *simulation based model* [9, 6]. The FRAP images are in general very noisy, with small signal to noise ratio (SNR), which requires an adequate technique assuring the reliable results.[2]

Our study describes the development of a method aiming to determine the phycobilisomes diffusivity on thylakoid membrane from FRAP experiments. As we know, this is usually done by experimental curve fitting to the analytical (closed form) models, see e.g. [1, 10, 7, 15]. However, the closed form models need some unrealistic assumptions. For example, C. W. Moulineaux *et al.* [10] have exploited the rotational symmetry of the cells by bleaching a plane across the short axis of the cell and reaching one-dimensional bleaching profiles along the long axis. Moreover, it was supposed that: (i) $x \in \mathcal{R}$, i.e. the infinite domain, (ii) the initial bleaching profile is Gaussian, and (iii) the recovery is complete for $t \to \infty$.[3] The calculation of diffusion coefficient D then resides in the weighted linear regression. The error analysis for this method, i.e. how the noise corrupts the result, we treat in paper [13].

As the analytical approach has several limitation (e.g. restriction to the specific cell geometry, bleach profile must be gaussian-like, full recovery is required, etc.), we model the FRAP process by the Fickian diffusion equation with realistic initial and boundary conditions instead. The estimation of diffusivity is further formulated as a single parameter optimization problem consisting in the minimization of an objective function representing the disparity between the experimental and simulated time-varying concentration profiles.

The paper is organized as follows. The model of the process (i.e. reaction-diffusion system) and the real data form we deal with are introduced in the second section. In the third section we define the optimization problem, describe a regularization method and its implementation. The results of the numerical simulations are contained in the fourth section, while in the fifth section the paper is concluded.

[1] I. F. Sbalzarini in [14] distinguishes between the molecular diffusion constant and the apparent diffusion constant; while the former is directly measured by single-molecule techniques, the latter is determined by coarse-grained methods such as FRAP, averaging over a certain observation volume.

[2] Let us mention that the fluorescence confocal microscope allows the selection of a thin cross-section of the sample by rejecting the information coming from the out-of-focus planes. However, the small energy level emitted by the fluorophore and the amplification performed by the photon detector introduces a measurement noise.

[3] Having $y(x,t_0) = y_{0,0} \exp \frac{-2x^2}{{r_0}^2}$, where r_0 is the half-width of the bleach at time $t_0 = 0$, the solution $y(x,t)$ of diffusion equation $\frac{\partial y}{\partial t} = D\frac{\partial^2 y}{\partial x^2}$ and the maximum depth at time t, i.e. $y(0,t)$ are as follows: $y(x,t) = \frac{y_{0,0} r_0}{\sqrt{{r_0}^2+8Dt}} \exp \frac{-2x^2}{{r_0}^2+8Dt}$, $y(0,t) = \frac{y_{0,0} r_0}{\sqrt{{r_0}^2+8Dt}}$. The calculation of diffusion coefficient D then resides in the weighted linear regression: a plot of $(\frac{y_{0,0}}{y(0,t)})^2$ against time should give a straight line with the tangent $\frac{8D}{{r_0}^2}$.

2. Problem formulation

2.1. Reaction-diffusion system

FRAP (Fluorescence Recovery After Photobleaching) technique is based on application of short, intense laser irradiation to a small target region of the cell that causes irreversible loss in fluorescence in this area without any damage in intracellular structures. After the “bleach” (or “bleaching”), the observed recovery in fluorescence in the “bleached area” reflects diffusion of fluorescence compounds from the area outside the bleach. For an arbitrary geometry of bleach spot and assuming (i) local homogeneity, i.e. assuring that the concentration profile of fluorescent particles is smooth, (ii) isotropy, i.e. diffusion coefficient is space-invariant, (iii) an unrestricted supply of unbleached particles outside of the target region, i.e. assuring the complete recovery,[4] the unbleached particle concentration C as a function of spatial coordinate $\vec{r}$ and time t is modeled with the following diffusion-reaction equation on two-dimensional domain Ω:

$$\frac{\partial C}{\partial t} - \nabla \cdot (D \nabla C) = R(C) \ , \tag{1}$$

where D is the fluorescent particle diffusivity within the domain Ω and $R(C)$ is a reaction term.

The initial condition and time varying Dirichlet boundary conditions are:

$$C_0 = f(\vec{r}, t_0) \ \text{ in } \Omega, \quad C(t) = g(\vec{r}, t) \text{ in } \partial\Omega \times [t_0, T]. \tag{2}$$

The reaction term $R(C)$ is often viewed as negligible under assumptions that diffusion of fluorescence compounds (proteins) is not restricted (e.g. by some binding to the medium) and that photobleaching of these molecules during recovery is negligible. In occasions where the binding reaction takes place, we can not reduce our process to the one component diffusion equation, but the dynamics of binding reaction and eventually the diffusion of bound complexes have to be modelled, see e.g. [15]. Consequently, if $R(C)$ is neglected, Eq. (1) becomes the Fickian diffusion equation. In contrast, under continual photobleaching during image acquisition, this reaction term could be described as a first order reaction: $R(C) = -k_S \ C$, where k_S is a rate constant describing bleaching during scanning [6].

It is of utmost importance to identify the relation between concentration of particles C and fluorescent signal ϕ. Although Eq. (1) and objective function J, cf. (10), works with concentrations, in fact we measure the fluorescence intensity level and not directly C. If the relation $C = k_F \phi$, where k_F is a constant, holds, than we can work with the measured signal without necessity of any recalculation. On the contrary, if k_F is space or time dependent, then we should design an experiment and estimate this dependence.

[4] The recovery is not always complete. It is usually modelled by introducing some correction term. More consistent method resides in the special time dependent Neumann boundary condition in form of a saturation curve.

Before bleaching, some number of so-called pre-bleach measurements are performed. Notice that the pre-bleach profile C_{pre} represents a steady state constant concentration profile which has to be gradually recovered for $t \to \infty$. Thereafter, based on the pre-bleach data ϕ_{pre}(e.g. its average value), we reach the coefficient k_F as follows: $k_F = \frac{C_{pre}}{\phi_{pre}}$. Consequently, in order to have experimental values C_{exp} representing the concentration profiles after bleaching, we have to divide the post-bleach fluorescence signal by its pre-bleach value, as it is explained in the following.

2.2. One-dimensional one component diffusion equation

For a linear bleach spot perpendicular to a longer axis (let this axis be denoted as r) and assuming local homogeneity and isotropy, the recovery of unbleached particle concentration as a function of spatial coordinate r and time t is modeled with a linear, diffusion-reaction equation

$$\frac{\partial C}{\partial t} - D\frac{\partial^2 C}{\partial r^2} = R(C) \ . \tag{3}$$

If we adopt the form of reaction term according to $R(C) = -k_S\, C$ and introduce the dimensionless spatial coordinate x, the dimensionless diffusion coefficient p, the dimensionless time τ and the dimensionless concentration y by

$$x := \frac{r}{L}, \quad p := \frac{D}{D_0}, \quad \tau := t\,\frac{D_0}{L^2}, \quad y := \frac{C}{C_{pre}} \ , \tag{4}$$

where L is the length of our specimen in direction perpendicular to bleach spot, D_0 is a constant with some characteristic value (unit: $\mathrm{m^2 s^{-1}}$), and C_{pre} is a pre-bleach concentration of C, we finally obtain the following form of dimensionless diffusion-reaction equation on one-dimensional domain, i.e. for $x \in [0, 1]$

$$\frac{\partial y}{\partial \tau} - p\frac{\partial^2 y}{\partial x^2} = -\frac{k_S L^2}{D_0} y \ . \tag{5}$$

The initial condition and time varying Dirichlet boundary conditions are:

$$y(x, \tau_0) = f(x), \quad x \in [0, 1], \tag{6}$$

$$y(0, \tau) = g_0(\tau), \quad y(1, \tau) = g_1(\tau), \quad \tau \geq \tau_0. \tag{7}$$

2.3. Experimentally measured data

Based on FRAP experiments, we have a 2D dataset in form of a table with experimental values $y_{exp}(r_i, t_j)$ (already normalized), where $(N + 1)$ rows correspond to the number of spatial points where the values are measured, and $(m^* + M + 1)$ columns correspond to the number of discrete time points, i.e. time instant when the data were measured:

$$y_{exp}(r_i, t_j), \ i = 0 \ldots N, \ j = -m^* \ldots M. \tag{8}$$

This can be read by columns as the concentration profiles (along r axis) in m^*+M+1 discrete time points, where m^* corresponds to the number of columns with pre-bleach data containing the information about the steady state and optical distortion, and $M+1$ columns of post-bleach data contain the information about the transport of unbleached particles (due to the diffusion process) through the boundary of bleach spot (our computational domain Ω).

The row data are further re-scaled in order to be in the following form:

$$y_{exp}(x_i, \tau_j),\ i = 0 \dots n,\ j = -m^* \dots m, \tag{9}$$

where space interval between first and last measurement points we take into account is chosen as $[a, b]$. Thus, $L = b - a$ is the length of space interval in physical units, i.e. [m], chosen by the person performing the measurment. The re-scaled dimensionless space interval is again $x \in [0, 1]$ and the re-scaled distance between two space measurements is $h = \frac{1}{n}$. Time interval between two measurements is T in [s], re-scaled dimensionless time interval is $\tau_t = \frac{TD_0}{L^2}$. For the further calculation, the number of post-bleach measurements can be also reduced, i.e. let $m \leq M$. Recall that τ_0 corresponds to the first post-bleach measurement, and $x_0 = 0$, $x_n = 1$. Consequently, $y_{exp}(x_i, \tau_0)$, $i = 0 \dots n$, represent the initial condition and $y_{exp}(0, \tau_j)$ and $y_{exp}(1, \tau_j)$, $j = 0 \dots m$, the left and right Dirichlet boundary conditions, respectively.

Recall that due to the measurement noise both the respective $j - profiles$ $y_{exp}(x_i, \tau_j)$, $i = 0 \dots n$, and the initial and boundary conditions cannot be simply approximated by a smooth function. The forthcoming task is to analyze the measurement noise from real data and to treat it correctly, i.e. to use it for the setting of the regularization parameter, see the following section 3.

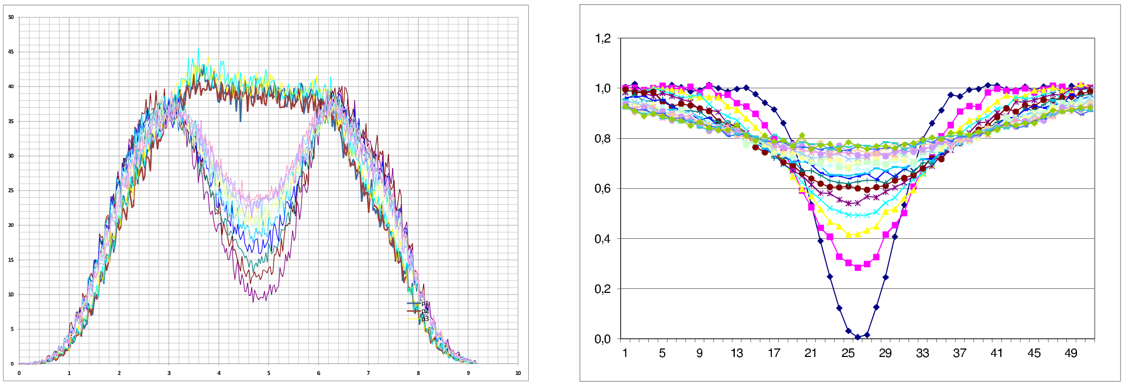


Figure 1: Left: Experimental data from FRAP experiments with red algae *Porphyridium cruentum* describing the phycobilisomes mobility on thylakoid membrane [7]. Right: Synthetic data used for numerical experiments. The y-axis represents the dimensionless concentration and x-axis the spatial coordinate, both in arbitrary units.

3. Inverse problem and its regularization

3.1. Determination of diffusivity as a parameter estimation problem

The problem of autofluorescence compound (e.g. phycobilisomes) diffusivity determination based on time series of FRAP experimental data will be further formulated as a parameter estimation problem. We construct an objective function J representing the disparity between the experimental and simulated time-varying concentration profiles, and then within a suitable method we look for such a value p minimizing J. The usual form of an objective function is the sum of squared differences between the experimentally measured and numerically simulated time-varying concentration profiles:

$$J(p) = \sum_{j=0}^{m} \sum_{i=0}^{n} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j)\right]^2 \ , \tag{10}$$

where $y_{sim}(x_i, \tau_j)$ are simulated values resulting from the solution of PDE (5) with the initial and boundary conditions (6)-(7) for the known parameter p, which is now the independent variable, i.e. $y_{sim} = y_{sim}(p)$. For the sake of clarity we further neglect the other parameter concerning the reaction term, i.e. we neglect the influence of bleaching during scanning, i.e. we put $\frac{k_S L^2}{D_0} = 0$.

Taking into account the biological reality residing in possible time dependence of phycobilisomes diffusivity, we further consider two cases:

1. First, we can take both sums for i and j in (10) together. In this case, the scalar p^* is a result of a minimization problem $p^* = \arg\min_p J(p)$.

2. Secondly, we can consider each j-th time instant separately. In this case, the m solutions $p_1^*, \ldots, p_m^*$ with values $J_1, \ldots, J_m$ correspond to each minimization problem for fixed j in sum (10), i.e. $p_j{}^* = \arg\min_{p_j} J_j(p_j)$, where $J_j(p_j) = \sum_{i=0}^{n} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p_j)\right]^2$, and we have a “dynamics” of diffusivity p evolution.

Our problem is ill-posed in the sense that the solution, i.e. the diffusion coefficients $D_j = p_j \ D_0, \ j = 1, \ldots, m$, does not depend continuously on the data and may be very sensitive to noise. This led us to the necessity of some stabilizing procedure[5] and the formulation of another cost function by adding the regularization term $\alpha||p - p_{reg}||^2$ to (10), see [3, 5, 17]. Here $\alpha \geq 0$ is a regularization parameter and p_{reg} is an expected regularized value. Doing this, we use an apriori information about the solution, in other words we assume that $p \equiv p(x, \tau)$ is almost constant with respect to x and τ and regularization term moves the minimum of functional $J(p) = \sum_{j=1}^{m} J_j(p_j)$, i.e. the solutions $p_1^*, \ldots, p_m^*$ towards a constant. In case $\alpha \to \infty$

[5] The “naive approach” consisting in the hope that the typical oscillation of $p_j{}^*$ can be suppressed by removing the noise from data, e.g. by smoothing using the Fourier transformation, was treated in [12] and further abandoned by the authors.

we obtain $p_j^* = p_{reg}, j = 1, \ldots, m$. Note that taking $\alpha = 0$, the regularization term vanishes, i.e. the functional (10) is the special case of a more general functional, see the next section.

3.2. Three types of optimization problem

Define the cost functions

$$J_j(p_j, \alpha) = \sum_{i=0}^{n} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p_j)\right]^2 + \alpha\,(p_j - p_{reg})^2, \quad j = 1, \ldots, m, \tag{11}$$

$$J(p_1, \ldots, p_m, \alpha) = \sum_{j=1}^{m} J_j(p_j, \alpha). \tag{12}$$

Three types of a one-dimensional optimization problem are considered:

1. Scalar p is a solution when taking both sums for i and j in together:

$$p^* = \arg\min_{p} \sum_{j=1}^{m} \sum_{i=0}^{n} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p)\right]^2 \tag{13}$$

2. Each j^{th} time instant separately without regularization ($\alpha = 0$):

$$p_j^* = \arg\min_{p_j} \sum_{i=0}^{n} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p_j)\right]^2 \tag{14}$$

3. Each j^{th} time instant separately using so-called Tikhonov regularization:

$$p_j^*(\alpha) = \arg\min_{p_j, p_{reg}} \left\{ \sum_{i=0}^{n} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p_j)\right]^2 + \alpha\,(p_j - p_{reg})^2 \right\} \tag{15}$$

We use a basic optimization method leading to values p^*, p_j^*, $p_j^*(\alpha)$ that minimize respective cost functional. Values p_j^*, $p_j^*(\alpha)$ are approximations of diffusion coefficients. We briefly describe a basic optimization method without loss of generality for the case of solving problem (13).

Basic optimization method is an iteration process starting from an initial point $p^{(0)}$ and generating a sequence of iterates $p^{(1)}, p^{(2)}, \ldots$ leading to a value p^* such that

$$p^{(l+1)} = p^{(l)} + \sigma^{(l)} d^{(l)},$$

where

- $d^{(l)}$ is a direction vector determined on the basis of values

$$p^{(j)},\ J(p^{(j)}),\ J'(p^{(j)}),\ J''(p^{(j)}), \quad 0 \le j \le l,$$

- $\sigma^{(l)} > 0$ is a step-length determined on the basis of behavior of the function J in the neighborhood of $p^{(l)}$.

There exist several methods for determination of direction vector and step-length selection (line-search or trust-region method) described e.g. in [11]. The trust-region method, implemented in the system for universal functional optimization [8], was used in our numerical test described in the next section.

3.3. Implementation

In this subsection we describe how we implemented both the direct problem, i.e. solution of problem (5)-(7), and the parameter estimation problem, i.e. minimization of a respective functional J.

In order to compute a function value $J_j(p_j^{(l)}, \alpha)$ in (12) for a given $p_j^{(l)}$ in the l^{th} iteration, we need to know both

- the experimental values $y_{exp}(x_i, \tau_j),\ i = 0 \dots n,\ j = 0 \dots m,$
- the simulated values $y_{sim}(x_i, \tau_j, p_j^{(l)}),\ i = 0 \dots n,\ j = 0 \dots m.$

It means that in each l^{th} iteration we need to solve the problem (we use the notation $y_{sim} \equiv y,\ p_j^{(l)} \equiv p$ for simplicity)

$$\frac{\partial y}{\partial \tau} - p\,\frac{\partial^2 y}{\partial x^2} = 0\ , \tag{16}$$

with the initial and boundary conditions defined by the experimental data

$$y(x, \tau_0, p) = y_{exp}(x, \tau_0) \quad \text{for} \quad x \in [0, 1], \tag{17}$$

$$y(0, \tau, p) = y_{exp}(0, \tau), \quad y(1, \tau, p) = y_{exp}(1, \tau) \quad \text{for} \quad \tau \geq \tau_0. \tag{18}$$

Problem (16)-(18) for simulated data $y(x_i, \tau_j, p_j)$ was solved numerically using two following finite difference schemes [2] for uniformly distributed nodes with the space steplength Δh and the variable time steplength $\Delta\tau$:

- The explicit scheme of order $\Delta\tau + \Delta h^2$:

$$y_{i,j} = \beta y_{i-1,j-1} + (1 - 2\beta) y_{i,j-1} + \beta y_{i+1,j-1}$$

- The Crank-Nicholson implicit (CN) scheme of order $\Delta\tau^2 + \Delta h^2$:

$$-\frac{\beta}{2} y_{i-1,j} + (1 + \beta) y_{i,j} - \frac{\beta}{2} y_{i+1,j} = \frac{\beta}{2} y_{i-1,j-1} + (1 - \beta) y_{i,j-1} + \frac{\beta}{2} y_{i+1,j-1}$$

Here $\beta = \frac{\Delta\tau}{\Delta h^2}\, p$ and $y_{i,j} \equiv y(x_i, \tau_j, p_j)$ are the computed values in nodes that enter the function J as values $y_{sim}(x_i, \tau_j, p_j)$. Recall that for the explicit scheme the condition $\beta \leq 1/2$ must hold.

Concerning the steplengths used in the numerical schemes, we set the space steplength to be $\Delta h = 1/n$ (smaller splitting $\Delta h = 1/(\kappa_s n)$ with $\kappa_s \in \mathcal{N}$ can also be considered). The time steplength $\Delta\tau$ is variable but should be ideally of the same order as Δh^2 (or Δh in the CN scheme) and in the explicit scheme has to fulfill the relation $\Delta\tau \leq \frac{\Delta h^2}{2p}$. In order to get from the $(j-1)$-th time instant to the j-th, we need to perform $\kappa_t = \lceil \frac{T D_0}{L^2 \Delta\tau} \rceil$ substeps of the above chosen scheme, where $\kappa_t \in \mathcal{N}$ is the smallest integer that is not less than $\frac{T D_0}{L^2 \Delta\tau}$.

4. Numerical simulation results

We have performed numerical experiments with the synthetic data corrupted by the 10% Gaussian noise with $n = 51$, $m = 19$ and consider each j-th time instant separately, i.e. j is fixed in sum (12). We report the results using the CN scheme (they are in fact independent of the used scheme) and illustrate the difficulties caused by the ill-posedness of our problem.

In Figure 2 we can see big jumps in computed approximated values $p_j^*, j=1,\ldots,m$ when using no regularization ($\alpha = 0$). In contrast, regularization technique ($\alpha > 0$) seems to cope with ill-posedness quite well. The solutions $p_1^*(\alpha),\ldots,p_m^*(\alpha)$ become smoother and tend to the estimated regularized value p_{reg} for larger α (larger weight of the regularization term). The regularized value corresponds to the exact solution $1/\pi \approx 0.3183$.

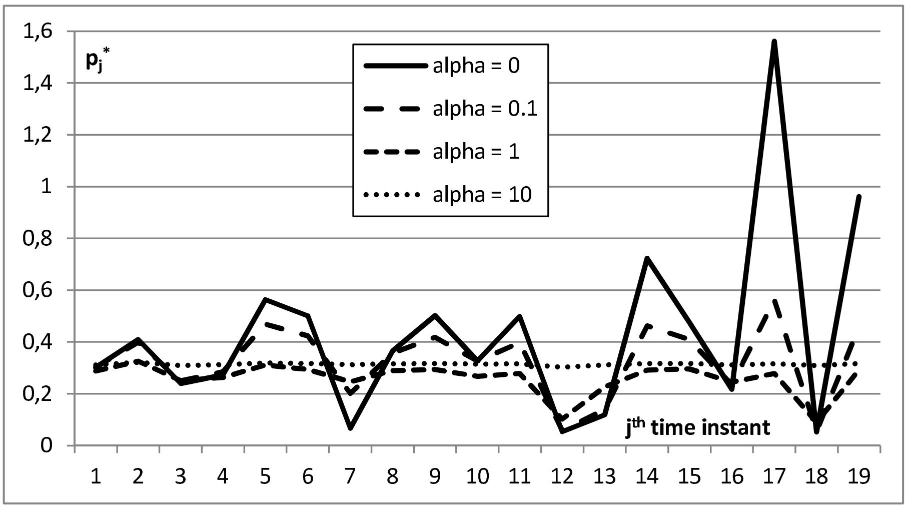


Figure 2: Dimensionless diffusivities $p_j^* = \frac{D_j}{D_0}$: Values $p_1^*(\alpha),\ldots,p_{19}^*(\alpha)$.

When using this approach, the variance of solutions $p_j^*(\alpha)$ tends to zero for $\alpha \to \infty$, i.e. $p_j^*(\alpha) \to p_{reg}$ $\forall j = 1,\ldots,m$, but the function values $J(p^*,\alpha)$, see (12), become larger (however there is a supremum). This fact is demonstrated in Figure 3, where we have used relative deviation from the average value (coefficient of variation[6]) as a solution norm:

$$c_v(\alpha) = \frac{1}{m\,\varnothing p_j^*(\alpha)}\sqrt{\sum_{j=1}^{m}[p_j^*(\alpha) - \varnothing p_j^*(\alpha)]^2}. \tag{19}$$

A proper choice of the regularization parameter α balances the above types of the curves. One of the possible criteria how to choose a proper α which is in some sense

[6] The coefficient of variation (c_v) is defined as the ratio of the standard deviation to the mean $c_v = \frac{\sigma}{\mu}$, which is the inverse of the signal-to-noise ratio.

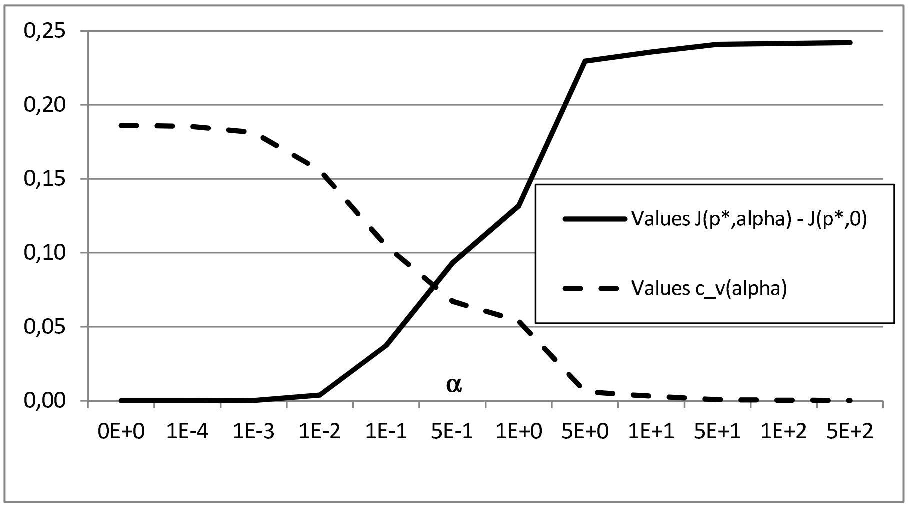


Figure 3: Values $J(p^*, \alpha) - J(p^*, 0)$ are increasing, values $c_v(\alpha)$ are decreasing.

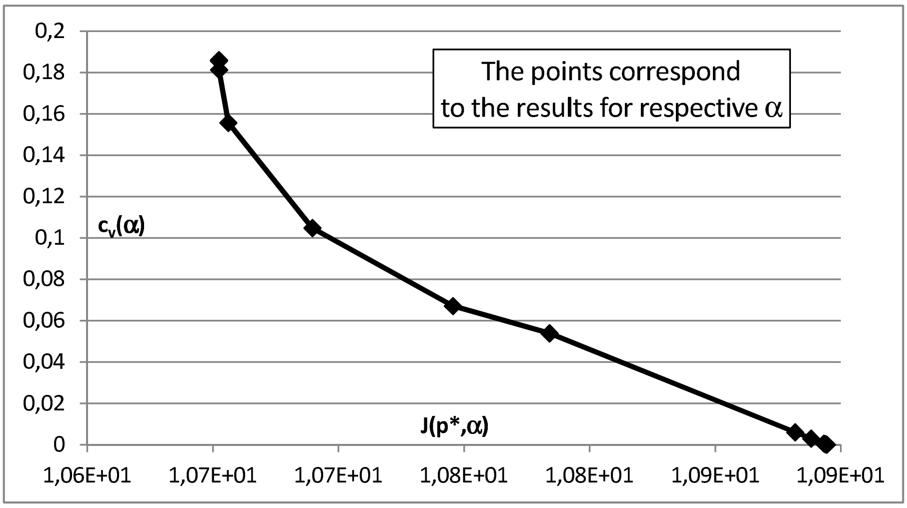


Figure 4: The L-curve – values $J(p^*, \alpha)$, see (12), against values $c_v(\alpha)$, see (19).

optimal is called the L-curve. We plot the value of objective function J against the value $c_v(\alpha)$. The *L-curve-optimal* parameter α^* usually corresponds to the point with maximal curvature. In Figure 4, we plot the L-curve resulting from our numerical tests for the 10% Gaussian noise, i.e. for $c_v = 0.1$. We see that for our "FRAP problem" and a particular noise level, there is not a sharp corner. Furthermore, the question of *optimal* value of α^* may also depend on what the user expects or prefers, if rather small function value $J(p^*, \alpha)$ or more constant solutions $p_1^*, \ldots, p_m^*$, i.e. small value $c_v(\alpha)$, see e.g. [4].

5. Conclusions

The purpose of this paper was to present the real problem residing in the estimation of diffusivity of phycobilisomes on thylakoid membrane based on spatio-temporal FRAP images. While the state-of-the-art methods in FRAP measurement of photosynthetic complexes mobility are usually based on the curve fitting to an analytical (closed form) models, which need some unrealistic conditions to be supposed, our method is based on finite difference approximation of diffusion process and on the minimization of an objective function evaluating both the disparity between the experimental and simulated time-varying concentration profiles and the smoothness of the time evolution of diffusivity. This approach naturally takes into account the time-dependent Dirichlet boundary conditions and can include also a reaction term (e.g. modeling the low level bleaching during scanning) and the time varying fluorescence signal as well.

Our program *CA-FRAP 4.0* is actually under testing, however, for the previously known diffusion coefficient and the synthetic data corrupted by the Gaussian noise it computes satisfactory results. Afterward, we determined the diffusivities for the real data of FRAP measurements (with the red algae *Porphyridium cruentum*). The range of result $10^{-15}\mathrm{m}^2\mathrm{s}^{-1}$ ($10^{-3}\mu\mathrm{m}^2\mathrm{s}^{-1}$) is in agreement with reference values.

Acknowledgements

This work was supported by the project Jihočeské výzkumné centrum akvakultury a biodiverzity hydrocenóz (CENAKVA CZ.1.05/2.1.00/01.0024), OP VaVpI and by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Axelrod D., Koppel D. E., Schlessinger J., Elson E., and Webb W. W.: Mobility measurement by analysis of fluorescence photobleaching recovery kinetics. Biophys. J. **16** (1976), 1055–1069.

[2] Babuška I., Práger M., and Vitásek E.: *Numerical processes in differential equations*. John Wiley & Sons, London, 1966.

[3] Chavent G. and Kunish K.: Regularization in state space. Mathematical Modelling and Numerical Analysis **27** (1995), 535–556.

[4] Engl W. and Grever W.: Using the *L*-curve for determining optimal regularization parameter. Numer. Math. **69** (1994), 25-31.

[5] Hinestroza D., Murio D. A., and Zhan S.: Regularization techniques for nonlinear problems. Comput. Math. Appl. **37** (1999), 145–159.

[6] Irrechukwu O. N. and Levenston M. E.: Improved estimation of solute diffusivity through numerical analysis of FRAP experiments. Cellular and Molecular Bioengineering **2** (2009), 104-117.

[7] Kaňa R., Prášil O., and Mullineaux C. W.: Immobility of phycobilins in the thylakoid lumen of a cryptophyte suggests that protein diffusion in the lumen is very restricted. FEBS letters **583** (2009), 670-674.

[8] Lukšan L., Tůma M., Vlček J., Ramešová N., Šiška M., Hartman J., and Matonoha C.: UFO 2011 – Interactive system for universal functional optimization. Technical Report V-1151, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague 2011 (`http://www.cs.cas.cz/luksan/ufo.html`).

[9] Mueller F., Mazza D., Stasevich T. J., and McNally J. G.: FRAP and kinetic modeling in the analysis of nuclear protein dynamics: what do we really know?. Current Opinion in Cell Biology **22** (2010), 1-9.

[10] Mullineaux C. W., Tobin M. J., and Jones G. R.: Mobility of photosynthetic complexes in thylakoid membranes. Nature **390** (1997), 421-424.

[11] Nocedal J. and Wright S. J.: *Numerical optimization, second edition.* Springer, New York, 2006.

[12] Papáček Š., Kaňa R., and Matonoha C.: Estimation of diffusivity of phycobilisomes on thylakoid membrane based on spatio-temporal FRAP images. Math. Compt. Modelling (2012), doi:10.1016/j.mcm.2011.12.029.

[13] Papáček Š. and Matonoha C.: Error analysis of three methods for the parameter estimation problem based on spatio-temporal FRAP measurement. SNA 2013, submitted.

[14] Sbalzarini I. F.: Analysis, modeling and simulation of diffusion processes in cell biology. VDM Verlag Dr. Muller, 2009.

[15] Sprague B. L., Pego R. L., Stavreva D. A., and McNally J. G.: Analysis of binding reactions by fluorescence recovery after photobleaching. Biophysical Journal. **86** (2004), 3473–3495.

[16] Thomas J. and Webb W. W.: Fluorescence photobleaching recovery: a probe of membrane dynamics. In: S. Grinstein and K. Foskett (Eds.), *Non-Invasive Techniques in Cell Biology*, pp. 129–152. Wiley-Liss, Inc., 1990.

[17] Tychonoff A. N. and Arsenin V. Y.: *Solution of Ill-posed problems.* Washington, Winston & Sons, 1977.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

NUMERICAL SIMULATION OF GENERALIZED NEWTONIAN AND OLDROYD-B FLUIDS FLOW

Radka Keslerová, Karel Kozel

CTU in Prague, Faculty of Mechanical Engineering,
Department of Technical Mathematics
Karlovo nám. 13, 121 35 Prague, Czech Republic
keslerov@marian.fsik.cvut.cz, Karel.Kozel@fs.cvut.cz

Abstract

This work deals with the numerical solution of generalized Newtonian and Oldroyd-B fluids flow. The governing system of equations is based on the system of balance laws for mass and momentum for incompressible laminar viscous and viscoelastic fluids. Two different definition of the stress tensor are considered. For viscous case Newtonian model is used. For the viscoelastic case Oldroyd-B model is tested. Both presented models can be generalized. In this case the viscosity is defined as a shear rate dependent viscosity function $\mu(\dot{\gamma})$. One of the most frequently used shear-thinning models is a cross model. Numerical solution of the described models is based on cell-centered finite volume method using explicit Runge Kutta time integration. The numerical results of generalized Newtonian and generalized Oldroyd-B fluids flow obtained by this method are presented and compared.

1. Mathematical model

In order to simulate the fluids flow in the channel the system of balance laws of mass and momentum for incompressible fluids are considered, [1], [4]:

$$\text{div } \boldsymbol{u} = 0 \tag{1}$$

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}.\nabla)\boldsymbol{u} = -\nabla P + \text{div } \mathbf{T} \tag{2}$$

where P is the pressure, ρ is the constant density, $\boldsymbol{u}$ is the velocity vector, $\boldsymbol{u} = (u, v, w)^T$. The symbol $\mathbf{T}$ represents the stress tensor.

1.1. Stress tensor

In this work the different definition of the stress tensor are used.

In the case of viscous fluids the used model corresponding to Newtonian fluid is *Newtonian model*:

$$\mathbf{T} = 2\mu\mathbf{D} \tag{3}$$

where μ is dynamic viscosity and tensor $\mathbf{D}$ is symmetric part of the velocity gradient defined by the relation $\mathbf{D} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$.

If viscoelastic fluids are considered *Maxwell model* as the simplest viscoelastic model is used:

$$\mathbf{T} + \lambda_1 \frac{\delta \mathbf{T}}{\delta t} = 2\mu \mathbf{D} \tag{4}$$

where λ_1 has dimension of time and denotes the relaxation time. The symbol $\frac{\delta}{\delta t}$ represents upper convected derivative (see (8))

By combination of these two models the behaviour of mixture of viscous and viscoelastic fluids can be described. Such a model is called *Oldroyd-B model* and it has the form

$$\mathbf{T} + \lambda_1 \frac{\delta \mathbf{T}}{\delta t} = 2\mu \left(\mathbf{D} + \lambda_2 \frac{\delta \mathbf{D}}{\delta t} \right) \tag{5}$$

the parameters λ_1, λ_2 are relaxation and retardation time.

The stress tensor $\mathbf{T}$ can be decomposed to the Newtonian part $\mathbf{T}_s$ and viscoelastic part $\mathbf{T}_e$ ($\mathbf{T} = \mathbf{T}_s + \mathbf{T}_e$) and

$$\mathbf{T}_s = 2\mu_s \mathbf{D}, \qquad \mathbf{T}_e + \lambda_1 \frac{\delta \mathbf{T}_e}{\delta t} = 2\mu_e \mathbf{D}, \tag{6}$$

where

$$\frac{\lambda_2}{\lambda_1} = \frac{\mu_s}{\mu_s + \mu_e}, \qquad \mu = \mu_s + \mu_e. \tag{7}$$

The upper convected derivative $\frac{\delta}{\delta t}$ is defined (for general tensor $\mathbf{M}$) by the relation (see [2])

$$\frac{\delta \mathbf{M}}{\delta t} = \frac{\partial \mathbf{M}}{\partial t} + (\boldsymbol{u}.\nabla)\mathbf{M} - (\mathbf{WM} - \mathbf{MW}) - (\mathbf{DM} + \mathbf{MD}) \tag{8}$$

where $\mathbf{D}$ is symmetric part of the velocity gradient

$$\mathbf{D} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T) = \frac{1}{2} \begin{pmatrix} 2u_x & u_y + v_x & u_z + w_x \\ u_y + v_x & 2v_y & v_z + w_y \\ w_x + u_z & w_y + v_z & 2w_z \end{pmatrix} \tag{9}$$

and $\mathbf{W}$ is antisymmetric part of the velocity gradient

$$\mathbf{W} = \frac{1}{2}(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^T) = \frac{1}{2} \begin{pmatrix} 0 & u_y - v_x & u_z - w_x \\ v_x - u_y & 0 & v_z - w_y \\ w_x - u_z & w_y - v_z & 0 \end{pmatrix}. \tag{10}$$

The governing system (1), (2) of equations is completed by the equation for the viscoelastic part of the stress tensor

$$\frac{\partial \mathbf{T}_e}{\partial t} + (\boldsymbol{u}.\nabla)\mathbf{T}_e = \frac{2\mu_e}{\lambda_1}\mathbf{D} - \frac{1}{\lambda_1}\mathbf{T}_e + (\mathbf{W}\mathbf{T}_e - \mathbf{T}_e\mathbf{W}) + (\mathbf{D}\mathbf{T}_e + \mathbf{T}_e\mathbf{D}). \tag{11}$$

Both models could be generalized. In this case the viscosity μ is no more constant, but is defined by viscosity function according to the cross model (for more details see [11])

$$\mu(\dot{\gamma}) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (\lambda\dot{\gamma})^b)^a} \tag{12}$$

where

$$\dot{\gamma} = 2\sqrt{\frac{1}{2}\mathrm{tr}\ \mathbf{D}^2} \tag{13}$$

$$\begin{array}{cc} \mu_0 = 1.6 \cdot 10^{-1} Pa \cdot s & \mu_\infty = 3.6 \cdot 10^{-3} Pa \cdot s \\ a = 1.23, b = 0.64 & \lambda = 8.2s. \end{array}$$

2. Numerical solution

In this work the steady state solution is considered. In this case an artificial compressibility method can be applied. It means that the continuity equation is completed by the time derivative of the pressure in the form (for more details see e.g. [3], [8]):

$$\frac{1}{\beta^2}\frac{\partial p}{\partial t} + \mathrm{div}\ \boldsymbol{u} = 0, \quad \beta \in \mathbb{R}^+. \tag{14}$$

The system of equations (including the modified continuity equation) could be rewritten in the conservative form.

$$\tilde{R}_\beta W_t + F^c_x + G^c_y + H^c_z = F^v_x + G^v_y + H^v_z + S, \qquad \tilde{R}_\beta = \mathrm{diag}(\frac{1}{\beta^2}, 1, \cdots, 1) \tag{15}$$

where W is the vector of unknowns, F^c, G^c, H^c are inviscid fluxes, F^v, G^v, H^v are viscous fluxes, and the source term S.

The following special parameters settings related to four specific models will be used in our numerical simulation:

Newtonian	$\mu(\dot{\gamma}) = \mu_s = const.$	$\mathbf{T}_e \equiv 0$
Generalized Newtonian	$\mu(\dot{\gamma})$	$\mathbf{T}_e \equiv 0$
Oldroyd-B	$\mu(\dot{\gamma}) = \mu_s = const.$	$\mathbf{T}_e$
Generalized Oldroyd-B	$\mu(\dot{\gamma})$	$\mathbf{T}_e$

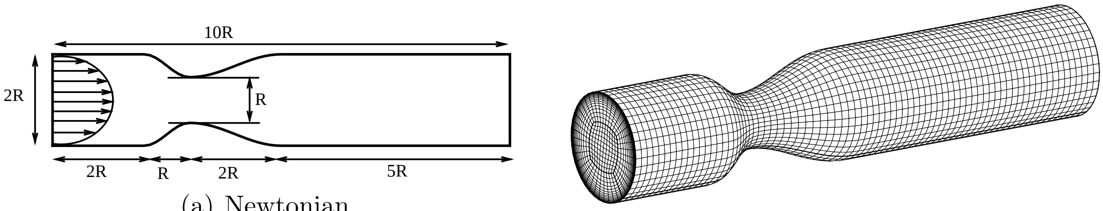
The (15) is discretized in space by the cell-centered finite volume method (see [7]) and the arising system of ODEs is integrated in time by the explicit multistage Runge–Kutta scheme (see [8], [10], [11]).

2.1. Boundary conditions

The flow is modelled in a bounded computational domain where a boundary is divided into three mutually disjoint parts: a solid wall, an outlet and an inlet. At the inlet Dirichlet boundary condition for velocity vector is used and for a pressure and the stress tensor Neumann boundary condition is used. At the outlet the pressure value is given and for the velocity vector and the stress tensor Neumann boundary condition is used. The homogeneous Dirichlet boundary condition for the velocity vector is used on the wall. For the pressure and stress tensor Neumann boundary condition is considered.

3. Numerical results

This section deals with the comparison of the numerical results of Newtonian and Oldroyd-B fluids. Numerical tests are performed in an idealized stenosed vessel. The stenosed vessel is assumed to be three-dimensional with circular cross-section. Figure 3 shows the shape of the tested domain. The computational domain is discretized using a structured, wall fitted mesh with hexahedral cells and uniform axial cell spacing. The similar numerical results can be found in [1], [2].



(a) Newtonian

(b) Generalized Newtonian

Figure 1: Structure of the computational domain.

The following model parameters are:

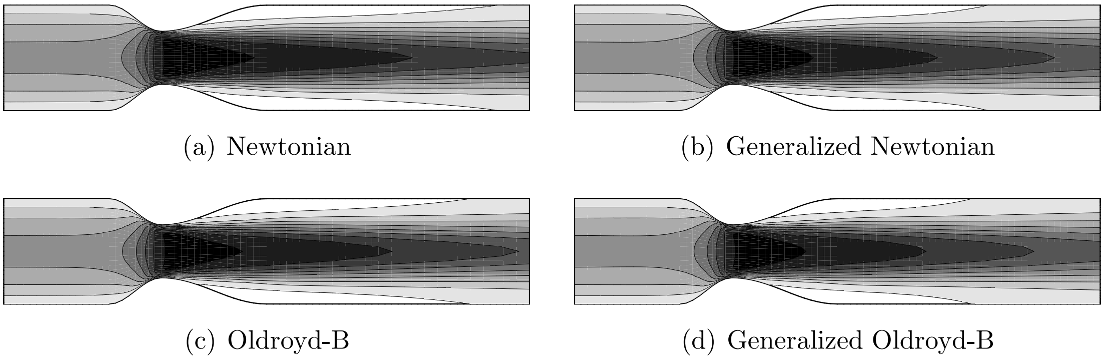
$$\begin{array}{cc}
\mu_e = 4.0 \cdot 10^{-4} Pa \cdot s & \mu_s = 3.6 \cdot 10^{-3} Pa \cdot s \\
\lambda_1 = 0.06 s & \lambda_2 = 0.054 s \\
U_0 = 0.0615 m \cdot s^{-1} & L_0 = 2R = 0.0062 m \\
\mu_0 = \mu = \mu_s + \mu_e & \rho = 1050 kg \cdot m^{-3}
\end{array}$$

Note that the fluid motion can be characterized by parameters: Reynolds number and Weissenberg number. Weissenberg number is proportional to the relaxation time of the fluid. These special data corresponds to Reynolds and Weissenberg numbers:

$$Re = \frac{\rho U_0 L_0}{\mu_0} = 100, \qquad We = \frac{\lambda_1 U_0}{L_0} = 0.6 \tag{16}$$

In Figure 2 the comparison of the axial velocity isolines is presented. To emphasize the flow separation behind the stenosis the regions of reversal flow (with respect to axial direction) are marked with white color.

Pressure and velocity distribution along the axis for both tested fluids models is shown in Figure 3. By simple observation one can conclude that the main effect of the Oldroyd-B fluids behavior is visible mainly in the recirculation zone.

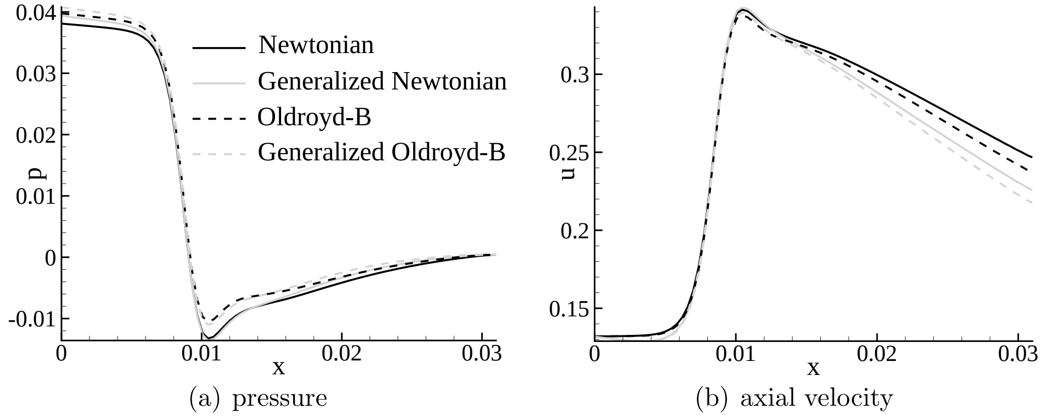

(a) Newtonian

(b) Generalized Newtonian

(c) Oldroyd-B

(d) Generalized Oldroyd-B

Figure 2: Axial velocity isolines for generalized Oldroyd-B fluids.



(a) pressure

(b) axial velocity

Figure 3: Pressure and axial velocity distribution along the central axis of the channel.

4. Conclusions

Newtonian and Oldroyd-B models with their generalized modification have been considered for numerical simulation of fluids flow in the idealized axisymmetric stenosis. The cell-centered finite volume solver for incompressible laminar viscous and viscoelastic fluids flow has been described. For time integration the explicit Runge–Kutta method was considered. The numerical results obtained by this method are presented. The differences between these tested fluids are given mainly in the separation region. These results clearly show that for shear-thinning flows the recirculation zone becomes shorter. This could be explained by the specific choice of the characteristic viscosity μ_∞ for the reference Newtonian and (non-generalized) Oldroyd-B solution.

Acknowledgements

This work was partly supported by the grant GACR P201/11/1304 and GACR 201/09/0917.

References

[1] Bodnar, T., Sequeira, A., and Prosi, M.: On the shear-thinning and viscoelastic effects of blood flow under various flow rates. Appl. Math. Comput. **217** (2011), 5055–5067.

[2] Bodnar, T. and Sequeira, A.: Numerical study of the significance of the non-Newtonian nature of blood in steady flow through stenosed vessel. In: R. Rannacher, A. Sequeira (Eds.), *Advances in Mathematical Fluid Mechanics*, pp. 83–104. Springer-Verlag, 2010.

[3] Chorin, A. J.: A numerical method for solving incompressible viscous flow problem. J. Comput. Phys. **135** (1967) 118–125.

[4] Dvořák, R. and Kozel, K.: *Mathematical modelling in aerodynamics (in Czech).* CTU, Prague, Czech Republic, 1996.

[5] Gaitonde, A. L.: A dual-time method for two dimensional unsteady incompressible flow calculations. International Journal for Numerical Methods in Engineering. **41** (1998), 1153–1166.

[6] Keslerová, R. and Kozel, K.: Numerical modelling of incompressible flows for Newtonian and non-Newtonian fluids. Math. Comput. Simulation **80** (2010), 1783–1794.

[7] LeVeque, R.: *Finite-volume methods for hyperbolic problems.* Cambridge University Press, 2004.

[8] Keslerová, R. and Kozel, K.: Numerical modelling of incompressible flows for Newtonian and non-Newtonian fluids. Math. Comput. Simulation **80** (2010), 1783–1794.

[9] Robertson, A. M., Sequeira, A., and Kameneva, M. V.: *Hemorheology.* Birkhäuser Verlag Basel, Switzerland, 2008.

[10] Jameson, A., Schmidt, W., and Turkel, E.: Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. AIAA 14th Fluid and Plasma Dynamic Conference California, 1981.

[11] Vimmr, J. and Jonášová, A.: Non-Newtonian effects of blood flow in complete coronary and femoral bypasses. Math. Comput. Simulation **80** (2010), 1324–1336.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

HEAT EXPOSURE OPTIMIZATION APPLIED TO MOULDING PROCESS IN THE AUTOMOTIVE INDUSTRY

Jiřina Královcová [1], Ladislav Lukšan [1, 2], Jaroslav Mlýnek [1]

[1] Technická Univerzita v Liberci
Hálkova 6, 461 17 Liberec
[2] Ústav informatiky AVČR
Pod Vodárenskou věží 2, 182 07 Praha 8
luksan@cs.cas.cz

Abstract

This contribution contains a description and comparison of two methods applied to exposure optimization applied to moulding process in the automotive industry.

1. Introduction

Consider an aluminium shape weighting approximately 300 kg. This shape should be uniformly warmed to $270^o C$ by approximately 100 heating lamps of the same power. Every lamp is defined by the coordinates of its endpoints A, B and the lighting direction u (9 parameters). All the lamps have the same length d. The shape surface is defined by using approximately 10000 plane elements. Every plane element is represented by the coordinates of its center T and its outer normal v (6 parameters). The initial coordinates of the lamps are given. To obtain a uniform exposure of the surface to the heat radiation, we optimize the lamp coordinates.

2. Formulation of a constrained optimization problem

2.1. Equations for the exposure of a plane element by a lamp

Let $x^T = (x_1^T, x_2^T, x_3^T)$ be the center of a plane element, $x^N = (x_1^N, x_2^N, x_3^N)$ be its outer normal, $x^A = (x_1^A, x_2^A, x_3^A)$, $x^B = (x_1^B, x_2^B, x_3^B)$ be the endpoints of the lamp and $x^S = (x_1^S, x_2^S, x_3^S)$ be the lighting direction of the lamp. We also denote $v = -x^N$, $u = x^S$ and use the following constraints

$$\sum_{i=1}^{3}(x_i^S)^2 = 1, \qquad \sum_{i=1}^{3} x_i^S(x_i^B - x_i^A) = 0, \qquad \sum_{i=1}^{3}(x_i^B - x_i^A)^2 = d^2, \tag{1}$$

where d is the length of the lamp. The first constraint ensures the unit length of vector x^S, the second its orthogonality to the axis of the lamp, and the third stabilizes the length of the lamp.

The lamp is a linear body of the length d, consisting of p lighting elements of lengths $d_k = d/p$, $1 \leq k \leq p$. The connecting line between the center of the lighting element and the center of the plane element is expressed as

$$w_k = x^T - (1-\lambda_k)x^A - \lambda_k x^B, \quad \lambda_k = \frac{2k-1}{2p}, \tag{2}$$

where $1 \leq k \leq p$. The exposure I of the selected plane element by the particular lamp is given by the formula

$$I = \sum_{k=1}^{p} I_k, \quad I_k = \left(3\alpha_k + \frac{1}{2}\sqrt{1-\alpha_k^2}\right)\frac{\beta_k}{\|w_k\|^2}d_k, \tag{3}$$

where

$$\alpha_k = \frac{u^T w_k}{\|u\|\|w_k\|} = \tilde{u}^T\tilde{w}_k, \quad \beta_k = \frac{v^T w_k}{\|v\|\|w_k\|} = \tilde{v}^T\tilde{w}_k,$$

and

$$\tilde{u} = u/\|u\|, \quad \tilde{v} = v/\|v\|, \quad \tilde{w_k} = w_k/\|w_k\|$$

(the expression for I_k has been obtained by measurements). Analytical expressions for the derivatives of the exposure I with respect to the elements of vectors x^A, x^B, x^S (elements of the vectors x^T, x^N are constants, since the heated surface is fixed) have the form

$$\begin{aligned}
\frac{\partial I}{\partial x_i^A} &= \sum_{k=1}^{p}\frac{\partial I_k}{\partial x_i^A} = -\sum_{k=1}^{p}(1-\lambda_k)\frac{\partial I_k}{\partial w_{ik}}, \\
\frac{\partial I}{\partial x_i^B} &= \sum_{k=1}^{p}\frac{\partial I_k}{\partial x_i^B} = -\sum_{k=1}^{p}\lambda_k\frac{\partial I_k}{\partial w_{ik}} \\
\frac{\partial I}{\partial x_i^S} &= \sum_{k=1}^{p}\frac{\partial I_k}{\partial x_i^S} = \sum_{k=1}^{p}\frac{\partial I_k}{\partial u_i},
\end{aligned}$$

so they can be easily computed from gradients

$$\begin{aligned}
\nabla_u I_k &= \left(3 - \frac{1}{2}\frac{\alpha_k}{\sqrt{1-\alpha_k^2}}\right)\frac{\beta_k d_k}{\|w_k\|^2}\nabla_u\alpha_k, \\
\nabla_{w_k} I_k &= \left(3 - \frac{1}{2}\frac{\alpha_k}{\sqrt{1-\alpha_k^2}}\right)\frac{\beta_k d_k}{\|w_k\|^2}\nabla_{w_k}\alpha_k \\
&+ \left(3\alpha_k + \frac{1}{2}\sqrt{1-\alpha_k^2}\right)\left(\frac{d_k}{\|w_k\|^2}\nabla_{w_k}\beta_k - 2\frac{\beta_k d_k}{\|w_k\|^4}w_k\right).
\end{aligned}$$

Furthermore, one has

$$\begin{aligned}
\nabla_u \alpha_k &= \frac{w_k}{\|u\|\|w_k\|} - \frac{u^T w_k}{\|u\|\|w_k\|}\frac{u}{\|u\|^2} = \frac{1}{\|u\|}(\tilde{w}_k - \alpha_k \tilde{u}), \\
\nabla_{w_k} \alpha_k &= \frac{u}{\|u\|\|w_k\|} - \frac{u^T w_k}{\|u\|\|w_k\|}\frac{w_k}{\|w_k\|^2} = \frac{1}{\|w_k\|}(\tilde{u} - \alpha_k \tilde{w}_k), \\
\nabla_{w_k} \beta_k &= \frac{v}{\|v\|\|w_k\|} - \frac{v^T w_k}{\|v\|\|w_k\|}\frac{w_k}{\|w_k\|^2} = \frac{1}{\|w_k\|}(\tilde{v} - \beta_k \tilde{w}_k),
\end{aligned}$$

and after substitution we obtain

$$\begin{aligned}
\nabla_u I_k &= \left(3 - \frac{1}{2}\frac{\alpha_k}{\sqrt{1-\alpha_k^2}}\right)\frac{\beta_k d_k}{\|u\|\|w_k\|^2}(\tilde{w}_k - \alpha_k \tilde{u}) && (4) \\
\nabla_{w_k} I_k &= \left(3 - \frac{1}{2}\frac{\alpha_k}{\sqrt{1-\alpha_k^2}}\right)\frac{\beta_k d_k}{\|w_k\|^3}(\tilde{u} - \alpha_k \tilde{w}_k) \\
&+ \left(3\alpha_k + \frac{1}{2}\sqrt{1-\alpha_k^2}\right)\frac{d_k}{\|w_k\|^3}(\tilde{v} - 3\beta_k \tilde{w}_k). && (5)
\end{aligned}$$

It is not necessary to known the elements of vectors u, v and w_k, $1 \le k \le p$. We use only their Euclidean norms and the elements of normalized vectors $\tilde{u}$, $\tilde{v}$ and $\tilde{w}_k$, $1 \le k \le p$, in our numerical algorithm.

2.2. Objective function and constraints for the uniform exposure

We have n_e plane elements and n_l lamps. Every plane element can be exposed by several lamps. Let L_j be a set of indices of the lamps that expose the jth plane element. Choose $1 \le j \le n_e$ and $l \in L_j$. If we denote I_{jl} the exposure of the jth element by the lth lamp, (this value corresponds to the value I from the previous subsection), then the total exposure I_j of the jth element is given by the formula

$$I_j = \sum_{l \in L_j} I_{jl}.$$

The derivatives of I_j are computed by the formulas

$$\begin{aligned}
&\frac{\partial I_j}{\partial x_{il}^A} = \frac{\partial I_{jl}}{\partial x_{il}^A}, \quad \frac{\partial I_j}{\partial x_{il}^B} = \frac{\partial I_{jl}}{\partial x_{il}^B}, \quad \frac{\partial I_j}{\partial x_{il}^S} = \frac{\partial I_{jl}}{\partial x_{il}^S}, \quad l \in L_j, \\
&\frac{\partial I_j}{\partial x_{il}^A} = 0, \qquad \frac{\partial I_j}{\partial x_{il}^A} = 0, \qquad \frac{\partial I_j}{\partial x_{il}^A} = 0, \qquad l \notin L_j,
\end{aligned}$$

where we substitute the previously defined quantities. Let $\overline{I}$ be the prescribed value of the exposure (the same for all elements of the shape surface). Then

$$F(x) = \frac{1}{2}\sum_{j=1}^{n_e}(I_j - \overline{I})^2, \tag{6}$$

where vector x has elements x_{1l}^A, x_{2l}^A, x_{3l}^A, x_{1l}^B, x_{2l}^B, x_{3l}^B, x_{1l}^S, x_{2l}^S, x_{3l}^S, $1 \le l \le n_l$ (nine for every lamp). One has

$$\frac{\partial F(x)}{\partial x_{il}^A} = \sum_{j=1}^{n_e} (I_j - \overline{I}) \frac{\partial I_j}{\partial x_{il}^A}, \quad \frac{\partial F(x)}{\partial x_{il}^B} = \sum_{j=1}^{n_e} (I_j - \overline{I}) \frac{\partial I_j}{\partial x_{il}^B}, \quad \frac{\partial F(x)}{\partial x_{il}^S} = \sum_{j=1}^{n_e} (I_j - \overline{I}) \frac{\partial I_j}{\partial x_{il}^S},$$

where we substitute quantities computed in the previous relations. The prescribed value of the exposure is determined by the initial positions of the lamps through the formula

$$\overline{I} = \frac{1}{n_e} \sum_{j=1}^{n_e} I_j.$$

The objective function $F(x)$ is minimized in the feasible region given by the equality constraints (1) (three constraints for every lamp). Computation of derivatives of these constraints with respect to the elements of vector x is easy. All constraints are sparse, so the memory size and the number of arithmetic operations are not large.

The described problem consists in the minimization of a sum of squares with respect to nonlinear equality constraints. The number of partial functions in the sum of squares is $n_e \sim 10000$ (the number of the plane elements). The number of variables is $9n_l \sim 900$ (nine for every lamp). The Hessian matrix of the objective function is not sparse. The number of nonlinear equality constraints is $3n_l \sim 300$ (three for every lamp). The Jacobian matrix of nonlinear equality constraints is sparse. These facts have an influence on the choice of the numerical method. We have used the recursive quadratic programming method with iterative solution of linear KKT system by indefinitely preconditioned conjugate gradient method (see [3]). This method uses partial derivatives derived above.

3. Formulation of an unconstrained optimization problem

In this section, we use constraints (1) to eliminate vector $u = x^S$ from the formula (3). For this purpose we assume that the basis of the warmed shape lies in the horizontal plane, the lamps are placed over the heated surface and the lighting directions of the lamps are mostly perpendicular to the basis of the shape. This assumption is not very restrictive and results obtained in this way are comparable with those obtained by approach used in the previous section.

Let y be a vector parallel to vector $x^B - x^A$. Then we can write $x^B - x^A = (y/\|y\|)d$ and $w_k = x^T - x^A - \lambda_k(y/\|y\|)d$, $1 \le k \le p$, where $d = \|x^B - x^A\|$ (see (2)). By our assumption, the angle between vector $u = x^S$, which is perpendicular to vector y, and the normal $e = (0, 0, -1)$ is minimal. If the norm of vector u is unit, it can be uniquely determined from vectors y and e.

Theorem 1 *Vector*

$$u = \frac{e + \lambda y}{\sqrt{e^T(e + \lambda y)}}, \qquad \lambda = -\frac{e^T y}{y^T y}.$$

is the solution of the optimization problem

$$\text{Maximize} \quad e^T u \quad \text{subject to} \quad y^T u = 0, \quad u^T u = 1.$$

Since the length of vector u can be arbitrary, we put

$$u = e - \frac{e^T y}{y^T y} y = e - e^T \tilde{y} \tilde{y},$$

where $\tilde{y} = y/\|y\|$ (vector $e = (0, 0, -1)$ has the unit norm). To compute the gradient of the objective function, we need the transposed Jacobian matrices of vectors u and w_k (with respect to y), which we denote $\nabla_y u$ and $\nabla_y w_k$.

Theorem 2 *One has*

$$\begin{aligned}
\nabla_y u &= \left(2 \frac{y\, y^T}{y^T y} - I \right) \frac{e^T y}{y^T y} - \frac{e\, y^T}{y^T y} = \frac{1}{\|y\|} \left((2\, \tilde{y} \tilde{y}^T - I)\, e^T \tilde{y} - e\, \tilde{y}^T \right) \\
\nabla_y w_k &= \frac{\lambda_k d}{\|y\|} \left(\frac{y\, y^T}{y^T y} - I \right) = \frac{\lambda_k d}{\|y\|} \left(\tilde{y} \tilde{y}^T - I \right)
\end{aligned}$$

The exposure (3) now depends on vectors $x = x^A$ and y (then $x^B = x^A + (y/\|y\|)d$ and vector $x^S = u$ is obtained by Theorem 1). Analytical expressions for gradients of the exposure I have the form

$$\nabla_x I = \sum_{k=1}^{p} \nabla_x I_k = -\sum_{k=1}^{p} \nabla_{w_k} I_k, \qquad \nabla_y I = \sum_{k=1}^{p} \nabla_y I_k = \sum_{k=1}^{p} (\nabla_y u \nabla_u I_k + \nabla_y w_k \nabla_{w_k} I_k),$$

where gradients $\nabla_u I_k$ and $\nabla_{w_k} I_k$ are computed by formulas (4) and (5). Note that using Theorem 2 we can write

$$\nabla_y u \nabla_u I_k + \nabla_y w_k \nabla_{w_k} I_k = -\frac{1}{\|y\|} \left(\gamma_e (\nabla_u I_k - 2\gamma_u \tilde{y}) + \gamma_u e + \lambda_k d (\nabla_{w_k} I_k - \gamma_{w_k} \tilde{y}) \right),$$

where $\gamma_e = \tilde{y}^T e$, $\gamma_u = \tilde{y}^T \nabla_u I_k$ and $\gamma_{w_k} = \tilde{y}^T \nabla_{w_k} I_k$.

Analogously to the previous section, we minimize the sum of squares (6), but now without constraints. The number of variables is $6n_l \sim 600$ (six for every lamp). The Hessian matrix of the objective function is not sparse. This fact have an influence to the choice of the numerical method. We have used the combination of the Gauss-Newton method and the BFGS variable metric method, which is described in [2]. This combination uses partial derivatives derived above.

4. Numerical comparison

The purpose if this section is to show that the elimination of constraints and the solution of the unconstrained optimization problem significantly increase the efficiency of the computation. To demonstrate this fact, we have used four test problems L1–L4 introduced in [1]. The following table contains the results corresponding to the two approaches described in the previous sections. Here NIT and NFV are the numbers of iterations and function evaluations, F_0 and F are the initial and the final values of the objective function. Computational time is given in seconds. The $*$ symbol means that 10000 function evaluations did not suffice for obtaining the solution. The results were obtained by the interactive system for universal functional optimization UFO described in [4].

The following figure demonstrates the solution of problem L1.

		Method with constraints				Method without constraints			
Problem	F_0	NIT	NFV	Time	F	NIT	NFV	Time	F
L1	169.53	1125	4653	396.14	27.68	74	165	18.67	29.16
L2	198.14	712	2456	218.68	31.22	83	186	21.22	32.75
L3	22.50	382	812	118.79	14.25	57	126	20.50	12.02
L4	11.86	1094	10007	742.15	2.03 *	43	98	9.71	1.27

Table 1: Comparison of two approaches for the heat exposure optimization.

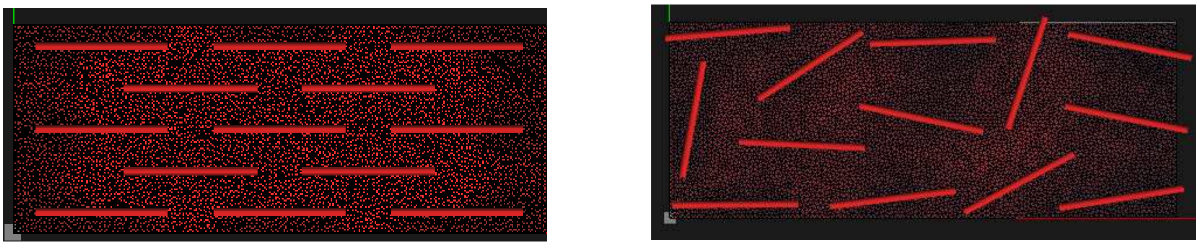


Figure 1: Initial (left) and final (right) positions of the lamps.

Acknowledgements

This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Královcová, J., Lukšan, L., and Mlýnek, J.: Optimalizace osvitu pro tepelný ohřev forem v automobilovém průmyslu. Tech. Rep. V-1050, ÚI AVČR, Praha, 2009.

[2] Lukšan, L.: Hybrid methods for large sparse nonlinear least squares. J. Optim. Theory Appl. **89** (1996), 575–595.

[3] Lukšan, L., Vlček, J.: Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems. Numer. Linear Algebra Appl. **5** (1998), 219–247.

[4] Lukšan, L., Tůma, M., Vlček, J., Ramešová, N., Šiška, M., Hartman, J., Matonoha, C.: UFO 2008 – Interactive system for universal functional optimization. Tech. Rep. V-1151, ÚI AVČR, Praha, 2011.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

TANGENTIAL FIELDS IN OPTICAL DIFFRACTION PROBLEMS

Jiří Krček, Jaroslav Vlček, Arnošt Žídek

Department of Mathematics and Descriptive Geometry
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava - Poruba, Czech Republic
jiri.krcek@vsb.cz, jaroslav.vlcek@vsb.cz, arnost.zidek@vsb.cz

Abstract

Optical diffraction for periodical interface belongs to relatively fewer exploited application of boundary integral equations method. Our contribution presents the formulation of diffraction problem based on vector tangential fields, for which the periodical Green function of Helmholtz equation is of key importance. There are discussed properties of obtained boundary operators with singular kernel and a numerical implementation is proposed.

1. Introduction

Development of optical micro- and nanostructures with periodical ordering takes important place in many branches of integrated optics or nano-technology. The geometrical and material optimization of the sensors, switching elements and many other devices depends on the accurate control of their parameters. Besides less or more complicated experiments, theoretical studies are carried out including mathematical models of electromagnetic wave interaction with geometrically or material-wise modulated media. Generally, these models consist in the solving of Maxwell equations with appropriate boundary conditions. Diffraction of optical wave on an interface between two different media is one of frequently solved problem, where the rigorous choice of theoretical approach plays important role.

In the last two decades, there were published numerous works treating of optical diffraction in periodical structures - see [1] and references therein. One of relatively new approaches is based on Boundary Integral Equations (BIE), theoretical background of which is referred e.g. in [2]. In this article, we aim to show the especial integral formulation of the boundary problem for system of Maxwell equations. To this purpose, we introduce tangential vector fields and study the properties of derived integral operators.

2. Formulation of the problem

Let's denote $\boldsymbol{X} = (x_1, x_2, x_3) \in \mathbb{R}^3$ and further $S : x_3 = f(x_1)$ a surface which we consider to be smooth with normal vector $\boldsymbol{\nu}$ and periodically modulated in coordinate x_1 with period Λ and uniform in the x_2 direction, see Fig.1.

The interface S divides the space into two semi-infinite homogeneous regions $\Omega^{(1)} = \{\boldsymbol{X} \in \mathbb{R}^3,\ x_3 > f(x_1)\}$, $\Omega^{(2)} = \{\boldsymbol{X} \in \mathbb{R}^3,\ x_3 < f(x_1)\}$ with constant relative permittivities $\varepsilon^{(1)} \neq \varepsilon^{(2)}$, $\varepsilon^{(1)} \in \mathbb{R}$ and $\varepsilon^{(2)} \in \mathbb{C}$, $\mathrm{Re}\,(\varepsilon^{(2)}) > 0$, $\mathrm{Im}\,(\varepsilon^{(2)}) \geq 0$, and, relative permeabilities $\mu^{(1)} = \mu^{(2)} = 1$ (materials are magnetically neutral).

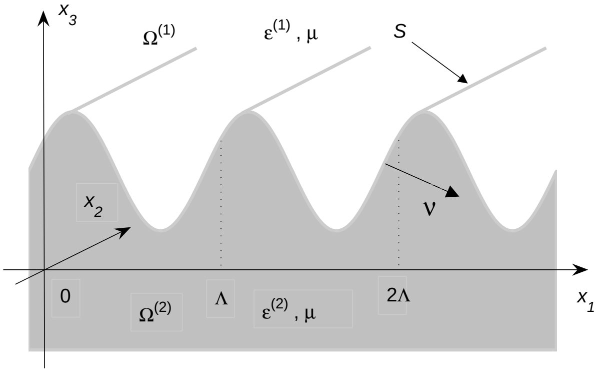


Figure 1: Structure of regions with common periodical boundary

We aim to solve optical diffraction problem for monochromatic plane wave with wavelength λ, i.e. with wave number $k_0 = 2\pi/\lambda$ that is incoming from $\Omega^{(1)}$ under the angle of incidence θ measured from x_3 direction. We seek for space-dependent amplitudes $\boldsymbol{E}^{(j)} = \boldsymbol{E}|_{\Omega^{(j)}}$, $\boldsymbol{H}^{(j)} = \boldsymbol{H}|_{\Omega^{(j)}}$ of the electromagnetic field intensity vectors $\boldsymbol{E}(\boldsymbol{X})\mathrm{e}^{-\mathrm{i}\omega t}$, $\boldsymbol{H}(\boldsymbol{X})\mathrm{e}^{-\mathrm{i}\omega t}$, where $\omega = c/\lambda$ and c represents the light velocity in the free space. Especially, we suppose the TM polarization of the incident wave, for which $\boldsymbol{E}^{(j)} = (E_1^{(j)},\ 0,\ E_3^{(j)})$, $\boldsymbol{H}^{(j)} = (0,\ H_2^{(j)},\ 0)$. Therefore, the Maxwell problem leads to the Helmholtz equations for the scalar components $H_2^{(j)}(\boldsymbol{X})$,

$$\Delta H_2^{(j)} + k_0^2 \varepsilon^{(j)} H_2^{(j)} = 0 \qquad \text{on} \quad \Omega^{(j)}\ , \quad j = 1, 2. \tag{1}$$

The tangential components of the fields are continuous on the boundary, i.e.

$$\boldsymbol{\nu} \times (\boldsymbol{E}^{(1)} - \boldsymbol{E}^{(2)}) = \boldsymbol{o}\ , \qquad \boldsymbol{\nu} \times (\boldsymbol{H}^{(1)} - \boldsymbol{H}^{(2)}) = \boldsymbol{o} \qquad \text{on } S\ . \tag{2}$$

For the far fields, the well-known Sommerfeld's radiation convergence conditions hold that enable to consider the problem on the common interface S only [3].

The incident field at zero diffraction order is characterized by the relation

$$\boldsymbol{H}_0^{(1-)} = \mathrm{e}^{-\mathrm{i}\omega t} \mathrm{e}^{\mathrm{i}(\alpha x_1 + \beta_0^{(1-)} x_3)} \boldsymbol{e}_2\ , \qquad \boldsymbol{e}_2 = (0, 1, 0)\ , \tag{3}$$

where $\alpha = k_0 \sqrt{\varepsilon^{(1)}} \sin\theta$ and $\beta_0^{(1-)}$ is the propagation constant defined below.

This optical beam is diffracted into reflected wave in $\Omega^{(1)}$ and transmitted one in $\Omega^{(2)}$, which are represented by countable sets of modes with wave vectors

$$\boldsymbol{k}_m^{(j\pm)} = (\alpha_m, 0, \beta_m^{(j\pm)})\,, \quad \alpha_m = \alpha + 2\pi m/\Lambda, \quad (\beta_m^{(j\pm)})^2 = k_0^2 \varepsilon^{(j)} - \alpha_m^2\,, \quad m \in \mathbb{Z}. \tag{4}$$

The sign in superscript denotes propagation direction with respect to the x_3 axis orientation: "+" means the forward wave (reflected), "–" the backward one (incident, transmitted). For example $\beta_m^{(j-)} < 0$, if $\beta_m^{(j-)} \in \mathbb{R}$, or, $\mathrm{Im}\,(\beta_m^{(j-)}) < 0$ otherwise with respect to radiation conditions and chosen convention $\mathrm{e}^{-\mathrm{i}\omega t}$ – see (3). In what follows stay 1 for 1+ and 2 for 2−.

Denoting $\boldsymbol{x} = (x_1, x_3)$, $\boldsymbol{y} = (y_1, y_3)$, the periodical fundamental solution of the Helmholtz equation in $\Omega^{(j)}$ can be written as [4]

$$\Psi^{(j)}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\mathrm{i}\Lambda} \sum_{m=-\infty}^{\infty} \Psi_m^{(j)}(\boldsymbol{x}, \boldsymbol{y})\,, \qquad \Psi_m^{(j)}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\beta_m^{(j)}}\, \mathrm{e}^{\mathrm{i}(\alpha_m(x_1-y_1)+\beta_m^{(j)}|x_3-y_3|)}\,. \tag{5}$$

In further considerations we exploit following well-known property of the functions $\Psi^{(j)}$.

Theorem 1. *For both of the function $\Psi^{(j)}(\boldsymbol{x}, \boldsymbol{y})$ defined by (5) the difference (6) is continuous in $\mathbb{R}^2$.*

$$\Psi^{(j)}(\boldsymbol{x}, \boldsymbol{y}) - \frac{1}{2\pi} \ln \frac{1}{\|\boldsymbol{x} - \boldsymbol{y}\|} \tag{6}$$

3. Boundary integral equations

The aim of this section is to formulate boundary integral equations for tangential fields

$$\boldsymbol{J} = \boldsymbol{\nu} \times \boldsymbol{E}^{(1)} = \boldsymbol{\nu} \times \boldsymbol{E}^{(2)}, \qquad \boldsymbol{I} = -\boldsymbol{\nu} \times \boldsymbol{H}^{(1)} = -\boldsymbol{\nu} \times \boldsymbol{H}^{(2)}\,, \tag{7}$$

where $\boldsymbol{\nu} = (f', 0, -1)/\sigma$ with $\sigma = \sqrt{1 + f'^2}$ is an unit normal vector of the reduced boundary $S : x_3 = f(x_1)$ oriented as shown in Fig.1. Similarly, $\boldsymbol{\tau} = (1, 0, f')/\sigma$ represents an unit tangential vector of S.

Thus, on the boundary we can write $\boldsymbol{J} = -J_2 \boldsymbol{e}_2$, where $J_2 = \boldsymbol{\tau} \cdot \boldsymbol{E}^{(1)} = \boldsymbol{\tau} \cdot \boldsymbol{E}^{(2)}$, and, $\boldsymbol{I} = \sigma I_1 \boldsymbol{\tau} = I_\tau \boldsymbol{\tau}$, where $I_\tau = \sigma I_1 = -H_2^{(1)} = -H_2^{(2)}$.

For boundary points $\boldsymbol{\xi} = (\xi_1, \xi_3)$, $\boldsymbol{\eta} = (\eta_1, \eta_3)$ on the interface $S : \eta_3 = f(\eta_1)$, $\eta_1 \in \langle 0, \Lambda \rangle$ we obtain following system of boundary integral equations [5]

$$J_2(\boldsymbol{\xi}) = -J_0(\boldsymbol{\xi}) - \mathrm{i}k_0 \boldsymbol{\tau}_\xi \cdot \int_S I_\tau \boldsymbol{\tau}_\eta (\Psi^{(1)} - \Psi^{(2)})\, dl_\eta$$

$$- \frac{1}{\mathrm{i}k_0} \boldsymbol{\tau}_\xi \cdot \int_S \frac{1}{\sigma} \frac{dI_\tau}{d\eta_1} \nabla_\eta \left(\frac{1}{\varepsilon^{(1)}} \Psi^{(1)} - \frac{1}{\varepsilon^{(2)}} \Psi^{(2)} \right) dl_\eta + \boldsymbol{\nu}_\xi \cdot \int_S J_2 \nabla_\eta (\Psi^{(1)} - \Psi^{(2)})\, dl_\eta\,, \tag{8}$$

$$I_\tau(\boldsymbol{\xi}) = -I_0(\boldsymbol{\xi}) - \mathrm{i}k_0 \int_S J_2 (\varepsilon^{(1)} \Psi^{(1)} - \varepsilon^{(2)} \Psi^{(2)})\, dl_\eta + \int_S I_\tau\, \boldsymbol{\nu}_\eta \cdot \nabla_\eta \left(\Psi^{(1)} - \Psi^{(2)} \right) dl_\eta\,, \tag{9}$$

where

$$J_0(\boldsymbol{\xi}) = -\boldsymbol{e}_2\cdot(\boldsymbol{\nu}_\xi\times\boldsymbol{E}_0^{(1-)}) = \boldsymbol{\tau}_\xi\cdot\boldsymbol{E}_0^{(1-)} , \qquad I_0(\boldsymbol{\xi}) = \boldsymbol{\tau}_\xi\cdot(\boldsymbol{\nu}_\xi\times\boldsymbol{H}_0^{(1-)}) = -H_{0,2}^{(1-)} , \tag{10}$$

thereby $\boldsymbol{E}_0^{(1-)}$, $\boldsymbol{H}_0^{(1-)}$ represent the incident wave in $\Omega^{(1)}$.

To derive these equations it is necessary to study properties of integral operators

$$\int\limits_S g(\boldsymbol{\eta})\,\psi(\boldsymbol{x},\boldsymbol{\eta})\,dl_\eta , \qquad \int\limits_S g(\boldsymbol{\eta})\,\frac{\partial\psi(\boldsymbol{x},\boldsymbol{\eta})}{\partial\boldsymbol{\nu}}\,dl_\eta , \qquad \int\limits_S g(\boldsymbol{\eta})\,\nabla_\eta\psi(\boldsymbol{x},\boldsymbol{\eta})\,dl_\eta \tag{11}$$

with the kernel

$$\psi(\boldsymbol{x},\boldsymbol{\eta}) = \frac{1}{2\pi}\ln\frac{1}{\|\,\boldsymbol{x}-\boldsymbol{\eta}\,\|} \tag{12}$$

when crossing from the inner point $\boldsymbol{x}$ to the boundary point $\boldsymbol{\xi}$ in the normal direction (the superscript (j) is omitted for simplicity).

Whereas the first and the second of them are the well-known single and double layer potentials, the third is worth to mention.

Theorem 2. *Let $\psi(\boldsymbol{x},\boldsymbol{\eta})$ is the function (12) and S is smooth boundary of the domain $\Omega\subset\mathbb{R}^2$ with unit outward normal $\boldsymbol{\nu}$. If $g\in \mathrm{C}(S)$, then*

$$\lim_{\boldsymbol{x}\to\boldsymbol{\xi}}\int\limits_S g(\boldsymbol{\eta})\,\nabla_\eta\psi(\boldsymbol{x},\boldsymbol{\eta})\,dl_\eta = \int\limits_S g(\boldsymbol{\eta})\,\nabla_\eta\psi(\boldsymbol{\xi},\boldsymbol{\eta})\,dl_\eta \pm \frac{1}{2}g(\boldsymbol{\xi})\boldsymbol{\nu}(\boldsymbol{\xi}) , \tag{13}$$

where $\boldsymbol{\xi}\in S$, minus holds for $\boldsymbol{x}\in\Omega$ and plus for $\boldsymbol{x}\in\mathbb{R}^2\setminus\bar{\Omega}$.

4. Operator form

Let $\boldsymbol{\pi}:\langle 0,2\pi\rangle\to\mathbb{R}^2$, $\boldsymbol{\pi}(t)=(p(t),q(t))$ be a parametrization of the boundary S. For the boundary points we have $\boldsymbol{\xi}=\boldsymbol{\pi}(s)$, $\boldsymbol{\eta}=\boldsymbol{\pi}(t)$, $s,t\in\langle 0,2\pi\rangle$ with corresponding unit normal vector $\boldsymbol{\nu}(t)=(\nu_1(t),\,\nu_3(t))=(q'(t),\,-p'(t))/\nu(t)$ and unit tangential vector $\boldsymbol{\tau}(t)=(p'(t),\,q'(t))/\nu(t)$, where $\nu(t)=\sqrt{p'(t)^2+q'(t)^2}$.

In the integral operators kernels the fundamental solution (5) of the Helmoltz equation takes place, hence the system (8), (9) can be written in operator form

$$\begin{bmatrix} \mathcal{V}_1+\mathcal{V}_2 & \mathcal{I}-\mathcal{V}_3 \\ \mathcal{I}-\mathcal{V}_4 & \mathcal{V}_5 \end{bmatrix}\begin{bmatrix} I_\tau \\ J_2 \end{bmatrix} = \begin{bmatrix} -J_{2,0} \\ -I_{\tau,0} \end{bmatrix} , \tag{14}$$

where $\mathcal{I}$ is the identity operator,

$$\mathcal{V}_1(I_\tau) = \frac{k_0}{2\Lambda\nu(s)}\int\limits_0^{2\pi} I_\tau(t)g_1(s,t)\sum_{m\in\mathbb{Z}}\left[\Psi_m^{(1)}(s,t)-\Psi_m^{(2)}(s,t)\right]dt , \tag{15}$$

$$\mathcal{V}_2(I_\tau) = \frac{\mathrm{i}}{2k_0\Lambda\nu(s)}\int\limits_0^{2\pi} I_\tau'(t)\sum_{m\in\mathbb{Z}}\left[\frac{g_{2,m}^{(1)}(s,t)}{\varepsilon^{(1)}}\Psi_m^{(1)}(s,t)-\frac{g_{2,m}^{(2)}(s,t)}{\varepsilon^{(2)}}\Psi_m^{(2)}(s,t)\right]dt , \tag{16}$$

$$\mathcal{V}_3(J_2) = \frac{1}{2\Lambda\nu(s)} \int_0^{2\pi} J_2(t) \sum_{m\in\mathbb{Z}} \left[g^{(1)}_{3,m}(s,t)\Psi^{(1)}_m(s,t) - g^{(2)}_{3,m}(s,t)\Psi^{(2)}_m(s,t) \right] \nu(t)\, dt \ , \tag{17}$$

$$\mathcal{V}_4(I_\tau) = \frac{1}{2\Lambda} \int_0^{2\pi} I_\tau(t) \sum_{m\in\mathbb{Z}} \left[g^{(1)}_{4,m}(s,t)\Psi^{(1)}_m(s,t) - g^{(2)}_{4,m}(s,t)\Psi^{(2)}_m(s,t) \right] dt \ , \tag{18}$$

$$\mathcal{V}_5(J_2) = \frac{k_0}{2\Lambda} \int_0^{2\pi} J_2(t) \sum_{m\in\mathbb{Z}} \left[\varepsilon^{(1)}\Psi^{(1)}_m(s,t) - \varepsilon^{(2)}\Psi^{(2)}_m(s,t) \right] \nu(t)\, dt \tag{19}$$

with

$$g_1(s,t) = \nu(s)\nu(t)\boldsymbol{\tau}(s)\cdot\boldsymbol{\tau}(t) \ , \qquad g^{(j)}_{2,m}(s,t) = \nu(s)\boldsymbol{\tau}(s)\cdot\boldsymbol{\kappa}^{(j)}_m(s,t) \ ,$$

$$g^{(j)}_{3,m}(s,t) = \nu(s)\boldsymbol{\nu}(s)\cdot\boldsymbol{\kappa}^{(j)}_m(s,t) \ , \qquad g^{(j)}_{4,m}(s,t) = \nu(t)\boldsymbol{\nu}(t)\cdot\boldsymbol{\kappa}^{(j)}_m(s,t) \ ,$$

$$\boldsymbol{\kappa}^{(j)}_m(s,t) = \Big(\alpha_m,\ \mathrm{sgn}(q(s)-q(t))\beta^{(j)}_m\Big) \tag{20}$$

The right-hand terms of (14) are obtained by parametrization of incident fields (10).

5. Properties of boundary integral operators

Now we need to discuss properties of integral operators kernels, which are written as differences $c_1\Psi^{(1)}(s,t) - c_2\Psi^{(2)}(s,t)$, or their gradients, where c_1, c_2 are generally complex constants. Because for $s \neq t$ this expression represents a continuous function, it suffices to analyse the singular case for $s = t$.

Theorem 3. *Let c_1, $c_2 \in \mathbb{C}$. Then for $s = t$ the functions*

$$c_1\Psi^{(1)}(s,t) - c_2\Psi^{(2)}(s,t) \ , \qquad \nabla_t\Big(c_1\Psi^{(1)}(s,t) - c_2\Psi^{(2)}(s,t)\Big) \tag{21}$$

are continuous for $c_1 = c_2$ and these have singularity of logarithmic type for $c_1 \neq c_2$.

The particular manner how to evaluate singular integrals depends on the choice of numerical method. The following theorems show one of possible methods - see [6], where also the proofs can be found ($\mathbb{Z}^* = \mathbb{Z} - \{0\}$).

Theorem 4. *Let $\boldsymbol{\pi} : \langle 0, 2\pi\rangle \to \mathbb{R}^2$ is a parametrization that satisfies*

$$p(0) = 0, \quad p(2\pi) = \Lambda, \quad q(0) = q(2\pi), \quad p(t+2\pi) = p(t) + \Lambda, \quad q(t+2\pi) = q(t).$$

Then

$$\ln\|\boldsymbol{\pi}(s) - \boldsymbol{\pi}(t)\| = \ln|2\sin\frac{s-t}{2}| = -\sum_{m\in\mathbb{Z}^*} \frac{\mathrm{e}^{-\mathrm{i}m(s-t)}}{2|m|} \ . \tag{22}$$

Theorem 5. *The series (23) is absolutely convergent.*

$$\sum_{m\in\mathbb{Z}^*} \left\{ \Psi^{(j)}_m(s,t) - \frac{1}{2\pi}\frac{\mathrm{e}^{-\mathrm{i}m(s-t)}}{2|m|} \right\} \tag{23}$$

These properties together with Theorem 1 allow us to split the fundamental solution as

$$\Psi^{(j)}(s,t) = \Psi_r^{(j)}(s,t) + \psi(s,t), \tag{24}$$

where

$$\Psi_r^{(j)}(s,t) = \Psi_0^{(j)}(s,t) + \sum_{m\in\mathbb{Z}^*} \left\{ \Psi_m^{(j)}(s,t) - \frac{1}{2\pi}\frac{\mathrm{e}^{-im(s-t)}}{2|m|} \right\} , \tag{25}$$

$$\psi(s,t) = \frac{1}{2\pi} \ln |2\sin\frac{s-t}{2}| \, . \tag{26}$$

In numerical implementations we work separately with regular integral kernels and with singular integrals which can be evaluated analytically.

6. Conclusion

The presented formulation of diffraction problem represents appropriate background of numerical solution by the Boundary Elements Method (BEM). Specific problem to discuss is the choice of basis functions; trigonometric polynomials can be used [3], for instance. For further work we prefer piecewise linear boundary elements.

Acknowledgements

This work was partially supported by Centrum Excelence IT4Innovations at the VŠB - Technical University of Ostrava.

References

[1] Bao, G., Cowsar, L. and Masters, W.: *Mathematical modeling in optical science.* SIAM, Philadelphia, 2001.

[2] Nedelec, J. C., Starling, F.: Integral equation methods in a quasi-periodic diffraction problem for the time-harmonic Maxwell's equations. SIAM, J. Math. Anal. **22** (1991), 1679–1701.

[3] Kleemann, B. H., Mitreiter, A. and Wyrowski, F.: Integral equation method with parametrization of grating profile. Theory and Experiments. J. Mod. Opt. **43** (7) (1996), 1323–1349.

[4] Linton, C. M.: The Green's function for the two-dimensional Helmholtz equation in periodic domains. J. Eng. Math. **33** (1998), 377–402.

[5] Dobson, D. C. and Cox, J. A.: An integral equation method for biperiodic diffraction structures. In: *Proc. of SPIE*, Vol. 1545, pp. 106-113, 1991.

[6] Žídek, A., Vlček, J., and Krček, J.: Solution of diffraction problems by boundary integral equations. In: *Proc. of 11th International Conference APLIMAT 2012, Febr. 7-9, 2012, Bratislava, Slovak Republic*, publ. by Faculty of Mechanical Engineering, Slovak University of Technology, Bratislava 2012, 221–229.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

COUPLED HEAT TRANSPORT AND DARCIAN WATER FLOW IN FREEZING SOILS

Lukáš Krupička[1], Radek Štefan[2], Michal Beneš[1]

[1] Department of Mathematics
[2]Department of Concrete and Masonry Structures
Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
lukas.krupicka@fsv.cvut.cz, radek.stefan@fsv.cvut.cz, benes@mat.fsv.cvut.cz

Abstract

The model of coupled heat transport and Darcian water flow in unsaturated soils and in conditions of freezing and thawing is analyzed. In this contribution, we present results concerning the existence of the numerical solution. Numerical scheme is based on semi-implicit discretization in time. This work illustrates its performance for a problem of freezing processes in vertical soil columns.

1. Introduction

Let $T > 0$ and $\ell > 0$ be the fixed values, $\Omega = (0, \ell)$, $I = (0, T)$, $\Omega_T = \Omega \times I$. We consider a mixed initial-boundary value problem for a general model of the coupled heat and mass flow in freezing soils. The mathematical model consists of the following system (cf. [1]):

$$\frac{\partial \theta_M(z, \vartheta, u)}{\partial t} = \frac{\partial}{\partial z}\left(k(z, \vartheta, u)\frac{\partial u}{\partial z}\right) \quad \text{in } \Omega_T, \tag{1}$$

$$C_a(z, \vartheta, u)\frac{\partial \vartheta}{\partial t} = \frac{\partial}{\partial z}\left(\lambda(z, \vartheta, u)\frac{\partial \vartheta}{\partial z}\right) + C_w k(z, \vartheta, u)\frac{\partial \vartheta}{\partial z}\frac{\partial u}{\partial z} \quad \text{in } \Omega_T, \tag{2}$$

$$u(0, t) = u_D(t) \quad \text{and} \quad \vartheta(0, t) = \vartheta_D(t) \quad \text{in } I, \tag{3}$$

$$-k(z, \vartheta, u)\frac{\partial u}{\partial z} = \beta_c(u - u_\infty) \text{ and } -\lambda(z, \vartheta, u)\frac{\partial \vartheta}{\partial z} = \alpha_c(\vartheta - \vartheta_\infty) \quad \text{in } I,\ z = \ell, \tag{4}$$

$$u(z, 0) = u_0(z) \quad \text{and} \quad \vartheta(z, 0) = \vartheta_0(z) \quad \text{in } \Omega. \tag{5}$$

This system describes the one-dimensional coupled water flow and heat transport involving freezing-thawing processes in a vertical soil column. Equations (1) and (2) represent conservation laws for mass and energy, (3) and (4) are prescribed boundary conditions of Dirichlet and Neumann type, respectively, and (5) represents appropriate initial conditions. In (1)–(5) $u = u(z, t)$ [m] and $\vartheta = \vartheta(z, t)$ [K] (single-valued functions of the time t and the spatial position $z \in \Omega$ positive upward) are the primary unknowns for the total pressure head and temperature, θ_M [-] is the total water content, k [m s^{-1}] represents the hydraulic conductivity, C_a [J m^{-3} K^{-1}] is the so called apparent heat capacity and λ [W m^{-1} K^{-1}] is the thermal conductivity of the soil. Material constant parameters in (1)–(5) are the volumetric heat capacity of water C_w (4.181×10^6 J m^{-3} K^{-1}), convective heat and mass transfer coefficients α_c [W m^{-2} K^{-1}] and β_c [s^{-1}].

2. Freezing and thawing

Define ψ [m] as the matric potential corresponding to the liquid water content θ_w [-] and the matric potential h [m] corresponding to the total water content θ_V [-] (liquid and ice). The amount of water present at a certain matric potential of the porous medium is characterized by the water retention curve $\theta(\cdot)$. In particular, $\theta_V = \theta(h)$, while $\theta_w = \theta(\psi)$. Here we use the relation proposed by van Genuchten [4] $\theta(h) = \theta_r + (\theta_s - \theta_r)[1 + |\alpha h|^n]^{-m}$, where θ_s is the soil saturated water content [-], θ_s is the soil residual water content [-], α [m^{-1}], m and n are parameters.

Water in soil pores does not freeze at 273.15 K, but is subject to a freezing-point depression caused by interaction between water, soil particles and solutes. The generalized Clapeyron equation is used to describe the condition for the co-existence of water and ice. The local freezing point of pore fluid can be obtained from the generalized Clapeyron equation [1, 2]

$$\mathrm{d}p = \frac{\rho_w L_f}{\vartheta}\,\mathrm{d}\vartheta, \tag{6}$$

where p [Pa] is the water pressure, $p = \rho_w g h$, $h = u - z$, g is the acceleration due to gravity (9.81 m s^{-2}), h [m] the pressure head (matric potential), ρ_w the density of liquid water (approximately 1000.0 kg m^{-3}) and L_f is the latent heat of fusion (3.34×10^5 J kg^{-1}). Let $\vartheta_0 = 273.15$ be the freezing temperature at zero pressure head. If the soil is unsaturated, the surface tension at the water/air interface decreases the water freezing temperature to $\vartheta_f < 273.15$ K. To obtain the value ϑ_f at the given pressure P integrate (6) in temperature from 273.15 to ϑ_f and from 0 to P in pressure to obtain

$$\int_0^P \mathrm{d}p = \int_{273.15}^{\vartheta_f} \frac{\rho_w L_f}{\vartheta}\mathrm{d}\vartheta, \text{ which yields } \vartheta_f = 273.15e^{hg/L_f} = 273.15e^{(u-z)g/L_f}. \tag{7}$$

Similarly, integrating (6) in temperature from ϑ_f to ϑ and from $P(= h\rho_w g)$ to $P_\psi(= \psi\rho_w g)$ in pressure yields (recall $u = h + z$)

$$\psi(z, \vartheta, u) = \psi(\vartheta, u - z) \equiv \psi(\vartheta, h) = h + \frac{L_f}{g}\ln\frac{\vartheta}{\vartheta_f} = u - z + \frac{L_f}{g}\ln\frac{\vartheta}{\vartheta_f}. \tag{8}$$

The above equation is valid for $\vartheta < \vartheta_f$. If $\vartheta \geqslant \vartheta_f$ all water is unfrozen and $h = \psi$ and $\theta_w = \theta_V$. Consequently, whenever $\vartheta < \vartheta_f$, the ice fraction θ_i [-] can be expressed as $\theta_i = \theta_V - \theta_w$ [-]. In addition, the total water content θ_M (present in (1)) as derived by the fraction of total mass of liquid water and ice (see [1, Appendix A]) reads

$$\theta_M(z,\vartheta,u) = \theta_w(\psi(z,\vartheta,u)) + \frac{\rho_i}{\rho_w}\theta_i(z,\vartheta,u) = \frac{\rho_i}{\rho_w}\theta_V(z,u) + \left(1 - \frac{\rho_i}{\rho_w}\right)\theta_w(\psi(z,\vartheta,u)),$$

where ρ_i is the density of ice (918 kg m^{-3}).

2.1. Structural conditions and assumptions on physical parameters

Let us present some properties and additional assumptions on physical parameters introduced in the model.

$\mathbb{A}_1$ The parameters ρ_w, ρ_i, θ_s, θ_r, C_w, L_f, α_c and β_c are real positive constants and $\rho_i < \rho_w$.

$\mathbb{A}_2$ The thermal conductivity λ, apparent thermal capacity C_a and hydraulic conductivity k are assumed to be positive continuous functions of their arguments (see [2] for specific examples). In addition,

$$0 < C_a(z,\xi,\zeta) \leqslant C_a^\sharp < +\infty \quad \forall \xi,\zeta \in \mathbb{R} \quad (C_a^\sharp = \text{const} > 0). \tag{9}$$

$\mathbb{A}_3$ Functions $\theta_w = \theta_w(z,\cdot)$ and $\theta_V = \theta_V(z,\cdot)$ (for $z \in \Omega$) are positive, nondecreasing, continuous and bounded functions such that

$$\theta_r \leqslant \theta_w(z,\xi) \leqslant \theta_s, \quad \theta_w(z,\xi) \leqslant \theta_V(z,\xi) \leqslant \theta_s \quad \forall \xi \in \mathbb{R}. \tag{10}$$

Consequently, θ_M is a positive continuous function such that

$$0 < \theta_M(z,\xi,\zeta) = \frac{\rho_i}{\rho_w}\theta_V(z,\xi) + \left(1 - \frac{\rho_i}{\rho_w}\right)\theta_w(z,\zeta) \leqslant \theta_s \quad \text{for all } \xi,\zeta \in \mathbb{R}.$$

$\mathbb{A}_4$ Functions u_D, ϑ_D, u_∞, ϑ_∞ are continuous on $[0,T]$, $u_0, \vartheta_0 \in W^{1,2}(\Omega)^2$ and the compatibility conditions $u_0(0) = u_D(0)$ and $\vartheta_0(0) = \vartheta_D(0)$ hold.

3. The approximate solution

Albeit the coupled problem (1)–(5) is essentially non-stationary in their nature, we shall formulate and analyze a weak form of the stationary problem. It has a significant mathematical interest because the time discretization of the evolution problem leads, in each time step, to a coupled system of stationary equations.

Let $0 = t_0 < t_1 < \cdots < t_N = T$ be an equidistant partitioning of time interval $[0;T]$ with step Δt. Set a fixed integer n such that $0 \leqslant n \leqslant N-1$. In what follows we abbreviate $\phi(z,t_n)$ by ϕ_n ($\equiv \phi(z)_n$) for any function ϕ. The time discretization of the continuous model is accomplished through a semi-implicit difference scheme.

Consequently, we have to solve, successively for $n = 0, \dots, N-1$, the following semilinear system with primary unknowns $[\vartheta_{n+1}, u_{n+1}]$

$$
\begin{aligned}
\frac{\theta_M(z,\vartheta_{n+1},u_{n+1}) - \theta_M(z,\vartheta_n,u_n)}{\Delta t} &= \frac{\partial}{\partial z}\left(k_n \frac{\partial u_{n+1}}{\partial z}\right), && (11)\\
(C_a)_{n+1}\frac{\vartheta_{n+1}-\vartheta_n}{\Delta t} &= \frac{\partial}{\partial z}\left(\lambda_n \frac{\partial \vartheta_{n+1}}{\partial z}\right) + C_w k_n \frac{\partial u_n}{\partial z}\frac{\partial \vartheta_n}{\partial z}, && (12)\\
u(0)_{n+1} &= (u_D)_{n+1} \quad \text{and} \quad \vartheta(0)_{n+1} = (\vartheta_D)_{n+1}, && (13)\\
-k_n \frac{\partial u_{n+1}}{\partial z}\bigg|_{z=\ell} &= \beta_c(u(\ell)_{n+1} - u_\infty(\ell)_{n+1}), && (14)\\
-\lambda_n \frac{\partial \vartheta_{n+1}}{\partial z}\bigg|_{z=\ell} &= \alpha_c(\vartheta(\ell)_{n+1} - \vartheta_\infty(\ell)_{n+1}). && (15)
\end{aligned}
$$

Here, we assume that the functions u_n and ϑ_n are known and (for the sake of simplicity) we put $k_n = k(z,\vartheta_n,u_n)$, $\lambda_n = \lambda(z,\vartheta_n,u_n)$, $(C_a)_{n+1} = C_a(z,\vartheta_{n+1},u_{n+1})$. In what follows we study the problem of the existence of the solution u_{n+1} and ϑ_{n+1}.

Let $\mathbb{V}$ be a closure of the space $\left\{\boldsymbol{v} \in C^\infty(\overline{\Omega})^2;\ \boldsymbol{v}(0) = \boldsymbol{0}\right\}$ in the norm of $W^{1,2}(\Omega)^2$. By $\langle\cdot,\cdot\rangle$ we denote the duality between $\mathbb{V}$ and $\mathbb{V}^*$, where $\mathbb{V}^*$ represents the dual space corresponding to $\mathbb{V}$. Define an operator $\mathcal{A} : W^{1,2}(\Omega)^2 \to \mathbb{V}^*$ given by the equation

$$
\begin{aligned}
\langle \mathcal{A}([\vartheta_{n+1},u_{n+1}]),\boldsymbol{\varphi}\rangle := \int_\Omega k_n \frac{\partial u_{n+1}}{\partial z}\frac{\partial \varphi_1}{\partial z} + \lambda_n \frac{\partial \vartheta_{n+1}}{\partial z}\frac{\partial \varphi_2}{\partial z}\,\mathrm{d}z \\
+ \frac{1}{\Delta t}\int_\Omega \theta_M(z,\vartheta_{n+1},u_{n+1})\varphi_1 + (C_a)_{n+1}(\vartheta_{n+1}-\vartheta_n)\varphi_2\,\mathrm{d}z \\
+ \beta_c u(\ell)_{n+1}\,\varphi_1(\ell) + \alpha_c\vartheta(\ell)_{n+1}\varphi_2(\ell) \quad (16)
\end{aligned}
$$

for every $\boldsymbol{\varphi} = [\varphi_1,\varphi_2] \in \mathbb{V}$ and the functional $\boldsymbol{f} \in \mathbb{V}^*$ by the equation

$$
\begin{aligned}
\langle \boldsymbol{f},\boldsymbol{\varphi}\rangle := \frac{1}{\Delta t}\int_\Omega \theta_M(z,\vartheta_n,u_n)\varphi_1 \mathrm{d}z + \int_\Omega C_w k_n \frac{\partial u_n}{\partial z}\frac{\partial \vartheta_n}{\partial z}\varphi_2\,\mathrm{d}z \\
+ \beta_c u_\infty(\ell)_{n+1}\varphi_1(\ell) + \alpha_c\vartheta_\infty(\ell)_{n+1}\varphi_2(\ell) \quad (17)
\end{aligned}
$$

for all $\boldsymbol{\varphi} = [\varphi_1,\varphi_2] \in \mathbb{V}$. It can be shown that the operator $\mathcal{A}$ and the functional $\boldsymbol{f}$ are well defined. Let $[\vartheta_n,u_n] \in [(\vartheta_D)_n,(u_D)_n] + \mathbb{V}$. We say that the couple $[\vartheta_{n+1},u_{n+1}] \in [(\vartheta_D)_{n+1},(u_D)_{n+1}] + \mathbb{V}$ is the weak solution of the problem (11)–(15) whenever $\langle \mathcal{A}([\vartheta_{n+1},u_{n+1}]),\boldsymbol{\varphi}\rangle = \langle \boldsymbol{f},\boldsymbol{\varphi}\rangle$ for all $\boldsymbol{\varphi} = [\varphi_1,\varphi_2] \in \mathbb{V}$.

Theorem 1. *For a given couple $[\vartheta_n,u_n] \in [(\vartheta_D)_n,(u_D)_n] + \mathbb{V}$ there exists a weak solution $[\vartheta_{n+1},u_{n+1}] \in [(\vartheta_D)_{n+1},(u_D)_{n+1}] + \mathbb{V}$ of the problem* (11)–(15).

Sketch of the proof. Note that the couple $[\vartheta_{n+1},u_{n+1}] \in [(\vartheta_D)_{n+1},(u_D)_{n+1}] + \mathbb{V}$ is the weak solution of the problem (11)–(15) iff it is a solution of the operator equation

$$\mathcal{A}([\vartheta_{n+1},u_{n+1}]) = \boldsymbol{f}.$$

Let us define $\overline{\mathcal{A}} : \mathbb{V} \to \mathbb{V}^*$ by $\overline{\mathcal{A}}([\bar{\vartheta}_{n+1}, \bar{u}_{n+1}]) := \mathcal{A}([\bar{\vartheta}_{n+1}, \bar{u}_{n+1}] + [(\vartheta_D)_{n+1}, (u_D)_{n+1}])$. The abstract equation $\overline{\mathcal{A}}([\bar{\vartheta}_{n+1}, \bar{u}_{n+1}]) = \boldsymbol{f}$ has a solution $[\bar{\vartheta}_{n+1}, \bar{u}_{n+1}] \in \mathbb{V}$ if and only if $[\vartheta_{n+1}, u_{n+1}] = [\bar{\vartheta}_{n+1}, \bar{u}_{n+1}] + [(\vartheta_D)_{n+1}, (u_D)_{n+1}] \in W^{1,2}(\Omega)^2$ is the solution of the equation $\mathcal{A}([\vartheta_{n+1}, u_{n+1}]) = \boldsymbol{f}$. Note that the equation $\overline{\mathcal{A}}([\bar{\vartheta}_{n+1}, \bar{u}_{n+1}]) = \boldsymbol{f}$ represents a variational formulation corresponding to the system of coupled semilinear equations. It can be shown that the operator $\overline{\mathcal{A}} : \mathbb{V} \to \mathbb{V}^*$ is pseudomonotone and coercive. Now [3, Theorem 3.3.42] yields the existence of the solution $[\bar{\vartheta}_{n+1}, \bar{u}_{n+1}] \in \mathbb{V}$ to the equation $\overline{\mathcal{A}}([\bar{\vartheta}_{n+1}, \bar{u}_{n+1}]) = \boldsymbol{f}$. □

4. Numerical solution and results

By means of the model described above, we briefly present numerical results for coupled water flow and heat transport involving freezing-thawing cycle in a vertical soil column. The soil thickness in the numerical simulation for the one-dimensional vertical transport is 1 m, see Fig. 1. The spatial discretization of the system (11)–(15) is carried out by means of the FE-method with piecewise linear elements with spatial discretization as indicated in Fig. 1. This resulting system is solved using the well-known Newton method at each time step with $\Delta t = 30$ s. Physical properties of soil are taken from [1, 2, 4]. The initial and boundary conditions are set as follows: $\vartheta_0 = \vartheta_D = 277.15$ K, $u_0 = -0.1241 + z$ m, u_∞ decreases from the value $u_0 + 1$ m to -100 m during the first two days and then taken constant, the distribution of ϑ_∞ is shown in Fig. 2. The progress of freezing and thawing in a soil column based on numerical simulation is clearly visible in Figures 3 and 4 which show the vertical distributions of the temperature, water content and ice during the 8-days period.

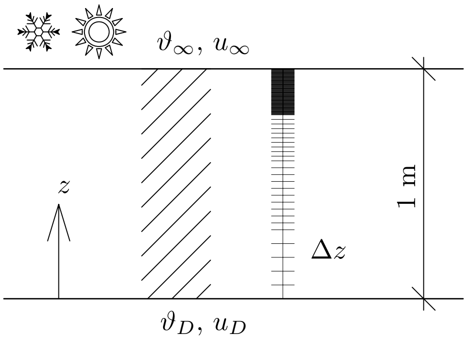


Figure 1: Analyzed soil profile.

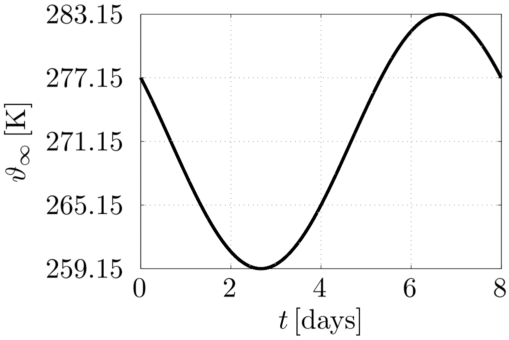


Figure 2: Temperature ϑ_∞.

Acknowledgements

This work was supported by the grant SGS12/001/OHK1/1T/11 provided by the Grant Agency of the Czech Technical University in Prague.

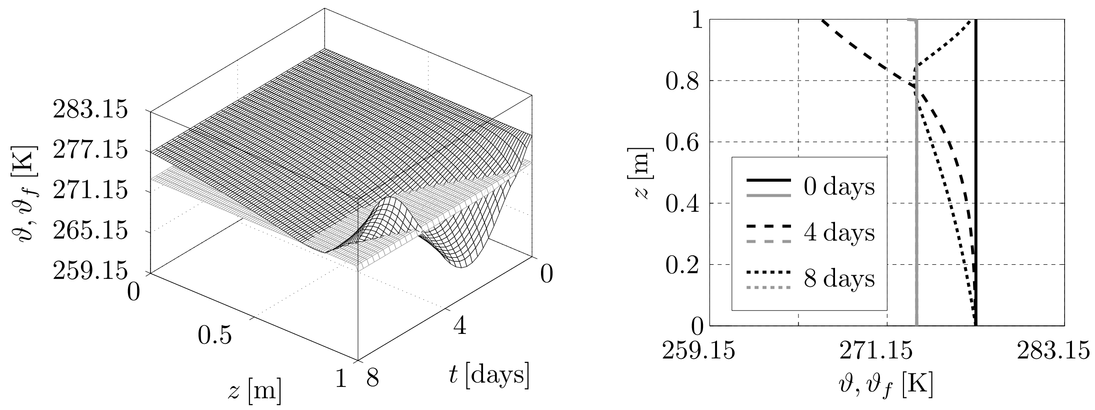


Figure 3: Spatial and time distribution of temperature (black lines) and freezing temperature (gray lines), ϑ and ϑ_f, respectively, for the analyzed soil profile.

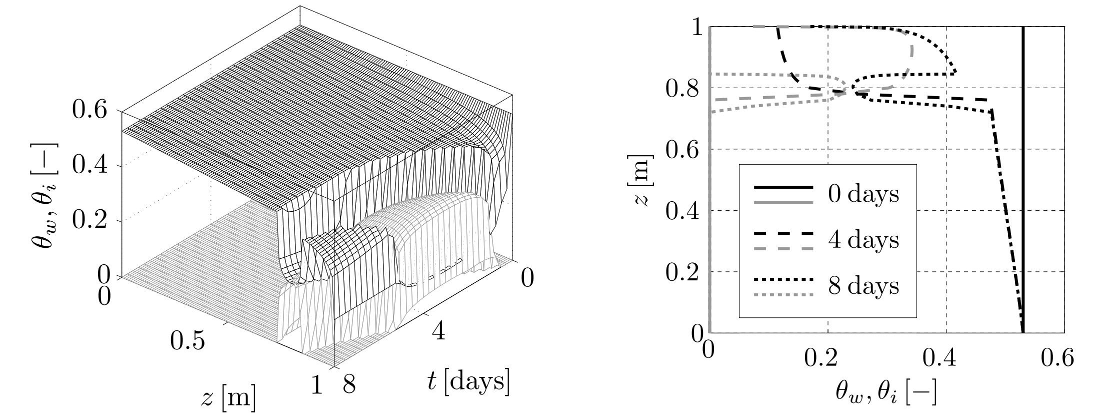


Figure 4: Spatial and time distribution of water (black lines) and ice (gray lines) content, θ_w and θ_i, respectively, for the analyzed soil profile.

References

[1] Dall'Amico, M., Endrizzi, S., Gruber, S., and Rigon, R.: A robust and energy-conserving model of freezing variably-saturated soil. The Cryosphere **2** (2011), 469–484.

[2] Hansson, K., Šimunek, J., and Mizoguchi, M.: Water flow and heat transport in frozen soil: numerical solution and freeze-thaw applications. Vadose Zone Journal **3** (2004), 693–704.

[3] Nečas, J.: *Introduction to the theory of nonlinear elliptic equations.* Teubner-Texte zur Mathematik, Leipzig, 1983.

[4] van Genuchten, M. Th.: A closed form equation for predicting the hydraulic conductivity of unsaturated soil. Soil Sci. Soc. Am. J. **44** (1980) 892–898.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

ERROR ESTIMATES FOR NONLINEAR CONVECTIVE PROBLEMS IN THE FINITE ELEMENT METHOD

Václav Kučera

Faculty of Mathematics and Physics, Charles University in Prague
Sokolovská 83, 186 75 Praha 8, Czech Republic
vaclav.kucera@email.cz

Abstract

We describe the basic ideas needed to obtain apriori error estimates for a nonlinear convection diffusion equation discretized by higher order conforming finite elements. For simplicity of presentation, we derive the key estimates under simplified assumptions, e.g. Dirichlet-only boundary conditions. The resulting error estimate is obtained using continuous mathematical induction for the space semi-discrete scheme.

1. Continuous problem

Let $\Omega \subset \mathbb{R}^d, d \in \mathbb{N}$, be a bounded open polyhedral domain. We treat the following nonlinear convective problem. Find $u : \Omega \times (0,T) \to \mathbb{R}$ such that

$$\text{a)} \quad \frac{\partial u}{\partial t} + \operatorname{div} \mathbf{f}(u) = g \quad \text{in } \Omega \times (0,T), \tag{1}$$

$$\text{b)} \quad u\big|_{\partial\Omega \times (0,T)} = 0, \tag{2}$$

$$\text{d)} \quad u(x,0) = u^0(x), \quad x \in \Omega. \tag{3}$$

Here $g : \Omega \times (0,T) \to \mathbb{R}$ and $u^0 : \Omega \to \mathbb{R}$ are given functions. We assume that the *convective fluxes* $\mathbf{f} = (f_1, \cdots, f_d) \in (C_b^2(\mathbb{R}))^d = (C^2(\mathbb{R}) \cap W^{2,\infty}(\mathbb{R}))^d$, hence $\mathbf{f}$ and $\mathbf{f}' = (f_1', \cdots, f_d')$ are *globally Lipschitz continuous.*

By $(\cdot\,,\cdot)$ we denote the standard $L^2(\Omega)-$scalar product and by $\|\cdot\|$ the $L^2(\Omega)$-norm. By $\|\cdot\|_\infty$, we denote the $L^\infty(\Omega)$-norm. For simplicity of notation, we shall drop the argument Ω in Sobolev norms, e.g. $\|\cdot\|_{H^{p+1}}$ denotes the $H^{p+1}(\Omega)$-norm. We shall also denote the Bochner norms over the whole interval $[0,T]$ in concise form, e.g. $\|u\|_{L^\infty(H^{p+1})}$ denotes the $L^\infty(0,T;H^{p+1}(\Omega))$-norm.

2. Discretization

Let $\mathcal{T}_h$ be a triangulation of $\overline{\Omega}$, i.e. a partition into a finite number of closed simplexes with mutually disjoint interiors. We assume standard conforming properties: two neighboring elements from $\mathcal{T}_h$ share an entire face, edge or vertex. We set $h = \max_{K\in\mathcal{T}_h} \operatorname{diam}(K)$.

We consider a system $\{\mathcal{T}_h\}_{h\in(0,h_0)}$, $h_0 > 0$, of triangulations of the domain Ω which are shape regular and satisfy the inverse assumption, cf. [2]. Let $p \geq 1$ be an integer. The approximate solution will be sought in the space of globally continuous piecewise polynomial functions $S_h = \{v \in C(\overline{\Omega});\ v|_{\Gamma_D} = 0,\ v|_K \in P^p(K)\forall K \in \mathcal{T}_h\}$, where $P^p(K)$ denotes the space of polynomials on K of degree $\leq p$.

We discretize the continuous problem in a standard way. Multiply (1) by a test function $\varphi_h \in S_h$, integrate over Ω and apply Green's theorem.

Definition 1. *We say that $u_h \in C^1([0,T];S_h)$ is the space-semidiscretized finite element solution of problem (1)–(3), if $u_h(0) = u_h^0 \approx u^0$ and*

$$\frac{d}{dt}\big(u_h(t),\varphi_h\big) + b\big(u_h(t),\varphi_h\big) = l\big(\varphi_h\big)(t), \quad \forall\varphi_h \in S_h,\ t \in (0,T). \tag{4}$$

Here, we have introduced an approximation $u_h^0 \in S_h$ of the initial condition u^0 and the *convective* and *right-hand side forms* defined for $v,\varphi \in H^1(\Omega)$:

$$b(v,\varphi) = -\int_\Omega \mathbf{f}(v)\cdot\nabla\varphi\,\mathrm{d}x, \qquad l(\varphi)(t) = \int_\Omega g(t)\varphi\,\mathrm{d}x.$$

We note that a sufficiently regular exact solution u of problem (1) satisfies

$$\frac{d}{dt}\big(u(t),\varphi_h\big) + b\big(u(t),\varphi_h\big) = l\big(\varphi_h\big)(t), \quad \forall\varphi_h \in S_h,\ \forall t \in (0,T), \tag{5}$$

which implies the *Galerkin orthogonality property* of the error.

3. Key estimates of the convective terms

As usual in apriori error analysis, we assume that the weak solution u is sufficiently regular, namely

$$u, u_t \in L^2\big(0,T;H^{p+1}(\Omega)\big), \quad u \in L^\infty(0,T;W^{1,\infty}(\Omega)), \tag{6}$$

where $u_t := \frac{\partial u}{\partial t}$. For $v \in L^2(\Omega)$ we denote by $\Pi_h v$ the $L^2(\Omega)$-projection of v on S_h:

$$\Pi_h v \in S_h, \quad (\Pi_h v - v,\ \varphi_h) = 0, \qquad \forall\,\varphi_h \in S_h.$$

Let $\eta_h(t) = u(t) - \Pi_h u(t) \in H^{p+1}(\Omega)$ and $\xi_h(t) = \Pi_h u(t) - u_h(t) \in S_h$ for $t \in (0,T)$. Then we can write the error e_h as $e_h(t) := u(t) - u_h(t) = \eta_h(t) + \xi_h(t)$. By C we denote a generic constant independent of h, which may have different values in different parts of the text. Also, for simplicity of notation, we shall usually omit the argument (t) and subscript h in $\xi_h(t)$ and $\eta_h(t)$. In our analysis, we shall need the following standard inverse inequalities and approximation properties of η, (cf. [2]):

Lemma 1. *There exists a constant $C_I > 0$ independent of h s.t. for all $v_h \in S_h$*

$$|v_h|_{H^1} \leq C_I h^{-1}\|v_h\|,$$
$$\|v_h\|_\infty \leq C_I h^{-d/2}\|v_h\|.$$

Lemma 2. *There exists a constant $C > 0$ independent of h s.t. for all $h \in (0, h_0)$*

$$\begin{aligned}\|\eta_h(t)\| &\le Ch^{p+1}|u(t)|_{H^{p+1}},\\ \|\tfrac{\partial\eta_h(t)}{\partial t}\| &\le Ch^{p+1}\big|\tfrac{\partial u(t)}{\partial t}\big|_{H^{p+1}},\\ \|\eta_h(t)\|_\infty &\le Ch|u(t)|_{W^{1,\infty}}.\end{aligned}$$

Lemma 3. *There exists a constant $C \ge 0$ independent of h, t, such that*

$$b\big(u_h(t), \xi(t)\big) - b\big(u(t), \xi(t)\big) \le C\Big(1 + \frac{\|e_h(t)\|_\infty}{h}\Big)\big(h^{2p+2}|u(t)|^2_{H^{p+1}} + \|\xi(t)\|^2\big). \quad (7)$$

Proof. The proof follows the arguments of [5], where similar estimates are derived for periodic boundary conditions or compactly supported solutions in 1D. The proof for mixed Dirichlet-Neumann boundary conditions is contained in [4]. We write

$$b(u_h, \xi) - b(u, \xi) = \int_\Omega \big(\mathbf{f}(u) - \mathbf{f}(u_h)\big)\cdot \nabla\xi \,\mathrm{d}x. \quad (8)$$

By the Taylor expansion of $\mathbf{f}$ with respect to u, we have

$$\mathbf{f}(u) - \mathbf{f}(u_h) = \mathbf{f}'(u)\xi + \mathbf{f}'(u)\eta - \frac{1}{2}\mathbf{f}''_{u,u_h} e_h^2, \quad (9)$$

where $\mathbf{f}''_{u,u_h}$ is the Lagrange form of the remainder of the Taylor expansion, i.e. $\mathbf{f}''_{u,u_h}(x,t)$ has components $f''_s\big(\vartheta_s(x,t)u(x,t) + (1-\vartheta_s(x,t))u_h(x,t)\big)$ for some $\vartheta_s(x,t) \in [0,1]$ and $s = 1, \cdots, d$. Substituting (9) into (8), we obtain

$$b(u_h, \xi) - b(u, \xi) = \underbrace{\int_\Omega \mathbf{f}'(u)\xi\cdot \nabla\xi \,\mathrm{d}x}_{Y_1} + \underbrace{\int_\Omega \mathbf{f}'(u)\eta\cdot \nabla\xi \,\mathrm{d}x}_{Y_2} \underbrace{- \frac{1}{2}\int_\Omega \mathbf{f}''_{u,u_h} e_h^2\cdot \nabla\xi \,\mathrm{d}x}_{Y_3}. \quad (10)$$

We shall estimate these terms individually.
(A) Term $\mathbf{Y_1}$: Due to Green's theorem and the boundedness of $\mathbf{f}''$ and the regularity of u, we have

$$\int_\Omega \mathbf{f}'(u)\xi\cdot \nabla\xi \,\mathrm{d}x = -\frac{1}{2}\int_\Omega \mathrm{div}\big(\mathbf{f}'(u)\big)\xi^2 \,\mathrm{d}x \le C\|\xi\|^2.$$

(B) Term $\mathbf{Y_2}$: We define $\Pi_h^1 : (L^2(\Omega))^d \to (S_h^1)^d = \{\mathbf{v} \in (C(\overline{\Omega}))^d;\ \mathbf{v}|_{\Gamma_D} = 0,\ \mathbf{v}|_K \in (P^1(K))^d,\ \forall K \in \mathcal{T}_h\}$, the $(L^2(\Omega))^d$-projection onto the space of continuous piecewise linear vector functions. From standard approximation results (similar to those of Lemma 2, cf. [2]), we obtain

$$\|\mathbf{f}'(u) - \Pi_h^1(\mathbf{f}'(u))\|_\infty \le Ch|\mathbf{f}'(u)|_{W^{1,\infty}} \le Ch\|\mathbf{f}''\|_{L^\infty(\mathbb{R})}|u|_{L^\infty(W^{1,\infty})} = \tilde{C}h.$$

Furthermore, due to the definition of η, we have $\int_\Omega \Pi_h^1(\mathbf{f}'(u))\cdot\nabla\xi\,\eta\,\mathrm{d}x = 0$, since $\Pi_h^1(\mathbf{f}'(u))\cdot\nabla\xi \in S_h$. Therefore, by Lemmas 1, 2 and Young's inequality

$$|Y_2| = \Big|\int_\Omega \big(\mathbf{f}'(u) - \Pi_h^1(\mathbf{f}'(u))\big)\cdot\nabla\xi\,\eta\,\mathrm{d}x\Big| \leq \|\mathbf{f}'(u) - \Pi_h^1(\mathbf{f}'(u))\|_\infty C_I h^{-1}\|\xi\|\|\eta\|$$
$$\leq \tilde{C}hC_I h^{-1}\|\xi\|\|\eta\| \leq \|\xi\|^2 + Ch^{2p+2}|u(t)|^2_{H^{p+1}}.$$

(C) Term $\mathbf{Y_3}$: We apply Lemmas 1, 2 and Young's inequality:

$$|Y_3| \leq C\|e_h\|_\infty\|e_h\|C_I h^{-1}\|\xi\| \leq Ch^{-1}\|e_h\|_\infty\big(Ch^{2p+2}|u(t)|^2_{H^{p+1}} + \|\xi\|^2\big).$$

□

4. Error analysis of the semidiscrete scheme

We proceed similarly as for a parabolic equation. By Galerkin orthogonality, we subtract (5) and (4) and set $\varphi_h := \xi_h(t) \in S_h$. Since $\big(\frac{\partial\xi_h}{\partial t}, \xi_h\big) = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\xi_h\|^2$, we get

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\xi_h(t)\|^2 = b\big(u_h(t),\xi_h(t)\big) - b\big(u(t),\xi_h(t)\big) - \Big(\frac{\partial\eta_h(t)}{\partial t}, \xi_h(t)\Big).$$

For the last right-hand side term, we use the Cauchy and Young's inequalities and Lemma 2 and Lemma 3 for the convective terms. We integrate from 0 to $t \in [0,T]$,

$$\|\xi_h(t)\|^2 \leq C\int_0^t\Big(1+\tfrac{\|e_h(\vartheta)\|_\infty}{h}\Big)\Big(h^{2p+1}|u(\vartheta)|^2_{H^{p+1}} + h^{2p+2}|u_t(\vartheta)|^2_{H^{p+1}} + \|\xi_h(\vartheta)\|^2\Big)\mathrm{d}\vartheta, \quad (11)$$

where $C \geq 0$ is independent of h, t. For simplicity, we have assumed that $\xi_h(0) = 0$, i.e. $u_h^0 = \Pi_h u^0$. Otherwise we must assume e.g. $\|\xi_h(0)\|^2 \leq Ch^{2p+1}|u^0|^2_{H^{p+1}}$ and include this term in the estimate.

We notice that if we knew *apriori* that $\|e_h\|_\infty = O(h)$ then the unpleasant term $h^{-1}\|e_h\|_\infty$ in (11) would be $O(1)$. Thus we could simply apply the standard Gronwall lemma to obtain the desired error estimates. We state this formally:

Lemma 4. *Let $t \in [0,T]$ and $p \geq d/2$. If $\|e_h(\vartheta)\| \leq h^{1+d/2}$ for all $\vartheta \in [0,t]$, then there exists a constant C_T independent of h, t such that*

$$\max_{\vartheta\in[0,t]}\|e_h(\vartheta)\|^2 \leq C_T^2 h^{2p+1}. \quad (12)$$

Proof. The assumptions imply, by the inverse inequality and estimates of η, that

$$\|e_h(\vartheta)\|_\infty \leq \|\eta_h(\vartheta)\|_\infty + \|\xi_h(\vartheta)\|_\infty \leq Ch|u(t)|_{W^{1,\infty}} + C_I h^{-d/2}\|\xi_h(\vartheta)\| \quad (13)$$
$$\leq Ch + C_I h^{-d/2}\|e_h(\vartheta)\| + C_I h^{-d/2}\|\eta_h(\vartheta)\| \leq Ch + Ch^{p+1-d/2}|u(\vartheta)|_{H^{p+1}(\Omega)} \leq Ch,$$

where the constant C is independent of h, ϑ, t. Using this estimate in (11) gives us

$$\|\xi_h(t)\|^2 \leq \tilde{C}h^{2p+1} + C\int_0^t\|\xi_h(\vartheta)\|^2\,\mathrm{d}\vartheta, \quad (14)$$

where the constants $\widetilde{C}, C$ are independent of h, t. Gronwall's inequality applied to (14) states that there exists a constant $\widetilde{C}_T$, independent of h, t, such that

$$\max_{\vartheta\in[0,t]} \|\xi_h(\vartheta)\|^2 + \frac{1}{2}\int_0^t |\xi_h(\vartheta)|^2_{\Gamma_N}\, \mathrm{d}\vartheta \le \widetilde{C}_T h^{2p+1},$$

which allong with similar estimates for η gives us (12). □

Now it remains to get rid of the *apriori* assumption $\|e_h\|_\infty = O(h)$. In [5] this is done for an explicit scheme using mathematical induction. Starting from $\|e_h^0\| = O(h^{p+1/2})$, the following induction step is proved:

$$\|e_h^n\| = O(h^{p+1/2}) \quad\Longrightarrow\quad \|e_h^{n+1}\|_\infty = O(h) \quad\Longrightarrow\quad \|e_h^{n+1}\| = O(h^{p+1/2}). \tag{15}$$

For the method of lines we have continuous time and hence cannot use mathematical induction straightforwardly. However, we can divide $[0, T]$ into a finite number of sufficiently small intervals $[t_n, t_{n+1}]$ on which "e_h *does not change too much*" and use induction with respect to n. This is essentially a *continuous mathematical induction* argument, a concept introduced in [1], which has many generalizations, cf. [3].

Lemma 5 (Continuous mathematical induction). *Let $\varphi(t)$ be a propositional function depending on $t \in [0, T]$ such that*

(*i*) $\varphi(0)$ *is true,*

(*ii*) $\exists\delta_0 > 0:\ \varphi(t)$ *implies* $\varphi(t+\delta)$, $\forall t \in [0, T]\ \forall\delta \in [0, \delta_0] : t + \delta \in [0, T]$.

Then $\varphi(t)$ holds for all $t \in [0, T]$.

Remark 1 Due to the regularity assumptions, the functions $u(\cdot)$, $u_h(\cdot)$ are continuous mappings from $[0, T]$ to $L^2(\Omega)$. Since $[0, T]$ is a compact set, $e_h(\cdot)$ is a *uniformly continuous* function from $[0, T]$ to $L^2(\Omega)$. By definition,

$$\forall\epsilon > 0\ \exists\delta > 0 :\ s, \bar{s} \in [0, T], |s - \bar{s}| \le \delta \implies \|e_h(s) - e_h(\bar{s})\| \le \epsilon.$$

Theorem 6 (Semidiscrete error estimate). *Let $p > (1 + d)/2$. Let $h_1 > 0$ be such that $C_T h_1^{p+1/2} = \frac{1}{2} h_1^{1+d/2}$, where C_T is the constant from Lemma 4. Then for all $h \in (0, h_1]$ we have the estimate*

$$\max_{\vartheta\in[0,T]} \|e_h(\vartheta)\|^2 \le C_T^2 h^{2p+1}. \tag{16}$$

Proof. Since $p > (1 + d)/2$, h_1 is uniquely determined and $C_T h^{p+1/2} \le \frac{1}{2} h^{1+d/2}$ for all $h \in (0, h_1]$. We define the propositional function φ by

$$\varphi(t) \equiv \Big\{ \max_{\vartheta\in[0,t]} \|e_h(\vartheta)\|^2 \le C_T^2 h^{2p+1} \Big\}.$$

We shall use Lemma 5 to show that φ holds on $[0, T]$, hence $\varphi(T)$ holds, which is equivalent to (16).

(i) $\varphi(0)$ holds, since this is the error of the initial condition.
(ii) *Induction step*: We fix an arbitrary $h \in (0, h_1]$. By Remark 1, there exists $\delta_0 > 0$, such that if $t \in [0,T), \delta \in [0, \delta_0]$, then $\|e_h(t+\delta) - e_h(t)\| \leq \frac{1}{2}h^{1+d/2}$. Now let $t \in [0,T)$ and assume $\varphi(t)$ holds. Then $\varphi(t)$ implies $\|e_h(t)\| \leq C_T h^{p+1/2} \leq \frac{1}{2}h^{1+d/2}$. Let $\delta \in [0, \delta_0]$, then by uniform continuity

$$\|e_h(t+\delta)\| \leq \|e_h(t)\| + \|e_h(t+\delta) - e_h(t)\| \leq \tfrac{1}{2}h^{1+d/2} + \tfrac{1}{2}h^{1+d/2} = h^{1+d/2}.$$

This and $\varphi(t)$ implies that $\|e_h(s)\| \leq h^{1+d/2}$ for $s \in [0,t] \cup [t, t+\delta] = [0, t+\delta]$. By Lemma 4, φ holds on $[0, t+\delta]$. As a special case, we obtain the "induction step" $\varphi(t) \Longrightarrow \varphi(t+\delta)$ for all $\delta \in [0, \delta_0]$. □

5. Conclusion

We have presented the basic ideas behind the apriori analysis of nonlinear convective problems. To keep things as simple as possible, we have presented the analysis only for a space-semidiscrete scheme, with Dirichlet boundary conditions only. The extension to mixed boundary conditions, the extension to implicit schemes via continuation, derivation of improved estimates under the assumption $\mathbf{f} \in (C_b^3(\mathbb{R}))^d$ and the generalization to *locally Lipschitz* $\mathbf{f} \in (C^2(\mathbb{R}))^d$ can be found in [4].

Acknowledgements

The work was supported by the project P201/11/P414 of the Czech Science Foundation.

References

[1] Chao, Y. R.: A note on "Continuous mathematical induction". Bull. Amer. Math. Soc. **26** (1) (1919), 17–18.

[2] Ciarlet, P. G: *The finite element method for elliptic problems.* North-Holland, Amsterdam, 1979.

[3] Clark, P. L.: Real induction, available online `http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.187.3514`.

[4] Kučera, V.: Finite element error estimates for nonlinear convective problems. The Preprint Series of the School of Mathematics, preprint No. MATH-knm-2012/1, `http://www.karlin.mff.cuni.cz/ms-preprints/prep.php`. Submitted to Numer. Math, 2012.

[5] Zhang, Q. and Shu, C.-W.: Error estimates to smooth solutions of Runge–Kutta discontinuous Galerkin methods for scalar conservation laws. SIAM J. Numer. Anal. **42** (2) (2004), 641–666.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

THE OPTIMIZATION OF HEAT RADIATION INTENSITY

Jaroslav Mlýnek[1], Radek Srb[2]
[1] Department of Mathematics and Didactics of Mathematics
jaroslav.mlynek@tul.cz
[2] Institute of Mechatronics and Computer Engineering
radek.srb@tul.cz
Technical University of Liberec
Studentská 2, 461 17 Liberec, Czech Republic

Abstract

This article focuses on the problem of calculating the intensity of heat radiation and its optimization across the surface of an aluminium or nickel mould. The inner mould surface is sprinkled with a special PVC powder and the outer mould surface is warmed by infrared heaters located above the mould. In this way artificial leathers are produced in the car industry (e.g., the artificial leather on a car dashboard). The article includes a description of how a mathematical model allows us to calculate the heat radiation intensity across the mould surface for every fixed location of the heaters. In calculating the intensity of the heat radiation, we use experimentally measured values of the heat radiation intensity by a sensor at the selected points in the vicinity of the heater. It is necessary to optimize the location of the heaters to provide approximately the same heat radiation intensity across the whole mould surface during the warming of the mould (to obtain a uniform material structure and colour tone of the artificial leather). The problem of optimization is more complicated (used moulds are often very rugged, during the process of optimization we avoid possible collisions of two heaters as well as of a heater and the mould surface). A genetic algorithm and the technique of hill climbing are used during the process of optimization. The calculations were performed by a Matlab code written by the authors. The article contains a practical example.

1. Introduction

This article describes a procedure for the calculation of radiation intensity across the whole mould surface for fixed locations of infrared heaters and the process of heat radiation intensity optimization on the mould surface. The problem of optimization is rather complicated, a genetic algorithm and the hill climbing technique are used to find suitable locations for the heaters over the mould and to optimize the heat radiation intensity across the whole outer mould surface.

2. A mathematical model of heat radiation

In this chapter a simplified mathematical model of heat radiation produced by infrared heaters and absorbed by the outer mould surface is described. The heaters and the heated mould are represented in 3-dimensional Euclidean space using the Cartesian coordinate system (O, x_1, x_2, x_3) with basis vectors $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$, $e_3 = (0, 0, 1)$.

Representation of a heater. A heater is represented by a line segment of length d. The location of a heater is defined by the following parameters: 1/ coordinates of the heater centre $S = [x_1^S, x_2^S, x_3^S]$, 2/ the unit vector $u = (x_1^u, x_2^u, x_3^u)$ of the heat radiation direction, where $x_3^u < 0$ (i.e., heater radiates "downward "), 3/ the vector of the heater axis $r = (x_1^r, x_2^r, x_3^r)$ (see Figure 1). The other way to determine r is by using the angle φ between the positive part of the x_1-axis and the vertical projection of r onto the x_1x_2-plane (the vectors u and r are orthogonal, $0 \le \varphi < \pi$). The location of every heater Z can be defined by the following 6 parameters

$$Z : (x_1^S, x_2^S, x_3^S, x_1^u, x_2^u, \varphi). \tag{1}$$

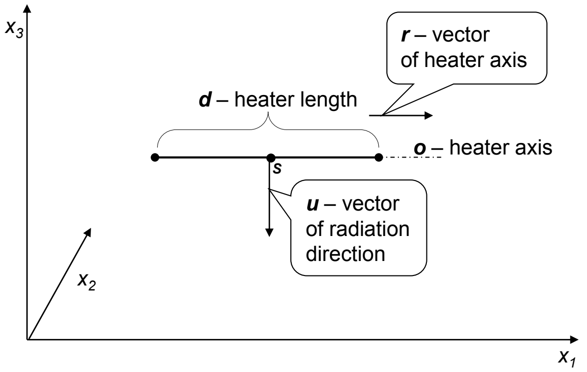


Figure 1: Representation of the heater in the model.

Representation of a mould. The mould surface P is described by the elementary surfaces p_j, where $1 \le j \le N$. It holds that $P = \cup p_j$, where $1 \le j \le N$ and $int\, p_i \cap int\, p_j = \emptyset$ for $i \ne j$, $1 \le i, j \le N$. Every elementary surface p_j is described by the following parameters: 1/ the center of gravity $T_j = [x_1^{T_j}, x_2^{T_j}, x_3^{T_j}]$, 2/ the unit outer normal vector $v_j = (x_1^{v_j}, x_2^{v_j}, x_3^{v_j})$ at the point T_j (we suppose v_j faces "upwards" and therefore is defined through the first two components $x_1^{v_j}$ and $x_2^{v_j}$), 3/ the size of its area s_j. Every elementary surface thus can be defined by the following 6 parameters:

$$p_j : (x_1^{T_j}, x_2^{T_j}, x_3^{T_j}, x_1^{v_j}, x_2^{v_j}, s_j). \tag{2}$$

3. The calculation of heat radiation intensity

We describe the process of calculating the heat radiation intensity on the mould surface for given fixed locations of the heaters. The heater manufacturer has not provided the distribution function of the heat radiation intensity in the heater surroundings. We realized the experimental measurement of the heat radiation intensity. The measured heater location was $Z:(0,0,0,0,0,0)$ in accordance with the relation (1), i.e., the center S of the heater was situated at the origin of the Cartesian coordinate system (O,x_1,x_2,x_3), the unit radiation vector had coordinates $u=(0,0,-1)$ and the vector of the heater axis had coordinates $r=(1,0,0)$. We suppose the heat radiation intensity across the elementary surface p_j is the same as at the centre of gravity T_j. The heat radiation intensity at T_j depends on the position of this point (determined by the first three parameters in the vector p_j given by the relation(2)) and on the direction of the outer normal vector v_j at the point T_j (determined by the fourth and fifth parameters in the vector p_j given by (2)). The heat radiation intensity in the surroundings of the heater was experimentally measured by a sensor placed at chosen points below the heater. We use a linear interpolation function of 5 variables to continuously interpolate the measured heat radiation intensity in the vicinity of the heater Z (for more detail see [4]).

For a heater in a general position, we briefly describe the transformation of the previous Cartesian coordinate system (O,e_1,e_2,e_3) into a positively oriented Cartesian system $(S,r,n,-u)$, where S is the centre of the heater, r is the heater axis vector, and u is the direction vector of the heat radiation. The vector n is determined by the vector product of the vectors $-u$ and r (see more detail in [1]) and is defined by the relation

$$n=(-u)\times r=\left(-\begin{vmatrix} x_2^u & x_3^u \\ x_2^r & x_3^r \end{vmatrix}, \begin{vmatrix} x_1^u & x_3^u \\ x_1^r & x_3^r \end{vmatrix}, -\begin{vmatrix} x_1^u & x_2^u \\ x_1^r & x_2^r \end{vmatrix}\right).$$

The vectors r, u and n are normalized to have the unit length. Then we can define an orthonormal transformation matrix

$$\mathbf{A}=\begin{pmatrix} x_1^r & x_1^n & -x_1^u \\ x_2^r & x_2^n & -x_2^u \\ x_3^r & x_3^n & -x_3^u \end{pmatrix}.$$

Let us recall that, for the elementary surface p_j, the respective triples T_j and v_j represent its centre of gravity and its outer normal vector in the Cartesian coordinate system (O,e_1,e_2,e_3). If S is the triple representing (in (O,e_1,e_2,e_3)) the center of the heater that determines the coordinate system $(S,r,n,-u)$, then T_j and v_j are transformed as follows:

$$\left(T_j^{'}\right)^T=\mathbf{A}^T\left(T_j-S\right)^T \quad \text{and} \quad \left(v_j^{'}\right)^T=\mathbf{A}^T v_j^T,$$

where $T_j^{'}$ and $v_j^{'}$ are the coordinates in $(S,r,n,-u)$. In this way, we transform the general case to the measured case.

Now we describe the procedure of numerical computation for the total heat radiation intensity on the mould surface. We denote L_j as the set of all heaters radiating on the jth elementary surface p_j $(1 \leq j \leq N)$ for the fixed locations of heaters, and I_{jl} the heat radiation intensity of the lth heater on the p_j elementary surface. Then the total radiation intensity I_j on the elementary surface p_j is given by the following relation (see in detail in [2])

$$I_j = \sum_{l \in L_j} I_{jl} \ . \tag{3}$$

The producer of artificial leathers recommends the constant value of heat radiation intensity across the whole outer mould surface. Let us denote this constant value as I_{rec}. We can define F, the aberration of the heat radiation intensity, by the relation

$$F = \frac{\sum_{j=1}^{N} |I_j - I_{rec}| s_j}{\sum_{j=1}^{N} s_j} \tag{4}$$

and the aberration $\tilde{F}$ by the relation

$$\tilde{F} = \sqrt{\sum_{j=1}^{N} (I_j - I_{rec})^2 \, s_j} \ . \tag{5}$$

4. The optimization of the location of the heaters

We use a genetic algorithm for global optimization and subsequently the hill climbing method for local optimization of the locations of heaters. These methods are described in more details in [3] and in [5]. The location of every heater is defined in accordance with the relation (1) by 6 parameters. Therefore $6M$ parameters are necessary to define the locations of all M heaters. One chromosome represents one individual (one possible location of the heaters). The population includes Q individuals. The generated individuals are saved in the matrix $\mathbf{B}_{Q \times 6M}$. Every row of this matrix represents one individual. We seek the individual $y_{min} \in C$ satisfying the condition

$$F(y_{min}) = min\{F(y); y \in C\}, \tag{6}$$

where $C \subset E_{6M}$ is the searched set. Every element of C is formed by a set of $6M$ allowable parameters and this set defines just one constellation of the heaters above the mould. The function F is defined by (4) or by (5). The identification of the individual y_{min} defined by (6) is not realistic in practice. But we are able to determine an optimized solution y_{opt}. Now we describe particular steps of the genetic algorithm that is used.

Genetic algorithm

Input: the specimen y_1 (initial individual), ε_1 - the specified accuracy of the calculation.

```
Internal computation:
1. create an initial population of $Q$ individuals,
2.a/ evaluate all the individuals of the population (calculate $F(y)$ for every individual $y$), b/ sort $F(y)$ in the ascending order and organize $y$ accordingly,
c/ store the individulas $y$ into the matrix $\mathbf{B}$,
3. repeat until min$\{F(y); y \in \mathbf{B}\} < \varepsilon_1$
a/ chose randomly between the crossover operation and the mutation operation,
b/ if the crossover operation is chosen then
        randomly select a pair of individuals (parents),
        execute the crossover operation and create two new individuals
                                            else
        randomly select an individual $y$, execute the mutation operation,
        create two new individuals
    end if,
c/ calculate $F(y)$ for the two new individuals (penalize an individual in the case of the collision of heaters or the collision of a heater and the mould surface), d/ sort as in 2.b/, e/ take the first $Q$ individuals with the best evaluation $F(y)$ and store them in the matrix $\mathbf{B}$
end repeat.
Output: the first row of matrix $\mathbf{B}$ contains the best found individual.
```

To further optimize y_{opt} delivered by the genetic algorithm, we apply the hill climbing method.

Hill climbing algorithm

```
Input: $y_{opt}$ - the solution provided by the genetic algorithm, real suitable increments $h_i$, where $1 \leq i \leq 6M$, $\varepsilon_2$ - the specified accuracy of the calculation.
Internal computation:
repeat until max$\{|h_i|; 1 \leq i \leq 6M\} < \varepsilon_2$
for i:= 1 to $6M$ do
a/ while $F(y_{opt}) > F(y_{opt} + h_i)$ do $y_{opt} := y_{opt} + h_i$
end while,
b/ $h_i := -h_i/2$
end for
end repeat.
Output: the best found individual.
```

The individual y_{opt} is the final optimized solution and includes information about the location of every heater in the form (1).

5. A practical example

Now we describe a practical example of the heating of an aluminium mould. The volume of the mould is $0.8 \times 0.4 \times 0.15[m^3]$, the number of elementary surfaces is $N = 2,064$; the recommended heat radiation intensity is $I_{rec} = 47[\mathrm{kW/m}^2]$. We use 16 infrared heaters of the same type (producer Philips, capacity $1,600$[W], length 15[cm], width 4[cm]). In the first step of our procedure we construct a specimen y_1 (this individual corresponds to the default locations of heaters). The centers of the heaters lie in the plane parallel to the x_1x_2-plane and at a distance of 10[cm] from the center of gravity T_j of the elementary surface p_j with the highest value $x_3^{T_j}$ $(1 \leq j \leq N)$. All the heaters have $r = (1, 0, 0)$ and $u = (0, 0, -1)$ (that is, all the heaters radiate downwards and they are parallel to the axis x_1). The population contains 30 individuals $(Q = 30)$.

For the initial specimen y_1 and F given by (4), we get $F(y_1) = 20.74$. We obtain the optimized individual y_{opt} with value $F(y_{opt}) = 3.39$ after $100,000$ iterations of the genetic algorithm and $5,000$ iterations of the hill climbing method (two individuls are generated during every iteration of the genetic algorithm and one individual is generated during every iteration of the hill climbing method). The value $F(y_{opt})$ depends on the number of iterations of the genetic algorithm and hill climbing method (see Figure 2).

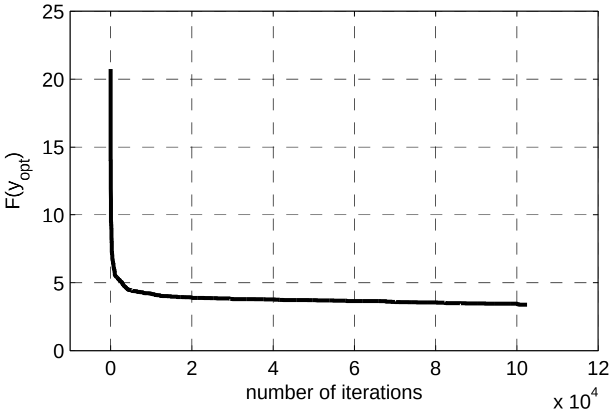


Figure 2: Dependence of $F(y_{opt})$ on the number of iterations.

The graphical representation of heat radiation on the mould surface (levels of radiation intensity in $[\mathrm{kW/m}^2]$ correspond to shades of grey colouring) and the locations of the heaters corresponding to the individual y_{opt} are displayed in Figure 3.

Let us replace F by $\tilde{F}$, see (5), and let us execute the same number of iterations. We get the following results: $\tilde{F}(y_1) = 25.13$; $\tilde{F}(y_{opt}) = 3.34$.

On the basis of our numerical tests, we have obtained results sufficiently accurate for the needs of production.

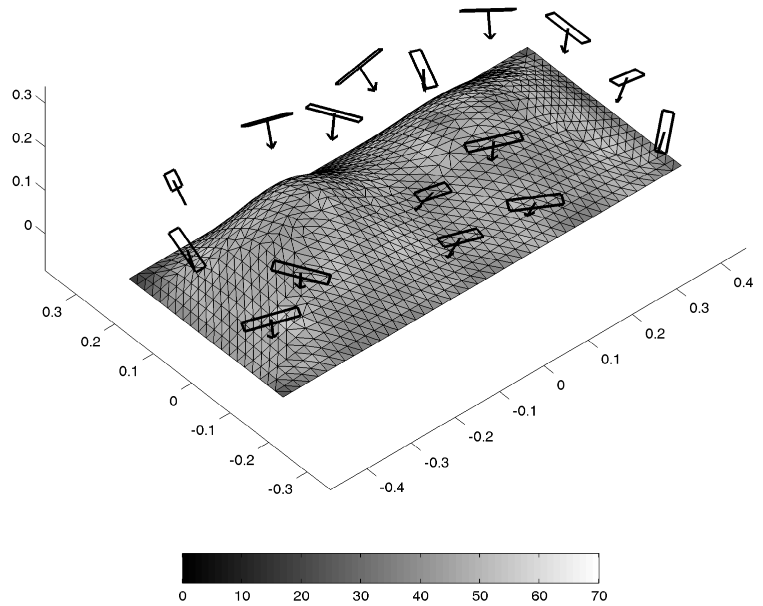


Figure 3: Heat radiation intensity([kW/m^2]) on the mould surface and the location of the heaters corresponding to the individual y_{opt}.

Acknowledgements

This work was supported by project MPO, No. FR-TI1/266.

References

[1] Budinský, B.: *Analytical and differential geometry.* SNTL, Prague, 1983 (in Czech).

[2] Cengel,Y. A.: *Heat and mass transfer.* McGraw-Hill, New York, 2007.

[3] Chambers, L.: *Genetic algorithms.* Chapman and Hall/CRC, Boca Raton, 2001.

[4] Mlýnek, J., Srb, R.: The Process of aluminium moulds warming in the car industry. In: *Journal of Automation, Mobile Robotics and Intelligent Systems*, Industrial Research Institute for Automation and Measurements PIAP, Warsaw, Vol. 6, No. 2, 2012, 47–51.

[5] Mlýnek, J., Srb, R.: The process of an optimized heat radiation intensity calculation on a mould surface. In: K. G. Troitzsch (Ed.), *Proceedings of the 29th European Conference on Modelling and Simulation*, Digitaldruck Pirrot GmbH, Koblenz, Germany, May 2012, 461–467.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

SHAPE FUNCTIONS AND WAVELETS – TOOLS OF NUMERICAL APPROXIMATION

Vratislava Mošová

Moravian College Olomouc,
Jeremenkova 40, 771 00 Olomouc, Czech Republic,
vratislava.mosova@mvso.cz

Abstract

Solution of a boundary value problem is often realized as the application of the Galerkin method to the weak formulation of given problem. It is possible to generate a trial space by means of splines or by means of functions that are not polynomial and have compact support. We restrict our attention only to RKP shape functions and compactly supported wavelets. Common features and comparison of approximation properties of these functions will be studied in the contribution.

1. Introduction

One of the possibilities to solve boundary value problems is the Galerkin method. Céa's lemma (1964) says that the error in the Galerkin method depends on how well the exact solution is approximated by elements of the trial space. There is a lot of possibilities how to generate such space. For example, it is possible to deal with compactly supported wavelets or with RKP shape functions. The solving of some boundary value problems by using wavelet bases can be found in [5], [2] and by using RKP shape functions for example in [4], [1]. Our aim is to introduce wavelets and RKP shape functions and compare their properties.

The outline of the next text is as follows. Some basic information on the construction and properties of the wavelet basis are presented in Section 2. The construction and properties of the RKP shape functions are described in Section 3. Finally, a comparision of properties of the wavelets and the RKP shape functions is shown in Section 4.

2. Wavelets

Wavelets have grown up not only from theoretical mathematical study but also from practical reasons. The technique of the wavelet transform is used in signal processing. It is a very effective tool, because it gives possibility to change window during the analysing of signal (in contrast with the Fourier transform). It allows to extract information from many different kinds of data, it can help to analyze voice or to compress pictures. It can also serve to analyze variability, to remove noise or to detect significant moments in the time series that are used in economy. In numerical mathematics, the wavelet bases can be used by the solution of boundary values problems, where they provide perfect space and spectral localization. They combine the advantage of the basis used in the FEM with the advantage of the basis used in spectral analysis.

Construction of the wavelet system

A function $\psi \in L^2(R)$ is called the *basic wavelet*, if the condition of stability

$$\int_R \frac{|\hat{\psi}(\xi)|^2}{|\xi|}\,\mathrm{d}\xi < \infty \tag{1}$$

is satisfied.

In this text, we will deal with two types of the basic wavelets – the *scaling function* φ and the *associated wavelet* ψ.

It is possible to receive an orthonormal basis in $L^2(R)$ by means of the *multiresolution analysis* (MRA). The MRA is an efective but not the only one way to obtain an orthogonal wavelet system. Each wavelet that quickly decreases to zero and that is smooth enough can be constructed by it.

In MRA, the spaces $V_j \subset L^2(R)$ $(j \in Z)$ that satisfy

$$\begin{gathered} V_j \subset V_{j+1};\ \bigcap_{j\in Z} V_j = \{0\};\ \bigcup_{j\in Z} V_j = L^2(R); \\ \exists\varphi \in V_0 : \{\varphi(x-k)\}_{k\in Z} \text{ is a complete orthogonal set in } L^2(R); \\ f \in V_0 \Leftrightarrow f(2^j x) \in V_j \end{gathered} \tag{2}$$

are constructed.

It follows from the properties given above that there exists the subspace W_j orthogonal to V_j such that $V_{j+1} = V_j \oplus W_j$. It means that $V_{j+1} = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_j$. Next, we put

$$V_j = \{\varphi_{j,k}\}_{j,k\in Z},\ \text{where } \varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k), \tag{3}$$

$$W_j = \{\psi_{j,k}\}_{j,k\in Z},\ \text{where } \psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k). \tag{4}$$

If a boundary value problem is solved numerically, it is suitable to generate the trial space by wavelets that have compact support. In this case, the scaling function φ and the associated wavelet have to satisfy

$$\varphi(x) = \sum_{k=0}^{D-1} a_k \varphi_{1,k}(x),\quad \psi(x) = \sum_{k=0}^{D-1} b_k \varphi_{1,k}(x),\ \text{where } b_k = (-1)^k a_{1-k}. \tag{5}$$

Example The class of Daubechies wavelets (including coiflets and symlets) can be received by the MRA. The compactly supported Daubechies wavelet of order 4 together with its scaling function are in Figure 1.

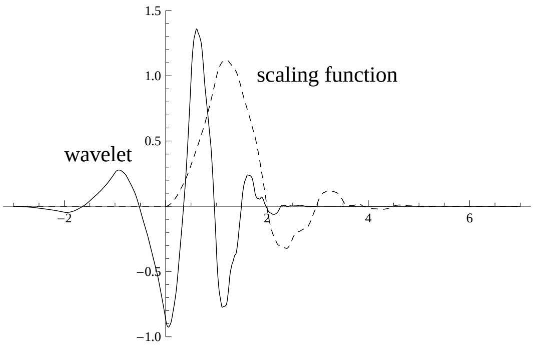


Figure 1: The Daubechies wavelet Db4

Properties of wavelets

1) It holds for wavelets defined by (5) that $\operatorname{supp}\varphi(x) = \langle 0, D-1\rangle$, $\operatorname{supp}\psi = \langle 1-\frac{D}{2}, \frac{D}{2}\rangle$. ($D = 2N$ for the Daubechies wavelets of order N.)

2) The functions $\{\varphi_{0,k}\}_{k\in\mathcal{Z}}$, $\{\psi_{l,k}\}_{k\in\mathcal{Z},\, l=1,\dots j}$ form an orthonormal basis in $V_{j+1} \subset L^2(R)$. It is possible to express an approximation of a function $u \in L^2(R)$ by means of

$$\tilde{u}(x) = \sum_{k\in\mathcal{Z}} c_{0,k}\varphi_{0,k}(x) + \sum_{l=1}^{j}\sum_{k\in\mathcal{Z}} c_{l,k}\psi_{l,k}(x). \tag{6}$$

3) We can see from the relation (3) that the functions $\{\varphi_{0,k}\}_{k\in Z}$ are translation invariant: $\varphi_{0,k+m}(x) = \varphi_{0,k}(x-m)$.

4) The approximation properties of the MRA are given in the next theorem (see [5]).

Theorem 1. *Let $\{V_j\}$ be the MRA with $\varphi \in L^1(R)$, φ be compactly supported, the value of the Fourier transform $\hat{\varphi}(0) = 1$ and $L \geq 1$, then the next conditions are equivalent*

(a) The Strang-Fix condition of order $L-1$: Function φ satisfies

$$\frac{\mathrm{d}^q}{\mathrm{d}\xi^q}\hat{\varphi}(2n\pi) = 0,\ n \neq 0,\ n \in Z,\ q = 0,\dots,L-1. \tag{7}$$

(b) The quasi-reproducing condition of order $L-1$: Function φ satisfies

$$\sum_{k\in Z} k^q\varphi(x-k) = x^q + p_{q-1}(x)\ for\ all\ x \in R,\ q = 0,\dots,L-1. \tag{8}$$

Here p_{q-1} is a polynomial that has order less or equal $q-1$.

(c) The vanishing moment condition: It holds for the q^{th} moment of the associated wavelet

$$M_q(\psi) = \int_R x^q\psi(x)\,\mathrm{d}x = 0 \quad \forall q = 1,\dots,L-1. \tag{9}$$

(d) There exist coeficients $c_{j,k}$, $j,k \in Z$, and constants C_s, such that it holds for all $u \in W^{L,2}(R)$

$$\|u - \sum_{k\in Z} c_{j,k}\varphi_{j,k}\|_{W^{s,2}(R)} \leq C_s 2^{-j(L-s)}|u|_{W^{L,2}(R)}\ for\ s = 0,\dots,L-1. \tag{10}$$

Remark The construction of orthogonal wavelet bases on the real line was described in the previous text. Note that if boundary value problems are solved, it is necessary to adapt wavelet bases to the interval. Some problems can occur when the wavelets are used directly as trial functions. For example the introduction of Dirichlet boundary conditions is difficult. Lower order wavelets cannot be employed due to the lack of regularity. Also the request for orthogonality in (2) is too strong. It appears better to use Riesz wavelet bases than orthonormal bases given above by solving BVP's. Especially the biorthogonal multiwavelets on the basis of splines are used successfully.

3. RKP-shape functions

Meshless methods were developed to find the solution of boundary value problems for differential equations that describe practical problems such as large deformation, crack propagation or moving boundary problems where it is necessary to overmesh during computation. The fact that meshless methods need no explicitly given mesh avoids or greatly simplifies this meshing task. The trial space is generated by shape functions in meshless methods. There is a lot of meshless methods and each of them constructs the shape functions in a different way. For instance the Reproducing Kernel Particle Method (RKPM) belongs to meshless methods that are based on kernel approximation.

Construction of shape functions

Let $x_1, \ldots, x_N$ be particles in $\langle a, b\rangle$, $w(x)$ be a weight function (i.e. continuous, compactly supported function) and $\mathbf{p}(x) = (p_0(x), \ldots, p_s(x))$ be a polynomial basis of order s (i.e. components $p_j \in P_{\leq s},\ s \geq 0$.)

The one dimensional *RKP shape function* $\Phi_j^{[\alpha]}(x)$ of order α, $0 \leq \alpha \leq s$, which is associated with the particle x_j, is defined by

$$\Phi_j^{[\alpha]}(x) = \alpha!\mathbf{p}\left(\frac{x - x_j}{\rho}\right)\mathbf{b}_\alpha^T(x)\, w\left(\frac{x - x_j}{\rho}\right)\Delta x_j. \tag{11}$$

Here $\rho > 0$ is a dilatation parameter, Δx_j is the quadrature weight and vector $\mathbf{b}_\alpha(x)$ is the solution of the linear equations

$$M(x)\mathbf{b}_\alpha^T(x) = \left(\mathbf{p}^{(\alpha)}(0)\right)^T, \tag{12}$$

where $M(x) = \sum_{j=1}^N \mathbf{p}^T\left(\frac{x-x_j}{\rho}\right)\mathbf{p}\left(\frac{x-x_j}{\rho}\right) w\left(\frac{x-x_j}{\rho}\right)\Delta x_j$ and $\mathbf{p}^{(\alpha)}(x) = \frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}\mathbf{p}(x)$

The vector $\mathbf{b}_\alpha(x)$ is constructed in such a way that the shape functions $\Phi_j^{[\alpha]}(x)$ reproduce polynomials of order $s - \alpha$.

If we use (12), (11) and put $p_\beta\left(\frac{x-x_j}{\rho}\right) = \left(\frac{x-x_j}{\rho}\right)^\beta,\ 0 \leq \beta \leq s$, we can see that the condition (12) leads to system

$$\sum_{j=1}^N \left(\frac{x - x_j}{\rho}\right)^\beta \Phi_j^{[\alpha]}(x) = \alpha!\delta_{\beta,\alpha},\quad 0 \leq \alpha, \beta \leq s. \tag{13}$$

Example The system of reproducing RKP shape functions $\Phi_3^{[0]}$ and $\Phi_3^{[1]}$ is given in Figure 2. They are constructed on the interval $\langle 0,1\rangle$ for $N=5$ equidistant particles, $\mathbf{p}(x)=(1,x)$, $w(x)=\begin{cases}(1-x^2)^2 & \text{if} \quad |x|\le 1\\ 0 & \text{if} \quad |x|>1\end{cases}$ and $\rho=0.3$.

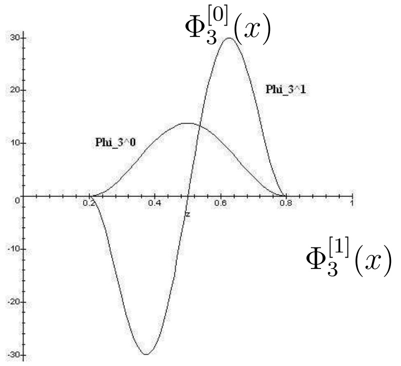


Figure 2: Shape functions

Properties of RKP shape functions

Suppose that RKP shape functions are defined by (11), (12).

1) The continuous version of function $\Phi_0^{[0]}$ satisfies the condition of stability (1) for the basic wavelet (see [3]).

2) The support and smoothness of $\Phi_j^{[0]}$ are the same as the support of the given weight function w.

3) The functions $\Phi_j^{[0]}$ are translation invariant for uniformly distributed particles, i.e. $\Phi_{j+k}^{[0]}(x)=\Phi_j^{[0]}(x-x_k)$, where $x_k=kh,\ k\in Z$ (see [1]).

4) From the conditon (13) one can receive that the shape functions $\Phi_j^{[0]}$ are reproducing of order s i.e. they reproduce polynomials from $P_{\le s}$ exactly (see [3]).

5) It is possible to receive from (13) that $\sum_{j=1}^N \Phi_j^{[0]}(x)=1$ and $\sum_{j=1}^N \Phi_j^{[\alpha]}(x)=0$. It means that the shape functions $\Phi_j^{[\alpha]}, 0\le\alpha\le s$, form the partition of unity and an approximation of a function $u\in W^{1,2}(\Omega)$ can be supposed in the form

$$\tilde{u}(x)=\sum_{j=1}^{N} c_{0,j}\Phi_j^{[0]}(x)+\sum_{\alpha=1}^{s}\sum_{j=1}^{N} c_{\alpha,j}\Phi_j^{[\alpha]}(x). \tag{14}$$

6) Because the property "reproducing order" is a particular case of "quasi-reproducing order", the error of approximation can be determined from the Strang-Fix theorem (see [1]).

Theorem 2. *Let particles $\{x_i\}$ be uniformly distributed, $\Phi_j^{[0]}\in W^{q,2}(R)$, $q\ge 0$, be reproducing of order s. Then for each $u\in W^{k+1,2}(R)$ there are $C, c_j\in R$ such that*

$$\|u-\sum_{j\in Z} c_j\Phi_j^{[0]}\|_{W^{s,2}}\le C\,h^{k+1-s}\|u\|_{W^{k+1,2}}\ for\ 0\le s\le\min\{q,k+1\}. \tag{15}$$

4. Conclusion

In this contribution the construction of compactly supported wavelet and RKP shape function systems is described. Then a short overview of properties of these systems is given. It is possible to say that even though these systems are built in different ways, they have some common features.

For example: The basic functions $\Phi_0^{[0]}$ behave similarly as the scaling functions φ. It is possible to obtain the constructed systems from these basic functions using translation and dilatation. The basic functions are able to approximate polynomials of the order, which corresponds to the order of reproducing conditions that they satisfy. The functions $\psi_{j,k}$ and $\Phi_j^{[\alpha]}, \alpha \neq 0$, satisfy the vanishing moment condition. It is possible to carry out the estimate of approximation errors using the Strang-Fix theorem.

However, it is possible to find some differences between wavelet bases and RKP shape functions that are used for solution of BVP's. For example, biorhotgonal wavelet bases are Riesz bases, but the sequence $\{\Phi_I^{[\alpha]}(x), \alpha \geq 0\}$ is only a frame. Wavelet basis provides the possibility to compute effectively coefficients of a stiffness matrix, but the RKP shape functions do not offer any similar advantage. On the other hand, it is possible to construct RKP shape functions that have the desired order of continuity and that are not linked to any explicitly given mesh.

Acknowledgements

This work was supported by project GACR P403/12/1811: Unconventional Managerials Decision making Methods Development in Enterprise Economics and Public Economy.

References

[1] Babuška, I., Banerjee, U., Osborn, J. E.: Survey of meshless and generalized finite element mehods: A unified approach, Acta Numer. (2003), 1–125.

[2] Černá, D., Finěk, V.: Adaptive frame methods with cubic spline-wavelet bases. In: J. Chleboun, P. Přikryl, K. Segeth, T. Vejchodský (Eds.), Programs and Algorithms of Numerical Mathematics 14, pp. 59–64, IM AS CR, Prague, 2008.

[3] Li, S., Liu, W. K.: Reproducing kernel hierarchical partition of unity - Part I: Foundation and theory, Internat. J. Numer. Methods Engrg. **45** (1999), 251–288.

[4] Li, S., Liu, W. K.: Reproducing kernel hierarchical partition of unity – Part II: Application, Internat. J. Numer. Methods Engrg. **45** (1999), 289–317.

[5] Najzar, K.: Základy teorie waveletů (Basics of wavelets theory), in Czech, Karolinum, Praha, 2004.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

AN OPTIMAL ALGORITHM WITH BARZILAI-BORWEIN STEPLENGTH AND SUPERRELAXATION FOR QPQC PROBLEM

Lukáš Pospíšil

FEECS VSB-Technical University of Ostrava
17. listopadu 15, CZ-70833 Ostrava, Czech Republic
lukas.pospisil@vsb.cz

Abstract

We propose a modification of MPGP algorithm for solving minimizing problem of strictly convex quadratic function subject to separable spherical constraints. This active set based algorithm explores the faces by the conjugate gradients and changes the active sets and active variables by the gradient projection with the Barzilai-Borwein steplength. We show how to use the algorithm for the solution of separable and equality constraints. The power of our modification is demonstrated on the solution of a contact problem with Tresca friction.

1. Motivation

Let us consider simple contact problem with given friction. The block of homogeneous material has prescribed zero displacements on boundary Γ_D and imposed traction $\boldsymbol{F}$ on Γ_F. The part Γ_C denotes the part of boundary that may get into contact with rigid obstacle. The block is attracted to obstacle by gravity force $\boldsymbol{F}_G$.

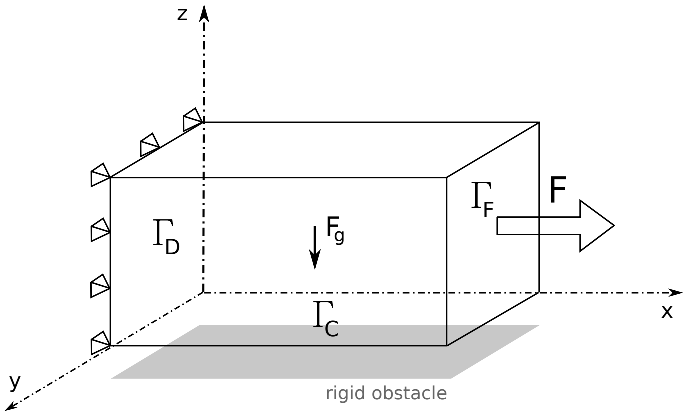


Figure 1: Contact problem with rigid obstacle and given friction.

We solve discretized form of the problem using FEM. This technique leads to optimizing problem (see [3])

$$\bar{u} := \min_{\boldsymbol{u}\in\Omega} \left(f(\boldsymbol{u}) + j_h(\boldsymbol{u})\right), \quad f(\boldsymbol{u}) := \frac{1}{2}\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}^T\boldsymbol{u}, \quad j_h(\boldsymbol{u}) := \sum_{i=1}^{m_c} \psi_i \|\boldsymbol{T}_i\boldsymbol{u}\|, \quad (1)$$

where $N \in \mathbb{N}$ is number of used nodes and $n = 3N$ is number of variables, $\boldsymbol{u} \in \mathbb{R}^n$ is a vector of unknown displacements, $\Omega := \{\boldsymbol{u} \in \mathbb{R}^n : u_z \geq -c\}$ is set of feasible $\boldsymbol{u}$, $c \in \mathbb{R}_0^+$ is a distance between body and rigid obstacle, $f : \mathbb{R}^n \to \mathbb{R}$ denotes function of total potential energy, $\boldsymbol{K} \in \mathbb{R}^{n,n}$ is a symmetric-positive definite stiffness matrix, $\boldsymbol{f} \in \mathbb{R}^n$ is vector of internal forces resulting from the stresses imposed on the structure during a displacement, $j_h : \mathbb{R}^n \to \mathbb{R}$ is numerical integration of functional describing the friction forces in the weak formulation of the problem, $\boldsymbol{T}_i \in \mathbb{R}^{2,n}$ are formed by appropriately placed multiples of the unit tangential vectors in such way that the jump of tangential displacement due to displacement $\boldsymbol{u}$ is given by $\boldsymbol{T}_i\boldsymbol{u}$, $\psi_i \in \mathbb{R}$ is slip bound associated with $\boldsymbol{T}_i$.

At first denote $m_c \leq N$ as number of FEM nodes in Γ_C.
Our problem has simple geometry, so we can simply choose $\boldsymbol{n} := [0, 0, -1]$ as normal vector and $\boldsymbol{t}_1 := [1, 0, 0], \boldsymbol{t}_2 := [0, 1, 0]$ as tangential vectors for every FEM node in Γ_C.

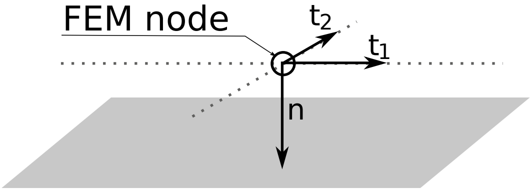


Figure 2: Normal and tangential vectors on Γ_C.

So for every contact node (i-th node from Γ_C) is $\boldsymbol{T}_i \in \mathbb{R}^{2,n}$ given by sparse matrix with 1 in first row on appropriate x-coordinate of i-th node and in second row on appropriate y-coordinate of i-th node. Then we assume that $\boldsymbol{T} := \left[\boldsymbol{T}_1^T, \ldots, \boldsymbol{T}_{m_c}^T\right]^T$ is the full rank matrix.

In our problem with Dirichlet conditions, f is strictly convex quadratic function (i.e. quadratic function with symmetric positive-definite matrix $\boldsymbol{K}$), so in next eductions, we can use standard inversion $\boldsymbol{K}^{-1}$.
We can express the non-differentiable term j_h in (1) by (see [7])

$$j_h(\boldsymbol{u}) = \sum_{i=1}^{m_c} \max_{\|\tau_i\|\leq\psi_i} \boldsymbol{\tau}_i^T\boldsymbol{T}_i\boldsymbol{u}, \quad (2)$$

where $\boldsymbol{\tau}_i \in \mathbb{R}^2$ are regulation variables.

2. Saddle point problem equivalency and dual formulation

At first, we denote function and vector

$$\tilde{L}(\boldsymbol{u}, \boldsymbol{\tau}) := f(\boldsymbol{u}) + \boldsymbol{\tau}^T \boldsymbol{T} \boldsymbol{u}, \quad \boldsymbol{\tau} := [\boldsymbol{\tau}_1^T, \dots, \boldsymbol{\tau}_{m_c}^T]^T. \tag{3}$$

Then the conditions $\|\boldsymbol{\tau}_i\| \leq \psi_i$ can by written in form

$$\sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, \quad i = 1, \dots, m_c, \tag{4}$$

where τ_j is j-th component of $\boldsymbol{\tau}$.
Now we can simplify the notation, we denote set of feasible $\boldsymbol{\tau}$ as

$$\Lambda_\tau := \left\{ \sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, i = 1, \dots, m_c \right\}. \tag{5}$$

After substituing (2) into (1) and using (3),(4) we get

$$\min_{\boldsymbol{u} \in \Omega} (f(\boldsymbol{u}) + j_h(\boldsymbol{u})) = \min_{\boldsymbol{u} \in \Omega} \left(f(\boldsymbol{u}) + \sum_{i=1}^{m_c} \max_{\|\tau_i\| \leq \psi_i} \boldsymbol{\tau}_i^T \boldsymbol{T}_i \boldsymbol{u} \right) = \min_{\boldsymbol{u} \in \Omega} \sup_{\boldsymbol{\tau} \in \Lambda_\tau} \tilde{L}(\boldsymbol{u}, \boldsymbol{\tau}). \tag{6}$$

If we consider $\tilde{L}(\boldsymbol{u}, \boldsymbol{\tau})$ as Lagrange function and $\boldsymbol{\tau}$ as vector of Lagrange multipliers (in notation (3)), we can use the classical duality theorem (see [4]) to reformulate problem (6) and get

$$\min_{\boldsymbol{u} \in \Omega} \sup_{\boldsymbol{\tau} \in \Lambda_\tau} \tilde{L}(\boldsymbol{u}, \boldsymbol{\tau}) = \max_{\boldsymbol{\tau} \in \Lambda_\tau} \min_{\boldsymbol{u} \in \Omega} \tilde{L}(\boldsymbol{u}, \boldsymbol{\tau}). \tag{7}$$

Now we can include condition $\boldsymbol{u} \in \Omega$ by creating new Lagrange multipliers.

$$\max_{\boldsymbol{\tau} \in \Lambda_\tau} \min_{\boldsymbol{u} \in \Omega_C} \tilde{L}(\boldsymbol{u}, \boldsymbol{\tau}) = \max_{\boldsymbol{\tau} \in \Lambda_\tau, \boldsymbol{\lambda}_C \geq 0} \min_{\boldsymbol{u} \in \mathbb{R}^n} \left(\tilde{L}(\boldsymbol{u}, \boldsymbol{\tau}) + \boldsymbol{\lambda}_C^T (\boldsymbol{B}\boldsymbol{u} - \boldsymbol{c}) \right), \tag{8}$$

where matrix $\boldsymbol{B} \in \mathbb{R}^{m_c, n}$ and vector $\boldsymbol{c} \in \mathbb{R}^{m_c}$ are constructed in such way, that

$$\{\boldsymbol{u} \in \mathbb{R}^n : \boldsymbol{B}\boldsymbol{u} \leq \boldsymbol{c}\} = \Omega.$$

Due to geometry in our problem we can construct $\boldsymbol{B}$ very simply. $\boldsymbol{B}$ is a sparse matrix with -1 in every i-th row (which is corresponding to i-th node in Γ_C) on appropriate z-coordinate of i-th node (see former choice of normal vectors for nodes in Γ_C).

So problem (1) is equivalent to the saddle point problem

$$(\bar{\boldsymbol{u}}, \bar{\boldsymbol{\lambda}}) := \arg \max_{\boldsymbol{\lambda} \in \Lambda} \min_{\boldsymbol{u} \in \mathbb{R}^n} L(\boldsymbol{u}, \boldsymbol{\lambda}), \tag{9}$$

where

$$L(\boldsymbol{u}, \boldsymbol{\lambda}) := f(\boldsymbol{u}) + \boldsymbol{\lambda}^T (\tilde{\boldsymbol{B}}\boldsymbol{u} - \tilde{\boldsymbol{c}}) \tag{10}$$

is Lagrange function, which includes both of friction and non-penetration conditions, and

$$\boldsymbol{\lambda} := \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda}_C \end{bmatrix}, \quad \tilde{\boldsymbol{B}} := \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{B} \end{bmatrix}, \quad \tilde{\boldsymbol{c}} := \begin{bmatrix} \boldsymbol{o} \\ \boldsymbol{c} \end{bmatrix},$$

$$\Lambda := \{[\boldsymbol{\tau}, \boldsymbol{\lambda}_C] \in \mathbb{R}^{3m_c} : \sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, i = 1, \ldots, m_c, \boldsymbol{\lambda}_C \geq \boldsymbol{o}\}.$$

Now we are going to solve problem (9) using dual formulation, dual function and KKT conditions (again can be found in [4]).
At first we induce first Karush-Kuhn-Tucker condition (the minimizer $\bar{\boldsymbol{u}}$ of function $L(\boldsymbol{u}, .)$ satisfy state of stationary point - we put part of gradient of L corresponding to derivation with respect to components of $\boldsymbol{u}$ equal to zero)

$$\nabla_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{\lambda}) = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f} + \tilde{\boldsymbol{B}}^T \boldsymbol{\lambda} = \boldsymbol{o} \quad \Rightarrow \quad \bar{\boldsymbol{u}} = \boldsymbol{K}^{-1} \left(\boldsymbol{f} - \tilde{\boldsymbol{B}}^T \boldsymbol{\lambda} \right) \tag{11}$$

and induct this into Lagrange function (10) and make some simplifications. We get

$$L(\bar{\boldsymbol{u}}, \boldsymbol{\lambda}) = L(\boldsymbol{K}^{-1} \left(\boldsymbol{f} - \tilde{\boldsymbol{B}}^T \boldsymbol{\lambda} \right), \boldsymbol{\lambda}) = -\frac{1}{2} \boldsymbol{\lambda}^T \tilde{\boldsymbol{B}} \boldsymbol{K}^{-1} \boldsymbol{T}^T \boldsymbol{\lambda} + \boldsymbol{\lambda}^T \tilde{\boldsymbol{B}} \boldsymbol{K}^{-1} \boldsymbol{f} - \frac{1}{2} \boldsymbol{f}^T \boldsymbol{K}^{-1} \boldsymbol{f}.$$

We get function of only one variable $\boldsymbol{\lambda}$. Our task is to find maximizer (see saddle-point problem (9)), so we can omit the constant term and change signs. Then $\bar{\boldsymbol{\lambda}}$ solves minimization problem

$$\bar{\boldsymbol{\lambda}} = \arg \min_{\boldsymbol{x} \in \Lambda} \Theta(\boldsymbol{x}), \quad \Theta(\boldsymbol{x}) := \frac{1}{2} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{b}, \tag{12}$$

where we denoted

$$\boldsymbol{A} := \tilde{\boldsymbol{B}} \boldsymbol{K}^{-1} \tilde{\boldsymbol{B}}^T, \quad \boldsymbol{b} := \tilde{\boldsymbol{B}} \boldsymbol{K}^{-1} \boldsymbol{f}.$$

After solving minimizing problem (12), the corresponding solution $\bar{\boldsymbol{u}}$ of primary problem (1) can be evaluated using (11).

Obviously $\boldsymbol{A} \in \mathbb{R}^{3m_c, 3m_c}$ is symmetric positive-definite matrix and problem (12) is the problem of minimizing strictly convex quadratic functions with separable quadratic constraints (QPQC) combined with bound constraints.

3. MPGP and projected Barzilai-Borwein algorithm

Now we are ready to introduce Modified proportioning with gradient projections algorithm (MPGP) (also included in [4, 3]), which convergence for QPQC was analysed in [5]. This active-set based algorithm solves problem on a free set using Conjugate gradient (CG) method (eventually do only *halfstep*) and finalize optimizing process on active set using gradient projection method with constant step-size.

Our modification lies in replacement of constant step-size in projection step by step-size used in recently developed Spectral Projected Gradient Method (SPG, see [2]). This method is based on projected version of Barzilai-Borwein algorithm (see [1]) combined with additional modified GLL line-search (see [6]). This additional line-search does not affect our algorithm, because it usually evokes leaving the border of feasible set, i.e. in our case it evokes extension of free set and restart CG method. So we use only first *spectral* projected step.

```
Choose $\boldsymbol{x}_0 \in \Omega, \alpha \in (0, 2\|\boldsymbol{A}\|^{-1}), \delta \in (0, 1/2\rangle$
$\alpha_{bb} := \alpha$
$k := 0$
while $\|\boldsymbol{x}_k - P(\boldsymbol{x}_k - \boldsymbol{g}_k)\| \geq \epsilon\|b\|$ do
    if $2\delta \boldsymbol{g}_k^T \boldsymbol{g}_k^P \leq \|\varphi(\boldsymbol{x}_k)\|^2$ then
        CG step or CG halfstep.
        make CG step to solve problem on free set.
        if this step means leaving $\Omega$, do only a half-step and restart CG.
        $k := k + 1$
    else
        Barzilai-Borwein gradient projection step.
        $\boldsymbol{x}_{k+1} := P(\boldsymbol{x}_k - \alpha_{bb}\boldsymbol{g}_k)$
        $\boldsymbol{s} := \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$
        $\alpha_{bb} := \boldsymbol{s}^T\boldsymbol{s}/\boldsymbol{s}^T\boldsymbol{A}\boldsymbol{s}$
        restart CG
        $k := k + 1$
    end if
end while
```

In our algorithm we use these notations

$$\boldsymbol{g}_k := \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \quad \tilde{\boldsymbol{g}}_k := \frac{1}{\alpha}\left(\boldsymbol{x}_k - P(\boldsymbol{x}_k - \alpha\boldsymbol{g}_k)\right), \quad \boldsymbol{g}_k^P := \varphi(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k),$$

φ and β are *free gradient* and *chopped gradient* defined in [4].

4. Numerical experiments

In our numerical experiment, we choose steel brick ($E = 2.10^5$, $\mu = 0.35$, $\rho = 7.85.10^{-2}$) and force $F = 5.10^3$.
For generating discretized problem we used MatSol library (see [8]).
We require accuracy $\epsilon = 10^{-4}$. We make two tests – in first we choose $\psi = 900$, in second $\psi = 15.10^3$.

For MPGP we used parameters $\delta := 1/2, \alpha := 1.95/\|A\|$. For SPG were used parameters $M := 1, \alpha_{\min} := 10^{-6}, \alpha_{\max} := 10^6, \gamma := 10^{-4}, \sigma_1 := 0.1, \sigma_2 := 0.9, \alpha_0 := 1$.

In Tables 1 and 2, N is discretization parameter. Every edge of brick was divided into N intervals, so the number of all FEM nodes in model is given by $(N+1)^3$. Because the problem is computed in 3D, the number of *primal* variables is $3(N+1)^3$. The number of FEM nodes in Γ_C is given by the number of nodes on bottom side

$\psi = 900$													
N	**primal**	**dual**	**SPG**			**MPGP**				**MPGP-BB**			
			it	GLL	f(x)	it	cg	half	proj	it	cg	half	proj
4	375	75	36	9	44	5176	0	1	5175	41	0	1	40
6	1029	147	45	20	64	2746	0	1	2745	57	0	1	56
8	2187	243	27	12	38	1236	0	1	1235	51	0	1	50
10	3993	363	33	15	47	661	0	1	660	40	0	1	39

Table 1: Test with small radius.

$\psi = 15000$													
N	**primal**	**dual**	**SPG**			**MPGP**				**MPGP-BB**			
			it	GLL	f(x)	it	cg	half	proj	it	cg	half	proj
4	375	75	1566	977	2542	43	33	9	1	43	33	9	1
6	1029	147	923	553	1475	48	29	18	1	48	29	18	1
8	2187	243	588	366	953	53	24	28	1	53	24	28	1
10	3993	363	1020	547	1566	101	40	46	15	73	27	40	6

Table 2: Test with larger radius.

of brick, i.e. $m_c = (N+1)^2$. So the number of all Lagrange multipliers is given by $3m_c = 3(N+1)^2$. This number is a dimension of *dual* problem.

For SPG algorithm we counted outer iterations and denoted this number by *it*. In the tables, one can find also number of all additional *GLL*-search iterations and a number of evaluations of cost function denoted by $f(x)$. For MPGP and MPGP-BB we denoted the number of all iterations by *it* and we counted also each type of iterations.

These tables show typical performance properties of algorithms.
If the radius of quadratic constraints is small (see Table 1), the type of the most of the iterations of MPGP and MPGP-BB is projection. Because MPGP-BB in projection uses similar rule for choosing step-size as SPG, the number of iterations of these two algorithms is similar. Choosing the constant step-size in MPGP is not so efficient.

If the radius of quadratic constraints is larger (see Table 2), MPGP and MPGP-BB are able to use more CG-iterations. That is the reason, why it is faster than non-monotone gradient descend method SPG.

5. Conclusions

Our numerical experiments predicate better performace of modified MPGP with BB step-size then original constant step-size for solving QPQC problems. But proof of convergence need be established, because the proof of convergence of original SPG in [2] is based on Armijo condition in GLL in additional line-search, but in our modification we did not use it.

Acknowledgements

This work was supported by grant GD103/09/H078 of Grant Agency of the Czech Republic. The author would like to express his thanks to Prof. RNDr. Zdeněk Dostál, DSc., Ing. Alexandros Markopoulos, Ph.D. and Ing. Tomáš Brzobohatý, Ph.D.

References

[1] Barzilai, J. and Borwein, J. M.: Two-point step size gradient methods. IMA Journal of Numerical Analysis **8** (1988), 141–148.

[2] Birgin, E., Martínez, J., and Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10** (2000), 1196–1211.

[3] Dostál, Z. et al.: A theoretically supported scalable tfeti algorithm for the solution of multibody 3d contact problems with friction. Computer Methods in Applied Mechanics and Engineering **205208** (2012), 110–120.

[4] Dostál, Z.: *Optimal quadratic programming algorithms: with applications to variational inequalities.* Springer Publishing Company, Incorporated, 2009, 1st edn.

[5] Dostál, Z. and Kozubek, T.: An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. Math. Program. **135** (2012), 195–220.

[6] Grippo, L., Lampariello, F., and Lucidi, S.: A nonmonotone line search technique for newtons method. SIAM Journal on Numerical Analysis **23** (1986), 707–716.

[7] Hlaváček, I., Haslinger, J., Nečas, J., and Lovíšek, J.: *Solution of variational inequalities in mechanics.* No. sv. 66 in Applied Mathematical Sciences Series, Springer Verlag, Berlin, 1988.

[8] Kozubek, T. et al.: Matsol – Matlab efficient solvers for problems in engineering. URL `http://www.am.vsb.cz/matsol`.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

GRAPHICS CARD AS A CHEAP SUPERCOMPUTER

Jan Přikryl

Institute of Information Theory and Automation
Pod Vodárenskou věží 2, CZ-18200 Praha 8, Czech Republic
prikryl@utia.cas.cz

Abstract

The current powerful graphics cards, providing stunning real-time visual effects for computer-based entertainment, have to accommodate powerful hardware components that are able to deliver the photo-realistic simulation to the end-user. Given the vast computing power of the graphics hardware, its producers very often offer a programming interface that makes it possible to use the computational resources of the graphics processors (GPU) to more general purposes. This step gave birth to the so-called GPGPU (general-purpose GPU) processors that – if programmed correctly – are able to achieve astonishing performance in floating point operations. In this paper we will briefly overview nVidia CUDA technology and we will demonstrate a process of developing a simple GPGPU application both in the native GPGPU style and in the add-ons for Matlab (Jacket and Parallel Toolbox).

1. Introduction

While ‘standard’ modern CPUs provide users with growing computational power, many scientists currently migrate towards general-purpose GPU (GPGPU) applications [3], using GPUs as parallel accelerators for memory-dense, floating-point intensive, applications. An accelerated linear algebra package exploiting the hybrid computation paradigm is currently under development [8] and GPGPU accelerators are becoming a tool of choice in many computationally-bound research tasks.

The concept of a GPGPU evolved from the needs of 3D-graphics-intensive applications that dictated the design of the processor such that most transistors were dedicated to the data processing, contrary to a regular CPU. The GPUs were then designed to be able to execute data-parallel algorithms on a stream of data, and consequently, the GPGPU processors are sometimes called ‘stream processors’ and are (not quite correctly) considered to be representatives of the SIMD processor architecture. The currently dominant architectures for GPGPU computing are the nVidia CUDA [5] and the AMD APP (formerly ATI Stream) [1].

The intrinsic parallel structure of a GPU (see Figure 1) allows a significant speedup in comparison to the multi-threaded single-processor architecture. The GPU programs are called *kernels* and the processor typically processes only one kernel at

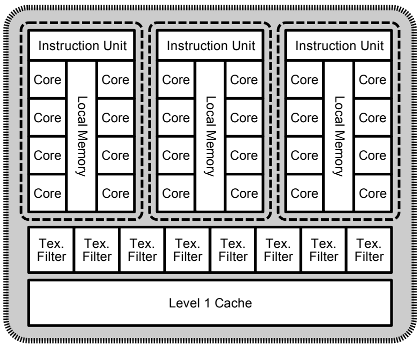


Figure 1: Thread processing cluster of a GTX280 GPU configured in 'compute mode'. The cluster contains three 8-core streaming multiprocessors, each of them has 16kB of fast local memory shared to all 8 cores. Adapted from [2].

a time by running it on several streaming multiprocessor units that form the so-called *thread block*. Every core in the GPU can access small but fast *shared* memory (local memory of a multiprocessor), large and slow *main* memory, constants can be placed to read-only and cached *constant* memory.

Although it is relatively easy to setup and perform basic operations with GPGPU even using the low-level programming (mostly ANSI C variants), it quickly becomes more complex when dealing with more demanding numerical problems – sometimes a small change in the order of instructions can have a dramatic impact on the overall performance. Additionally, special care must be taken when performing memory operations:

- due to the relatively slow memory transfer, data transfers between the host system and the GPU device shall be as few as possible, and shall be asynchronous if possible,
- improper kernel code design with respect to the operation on different memory types and ignoring memory access coalescing on the GPU device can cause a significant performance loss,
- shared memory is organised into banks and accessing elements not consecutively will cause a bank conflict.

The paper is composed as follows. The next section will introduce the covariance function, which is one of the bottlenecks of the modelling systems with Gaussian-process models. Different configurations of computation are described in Section 3, and the demonstration with a case study is described in Section 4. Conclusions are given at the end of the paper.

2. Modelling of dynamic systems with Gaussian processes

A Gaussian process [7] is a collection of random variables that have a joint multivariate Gaussian distribution. Assuming a relationship of the form $y = f(\mathbf{x})$

between an input $\mathbf{x}$ and an output y, we have $y_1, \ldots, y_n \sim \mathcal{N}(\mu(\mathbf{x}), \Sigma_{pq})$, where $\Sigma_{pq} = C(\mathbf{x}_p, \mathbf{x}_q)$ gives the covariance between the output points corresponding to the input vectors $\mathbf{x}_p$ and $\mathbf{x}_q$ and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate Gaussian distribution with the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

$C(\mathbf{x}_p, \mathbf{x}_q)$ can be any function having the property of generating a positive definite covariance matrix. A common choice is [7]

$$C(\mathbf{x}_p, \mathbf{x}_q) = v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_{dp} - x_{dq})^2\right] + \delta_{pq} v_0, \tag{1}$$

where $\boldsymbol{\Theta} = [w_1, \ldots, w_D, v_0, v_1]^T$ are the 'hyperparameters' of the covariance function, D is the dimension of the input regressors and $\delta_{pq} = 1$ if $p = q$ and 0 otherwise. The square exponential covariance function represents the smooth and continuous functional part and the constant covariance function represents the noise part, when it is presumed to be the white noise.

For a given problem, $\boldsymbol{\Theta}$ is identified using the data at hand and the function (1) is being evaluated many times before the process converges. This is one of the bottlenecks of the whole identification process of $\boldsymbol{\Theta}$ (although it is not the major one, unfortunately there are operations that can reach even $\mathcal{O}(n^3)$ [6], where n is the number of data used for identification).

3. Acceleration with various programming effort

The identification of a Gaussian-process model can be accomplished using a set of Matlab routines [4] that are an upgrade to the GPML toolbox [7] for machine learning with Gaussian processes. We will use this code base to demonstrate the process of upgrading the standard Matlab code to GPGPU code both with Jacket and Parallel Toolbox.

The code of the GPML toolbox relies heavily on linear algebra operations, which are considered to be fairly optimised even in the interpreted Matlab environment. We will therefore study the following scenarios which are ordered according to the working effort that has to be spent before actual computation:

Matlab on CPU only. We will use the native Matlab code on a multiple-core CPU. No changes are necessary.

Matlab on CPU using MEX file. We will use the original GPML MEX code on a multiple-core CPU. The publicly available ANSI C source code of a single MEX subroutine has to be compiled for the target architecture.

Matlab using Parallel Toolbox. We will use Mathworks' original interface to GPU and create our own replacement of the covariance code to compute the covariance matrix. This can be accomplished by simply retyping all GPU variables to `gpuArray`, carrying out the computation, and calling `gather` to transfer the covariance matrix back to the CPU.

Matlab using Jacket. We will update the code of the covariance routine to use the Jacket library, a third-party extension for GPU acceleration of Matlab code (see `http://www.accelereyes.com/`). We will compute the covariance matrix on a GPU using small modifications of the original GPML code: (1) all variables that will reside on GPU have to be retyped to `gdouble`, (2) we have to check that CPU and GPU variables do not occur within a single formula, and (3) the resulting covariance matrix has to be fetched back to the CPU by retyping it back to `double`.

Matlab using GPU MEX file. We will use our own replacement of covariance code to compute the covariance matrix on GPU using a hand-optimised GPU kernel. The kernel has been written in ANSI C, manually debugged and hand optimised for performance. Then a MEX file has to be created that takes care of moving data to GPU, calling the kernel and copying the result back to the CPU memory. The custom GPU kernel for the covariance function (1) relies on a coalesced memory access to move up to 16 elements of $\mathbf{x}_p$ and $\mathbf{x}_q$ to the shared memory of the thread block and computing an up to 16×16 sub-matrix of C in a single GPU kernel block. The main speedup is achieved by utilising as many kernels in a block as possible for a coalesced read of the elements from $\mathbf{x}$ into the shared memory, and by moving the elements of $\mathbf{\Theta}$ to the constant memory as they are used by all the invoked kernels.

In our tests, a standard PC equipped with an Intel i5/750 processor (42.56 GFLOPS in both single and double precision) and 4GB of RAM (bandwidth 17 GB/s) will be used. The GPU was nVidia GTX 275, which includes 240 processor cores (1010 GFLOPS in single, but only 124 GFLOPS in double precision; the double-precision performance is by design 8× lower than that of a single-precision computation [2]) running at 1404 MHz, with the memory interface running at 1134 MHz. The board contains 896 MB of GDDR3 memory (bandwidth 127 GB/s), every processor may use up to 16 kB of fast shared memory. All computations will be carried out in Matlab R2012a in double-precision arithmetics as most current GPUs have already an unlimited support for doubles.

4. Case study

The following example demonstrates the potential of the above described scenarios for accelerating the computation of covariance function (1). We will consider computing mutual covariances of an output sequence $y[k]$ generated by

$$y[k+1] = \frac{y[k]}{1+y^2[k]} + u^3[k] + \epsilon \tag{2}$$

where ϵ is the normally-distributed white noise with $\sigma = 0.05$ that contaminates the system response and the sampling time is one second. The input signal $u[k]$ is uniformly distributed noise in the interval $[-1.5, 1.5]$ sampled every 10-th step to prevent oscillations in the system.

The comparison of the computation times for the computation of one covariance matrix as a function of the length of the $y[k]$ sequence is given in Figure 2. We

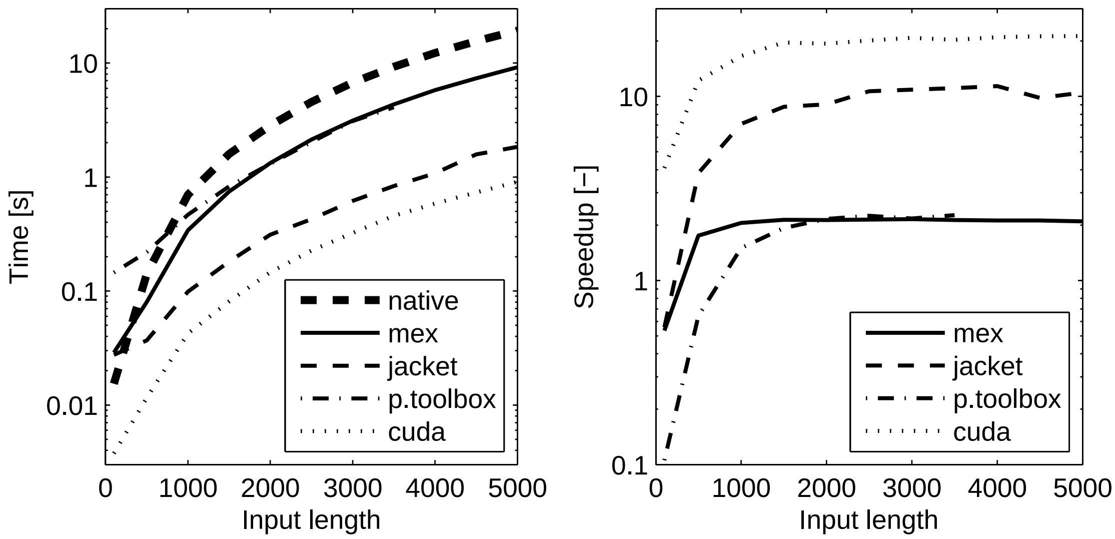


Figure 2: Computation times for the model identification versus input data dimension for different hardware configurations (left). Relative speed-ups of different hardware configurations with respect to the native CPU computation (right). Note that the GTX275 GPU has been used in the double-precision mode, where it reaches only 1/8 of the single-precision performance – hence, in a single-precision arithmetic, a GPU would be even more significantly faster than a CPU.

can see that for smaller dataset sizes below approximately 500 elements the native CPU computations may be faster than the MEX and GPGPU code, while for larger datasets the GPU-accelerated computations outperform the CPU by a factor up to 20.

The relatively poor performance for smaller input sizes is mainly due to the initialisation overhead required by the GPU and MEX code and due to the overhead of GPU data transfer (the overhead is almost 90 % of the total time for input length 100 and it is still about 30 % for input length 5000). The computation is faster on the host CPU unless this overhead can be eliminated or unless it represents a minor part of the whole computation time. Notice also the poor performance of the Parallel Toolbox code which is due to poor implementation of `repmat()` on GPU and the fact that probably due to memory leaks in the GPU code the maximum length of the input vector was 3500.

5. Conclusions

This paper provides a computational-time demonstration of how general-purpose graphics processors (GPGPU) may be used to accelerate a computation by offloading the most computational intensive parts of the code to the graphics hardware. The demonstration was performed from the user point of view to test the usability of different computational platforms for the Gaussian process model identification and simulation. We can see that using a GPGPU computing architecture has its benefits, even in cases when the user is no expert in GPU computing: using the commercial Jacket library for Matlab or possibly Matlab Parallel Toolbox makes it possible to achieve speedup over 10 with virtually no or moderate Matlab code changes. The best results are of course provided by the hand-crafted code that has been optimised for the GPU. However, producing such a code requires a significant programming effort.

Source codes of all tested scenarios can be downloaded from `http://staff.utia.cas.cz/prikryl/panm16.zip`.

Acknowledgments

This work has been supported by the Technology Agency of the Czech Republic under project no. TA01030603 and in part by a bilateral project between Slovenia and Czech Republic no. MEB091015.

References

[1] Advanced Micro Devices, Inc., Sunnyvale, CA: *AMD Accelerated Parallel Processing OpenCL Programming Guide*, 2011.

[2] NVIDIA GeForce® GTX 200 GPU Architectural Overview. TB-04044-001, nVidia, 2008. URL `http://www.nvidia.com/object/io_1213615494642.html`.

[3] Kirk, D. B. and Hwu, W. W.: *Programming Massively Parallel Processors A Hands-on Approach.* Morgan Kaufmann, 2010, 1 edn.

[4] Kocijan, J., Ažman, K., and Grancharova, A.: The concept for Gaussian process model based system identification toolbox. In: *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech).* Rousse, Bulgaria, 2007 pp. IIIA.23–1–IIIA.23–6.

[5] NVIDIA Corporation, Santa Clara, CA: *CUDA Programming Guide Version 2.3.1*, 2009.

[6] Quiňonero-Candela, J. and Rasmussen, C. E.: A unifying view of sparse approximate Gaussian process regression. J. Mach. Learn. Res. **6** (2005), 1939–1959.

[7] Rasmussen, C.E. and Williams, C.K.I.: *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA, 2006.

[8] Tomov, S., Dongarra, J., and Baboulin, M.: Towards dense linear algebra for hybrid GPU accelerated manycore systems. Parallel Comput. **36** (2010), 232–240.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

FOURIER ANALYSIS OF ITERATIVE AGGREGATION-DISAGGREGATION METHODS FOR NEARLY CIRCULANT STOCHASTIC MATRICES

Ivana Pultarová
Czech Technical University in Prague, Faculty of Civil Engineering
Thákurova 7, 166 29 Praha 6, Czech Republic
ivana@mat.fsv.cvut.cz

Abstract

We introduce a new way of the analysis of iterative aggregation-disaggregation methods for computing stationary probability distribution vectors of stochastic matrices. This new approach is based on the Fourier transform of the error propagation matrix. Exact formula for its spectrum can be obtained if the stochastic matrix is circulant. Some examples are presented.

1. Introduction

Iterative aggregation-disaggregation (IAD) methods are a popular tool for numerical solution of stationary probability distribution vectors of stochastic matrices: they search for a sufficiently good approximation of x fulfilling

$$Bx = x, \qquad e^T x = 1, \tag{1}$$

where B is an irreducible column stochastic matrix and e is a vector of all ones. B is column stochastic if $B \geq 0$ and $e^T B = e^T$. It is well known that the solution x exists, is unique and positive [12].

The IAD methods work in a multilevel fashion. A set of aggregation groups of unknowns is chosen. Each group represents one unknown on the coarse level. A solution of the coarse problem is used for improving the approximate solution of the original problem on the fine level. The idea is similar to the classical algebraic multigrid (AMG) used for the solution of symmetric positive definite (SPD) problems [1, 2, 3, 4, 5, 7, 13]. The main difference is caused by the nonsymmetry of stochastic matrices. While for the AMG methods the estimates in a corresponding energy norm are utilized, the theoretical justifying the convergence of the IAD methods exploit completely different approaches. Unfortunately, there are no convergence conditions for general IAD methods and for general stochastic matrices. In spite of this, there are many numerical experiments confirming good efficiency of various

IAD methods. The aim of this paper is to provide a theoretical background for some observations made e.g. in [1, 3, 4, 5].

Let $[B]_{rs}$ denote the element of B in the row r and column s, similarly $[x]_r$ is the rth element of vector x. If B is nonsymmetric, the preferable algorithm of aggregation of unknowns into aggregation groups is according to their strong connection [1, 3, 4, 5]: the unknowns $[x]_r$ and $[x]_s$ are strongly connected if $[B]_{rs} + [B]_{sr} \gg 0$. Then the IAD methods are reported to converge fast. But there is no theoretical background given in the literature. In this paper we consider a special $N \times N$ stochastic matrix B, where

$$[B]_{rs} = 1 \quad \text{if} \quad (r - s - 1) \bmod N = 0, \quad \text{and} \quad [B]_{rs} = 0 \quad \text{otherwise}. \tag{2}$$

Adding small perturbations to B gives rise to typical examples of slowly mixing stochastic matrices for which the stationary iterative methods converge slowly. Such matrices appear for example in queuing network applications. At the same time B is a circulant matrix. While the stationary probability distribution of B is $x = e/N$, the solution for perturbations of B are not known a priori. But from the continuity, similar quality is achieved for perturbations of B. Motivated by the Fourier transform of AMG operators for circulant and Toeplitz SPD matrices [2], we use the Fourier transform for the IAD methods and for circulant matrices. A scope of this paper allows us to consider only two-level IAD methods. Our particular goal is to find the optimal parameters in the IAD methods for B defined by (2).

The paper is organized as follows. In the next section the IAD methods and the error propagation formula are recalled. In Section 3 the Fourier transform is used for the error propagation matrix and its spectrum is computed. The optimal IAD parameters are computed in Section 4. A short discussion concludes the paper.

2. Two-level IAD methods

Let us assume an irreducible $N \times N$ stochastic matrix B. Let pairwise disjoint aggregation groups $G_1, \ldots, G_n$ be chosen, $\cup_{k=1}^{n} G_k = \{1, \ldots, N\}$. Then a reduction matrix $R \in \mathcal{R}^{n \times N}$ is given by

$$\begin{aligned} [R]_{ij} &= 1 \quad \text{if} \quad j \in G_i, \\ &= 0 \quad \text{otherwise}. \end{aligned}$$

A prolongation matrix $S(y) \in \mathcal{R}^{N \times n}$ is defined for any positive vector $y \in \mathcal{R}^N$ by

$$\begin{aligned} [S(y)]_{ij} &= \frac{y_i}{\sum_{k \in G_j} y_k} \quad \text{if} \quad i \in G_j, \\ &= 0 \quad \text{otherwise}. \end{aligned}$$

Matrix $B_1 = RBS(y)$ is an aggregated matrix corresponding to B and y. Of course, $P(y) = S(y)R$ is a projection.

On the fine level, μ steps of some stationary iteration (we call it a basic iteration) with matrix T are performed. We use Richardson iteration with $T = \alpha B + (1-\alpha)I$, where I is the identity matrix and $\alpha \in (0, 1\rangle$. A solution of the coarse problem with matrix B_1 is carried out exactly. One cycle of the IAD method is as follows.

One cycle of the IAD method: input $x^m > 0$; output x^{m+1}.

1. set $B_1 := RBS(x^m)$ and solve $B_1 z = z$, $e^T z = 1$ (coarse step)
2. $y := S(x^m)z$ (prolongation)
3. $x^{m+1} := T^\mu y$ (basic iterations)

It can be easily shown that the exact solution x is a fixed point of this computing process. Moreover, the error of the approximation x^{m+1} is

$$x^{m+1} - x = J(x^m)(x^m - x)$$

[6], where

$$J(x^m) = T^\mu (I - P(x^m)(B - xe^T))^{-1}(I - P(x^m)). \tag{3}$$

Since spectral radii $\rho(J(x^m))$ are greater than one in general, we can study the asymptotic (local) convergence properties by substituting the exact solution into (3) instead of x^m and computing the spectral radii of $J(x)$. We say that the IAD method is locally convergent if there exists a neighborhood U of x such that for any $x^0 \in U$, the IAD method yields a convergent sequence with a limit x. A sufficient condition for the local convergence is of course $\rho(J(x)) < 1$.

3. Fourier transform of the error propagation formula

The spectral analysis of the AMG methods for circulant and Toeplitz matrices is based on the Fourier transform of the error propagation operator [2]. We apply this idea to the IAD methods and compute spectra of matrices $J(x)$ given by (3) if the stochastic matrix B is circulant. As the first type we consider B defined by (2). According to Theorem 1 a spectrum of $J(x)$ can be expressed exactly which helps us to see what are the values of μ and α resulting in the smallest $\rho(J(x))$. Adding small perturbations to B does not change the convergence rates of the IAD method significantly. Such matrices represent a kind of slowly mixing Markov chains [12]. For the sake of simplicity we consider $n = N/2$ and $G_k = \{2k-1, 2k\}$, $k = 1, \dots, n$, which corresponds to the aggregation of unknowns according to their strong connections.

Let us denote the $N \times N$ Fourier matrix by F_N, where

$$[F_N]_{rs} = \frac{1}{\sqrt{N}} \mathrm{e}^{-2\pi(r-1)(s-1)\mathrm{i}/N}.$$

The superscript ^{H} indicates the adjoint matrix.

Theorem 1. *Let B be defined by (2). Assume the IAD method with the basic iteration matrix $T = \alpha B + (1-\alpha)I$, $\alpha \in (0,1\rangle$, and with μ steps of basic iterations in each cycle. Let the aggregation groups be $G_k = \{2k-1, 2k\}$, $k = 1,\ldots,n$, $n = N/2$. Then the spectrum of the error propagation matrix $J(x)$ is*

$$\sigma(J(x)) = \{0, v_0, v_1, \ldots, v_{n-1}\},$$

where

$$v_k = \frac{1}{2}\left(\left(1 - \mathrm{e}^{2\pi k\mathrm{i}/N}\right)\left(1 - \alpha + \alpha\mathrm{e}^{-2\pi k\mathrm{i}/N}\right)^{\mu} + \left(1 + \mathrm{e}^{2\pi k\mathrm{i}/N}\right)\left(1 - \alpha - \alpha\mathrm{e}^{-2\pi k\mathrm{i}/N}\right)^{\mu}\right). \tag{4}$$

Proof. The proof aims to compute the spectra of $F_N^H J(x) F_N$. We show only two crucial points of the proof. The first one is the well known formula

$$B = F_N D F_N^H,$$

where D is diagonal and $[D]_{rr} = \mathrm{e}^{2\pi(r-1)\mathrm{i}/N}$. The second one is that for the exact solution $x = e/N$

$$P(x) = \frac{1}{2}R^T R = \frac{1}{4}F_n \begin{pmatrix} \tilde{D}_1 & 0 \\ 0 & \tilde{D}_2 \end{pmatrix} \begin{pmatrix} I & I \\ I & I \end{pmatrix} \begin{pmatrix} \tilde{D}_1^H & 0 \\ 0 & \tilde{D}_2^H \end{pmatrix} F_n^H, \tag{5}$$

where the matrices $\tilde{D}_1$ and $\tilde{D}_2$ are diagonal and $[\tilde{D}_1]_{rr} = 1 + \mathrm{e}^{2\pi(r-1)\mathrm{i}/N}$ and $[\tilde{D}_2]_{rr} = 1 - \mathrm{e}^{2\pi(r-1)\mathrm{i}/N}$, $r = 1,\ldots,n$. Find more about this technique in [2]. □

Though the spectrum of $J(x)$ is computable for B defined by (2), it is not straightforward to simplify the term (4) for an arbitrary μ.

4. Optimal parameters μ and α

Under the assumptions of Theorem 1 let $\mu \in \{1,2,3\}$. Let the spectra of the corresponding matrices $J(x)$ be σ_1, σ_2, σ_3 and the spectral radii ρ_1, ρ_2, ρ_3. Then

$$\begin{aligned} \sigma_1 &= \{0, 1-2\alpha\}, \\ \sigma_2 &= \{0\} \cup \left(\alpha^2 M + (1-\alpha)(1-3\alpha)\right), \\ \sigma_3 &= \{0\} \cup \left((3\alpha^2 - 4\alpha^3)M + (1-\alpha)^2(1-4\alpha)\right), \end{aligned}$$

where $M = \{\mathrm{e}^{-4\pi k\mathrm{i}/N}\}_{k=0}^{n-1}$.

For $\alpha \approx 1$ we have $\rho_3 < \rho_1 < \rho_2$, see also Figure 1. Thus in case of B nearly of the type (2) and of the aggregation groups with two elements strongly connected, and for $T = \alpha B + (1-\alpha)I$, $\alpha \approx 1$, the most advantageous number of basic iterations (among $1,2,3$) in every IAD cycle is $\mu = 3$.

Theorem 1 also allows to find the best parameter α if μ is given. Note that it does not depend on N. For example, for $\mu = 1$ the best is $\alpha = 1/2$ which leads to $\rho_1 = 0$. For $\mu = 2$ the best spectral radius is $\rho_2 = 1/9$ for

$$\alpha = \arg\min_{\alpha\in(0,1\rangle} \max\left(|(1-\alpha)(1-3\alpha) + \alpha^2|, |(1-\alpha)(1-3\alpha) - \alpha^2|\right) = 1/3.$$

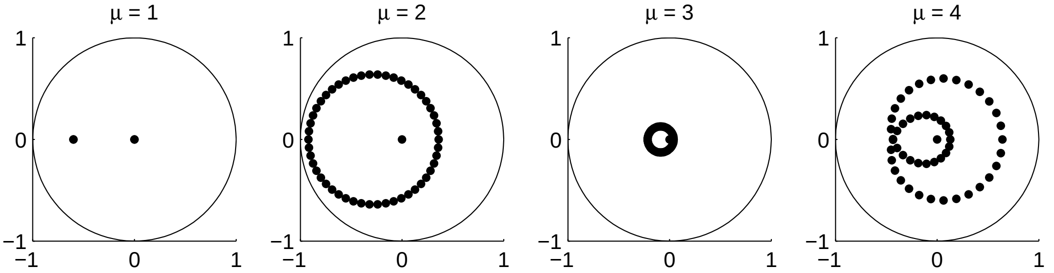


Figure 1: Eigenvalues of $J(x)$ for B defined by (2), $x = e/N$, $N = 100$, aggregation groups $G_k = \{2k-1, 2k\}$, $k = 1, \ldots, N/2$, parameters $\alpha = 0.8$ and $\mu \in \{1, 2, 3, 4\}$. The solid line is a reference unit cycle.

5. Discussion

We contribute to the theory of the IAD methods. Our results are applicable to the theory of the AMG for nonsymmetric problems as well. The introduced approach is based on the Fourier transform.

The introduced analysis can be generalized in several directions. More than two elements in each aggregation group can be considered. Then instead of the 2×2 block form in (5) we get an $m \times m$ block form if m elements are contained in every aggregation group. Also block-circulant matrices can be studied [2].

We would like to emphasize that the local convergence of the IAD methods is not necessarily obtained in general [8]. There are several examples where the spectral radius of $J(x)$ can be arbitrarily large [10]. It was shown that even B in the form (2) can yield the spectral radius of $J(x)$ arbitrarily close to two [9]. These examples should be understood and avoided in the real life computation.

A promising utilization of our approach is in the theory of multi-level IAD methods. Presently we are not able to find any exact criteria for the local convergence of the IAD methods with more than two levels. Our new approach could simplify the involved formulae [11] and help us to find the optimal IAD parameters for at least some special stochastic matrices.

Acknowledgement

The author thanks to the anonymous referee for his/her careful reading the manuscript and for his/her comments.

References

[1] Bolten, M., Brandt, A., Brannick, J., Frommer A., Kahl K., and Livshits I.: A bootstrap algebraic multilevel method for Markov chains. SIAM J. Sci. Comput. **33** (2011), 3425–3446.

[2] Bolten, M., Donatelli, M., and Huckle, T.: Aggregation-based multigrid methods for circulant and Toeplitz matrices. Preprint BUW-IMACM 12/10, Bergische Universität Wuppertal, 2012.

[3] Brezina, M., Manteuffel, T., McCormick, S., Ruge, J., and Sanders, G.: Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems. SIAM J. Sci. Comput. **32** (2010), 14–39.

[4] De Sterck, H., Manteuffel, T. A., McCormick, S. F., Miller, K., Ruge, J., and Sanders, G.: Algebraic multigrid for Markov chains. SIAM J. Sci. Comput. **32** (2010), 544–562.

[5] De Sterck, H., Miller, K., Treister, E., and Yavneh, I.: Fast multilevel methods for Markov chains. Numer. Linear Algebra Appl. **18** (2011), 961–980.

[6] Marek, I. and Mayer, P.: Convergence analysis of an iterative aggregation/disaggregation method for computing stationary probability vectors of stochastic matrices. Numer. Linear Algebra Appl. **5** (1998), 253–274.

[7] Notay, Y.: Algebraic analysis of two-grid methods: The nonsymmetric case. Numer. Linear Algebra Appl. **17** (2010), 73–96.

[8] Pultarová, I.: Necessary and sufficient local convergence condition of one class of iterative aggregation-disaggregation methods. Numer. Linear Algebra Appl. **15** (2008), 339–354.

[9] Pultarová, I.: Ordering of matrices for iterative aggregation-disaggregation methods. Lecture Notes in Control and Inform. Sci. **389** (2009), 379–386.

[10] Pultarová, I., Marek, I.: Physiology and pathology of iterative aggregation-disaggregation methods. Numer. Linear Algebra Appl. **18** (2011), 1051–1065.

[11] Pultarová, I.: Error propagation formula of multi-level iterative aggregation-disaggregation methods for non-symmetric problems. Electron. J. Linear Algebra **25** (2012), 9–21.

[12] Stewart, W. J.: *Introduction to the numerical solution of Markov chains.* Princeton University Press, Princeton, 1994.

[13] Vaněk, P., Brezina, M., and Mandel, J.: Convergence of algebraic multigrid based on smoothed aggregation. Computing **56** (1998), 179–196.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

OPTIMIZATION OF PLUNGER CAVITY

Petr Salač

Technical University of Liberec
Studentská 2, 461 17 Liberec, Czech Republic
petr.salac@tul.cz

Abstract

In the contribution we present a problem of shape optimization of the cooling cavity of a plunger that is used in the forming process in the glass industry. A rotationally symmetric system of the mould, the glass piece, the plunger and the plunger cavity is considered. The state problem is given as a stationary heat conduction process. The system includes a heat source representing the glass piece that is cooled from inside by water flowing through the plunger cavity and from outside by the environment surrounding the mould. The design variable is the shape of the inner surface of the plunger cavity.

The cost functional is defined as the squared L_r^2 norm of the difference between a prescribed constant and the temperature on the outward boundary of the plunger.

1. Introduction

This work deals with the optimal design of the shape of a plunger cavity that controls the cooling of a glass piece during the manufacturing process. The aim of the optimization is to find such a shape of the inner plunger cavity that allows for cooling in such a way that a constant distribution of the temperature is achieved across the surface of the moulding device at the moment of separation of the plunger from the moulded piece.

2. Formulation of the problem

We rotate the system to the horizontal position to be able to describe the optimized plunger cavity surface by a function of one variable.

We define

$$F_2^e(x) = \begin{cases} 0 & \text{for} \quad x \in [0,\, x_2^e] \\ f_2^e(x) & \text{for} \quad x \in [x_2^e,\, 1] \end{cases} \quad , \tag{1}$$

where $x_2^e \in [s_{\min},\, 1]$ ($s_{\min} > 0$ is a fixed constant given by the minimal thickness of the plunger wall), $f_2^e \in C^{(0),1}([x_2^e,\, 1])$, $f_2^e(x_2^e) = 0$ and $0 \leq f_2^e(x) \leq f_1(x) - s_{\min}$, $|f_2^{e\prime}(x)| < C_D$ for $x \in]x_2^e, 1]$, where f_1 is a fixed function. Further we assume that $a \leq f_2^e(x) - s_2$ for $x \in [x_3^e, 1]$, where $a > 0$ represents the radius of a supply tube and

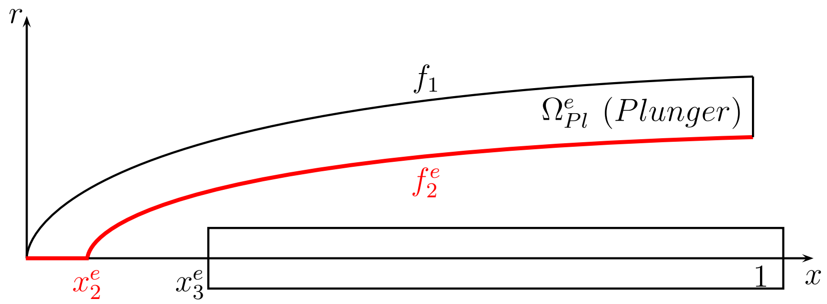


Figure 1: Scheme of the plunger with the optimized part of the boundary.

$s_2 > 0$ is the minimal admissible split width between the inner wall of the plunger cavity and the water supply tube, and $x_3^e \in]x_2, 1]$ is the deepness of the insertion of the tube.

Further we define the set of admissible functions as

$$U^e_{ad} = \{\ F^e_2(x) \in C^{(0),1}([0,1])\ ;\ F^e_2(x) = \begin{cases} 0 & \text{for} \quad x \in [0,\ x^e_2] \\ f^e_2(x) & \text{for} \quad x \in [x^e_2,\ 1] \end{cases} \quad ,$$
$$x^e_2 \in [s_{\min},\ 1],\ s_{\min} > 0,\ f^e_2 \in C^{(0),1}([x^e_2,\ 1]),\ f^e_2(x^e_2) = 0,$$
$$0 \le f^e_2(x) \le f_1(x) - s_{\min},\ |f^{e\prime}_2(x)| < C_D \text{ for } x \in]x^e_2, 1],$$
$$f_1 \text{ given},\ a \le f^e_2(x) - s_2 \text{ for } x \in [x^e_3, 1],\ a > 0,\ s_2 > 0,\ x^e_3 \in]x_2, 1]\}\,,$$

where the function F^e_2 describes the technological constraint for the inner cavity surface.

We assume the region Ω^e_{Pl} that depends on the design function $F^e_2(x)$, and that is defined by the formula

$$\Omega^e_{Pl} = \{(x,\ r) \in R^2\ ;\ F^e_2(x) < r < f_1(x), \quad \text{for} \quad x \in [0,\ 1]\}\ .$$

Denote by Θ the set of all admissible regions $\Omega^e_{Pl} \subset R^2$, i.e., regions characterized by $F^e_2 \in U^e_{ad}$. Let us define the convergence on the set Θ. Since each Ω^e_{Pl} is uniquely related to F^e_2, we can say that a sequence $\Omega^n_{Pl} \in \Theta$ converges to a region $\Omega^e_{Pl} \in \Theta$ if and only if the sequence of functions ${}^nF^e_2(x)$ converges uniformly in $[0,1]$ to the function $F^e_2(x)$ that defines Ω^e_{Pl}.

Let us consider the union of four planar regions $\Omega = \Omega_{Mo} \cup \Omega_{Gl} \cup \Omega^e_{Pl} \cup \Omega^e_{Ca}$ that represents the planar cross section of the mould, the glass piece, the plunger and the cooling channel of the plunger (see Figure 2).

Furthermore, we denote by Γ_1 the boundary between the plunger Ω^e_{Pl} and the moulded piece Ω_{Gl} and Γ^e_2 the boundary between the plunger Ω^e_{Pl} and the plunger cavity Ω^e_{Ca}. We denote by Γ_3 the part of the boundary connecting the mould, the moulded piece and the plunger with the presser, by Γ_4 a part of the axis of symmetry (see Figure 2), by Γ_5 the part of the boundary formed by the tube. Γ_6 is the notation for the part of the boundary between the moulded piece Ω_{Gl} and the mould Ω_{Mo}

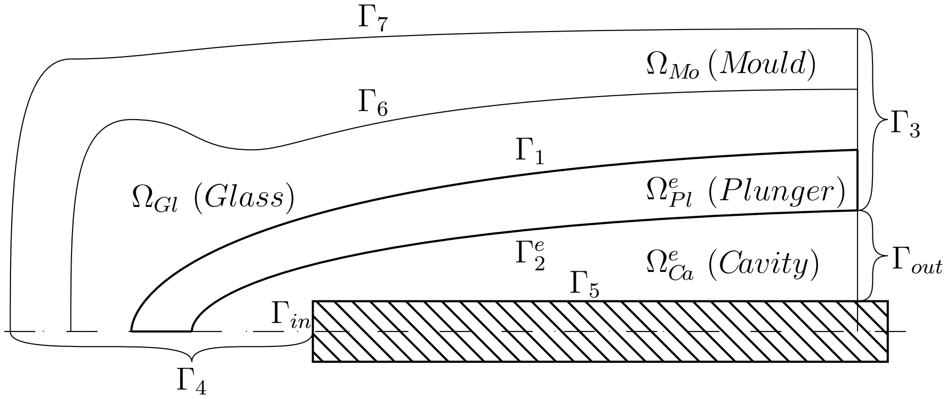


Figure 2: Scheme of the mould, the glass piece, the plunger, the cavity of plunger and the supply tube.

and Γ_7 is the outward boundary of the mould, which is surrounded by an external environment. Γ_{in} denotes the part of the boundary, where the cooling water comes into the cooling channel of the plunger, and Γ_{out} stands for the part of the boundary, where the water exits the channel.
In the three dimensional region G_{Ca}^e, which is created by the rotation of Ω_{Ca}^e around the x axis, we assume an incompressible potential water flow that is rotationally symmetric with respect to the x axis. We split the boundary ∂G_{Ca}^e into the union of four parts as

$$\partial G_{Ca}^e = \Gamma_2^{3D} \cup \Gamma_5^{3D} \cup \Gamma_{in}^{3D} \cup \Gamma_{out}^{3D} \ , \tag{2}$$

where Γ_2^{3D}, Γ_5^{3D}, Γ_{in}^{3D}, and Γ_{out}^{3D} denote the respective parts of the boundary of ∂G_{Ca}^e created by the rotation of Γ_2^e, Γ_5, Γ_{in}, and Γ_{out} around the x axis.

The potential Φ describing the water flow is given as a solution of the Neumann problem

$$\Delta\Phi = 0 \quad \text{in} \qquad G_{Ca}^e \ , \tag{3}$$

$$\frac{\partial\Phi}{\partial n} = g \quad \text{on} \qquad \partial G_{Ca}^e \ , \tag{4}$$

where $g \in L^2(\partial G_{Ca}^e)$, representing the normal component of the water flow velocity at the entrance to and the exit from the plunger cavity, is in the form

$$g = \begin{cases} 0 & \text{on} \quad \Gamma_2^{3D} \cup \Gamma_5^{3D} \ , \\ h_{velo}^{in} & \text{on} \quad \Gamma_{in}^{3D} \ , \\ h_{velo}^{out} & \text{on} \quad \Gamma_{out}^{3D} \ , \end{cases} \tag{5}$$

h_{velo}^{in} is the normal velocity at the entrance Γ_{in}^{3D} ($h_{velo}^{in} < 0$) and h_{velo}^{out} is the normal velocity at the exit Γ_{out}^{3D}. Further we assume

$$\int_{\Gamma_{in}^{3D} \cup \Gamma_{out}^{3D}} g \, dS = 0 \ . \tag{6}$$

The variational formulation for the potential function has the form:
We look for the function $\Phi \in H^1(G^e_{Ca})$ such that

$$\int_{G^e_{Ca}} \left(\frac{\partial \Phi}{\partial x_1}\frac{\partial \varphi}{\partial x_1} + \frac{\partial \Phi}{\partial x_2}\frac{\partial \varphi}{\partial x_2} + \frac{\partial \Phi}{\partial x_3}\frac{\partial \varphi}{\partial x_3} \right) dV = \int_{\Gamma^{3D}_{in} \cup \Gamma^{3D}_{out}} g\varphi \, dS \quad \forall \varphi \in H^1(G^e_{Ca}) \; . \tag{7}$$

In the cavity G^e_{Ca}, the flowing water velocity field $\boldsymbol{u} = (u_1, u_2, u_3)$ is given as

$$\boldsymbol{u} = \operatorname{grad} \Phi \; . \tag{8}$$

Theorem 1. *(existence and uniqueness of the velocity field)* Under the assumption (6) there exists a unique velocity field of the form (8) satisfying

$$|||\boldsymbol{u}|||_{L^2(G^e_{Ca})} \le c \left(\|h^{in}_{velo}\|_{L^2(\Gamma^{3D}_{in})} + \|h^{out}_{velo}\|_{L^2(\Gamma^{3D}_{out})} \right) \; , \tag{9}$$

where

$$|||\boldsymbol{u}|||_{L^2(G^e_{Ca})} = \left\| \sqrt{u_1^2 + u_2^2 + u_3^2} \right\|_{L^2(G^e_{Ca})} \; . \tag{10}$$

Proof. See [3]. □

Let us consider the union of four regions $G = G_{Mo} \cup G_{Gl} \cup G^e_{Pl} \cup G^e_{Ca}$ that is created by the rotation of the union $\Omega = \Omega_{Mo} \cup \Omega_{Gl} \cup \Omega^e_{Pl} \cup \Omega^e_{Ca}$ around the x axis. We split ϑ, the searched function representing the distribution of the temperature, into four functions

$$\vartheta = \vartheta_0 + \vartheta_1 + \vartheta_2 + \vartheta_3 \; , \tag{11}$$

where

$$\vartheta_i = \begin{cases} \vartheta|_{G_i} & \text{in } \; G_i \\ 0 & \text{in } \; G \setminus G_i \end{cases} \qquad \text{for} \quad i = 0,\, 1,\, 2,\, 3 \; , \tag{12}$$

$(G_0 \equiv G^e_{Pl},\ G_1 \equiv G_{Gl},\ G_2 \equiv G^e_{Ca},\ G_3 \equiv G_{Mo})$.
Further we denote by $\vartheta_i|_{\Gamma^{3D}_j}$ the trace of the solution ϑ_i on the boundary Γ^{3D}_j if Γ^{3D}_j is a part of the boundary of G_i for $i = 0, 1, 2, 3,\ j = 1, 2, 3, 4, 5, 6, 7, 8, 9$ $(\Gamma^{3D}_8 = \Gamma^{3D}_{in},\ \Gamma^{3D}_9 = \Gamma^{3D}_{out})$.

By virtue of the rotational symmetry of both the state problem and the function ϑ, the state problem can be formulated variationally in two dimensions. We define the operators

$$\mathrm{Energy}^{velo}_{\Omega}(\vartheta,\, \boldsymbol{w},\, \psi) = c_v \varrho_2 \int_{\Omega^e_{Ca}} \left(\frac{\partial \vartheta_2}{\partial x} w_1 + \frac{\partial \vartheta_2}{\partial r} w_2 \right) \psi r \, d\Omega \; , \tag{13}$$

$$\begin{aligned} \mathrm{Energy}^{cond}_{\Omega}(\vartheta,\, \psi) = \; & k_0 \int_{\Omega^e_{Pl}} \left(\frac{\partial \vartheta_0}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial \vartheta_0}{\partial r}\frac{\partial \psi}{\partial r} \right) r \, d\Omega + \\ + \; & k_1 \int_{\Omega_{Gl}} \left(\frac{\partial \vartheta_1}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial \vartheta_1}{\partial r}\frac{\partial \psi}{\partial r} \right) r \, d\Omega + \end{aligned} \tag{14}$$

$$
\begin{aligned}
&+\ k_2 \int_{\Omega^e_{Ca}} \left(\frac{\partial \vartheta_2}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial \vartheta_2}{\partial r}\frac{\partial \psi}{\partial r} \right) r\, d\Omega + \\
&+\ k_3 \int_{\Omega_{Mo}} \left(\frac{\partial \vartheta_3}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial \vartheta_3}{\partial r}\frac{\partial \psi}{\partial r} \right) r\, d\Omega\ ,
\end{aligned}
$$

$$
\mathrm{Environment}_{\Omega}(\vartheta,\, \psi) = \int_{\Gamma_7} \alpha \vartheta_3|_{\Gamma_7} \psi r\, d\Gamma\ , \tag{15}
$$

$$
\mathrm{Source}_{\Omega}(\psi) = \varrho_1 \int_{\Omega_{Gl}} q\psi r\, d\Omega\ , \tag{16}
$$

$$
\mathrm{Coeff}_{\Omega}(\psi) = \int_{\Gamma_1} \beta_1 \psi r\, d\Gamma + \int_{\Gamma_6} \beta_6 \psi r\, d\Gamma + \int_{\Gamma_7} \alpha \vartheta_4 \psi r\, d\Gamma\ , \tag{17}
$$

where c_v is the specific heat capacity per unit volume, ϱ_1 is the density of glass, ϱ_2 is the density of water, w_1, w_2 are the water velocity field components expressed in cylindrical coordinates, k_0, k_1, k_2, k_3 are the coefficients of thermal conductivity, α is the coefficient of heat-transfer between the mould and the environment, ϑ_4 is the temperature of the environment, β_1, β_6 are the average power conversion of the unit volume of the glass body (see [4, page 128]) and q is the density of heat sources. Further we denote by

$$
\begin{aligned}
A_{\Omega}(\vartheta,\, \boldsymbol{w},\, \psi)\ &=\ \mathrm{Energy}^{velo}_{\Omega}(\vartheta,\, \boldsymbol{w},\, \psi) + \mathrm{Energy}^{cond}_{\Omega}(\vartheta,\, \psi) + \\
&+\ \mathrm{Environment}_{\Omega}(\vartheta,\, \psi)
\end{aligned} \tag{18}
$$

and

$$
F_{\Omega}(\psi) = \mathrm{Source}_{\Omega}(\psi) + \mathrm{Coeff}_{\Omega}(\psi)\ . \tag{19}
$$

We introduce the weighted Sobolev space $H^1_r(\Omega_i)$ (see [2]) provided with the norm

$$
\|v\|_{1,r,\Omega_i} = \left(\int_{\Omega_i} \left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + v^2 \right] r\, d\Omega \right)^{\frac{1}{2}} \quad i = 0, 1, 2, 3\ , \tag{20}
$$

($\Omega_0 \equiv \Omega^e_{Pl}$, $\Omega_1 \equiv \Omega_{Gl}$, $\Omega_2 \equiv \Omega^e_{Ca}$, $\Omega_3 \equiv \Omega_{Mo}$).
Further we introduce

$$
\begin{aligned}
\boldsymbol{H}(\Omega) = \{\ \vartheta;\ \vartheta \text{ defined in (12)},\ \vartheta_i \in H^1_r(\Omega_i) \text{ for any } i = 0, 1, 2, 3, \\
\vartheta_3|_{\Gamma_6} = \vartheta_1|_{\Gamma_6},\, \vartheta_1|_{\Gamma_1} = \vartheta_0|_{\Gamma_1},\, \vartheta_0|_{\Gamma^e_2} = \vartheta_2|_{\Gamma^e_2}\}\ ,
\end{aligned}
$$

where $\vartheta_i|_{\Gamma_j}$ denotes the trace of the function ϑ_i on the boundary Γ_j.
We define the norm in $\boldsymbol{H}(\Omega)$ as

$$
\|\vartheta\|_{\boldsymbol{H}} = \left(\|\vartheta_0\|^2_{1,r,\Omega_0} + \|\vartheta_1\|^2_{1,r,\Omega_1} + \|\vartheta_2\|^2_{1,r,\Omega_2} + \|\vartheta_3\|^2_{1,r,\Omega_3} \right)^{\frac{1}{2}}\ . \tag{21}
$$

Theorem 2. *The set $\boldsymbol{H}(\Omega)$ with the norm (21) is a Hilbert space.*

We denote by $\boldsymbol{H}^*(\Omega)$ the dual space to the space $\boldsymbol{H}(\Omega)$ with the norm

$$\|\psi\|_{\boldsymbol{H}^*} = \sup_{\varphi\neq 0} \frac{A_\Omega(\varphi,\, \boldsymbol{w},\, \psi)}{\|\varphi\|_{\boldsymbol{H}}} \ .$$

We define the sets

$$\Omega_H = \Omega \cup \Gamma_3 \cup \Gamma_{in} \cup \Gamma_{out}$$

and

$${}^e\mathcal{H}^{2D} = \{ v \in C^\infty(\Omega_H);\ v|_{\Gamma_3 \cup \Gamma_{in} \cup \Gamma_{out}} = 0\,\} \,.$$

Let $\boldsymbol{H}_{\boldsymbol{0}}(\Omega)$ be the closure of the set ${}^e\mathcal{H}^{2D}$ in $\boldsymbol{H}(\Omega)$.

We assume the existence of a function $\vartheta^e_\Gamma \in \boldsymbol{H}(\Omega)$ such that

$$\vartheta^e_\Gamma|_{\Gamma_{in}} = 288 \qquad \text{on } \Gamma_{in}, \tag{22}$$

$$\vartheta^e_\Gamma|_{\Gamma_{out}} = h^e_{out} \qquad \text{on } \Gamma_{out}, \tag{23}$$

$$\vartheta^e_\Gamma|_{\Gamma_3} = h_3 \qquad \text{on } \Gamma_3, \tag{24}$$

where $h_3 \in C(\Gamma_3)$ is a given function representing the steady temperature on the boundary Γ_3 (see Figure 2) and $h^e_{out} \in C(\Gamma_{out})$ is a given function representing the temperature distribution on the cavity output Γ_{out}.

We use the variational formulation of the energy equation to formulate

The State Problem:

We look for the function $\vartheta \equiv \vartheta(F^e_2) \in \boldsymbol{H}(\Omega)$ such that

$$A_\Omega(\vartheta,\, \boldsymbol{w}^{\boldsymbol{e}},\, \psi) \ = \ F_\Omega(\psi) \qquad \forall \psi \in \ \boldsymbol{H}_0(\Omega) \ , \tag{25}$$

$$\vartheta - \vartheta^e_\Gamma \in \ \boldsymbol{H}_0(\Omega) \ , \tag{26}$$

where $F^e_2 \in U^e_{ad}$ and $\boldsymbol{w}^{\boldsymbol{e}}$ is the corresponding flow pattern given as the gradient of the solution to (7).

Remark. The state problem is solved in two steps. First, the potential Φ of the water velocity is found as a solution of the problem (7) in the region G^e_{Ca}. The components of the velocity field $\boldsymbol{u}$ are computed from (8), transformed to cylindrical coordinates and substituted into (13). Then the distribution of the temperature ϑ in the whole system Ω is found as the solution of the state problem (25), (26).

Theorem 3. *(the existence and uniqueness of the solution of the state problem) The state problem (25), (26) has a unique solution $\vartheta(F^e_2)$ for each $F^e_2 \in U^e_{ad}$ and the associated flow pattern $\boldsymbol{w}^{\boldsymbol{e}}$ obtained as the gradient of the unique solution of (7), moreover, there exists a constant $C > 0$ such that*

$$\|\vartheta(F^e_2)\|_{\boldsymbol{H}} \leq C \|F_\Omega\|_{\boldsymbol{H}^*} \ . \tag{27}$$

Proof. It is sufficient to verify the assumptions of the Lax-Milgram Theorem (see [3]). □

We formulate the **problem of the optimal design for the plunger cavity shape:** We define the **cost functional** as

$$\mathcal{J}^S(F_2^e) = \|\vartheta(F_2^e)|_{\Gamma_1} - T_{\Gamma_1}\|^2_{0,r,\Gamma_1} \ , \tag{28}$$

where $\vartheta(F_2^e)|_{\Gamma_1}$ is the Γ_1-trace of the solution $\vartheta(F_2^e)$ of the state problem (25), (26) in the region Ω^e_{Pl}, where T_{Γ_1} is a given constant representing the known optimal temperature of the plunger surface. We look for the **optimal design** $F_{Opt} \in U^e_{ad}$ such that

$$\mathcal{J}^S(F_{Opt}) \leq \mathcal{J}^S(F_2^e) \qquad \forall\ F_2^e \in U^e_{ad} \ . \tag{29}$$

Theorem 4. *The optimal design problem (29) has at least one solution.*

Proof. We refer to Theorem 2.1 [1, page 29], see [3]. □

Remark. A sensitivity analysis can be performed on the basis of temperature evaluation along the boundary Γ_1. Let us introduce a homeomorphism between the outward plunger boundary Γ_1 and the plunger cavity boundary Γ_2^e defined by the gradient lines of the temperature field in the plunger. In the parts of Γ_1 where we need to decrease the temperature, we narrow "the wall" by moving the points of Γ_2^e along the gradient lines to locally achieve more intensive cooling. On the other hand, in places of Γ_1 where we need higher temperature, we increase "the wall thickness" to locally decrease the intensity of cooling. By the term "the wall thickness" we understand the length of the temperature gradient line that connects the related points of Γ_1 and Γ_2^e.

Acknowledgements

The paper was supported by the ESF project CZ.1.07/2.3.00/09.0155, "Constitution and improvement of a team for demanding technical computations on parallel computers at TU Liberec".

References

[1] Haslinger, J. and Neittaanmäki, P.: *Finite element approximation for optimal shape design: Theory and applications.* John Wiley & Sons Ltd., Chichester, 1988.

[2] Kufner, A.: *Weighted Sobolev spaces.* John Wiley & Sons, New York, 1985.

[3] Salač, P.: Optimal design of the cooling plunger cavity. Appl. Math. (accepted for publication).

[4] Šorin, S. N.: *Sdílení tepla.* SNTL, Praha, 1968.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

SMOOTH APPROXIMATION OF DATA WITH APPLICATIONS TO INTERPOLATING AND SMOOTHING

Karel Segeth

Institute of Mathematics, Academy of Sciences
Žitná 25, Prague 1, Czech Republic
segeth@math.cas.cz

Abstract

In the paper, we are concerned with some computational aspects of smooth approximation of data. This approach to approximation employs a (possibly infinite) linear combinations of smooth functions with coefficients obtained as the solution of a variational problem, where constraints represent the conditions of interpolating or smoothing. Some 1D numerical examples are presented.

1. Introduction

Smooth approximation [2] is an approach to data interpolation that employs the variational formulation of the problem in an inner product space, where constraints represent the interpolation conditions. A possible criterion is to minimize the integral of the squared magnitude of the interpolating function. A more sophisticated criterion is then to minimize, with some weights chosen, the integrals of the squared magnitude of some (or possibly all) derivatives of the interpolating function. We are thus concerned with the exact interpolation of the data at nodes and, at the same time, with the smoothness of the interpolating curve and its derivatives.

Smooth approximation has numerous applications as measurements of the values of a continuous function of one, two, or three independent variables are carried out in many branches of science and technology. We always get a finite number of function values measured at a finite number of points but we are interested also in intermediate values corresponding to other points. Apparently, except for the fixed constraints to be satisfied, the formulation of the problem of smooth approximation can vary and give the resulting interpolant of different smoothness. The cubic spline interpolation is known to be the approximation of this kind.

We confine ourselves to the case of 1D independent variable. We introduce the proper inner product space in Section 2. We formulate the problem and present the existence and uniqueness theorem in Section 3. In the next section, we show results of numerical experiments comparing the classical interpolation formulae and various basis systems for the smooth approximation. We finally sum up the results presented that show some properties of smooth approximation.

A paper containing all proofs has been prepared for a numerical analysis journal.

2. Notation

Let us consider the linear vector space $\widetilde{W}$ of complex functions g continuous together with their derivatives of all orders on the interval (a, b), which may be infinite. Let $\{B_l\}_{l=0}^{\infty}$ be a sequence of nonnegative numbers and let there be the smallest nonnegative integer L such that $B_L > 0$ while $B_l = 0$ for all $l < L$. For $g, h \in \widetilde{W}$ we construct the expression

$$(g, h)_L = \sum_{l=L}^{\infty} B_l \int_a^b g^{(l)}(x)[h^{(l)}(x)]^* \,\mathrm{d}x, \tag{1}$$

where * denotes complex conjugation. Let us further put

$$|g|_L^2 = \sum_{l=L}^{\infty} B_l \int_a^b |g^{(l)}(x)|^2 \,\mathrm{d}x. \tag{2}$$

If $B_0 > 0$ (i.e. $L = 0$) the expression $|g|_0 = \|g\|$ is the *norm* and $(g, h)_0 = (g, h)$ the *inner product*, and the set of all such functions forms a Hilbert space W corresponding to the sequence $\{B_l\}$.

If $L > 0$ then $|g|_L$ is a *seminorm* on W. We construct the quotient space W/P_{L-1} where the subspace $P_{L-1} \subset W$ is the space of polynomials of degree at most $L - 1$. Then $|g|_L$ is the norm and $(g, h)_L$ the inner product on the quotient space W/P_{L-1} of equivalence classes. The choice of the sequence $\{B_l\}$ defines weights of the individual derivatives in the expression (2) and guarantees the convergence of the series (2) as well.

Let us introduce some more notation to be able to formulate the problem of smooth approximation. Let us choose a system of functions $\{g_k\} \subset W$, $k = 1, 2, \ldots$, that is complete and orthogonal (with respect to the inner product (1)), i.e.,

$$(g_k, g_m)_L = 0 \ \text{ for } \ k \neq m, \quad (g_k, g_k)_L = |g_k|_L^2 > 0.$$

3. Problem of smooth interpolation

Let us have N (complex, in general) measured (sampled) function values f_1, f_2, $\ldots$, $f_N \in C$ measured at N mutually distinct nodes $X_1, X_2, \ldots, X_N \in R^n$. We are interested also in the intermediate values corresponding to other points. Assume that these $f_j = f(X_j)$ are measured values of some continuous function f while z is an approximating function to be constructed. We put $n = 1$ in what follows.

If $L > 0$ we construct the set $\{\varphi_p\}$, $p = 1, \ldots, L$, of mutually orthogonal complex functions from W such that

$$(\varphi_p, \varphi_q)_L = 0 \ \text{ for } \ p, q = 1, \ldots, L. \tag{3}$$

This implies $|\varphi_p|_L = 0$. Moreover, assume

$$(\varphi_p, g_k)_L = 0 \ \text{ for } \ p = 1, \ldots, L, \quad k = 1, 2, \ldots. \tag{4}$$

The natural choice is $\varphi_p(x) = x^{p-1}$, $p = 1, \ldots, L$. The relations (3) and (4) are then satisfied. The set $\{\varphi_p\}$ is empty for $L = 0$.

Put

$$z(x) = \sum_{k=1}^{\infty} A_k g_k(x) + t(x), \quad t(x) = \sum_{p=1}^{L} a_p \varphi_p(x). \tag{5}$$

Problem of smooth interpolation. Let us fix nonnegative integers L and N of the above properties. The problem of smooth interpolation of a continuous function f given by its N values $f_j = f(X_j)$ is to find the coefficients a_p and A_k of the expressions (5) such that

$$z(X_j) = f_j, \quad j = 1, \ldots, N, \tag{6}$$

and

$$\text{the quantity } \; |z|_L^2 \; \text{ attains its minimum.} \tag{7}$$

The smooth interpolation problem thus consists of the variational problem (7), i.e. minimizing the functional $|z|_L^2$, with constraints (6).

Note that when minimizing $\|z\|^2$, we minimize not only the $L^2(a,b)$ norm of z but also (with a weight B_1 chosen) the $L^2(a,b)$ norm of z', i.e. of the first derivative of z. This can be of importance in processing of such measured data where also a good approximation of the first derivative is needed.

Put

$$R_L(x,y) = \sum_{k=1}^{\infty} \frac{g_k(x) g_k^*(y)}{|g_k|_L^2}. \tag{8}$$

If $L > 0$, introduce the rectangular $N \times L$ matrix Φ with entries $\Phi_{jp} = \varphi_p(X_j)$, $j = 1, \ldots, N$, $p = 1, \ldots, L$. Now we can formulate the following theorem.

Theorem 1. *Let $X_i \neq X_j$ for all $i \neq j$. Assume that the series* (8) *converges for all $x, y \in (a,b)$. Moreover, let* rank $\Phi = L$. *Then the problem* (5), (6), *and* (7) *of smooth interpolation has the unique solution*

$$z(x) = \sum_{j=1}^{N} \lambda_j R_L(x, X_j) + \sum_{p=1}^{L} a_p \varphi_p(x),$$

where the coefficients λ_j, $j = 1, \ldots, N$, and a_p, $p = 1, \ldots, L$, are the unique solution of the linear algebraic system

$$\begin{aligned} \sum_{j=1}^{N} \lambda_j R_L(X_i, X_j) + \sum_{p=1}^{L} a_p \varphi_p(X_i) &= f_i, \quad i = 1, \ldots, N, \\ \sum_{j=1}^{N} \lambda_j \varphi_p^*(X_j) \qquad\qquad\qquad\qquad &= 0, \quad p = 1, \ldots, L. \end{aligned}$$

Proof. The proof is based on the method of Lagrange multipliers for constrained minimization.

4. Numerical examples

We have used three systems $\{g_k\}$ defined in different spaces W with different sequences $\{B_l\}$, cf. [1], [2]. It is $B_l = (\frac{1}{3})^{2l}/(2l)!$, $l = 0, 1, \dots,$ for I and II.

I dashed line The system of transformed complex exponential functions $\exp(\mathrm{i}kx)$, $L = 0$, and the function $R_0(x, y)$ analytically known.

II dotted line The system of monomials x^k orthonormalized numerically on $(-1, 1)$ by the Gram-Schmidt procedure. The function $R_0(x, y)$ computed in double precision by summation until the module of the increment is less than 10^{-12} but at most 40 terms are considered.

III dashed line The same transformed complex exponential functions like in I. $B_l = 0$ for all l except for $B_2 = 1$, i.e. the L^2 norm of z'' is minimized. $R_2(x, y) = |y - x|^3$, $t(x) = a_0 + a_1 x$. This is the well-known *cubic spline interpolation.*

Moreover, we computed the results of

IV dotted line Polynomial interpolation.

V dash-dot line Rational interpolation.

Solid line shows the true solution, i.e. the function f given. We tried two of them, the smooth even function

$$f(x) = \frac{1}{1 + 16x^2} \tag{9}$$

with its maximum at $x = 0$ and the function

$$f(x) = 3(x + 1)^2 + \ln((\tfrac{1}{10}x)^2 + 10^{-5}) + 1 \tag{10}$$

with "almost a singularity" at $x = 0$. The grid is equidistant. Very "wavy" interpolants obtained are not shown.

Numerical experiments performed to present the properties of smooth interpolation show that it is an efficient method.

We were concerned only with the problem of smooth *exact interpolation of function values* at nodes which is controlled by the constraints (6) and, in addition, by the minimum condition (7). Moreover, the smooth approximation approach can be employed also in the *exact Hermite interpolation* and in the *smoothing of data.* The 2D case is much more interesting and makes many important applications possible.

Acknowledgement

This work was supported by RVO:67985840.

References

[1] Segeth, K.: Smooth approximation and its application to some 1D problems. In: *Proc. of Conference Applications of Mathematics 2012*, pp. 243–252. Institute of Mathematics, Academy of Sciences, Prague, 2012.

[2] Talmi, A. and Gilat, G.: Method for smooth approximation of data. J. Comput. Phys. **23** (1977), 93–123.

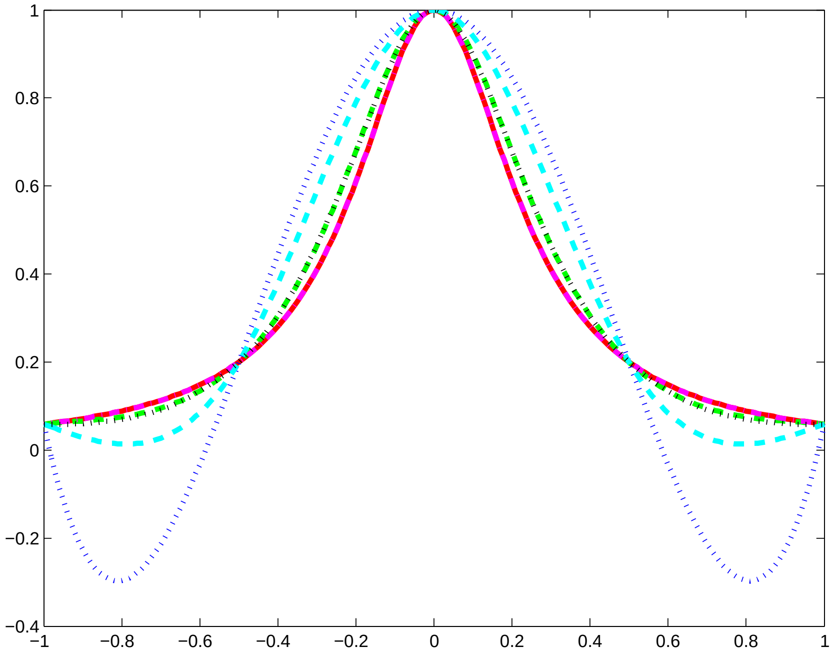


Figure 1: Interpolants of the function (9), $N = 5$. Curves at $x = 0.2$ from top to bottom: IV, III, I identical to II, true identical to V.

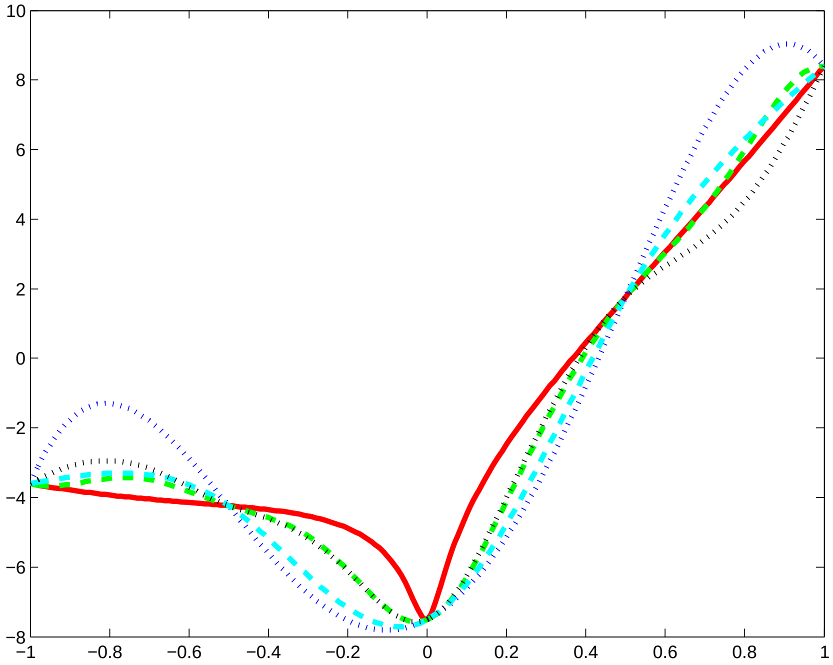


Figure 2: Interpolants of the function (10), $N = 5$. Curves at $x = -0.8$ from top to bottom: IV, II, III, I, true. V not shown.

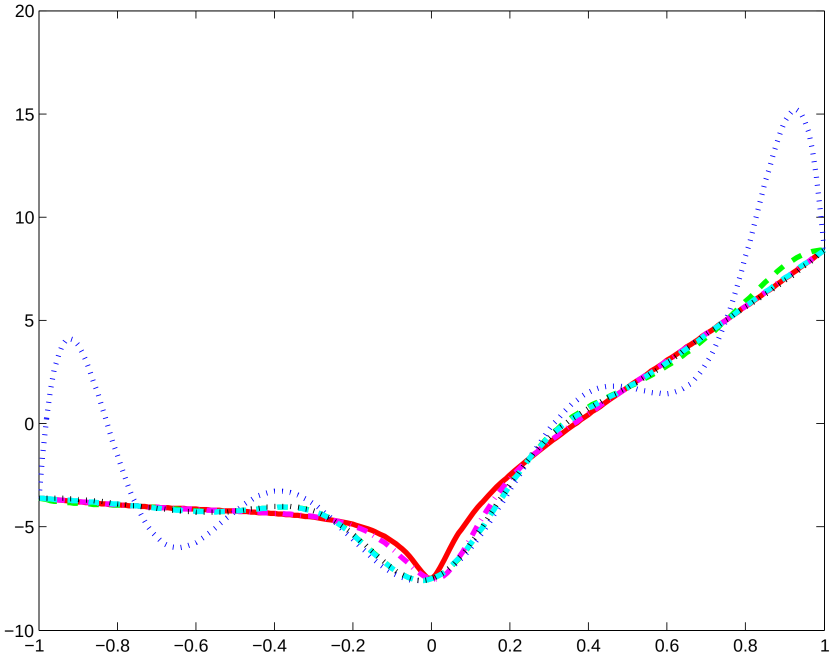


Figure 3: Interpolants of the function (10), $N = 9$. Curves at $x = 0.9$ from top to bottom: IV, I, II identical to III and to true. At $x = -0.1$, the first two from top: true, V. Notice different y scale.

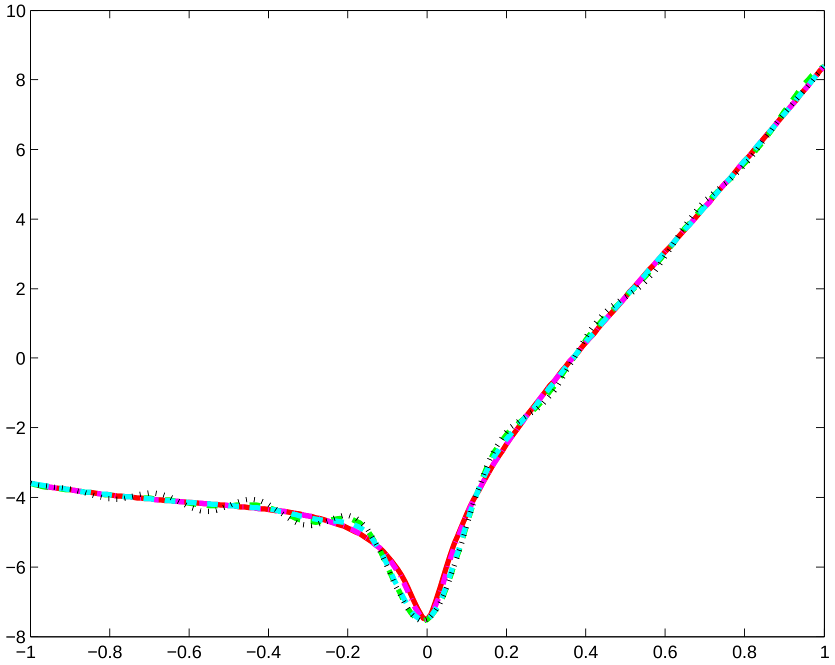


Figure 4: Interpolants of the function (10), $N = 17$. IV not shown, the rest almost identical.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

NUMERICAL ASPECTS OF THE IDENTIFICATION OF THERMAL CHARACTERISTICS USING THE HOT-WIRE METHOD

Jiří Vala
Brno University of Technology, Faculty of Civil Engineering
602 00 Brno, Veveří 95, Czech Republic
vala.j@fce.vutbr.cz

Abstract

The hot-wire method, based on the recording of the temperature development in time in a testing sample, supplied by a probe with its own thermal source, is useful to evaluate the thermal conductivity of materials under extremal loads, in particular in refractory brickworks. The formulae in the technical standards come from the analytical solution of the non-stationary equation of heat conduction in cylindric (finally only polar) coordinates for a simplified formulation of boundary conditions, neglecting everything except the first terms of the decomposition of related exponential integrals to infinite series, and least-squares based data fitting; such approach reduces the validity of results and obstructs the simultaneous evaluation of heat capacity.

This paper demonstrates that substantial improvements can be obtained without any requirements to additional measurements, both i) under the assumption of a wire of zero-thickness and an infinite sample (following the valid Czech technical standard) with proper exponential integrals and ii) for a more realistic geometrical configuration and physical simplification (taking into account the thermal characteristics of the wire), based on the properties of Bessel functions. The suggested algorithms have been implemented in the MATLAB environment.

1. Introduction

Reliable evaluation of thermal characteristics of materials used in mechanical, civil, etc. engineering, including their dependence on temperature, moisture, strain and other fields, even for advanced materials, structures and technologies where no reasonable values from practical experience are available, determines the range of applications of computational modelling of all multi-physical processes. In particular, identification of thermal properties of refractory brickworks (discussed later in more details), of hardening cement pastes and concrete structures [14], as well as of foods stored in freezing and cooling plants [9], requires some simple methodology, applicable under hard conditions, with negligible disturbing effect of other physical processes.

For simplicity, let us restrict to the identification of two basic characteristics of heat conduction in engineering materials: the thermal conductivity λ [W/(m·K)] (as a crucial thermal insulation characteristic) and the volumetric heat capacity κ [J/(m^2·K)] (important for thermal accumulation); the thermal diffusivity α [m^2/s] can be then introduced as $\alpha := \lambda/\kappa$. For the evaluation of λ, European technical standards offer the i) hot-plate, ii) hot-wire and iii) hot-ball approaches. The physical background of all these approaches is very similar: temperature (or temperature difference) is recorded in some (sufficiently small) range, whose development is forced by the carefully controlled generation of heat fluxes, during a (rather short) time interval. The principal difference consists in the geometrical configuration: in the case i) we have one or more parallel heating (or also additional non-heating) thin plates [11], in the case ii) a thin heating wire (see [1] and the following section) and in the case iii) a small heating ball (see [8]); the heat fluxes generated into the measurement system is controlled by direct voltage in all cases. The arrangement should be as simple as possible, with the aim to reduce the dimensions of corresponding heat transfer problems as much as possible; consequently (most frequently) working i) with Cartesian coordinates, ii) with cylindrical and iii) with spherical ones.

Our more detailed analysis will be devoted to the case ii). The relevant European standard [4] contains a (seemingly strange) explicit logarithmic formula for the evaluation of λ, supplied (for uncertain measurements) by the least-square (linear regression) approach to data fitting. However, as shown in [1], this formula can be identified with the fundamental solution of a heat conduction equation, satisfying the realistic boundary conditions in certain limit sense, well-known from [2], where in the additive decomposition of an exponential all terms except the first two are removed; this can be justified by the location of temperature sensors close to the heating wire. Such approach enables us to calculate (approximately) λ without the a priori knowledge of α; unfortunately, no information referring to κ is then available (because it was hidden in the removed terms containing α). We shall demonstrate that the proper analysis of the above sketched problems offers a possibility to identify both λ and κ from the same data set. Moreover, we shall show how some unpleasant physical and geometrical assumptions can be modified to be more realistic, using the properties of Bessel functions by [3] instead of the classical analytical results from [2].

2. Improved computations with exponential integrals

Following [4], let us assume that some constant heat Q [W/m], starting from the zero initial time, is generated per unit length of a very long and thin wire, located in the axis of the circular cylinder with a very large radius, occupied by the material specimen. Let $T(r,t)$ be the temperature field defined for any positive radius r (distance from the axis of rotation) and any positive time t (in practice for some measurement time interval) and T_0 the constant temperature of the surrounding environment. Then, by [1] (referring to [2]), using the notation $\beta_0 := Q/(4\pi\lambda)$, $\beta := 1/(4\alpha)$, we have

$$T = \beta_0 \,\mathrm{Ei}\,(\beta r^2/t) + T_0 \qquad \text{with } \;\mathrm{Ei}\,(.) := \int_{.}^{\infty} \frac{\exp(-u)}{u}\,\mathrm{d}u\,. \tag{1}$$

Indeed, using dot symbols for partial derivatives with respect to t and prime symbols for those with respect to r, it is easy to verify that T from (1) satisfies the classical Fourier equation of heat conduction (without internal heat sources) with constant characteristics λ and κ in polar coordinates

$$\kappa\dot{T} + \frac{\lambda}{r}(rT')' = 0 \tag{2}$$

together with the obvious initial condition $T(.,0) = T_0$ and the with the couple of boundary conditions

$$\lim_{r\to\infty} T(r,.) = 0\,, \qquad \lim_{r\to 0+} \frac{-\lambda T'(r,.)}{Q/(2\pi r)} = 1 \tag{3}$$

where the first limit guarantees the absence of heat fluxes from external environment and both the numerator and the denominator in the second limit represent the heat flux [W/m^2] on the surface of cylinder with a fixed small radius (this is just the announced way how to avoid the realistic finite radius and material characteristics of a wire). Clearly the data for $t = 0$ (and also $t \to 0$ in practice), thanks to the discontinuity of heat generated into the system (forcing the application of Dirac measures and Heaviside functions in [2]), are then not employable in any credible identification procedure for λ and κ, in particular for λ and α from (1); for the special case of the simplified evaluation of λ this observation is reflected by [4], too.

Let us assume that all sensors recording the temperature are located at $r = \delta$ where distance δ must be very small positive number by [4] (the measurement could be performed as close as possible to the wire surface), but is allowed to be finite in our considerations. Let m be a number of measurement time steps; the initial time $t = 0$ is not included. Using the notation $t_1, \ldots, t_m$ $(0 < t_1 < \ldots < t_m)$ for discrete measurement times and T_s for corresponding temperature values at $r = \delta$. All differences $T_s - T_{s-1}$ with $s \in \{2, \ldots, m\}$ should correspond to the experimental temperature differences τ_s; for simplicity, only one recorded temperature value is considered in every discrete time; the generalization over all available data is obvious. Thus, using the notation $\beta_1 := \beta\delta^2$, we have to minimize a function

$$\Phi = (1/2)\sum_{s=2}^{m}(\tau_s - (T_s - T_{s-1}))^2 \tag{4}$$

of two positive variables β_0 and β_1 (transformed from λ and α easily) where, for simplicity, only one recorded temperature value is considered in every discrete time; the generalization over all available data is obvious.

Let $\Phi_{,i}$ and $\Phi_{,ij}$ denote the derivatives $\partial\Phi/\partial\beta_i$ and $\partial^2\Phi/\partial\beta_i\partial\beta_j$ with $i, j \in \{0, 1\}$. For $\beta_{1s} := \mathrm{Ei}\,(\beta_1/t_s)$, $\widetilde{\beta}_{1s} = \exp(-\beta_1/t_s) - \exp(-\beta_1/t_{s-1})$ and $\varepsilon_s := \beta_0\beta_{1s} - \tau_s$ with $s \in \{2, \ldots, m\}$ we receive explicit formulae (the MAPLE support is welcome)

$$\Phi = (1/2)\sum_{s=2}^{m}\varepsilon_s^2\,, \qquad \Phi_{,0} = \sum_{s=2}^{m}\varepsilon_s\beta_{1s}\,, \qquad \Phi_{,1} = -(\beta_0/\beta_1)\sum_{s=2}^{m}\varepsilon_s\widetilde{\beta}_{1s}\,,$$

$$\Phi_{,00} = \sum_{s=2}^{m} \beta_{1s}^2\,, \qquad \Phi_{,01} = -(1/\beta_1)\sum_{s=2}^{m}(2\beta_0\beta_{1s} - \tau_s)\widetilde{\beta}_{1s}\,, \qquad \Phi_{,11} = (\beta_0/\beta_1)^2 \sum_{s=2}^{m} \widetilde{\beta}_{1s}^2$$
$$+(\beta_0/\beta_1)\sum_{s=2}^{m} \varepsilon_s \left(\mathrm{Ei}\,(\beta_1/t_s)/t_s - \mathrm{Ei}\,(\beta_1/t_{s-1})/t_{s-1}\right) + (\beta_0/\beta_1^2)\sum_{s=2}^{m} \varepsilon_s \widetilde{\beta}_{1s}\,.$$

Clearly we need $\Phi_{,0} = \Phi_{,1} = 0$. Taking (for sufficiently small δ) $\beta_1 \approx 0$ together with $\mathrm{Ei}\,(.) \approx -C_e - \ln(.)$ (the Euler-Mascheroni constant C_e is not needed in numerical calculations), for $\gamma_s := \ln(t_s/t_{s-1})$ with $s \in \{2,\dots,m\}$ we receive the very simple formula

$$\beta_0 \approx \sum_{s=2}^{m} \gamma_s \tau_s \Big/ \sum_{s=2}^{m} \gamma_s^2\,, \tag{5}$$

which is identical with that for the identification of λ from [4]. More generally, we are allowed to choose β_0 from (5) as the first estimate together with

$$\beta_1 \approx \sum_{s=2}^{m} (1/t_s - 1/t_{s-1})(\tau_s/\beta_0 - \gamma_s) \Big/ \sum_{s=2}^{m} (1/t_s - 1/t_{s-1})^2$$

and apply the Newton iteration algorithm

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \leftarrow \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} - \begin{bmatrix} \Phi_{,00} & \Phi_{,01} \\ \Phi_{,01} & \Phi_{,11} \end{bmatrix}^{-1} \begin{bmatrix} \Phi_{,0} \\ \Phi_{,1} \end{bmatrix};$$

this enables us to determine (more exactly) both β_0 and β_1, consequently also λ and κ (even without evaluations of inverse matrices in the computational practice).

3. A generalized approach applying Bessel functions

The generalization of the above sketched approach, removing mathematical and physical simplifications, can be done in more directions. However, being motivated from the results of MATLAB supported practical calculations with data coming from experiments with fire-clay bricks at high temperatures, we shall try to replace rather artificial boundary conditions (3) by more realistic ones. Let a be the outer radius of a specimen and $\delta < a$ a wire radius. Following [5], let us introduce the brief notation for scalar products in the special Lebesgue weighted spaces

$$(\phi, \widetilde{\phi})_r = \int_0^a \phi(.) r \widetilde{\phi}(.)\,\mathrm{d}r \qquad \text{for all } \phi, \widetilde{\phi} \in L_r^2(0,a)\,,$$

$$(\phi, \widetilde{\phi})_{r0} = \int_0^\delta \phi(.) r \widetilde{\phi}(.)\,\mathrm{d}r \qquad \text{for all } \phi, \widetilde{\phi} \in L_r^2(0,\delta)\,,$$

$$(\phi, \widetilde{\phi})_{r1} = \int_\delta^a \phi(.) r \widetilde{\phi}(.)\,\mathrm{d}r \qquad \text{for all } \phi, \widetilde{\phi} \in L_r^2(\delta,a)\,.$$

Material characteristics λ, κ, α will be taken as simple functions of r, with values equal to a priori known constants $\lambda_0, \kappa_0, \alpha_0$ for $0 \le r \le \delta$ and unknown ones $\lambda_1, \kappa_1, \alpha_1$ for $\delta \le r \le a$ (although their rather good estimates may be available by the previous section); moreover we shall need $\lambda_* := \lambda_1/\lambda_0$, $\kappa_* := \kappa_1/\kappa_0$ and $\alpha_* := \alpha_1/\alpha_0$.

Let $\mathcal{V}$ be the space of admissible test functions, i. e., applying the notation of special Sobolev weighted spaces from [5] again, the space of all $v \in W_r^{1,2}(0,a)$ such that $v(r) = v_0(r)$ for $0 \leq r \leq \delta$ and some $v_0 \in W_r^{1,2}(0,\delta)$, as well as $v(r) = v_1(r)$ for $\delta \leq r \leq a$ and some $v_1 \in W_r^{1,2}(\delta,a)$ satisfying $v_1(a) = 0$. Let $\mathcal{H}$ be the space introduced in the same way as $\mathcal{V}$ except L_r^2 inserted instead of $W_r^{1,2}$ everywhere. Using such notation, we are able to convert (2) into the form

$$(v, \kappa\dot{T})_r = (v, \lambda(rT')'/r)_r + (v,g)_r \tag{6}$$

where $g := Q/(\pi\delta^2)$ for $0 \leq r \leq \delta$ (any better information on the distribution of g in a wire is usually missing), zero otherwise. For positive times t we have to find $T(.,t) - T_0$ from $\mathcal{V}$ with $\dot{T}(.,t)$ from $\mathcal{H}$.

Let us consider the decomposition $T(r,t) = T_\sigma(r) + \theta(r,t)$ where

$$T(r,t) = T_\sigma(r) + \theta(r,t) \qquad \text{with } \theta(r,t) = \sum_{i=1}^{\infty} \varphi_i(r)\psi_i(t)\,; \tag{7}$$

the corresponding initial conditions are $T(.,0) = T_0$ and $\theta(.,0) = T_0 - T_\sigma(.)$ and the boundary (including the internal interface) ones are

$$\begin{aligned} T'(0,.) = 0\,, \quad \lambda_0 T'(\delta_-,.) &= \lambda_1 T'(\delta_+,.)\,, \quad T(a,.) = 0\,, \\ \theta'(0,.) = 0\,, \quad \lambda_0\theta'(\delta_-,0) &= \lambda_1\theta'(\delta_+,0)\,, \quad \theta(a,.) = 0\,, \\ \lambda_0 T_\sigma'(\delta_-) &= \lambda_1 T_\sigma'(\delta_+)\,, \quad T_\sigma(a) = T_0\,. \end{aligned} \tag{8}$$

(δ_+ and δ_- refer to left and right limits for r tending to δ). Here T_σ can be derived as an analytical solution for the stationary case (with zero κ formally)

$$T_\sigma(r) = \begin{cases} Q/(2\pi\lambda_1)\ln(a/\delta) + Q/(4\pi\lambda_0)(1-(r/\delta)^2) & \text{for } 0 \leq r \leq \delta\,, \\ Q/(2\pi\lambda_1)\ln(a/r) & \text{for } \delta \leq r \leq a\,. \end{cases} \tag{9}$$

Utilizing the properties of Bessel functions

$$J_n(r) = \frac{1}{\pi}\int_0^\pi \cos(r\sin\xi - n\xi)\,\mathrm{d}\xi \qquad \text{with } n \in \{0,1,2,\ldots\}\,,$$

namely $J_0'(r) = -J_1(r)$, $J_1'(r) = J_0(r) - J_1(r)/r$, etc., by [3], we can see that

$$r^{-1}(rJ_0'(\omega r))' + \omega^2 J_0(\omega r) = 0 \tag{10}$$

for any real ω, it is natural to find the zero points of Bessel functions, i. e. to solve $J_0(\omega_i a/\sqrt{\alpha_*}) = 0$ for unknown parameters ω_i with $i \in \{1,2,\ldots\}$, and to choose

$$\varphi_i(r) = \begin{cases} \beta_i J_0(\gamma_i\omega_i r) & \text{for } 0 < r < \delta\,, \\ J_0(\omega_i r/\sqrt{\alpha_*}) & \text{for } \delta < r < a\,, \end{cases} \tag{11}$$

to satisfy boundary conditions $\varphi_i'(0) = 0$, $\varphi_i(a) = 0$ automatically and

$$\varphi_i(\delta_-) = \varphi_i(\delta_+)\,, \qquad \lambda(\delta_-)\varphi_i'(\delta_-) = \lambda(\delta_+)\varphi_i'(\delta_+) \tag{12}$$

for a priori unknown values of β_i and γ_i, coming from the auxiliary systems of two nonlinear equations

$$\beta_i J_0(\gamma_i\omega_i\delta) = J_0(\omega_i\delta/\sqrt{\alpha_*})\,, \qquad \beta_i\gamma_i J_1(\gamma_i\omega_i\delta) = (\lambda_*/\sqrt{\alpha_*})J_1(\omega_i\delta/\sqrt{\alpha_*})\,. \tag{13}$$

It is easy to see that β_i can be evaluated from (13) as a function of γ_i directly. Consequently (13) degenerates to just one nonlinear equation for the evaluation of γ_i; all technical details for the Newton iterative algorithm can be found in [13].

Inserting (11) and (7) into (6), for any $v \in \mathcal{V}$ we receive

$$[(v,\varphi_i)_{r0} + \kappa_*(v,\varphi_i)_{r1}]\dot{\psi}_i - \alpha_0[(v,(r\varphi_i')'/r)_{r0} + \lambda_*(v,(r\varphi_i')'/r)_{r1}]\psi_i = 0\,. \tag{14}$$

Taking (10) into account, (14) gets tho form

$$[(v,\varphi_i)_{r0} + \kappa_*(v,\varphi_i)_{r1}]\dot{\psi}_i + \alpha_0\omega_i^2[\gamma_i^2(v,\varphi_i)_{r0} + \kappa_*(v,\varphi_i)_{r1}]\psi_i = 0\,. \tag{15}$$

Simultaneously, applying the Green-Ostrogradskiĭ theorem, (14) yields

$$\begin{aligned}[(v,\varphi_i)_{r0} + \kappa_*(v,r\varphi_i)_{r1}]\dot{\psi}_i + \alpha_0[(v',\varphi_i')_{r0} + \lambda_*(v',\varphi_i')_{r1}]\psi_i \\ = \alpha_0[(v(\delta_-)\varphi_i'(\delta_-) - \lambda_* v(\delta_+)\varphi_i'(\delta_+)]\,.\end{aligned} \tag{16}$$

In particular for $v = \varphi_j$ with arbitrary $j \in \{1,2,\ldots\}$, comparing (15) and (16), we have

$$(\varphi_j',\varphi_i')_{r0} + \lambda_*(\varphi_j',\varphi_i')_{r1} = \omega_i^2[\gamma_i^2(\varphi_j,\varphi_i)_{r0} + \kappa_*(\varphi_j,\varphi_i)_{r1}]\,.$$

The mutual exchange of indices i and j then results certain quasi-orthogonality condition

$$(\omega_i^2 - \omega_j^2)\kappa_*(\varphi_i,\varphi_j)_{r0} + (\omega_i^2\gamma_i^2 - \omega_j^2\gamma_j^2)(\varphi_i,\varphi_j)_{r1} = 0\,;$$

in practice $\gamma_i^2 \approx \gamma_j^2 \approx \kappa_*$ can be considered.

To find all ψ_i contained in (7), we must solve an eigenproblem $M_{ji}\dot{\psi}_i + K_{ji}\psi_i = 0$ for $M_{ji} := (\varphi_j,\varphi_i)_{r0} + \kappa(\varphi_j,\varphi_i)_{r1}$, $K_{ji} := \alpha_0\omega_i^2\,[(\varphi_j,\varphi_i)_{r0} + \kappa(\varphi_j,\varphi_i)_{r1})]$ and for the decomposition $\psi_i = V_{ip}\exp(-\Lambda_p t)C_p$, using the Einstein summation rule for all indices $i,j,p \in \{1,2,\ldots\}$; Λ_p here are eigenvalues, $V_{i1}, V_{i2},\ldots$ eigenvectors (in the matrix form we could write $MV\Lambda = KV$ only) and C_p unknown parameters, needed to be set due to our initial condition. The resulting formulae (assuming $i \neq j$) for effective numerical evaluation (obtained with the support of MAPLE) for the effective evaluation of D_{ji}, M_{ji} and K_{ji}, separately for diagonal and non-diagonal terms, can be found in [13]. The evaluation of constants C_p then comes from the equation

$$(v,T_0 - T_\sigma)_{r0} + \kappa_*(v,T_0 - T_\sigma)_{r1} = [(v,\varphi_i)_{r0} + \kappa_*(v,\varphi_i)_{r1}]V_{ip}C_p\,,$$

i. e. $F = MVC$, consequently $C = (MV)^{-1}F$, where most parts of integrals F_j with $j \in \{1,2,\ldots\}$, coming from (9), as presented in all details in [13], can be evaluated analytically, thanks to the properties of Bessel functions J_0 J_1, J_2 and J_3.

Our final aim is, exploiting the same data as in the preceding section, to minimize a function Φ from (4) of two positive variables λ_* and κ_* (transformed from λ_1 and α_1). Clearly a (sufficiently large) finite number of Bessel functions is considered in (7) in numerical calculations, thus all matrices M and K, vectors F, etc. are finite. However, it is not so easy to perform the minimization procedure because no simple explicit formulae employable in the Newton iterations are available, thus numerical evaluations of approximate first and second derivatives of Φ are necessary. Fortunately, this can be done e. g. with the support of selected functions from the MATLAB optimization toolbox.

4. Applications, conclusions and generalizations

In addition to the i) simplified approach recommended by [4], both algorithms presented in ii) Section 2 and iii) Section 3 of this paper have been implemented in MATLAB environment as the support of measurement tools in the Laboratory of Building Physics at Brno University of Technology. The limited extent of this paper does not allow to present results of practical calculations; the reader can find corresponding figures and graphs, together with more detailed description (and photo) of the original hot-wire measurement equipment in [10], devoted to the material design for the high-temperature thermal accumulator, as one part of the large Swedish-Czech research project of the efficient exploitation of solar energy using optical fibers.

Up to now, the computational results under hard conditions (far from room temperatures) demonstrate that i) gives only the rough estimate of λ, but no reasonable value of κ at all, whereas ii) is able to improve this estimate substantially. The system error of ii), coming from the neglected size and heat capacity of a hot wire, can be removed by iii) effectively, but making use of much more numerical computations. Nevertheless, other disturbing effects, coming from thermal convection and radiation, namely from the heat transfer at the wire / specimen interface, as well as those connected with the more complicated real geometrical conditions, cannot be handled in this way. More general formulations of heat transfer (together with other physical, chemical, etc. processes) need extensive applications of finite element, volume or difference methods, accompanied by the proper uncertainty analysis, as that based on Sobol sensitivity indices and Monte Carlo stochastic simulations like [7], or that substituting the Lebesgue measure by some probabilistic one, directed to stochastic finite element, etc. approaches, like [15]. Consequently Φ the optimization problem of the type (4) is not a function of two (or finite, for the best low) number of positive parameters, but a rather general functional in some space of abstract functions; some results and open questions of such analysis, containing direct, sensitivity and adjoint problems, have been presented in [12].

Acknowledgements

This work was supported by the grant No. FAST-S-12-25/1660 of the specific university research at Brno University of Technology.

References

[1] Bilek, J., Atkinson, J. K., and Wakeham, W. A.: Repeatability and refinement of a transient hot wire instrument for measuring the thermal conductivity of high temperature melts. International J. of Thermophysics **27** (2006), 1626–1637.

[2] Carslaw, H. C. and Jaeger, J. C.: *Conduction of Heat in Solids*. Oxford University Press, 1946.

[3] Culham, J. R.: Bessel functions of the first and second kind. In: *Special Functions*, Chap. 7-8. University of Waterloo, 2004.

[4] EN ISO 8894-1: *Refractory materials – Determination of thermal conductivity – Part 1: Hot-wire method (cross-array and resistance thermometer)*. European Committee for Standardization, 2010.

[5] Gopalakrishnan, J., and Pasciak, J. E.: The convergence of V-cycle multigrid algorithms for axisymmetric Laplace and Maxwell equations. Math. Comp. **75** (2006), 1697–1719.

[6] Gutierrez-Miravete, E.: Heat conduction in cylindrical and spherical coordinates. In: *Conduction Heat Transfer*, Chap. 3, Hartford University, 2006.

[7] Kala, Z.: Sensitivity analysis of steel plane frames with initial imperfections. *Engineering Structures* **33** (2011), 2342–2349.

[8] Kubičár, Ľ., Bágel', Ľ., Vretenár, V., and Štofanik, V.: Validation test of the hot ball method for setting of the cement paste. *Proceedings of Thermophysics* in Kočovce, Slovak Technical University Bratislava, 2007, 38–42.

[9] Kumcuoglu, S., Turgut, A., and Tavman, S.: The effect of temperature and muscle composition on the thermal conductivity of frozen meats. *Journal of Food Processing and Preservation* **34** (2010), 425–438.

[10] Šťastník, S., and Vala, J.: Identification of thermal characteristics of a high-temperature thermal accumulator. *Proceedings of Thermophysics* in Podkylava, Slovak Technical University Bratislava, 2012, 214–222.

[11] Šťastník, S., Vala, J., and Kmínová, H.: Identification of basic thermal technical characteristics of building materials. *Kybernetika* **43** (2007), 561–576.

[12] Vala, J.: Least-squares based technique for identification of thermal characteristics of building materials. *International Journal of Mathematics and Computers in Simulation* **5** (2011), 126–134.

[13] Vala, J.: Identification of thermal characteristics of building materials from simple measurements under hard experimental conditions. *Algoritmy* in Podbanské, conference poster, Slovak Technical University Bratislava, 2012, 20 pp.

[14] Vala, J., Šťastník, S., and Kozák, V.: Micro- and macro- scale thermomechanical modelling of bulk deformation in early-age cement-based materials. Key Engineering Materials **465** (2011), 111–114.

[15] Zabaras, N.: Inverse problems in heat transfer. In: *Handbook on Numerical Heat Transfer* W. J. Minkowycz, E. M. Sparrow, and J. S. Murthy, (eds.), Chap. 17. John Wiley & Sons, Hoboken, 2004.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

GUARANTEED AND FULLY COMPUTABLE TWO-SIDED BOUNDS OF FRIEDRICHS' CONSTANT

Tomáš Vejchodský

Institute of Mathematics, Academy of Sciences
Žitná 25, Czech Republic
vejchod@math.cas.cz

Abstract

This contribution presents a general numerical method for computing lower and upper bound of the optimal constant in Friedrichs' inequality. The standard Rayleigh-Ritz method is used for the lower bound and the method of *a priori-a posteriori inequalities* is employed for the upper bound. Several numerical experiments show applicability and accuracy of this approach.

1. Introduction

From the numerical point of view, the guaranteed and fully computable two-sided bounds always provide a strong information about the computed quantity. Their difference is a reliable bound on the approximation error and in applications they allow to stay on the safe side by using properly either the lower or the upper bound as the approximation.

In this contribution, we concentrate on the optimal constant in Friedrichs' inequality. The presented two-sided bounds are guaranteed up to round-off errors. The chosen approach is quite general and theoretically it can be used in arbitrary dimension, for any domain, and for different variants of Friedrichs' inequality. Practically, we are limited by particular choices of discretization methods. For instance, the presented numerical examples are limited to polygonal domains in two dimensions.

The optimal constant in Friedrichs' inequality is called Friedrichs' constant and its value is connected with the smallest eigenvalue of the corresponding differential operator. The classical Rayleigh-Ritz method provides an upper bound on the exact eigenvalue and consequently a lower bound for Friedrichs' constant.

Computing a lower bound of the smallest eigenvalue and hence computing the upper bound of Friedrichs' constant is considerably more difficult task. We use the method of *a priori-a posteriori inequalities* [5, 9]. The original idea relies on C^2-smooth test and trial functions, which are technically difficult to work with. Therefore, we proposed in [11] an alternative approach based on complementarity and standard Raviart-Thomas finite element method.

We briefly review this approach in Sections 2–4 and provide several numerical experiments in Sections 5–7.

2. Friedrichs' inequality

Let us consider a domain $\Omega \subset \mathbb{R}^d$ with Lipschitz boundary. Further, let Γ_{D} and Γ_{N} be two relatively open and disjoint subsets of the boundary $\partial\Omega$ such that $\partial\Omega = \overline{\Gamma}_{\mathrm{D}} \cup \overline{\Gamma}_{\mathrm{N}}$. Let the $(d-1)$-dimensional measure of Γ_{D} be positive. We will refer Γ_{D} and Γ_{N} to as Dirichlet and Neumann parts of the boundary, respectively. Further, we consider Sobolev space $H^1(\Omega) = \{v \in L^2(\Omega) : \nabla v \in [L^2(\Omega)]^d\}$ and its subspace $V = \{v \in H^1(\Omega) : v|_{\Gamma_{\mathrm{D}}} = 0\}$ of functions with vanishing traces on Γ_{D}.

In this contribution, we will assume the following variant of Friedrichs' inequality:

$$\|v\|_{0,\Omega} \leq C_{\mathrm{F}} \|\nabla v\|_{0,\Omega} \quad \forall v \in V, \tag{1}$$

where $\|\cdot\|_{0,\Omega}$ stands for the $L^2(\Omega)$-norm. Let us note that this inequality is named after Kurt O. Friedrichs [3]. The optimal (smallest) possible value of the constant C_{F} such that inequality (1) holds is called Friedrichs' constant and the symbol C_{F} will denote this optimal value throughout the paper. The particular value of C_{F} depends on the domain Ω and on the Dirichlet part of the boundary Γ_{D}.

Friedrichs' constant scales naturally with the size of Ω. Namely, if $\widetilde{\Omega} = k\Omega$, $\widetilde{\Gamma}_{\mathrm{D}} = k\Gamma_{\mathrm{D}}$, and $\widetilde{\Gamma}_{\mathrm{N}} = k\Gamma_{\mathrm{N}}$ for some $k \in \mathbb{R}$ then Friedrichs' constants $\widetilde{C}_{\mathrm{F}}$ and C_{F} corresponding to $\widetilde{\Omega}$ and Ω, respectively, satisfy $\widetilde{C}_{\mathrm{F}} = kC_{\mathrm{F}}$.

In special cases, Friedrichs' constant can be computed analytically. For example, it was computed for a rectangle, circle, and a circular wedge in [6] for $\Gamma_{\mathrm{D}} = \partial\Omega$. Result [8] can be used for analytic computation of C_{F} for equilateral and right-angle triangles. In certain simple cases (e.g. rectangle) it can be computed even if $\Gamma_{\mathrm{D}} \neq \partial\Omega$. In less special cases there are analytic upper bounds for Friedrichs' constant. The Faber-Kran inequality [2, 4] yields upper bound $C_{\mathrm{F}} \leq \sqrt{|\Omega|}/(j_{0,1}\sqrt{2\pi})$, where $|\Omega|$ is the area of the two-dimensional domain Ω and $j_{0,1} \doteq 2.404826$ is the first positive root of the Bessel function J_0. Similarly, in [7] we can find an estimate $C_{\mathrm{F}} \leq \pi^{-1} \left(|a_1|^{-2} + \cdots + |a_d|^{-2}\right)^{-1/2}$, where $|a_1|, \ldots, |a_d|$ are lengths of sides of a d-dimensional box in which the domain Ω is contained. Note that both these estimates require $\Gamma_{\mathrm{D}} = \partial\Omega$. However, in more general cases the value of Friedrichs' constant has to be computed numerically.

3. Lower bound on Friedrichs' constant

Friedrichs' constant C_{F} from (1) is connect with the smallest eigenvalue of the Laplace eigenvalue problem that can be formulated in a weak sense as: find $\lambda_i \in \mathbb{R}$ and $u_i \in V$, $u_i \neq 0$, $i = 1, 2, \ldots$, such that

$$(\nabla u_i, \nabla v) = \lambda_i (u_i, v) \quad \forall v \in V, \tag{2}$$

where the parenthesis denote the $L^2(\Omega)$ inner product. If $\lambda_1 = \min_i \lambda_i$ stands for the smallest eigenvalue of (2) then it can be easily shown, see e.g. [10, 11], that

$$C_{\mathrm{F}} = 1/\sqrt{\lambda_1}. \tag{3}$$

A standard method for computing approximations of the eigenvalues λ_i is the Rayleigh-Ritz method. In this method we consider a finite dimensional subspace $V^h \subset V$ and seek $\lambda_i^h \in \mathbb{R}$ and $u_i^h \in V^h$, $u_i^h \neq 0$ such that

$$(\nabla u_i^h, \nabla v^h) = \lambda_i^h (u_i^h, v^h) \quad \forall v^h \in V^h.$$

This is equivalent to the generalized eigenvalue problem $Ax_i = \lambda_i^h M x_i$ for the stiffness and mass matrices A and M. If a standard finite element method is used then matrices A and M are sparse and efficient methods of numerical linear algebra can be used. The Rayleigh-Ritz method is well known for providing an upper bound on the smallest eigenvalue. Indeed, since the differential operator in (2) and the corresponding matrices A and M are symmetric, we can express λ_1 and λ_1^h as minima of (generalized) Rayleigh quotients over V and V^h, respectively, and we obtain

$$\lambda_1 = \min_{\substack{v \in V \\ v \neq 0}} \frac{(\nabla v, \nabla v)}{(v, v)} \leq \min_{\substack{v^h \in V^h \\ v^h \neq 0}} \frac{(\nabla v^h, \nabla v^h)}{(v^h, v^h)} = \min_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{x^T A x}{x^T M x} = \lambda_1^h,$$

where $n = \dim V^h$. Consequently, the approximation $C_{\mathrm{F}}^{\mathrm{low}} = (\lambda_1^h)^{-1/2}$ of Friedrichs' constant, see (3), is a lower bound on the exact value C_{F}, i.e.,

$$C_{\mathrm{F}}^{\mathrm{low}} = (\lambda_1^h)^{-1/2} \leq C_{\mathrm{F}}.$$

4. Upper bound on Friedrichs' constant

Computing an upper bound of Friedrichs' constant is a more difficult task, because it corresponds to the computation of a lower bound of the smallest eigenvalue. We employ the method of *a priori-a posteriori inequalities* [5, 9] enhanced by the complementary approach. Mathematical details, relations, and derivations can be found in [11]. Here, we just briefly describe the algorithm.

First, use the Rayleigh-Ritz method and compute approximations $\lambda_1^h \in \mathbb{R}$ and $u_1^h \in V$ of the smallest eigenvalue λ_1 and the corresponding eigenfunction u_1. Second, choose a flux reconstruction $\boldsymbol{q}_h \in \mathbf{H}(\mathrm{div}, \Omega) = \{\boldsymbol{q} \in [L^2(\Omega)]^d : \mathrm{div}\, \boldsymbol{q} \in L^2(\Omega)\}$. Third, compute

$$\alpha = \frac{\|\nabla u_1^h - \boldsymbol{q}_h\|_{0,\Omega}}{\|u_1^h\|_{0,\Omega}}, \quad \beta = \frac{\|\lambda_1^h u_1^h + \mathrm{div}\, \boldsymbol{q}_h\|_{0,\Omega}}{\|u_1^h\|_{0,\Omega}}, \quad X_2 = \frac{1}{2}\sqrt{\alpha^2 + 4(\lambda_1^h - \beta)} - \frac{\alpha}{2}.$$

The lower bound on the smallest eigenvalue and the corresponding upper bound on Friedrichs' constant are then given as

$$X_2^2 \leq \lambda_1 \quad \text{and} \quad C_{\mathrm{F}} \leq C_{\mathrm{F}}^{\mathrm{up}} = 1/X_2.$$

Although any $\boldsymbol{q}_h \in \mathbf{H}(\mathrm{div}, \Omega)$ provides an upper bound on C_{F}, an accurate approximation is obtained for an appropriate choice of $\boldsymbol{q}_h$, only. In this contribution, we consider a Raviart-Thomas finite element subspace $W_h \subset \mathbf{H}(\mathrm{div}, \Omega)$ based on a triangulation of Ω and minimize $\alpha^2 + \beta^2$ over W_h. This minimization is equivalent to finding $\boldsymbol{q}_h \in W_h$ such that

$$(\operatorname{div} \boldsymbol{q}_h, \operatorname{div} \boldsymbol{\psi}_h) + \lambda_1^h(\boldsymbol{q}_h, \boldsymbol{\psi}_h) = \lambda_1^h(\nabla u_1^h, \boldsymbol{\psi}_h) - \lambda_1^h(u_1^h, \operatorname{div} \boldsymbol{\psi}_h) \quad \forall \boldsymbol{\psi}_h \in W_h.$$

This problem can be solved by standard finite element technology, see e.g. [1]. We note that this particular flux reconstruction is a brute force solution and if the efficiency is an issue then a local reconstruction based on ∇u_1^h has to be used.

Further, it is important to note that the method of a priori-a posteriori inequalities is justified only if the approximation λ_1^h is sufficiently accurate. In particular, the closest eigenvalue to λ_1^h must be λ_1. If λ_1 and the second smallest eigenvalues λ_2 are well separated then sufficiently accurate Rayleigh-Ritz approximations of λ_1 and λ_2 can provide good confidence about the validity of this assumption. In all numerical experiments present below we experienced exactly this situation.

5. Example A: Friedrichs' constant for triangles

Friedrichs' constant C_F depends on the size and shape of the domain Ω and on the size, shape, and position of Γ_D. The dependence on the size of Ω is well known, see Section 2. Therefore, the following numerical experiments concentrate on the dependence of C_F on the shape of Ω (Examples A and B) and on Γ_D (Example C).

In all experiments below, the Rayleigh-Ritz approximations λ_1^h and u_1^h are computed by linear finite elements on triangular meshes and the reconstructed fluxes $\boldsymbol{q}_h$ by quadratic Raviart-Thomas finite elements on the same triangular mesh.

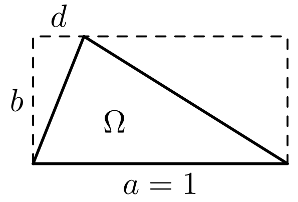


Figure 1: The shape of triangle Ω is given by the parameters b and d.

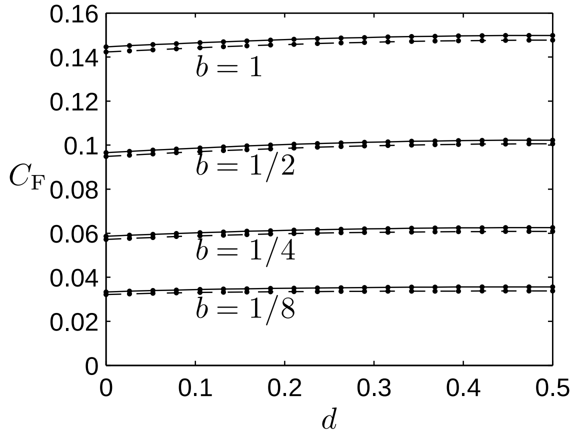


Figure 2: Friedrichs' constant for triangles with vertices $[0, 0]$, $[1, 0]$, $[d, b]$. Solid and dashed lines correspond to upper and lower bounds of C_F, respectively.

First, we consider Ω to be a nonobtuse triangle and assume $\Gamma_\mathrm{D} = \partial\Omega$. We investigate the dependence of C_F on the shape of this triangle. In particular, we consider triangles inscribed into a rectangle with lengths of sides a and b. The triangles have vertices with coordinates $(0, 0)$, $(a, 0)$, (d, b), see Figure 1. In particular, we fix $a = 1$, consider four values of $b \in (0,1]$, namely $b = 1, 1/2, 1/4, 1/8$, and 20 equidistributed values of d in $[0, 1/2]$. The two-sided bounds on C_{F} for the resulting triangles are presented in Figure 2. These bounds were computed on uniform meshes obtained by six successive uniform refinement steps of the original triangle. Thus, all these meshes have $4^6 = 4096$ triangles. We see that a fixed value of b yields triangles with the same area and the parameter d then controls the shape only. However, the observed dependence of C_{F} on the shape is negligible. We see a considerable dependence of C_{F} on b, but it is connected with the size of Ω as mentioned in Section 2.

6. Example B: Friedrichs' constant for regular stars

The value of Friedrichs' constant is of interest especially for nonconvex domains. Therefore, we consider Ω to be n-fold regular star with $n = 3, 4, \ldots, 30$ and choose $\Gamma_{\mathrm{D}} = \partial\Omega$. We put the outer vertices of stars Ω on a circle with radius $r_{\mathrm{out}} = 1$ and the inner vertices on a circle with radius $r_{\mathrm{in}} = 1/3$, see Figure 3. We use uniform mesh with $4^6 \cdot 2n$ triangles and compute both lower and upper bound on C_{F}.

Figure 4 shows the dependence of C_{F} on n. The value of C_{F} decreases with n and it seems that in the limit $n \to \infty$ it converges to Friedrichs' constant of a circle with radius $r_{\mathrm{in}} = 1/3$, which is approximately 0.138610. We note that Friedrichs' constant for a circle with radius $r_{\mathrm{out}} = 1$ is approximately 0.415831. The increasing gap between the lower and upper bound of C_{F} is probably caused by singularities of the eigenfunction u_1 at the obtuse angles. The strength of these singularities increases with the size of these angles, but the resolution of the used meshes stays the same.

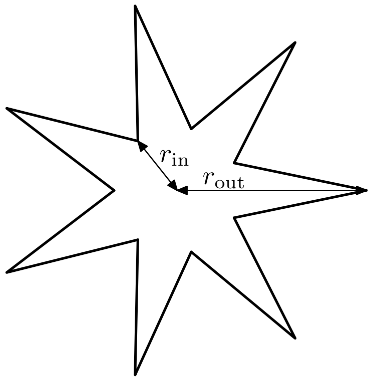


Figure 3: Illustration of 7-fold regular star with inner and outer radii.

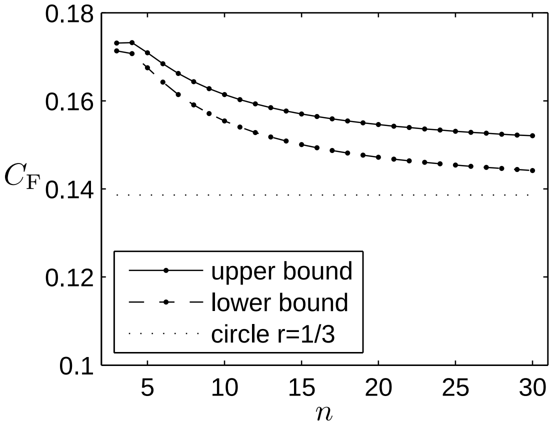


Figure 4: Values of Friedrichs' constant for n-fold regular stars.

7. Example C: Dependence on the Dirichlet part

In the final example we investigate the dependence of C_{F} on the Dirichlet part Γ_{D} of the boundary $\partial\Omega$. We consider a fixed L-shaped domain Ω and vary the position and size of Γ_{D}. The boundary $\partial\Omega$ is split into 16 segments of unit length. The part Γ_{D} is chosen as a connected curve of length $|\Gamma_{\mathrm{D}}| = \ell$, i.e. it consists of ℓ segments. In this experiment we consider $|\Gamma_{\mathrm{D}}| = 1, 5, 11$, and 15. For each length we compute the lower and upper bound of C_{F} for all 16 positions of Γ_{D} on $\partial\Omega$. The positions are indexed by the number of the first segment of Γ_{D} in the counterclockwise sense, see Figure 5.

Figure 6 presents the dependence of C_{F} on the position of Γ_{D} for the four considered sizes $|\Gamma_{\mathrm{D}}|$. We observe strong dependence both on the position and size. We also see similar values of C_{F} for almost symmetric positions, for instance for $|\Gamma_{\mathrm{D}}| = 1$ and positions 4 and 11 or for $|\Gamma_{\mathrm{D}}| = 11$ and positions 7 and 14 (these positions correspond to peeks in the graphs in both cases).

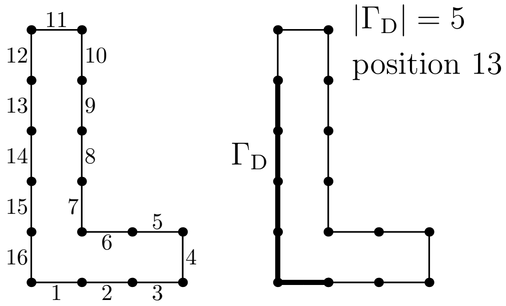


Figure 5: The L-shaped domain and enumeration of boundary segments (left). An example of a position and size of Γ_{D} (right).

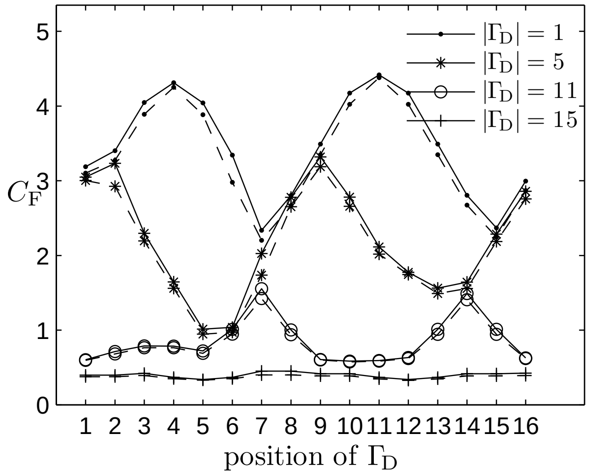


Figure 6: Dependence of Friedrichs' constant on the position and size of Γ_{D}. Upper bounds are indicated by solid lines and lower bounds by dashed lines.

8. Conclusions

In this contribution we present a method for computing guaranteed lower and upper bounds of Friedrichs' constant or equivalently for lower and upper bounds of eigenvalues of the corresponding differential operator. The main output is a numerical study of the value of Friedrichs' constant C_{F} in various cases including convex and nonconvex domains. We observe the dependence of C_{F} on the shape of the domain Ω and on the size and position of the Dirichlet part Γ_{D} of the boundary $\partial\Omega$. While we observed negligible dependence of C_{F} on the shape of nonobtuse triangles, the dependence on the size and position of the Dirichlet part Γ_{D} is significant in majority of tested cases.

Let us conclude this contribution by a note that the presented method can be easily generalized to compute two-sided bounds of the optimal constants in similar inequalities, like the trace inequalities, Poincaré inequality, and Korn's inequality. For all these inequalities the computation of the optimal constant reduces to the computation of the smallest eigenvalue of a differential operator.

Acknowledgements

This work has been supported by the Grant Agency of the Academy of Sciences by grant no. IAA100190803 and by RVO 67985840.

References

[1] Brenner, S. C. and Scott, L. R.: *The mathematical theory of finite element methods*, *Texts in Applied Mathematics*, vol. 15. Springer, New York, 2008, 3rd edn.

[2] Faber, G.: Beweis, daß unter allen homogenen Membranen von gleicher Fläche und gleicher Spannung die kreisförmige den tiefsten Grundton gibt. Sitz. bayer. Akad., Wiss. (1923), 169–172.

[3] Friedrichs, K.: Eine invariante Formulierung des Newtonschen Gravitationsgesetzes und des Grenzüberganges vom Einsteinschen zum Newtonschen Gesetz. Math. Ann. **98** (1927), 566–575.

[4] Krahn, E.: Über eine von Rayleigh formulierte Minimaleigenschaft des Kreises. Math. Ann. **94** (1925), 97–100.

[5] Kuttler, J. R. and Sigillito, V. G.: Bounding eigenvalues of elliptic operators. SIAM J. Math. Anal. **9** (1978), 768–778.

[6] Kuttler, J. R. and Sigillito, V. G.: Eigenvalues of the Laplacian in two dimensions. SIAM Rev. **26** (1984), 163–193.

[7] Mikhlin, S. G.: *Constants in some inequalities of analysis.* John Wiley & Sons., 1986.

[8] Práger, M.: Eigenvalues and eigenfunctions of the Laplace operator on an equilateral triangle. Appl. Math. **43** (1998), 311–320.

[9] Sigillito, V. G.: *Explicit a priori inequalities with applications to boundary value problems.* Pitman Publishing, London-San Francisco, Calif.-Melbourne, 1977.

[10] Valdman, J.: Minimization of functional majorant in a posteriori error analysis based on $H(\mathrm{div})$ multigrid-preconditioned CG method. Adv. Numer. Anal. (2009), Art. ID 164 519, 15.

[11] Vejchodský, T.: Computing upper bounds on Friedrichs' constant. In: J. Brandts, J. Chleboun, S. Korotov, K. Segeth, J. Šístek, and T. Vejchodský (Eds.), *Applications of Mathematics 2012*, pp. 278–289. Institute of Mathematics, ASCR, Prague, 2012.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

A PRIORI DIFFUSION-UNIFORM ERROR ESTIMATES FOR SINGULARLY PERTURBED PROBLEMS: MIDPOINT-DG DISCRETIZATION

Miloslav Vlasák, Václav Kučera

Charles University in Prague, Faculty of Mathematics and Physics
Department of Numerical Mathematics
Sokolovská 83, 18675 Prague 8, Czech Republic
vlasak@karlin.mff.cuni.cz, kucera@karlin.mff.cuni.cz

Abstract

We deal with a nonstationary semilinear singularly perturbed convection–diffusion problem. We discretize this problem by discontinuous Galerkin method in space and by midpoint rule in time. We present diffusion–uniform error estimates with sketches of proofs.

1. Introduction

Our aim is development of sufficiently robust, accurate and efficient numerical schemes for solving nonlinear singularly perturbed convection–diffusion equations, which describe many important topics, e.g. fluid dynamics.

Singularly perturbed convection–diffusion equations represent very difficult problems, since these problems lie on the edge between elliptic and hyperbolic problems. From numerical point of view these problems are unpleasant, since they have steep gradients or discontinuities in the solution even for smooth data. To overcome these difficulties we employ discontinuous Galerkin method, which uses piecewise discontinuous polynomial functions. It seems that such a weaker inter–element connection partially suppresses spurious oscillations in the discrete solution, which are present in the standard finite element solution.

Applying standard parabolic techniques to this problem we obtain diffusion dependent error estimates – typically with the constant $e^{1/\varepsilon}$, where ε is the diffusion parameter, see e.g. [1] or [4]. In practical cases from compressible fluid dynamics, where ε is about 10^{-5} to 10^{-9}, these error estimates are useless.

Our aim is to derive a priori error estimates that are uniform with respect to the diffusion parameter. A majority of analysis of singularly perturbed problems devoted to the uniform a priori error estimates concerns linear problems only, see e.g. [5].

The technique, how to overcome the nonlinearity in the convective part, is presented in [8], with applications to the explicit time stepping schemes. The technique is based on the linearization of the problem by Taylor expansion, where the problem is divided into linear part and higher order nonlinear reminder. How to deal with the linear part is known from the analysis of purely linear problems. The analysis of the nonlinear reminder is more tricky and takes advantage of higher order of the reminder and of higher order of the error at previous time levels. In [6] we can find the extension of this result to the semidiscrete problem and to the backward Euler method, where (in contrast to explicit schemes) one needs higher order of the error at the actual time level and not at the previous one. This problem is solved by continuous mathematical induction. This paper extends the technique from [6] to midpoint rule.

2. Continuous problem

Let $\Omega \subset \mathbb{R}^d$ be a bounded polyhedral domain and $T > 0$. We set $Q_T = \Omega \times (0, T)$. Let us consider the following problem: Find $u : Q_T \to \mathbb{R}$ such that

$$\begin{aligned} &\frac{\partial u}{\partial t} + \nabla \cdot f(u) - \varepsilon\, \Delta u = g \quad \text{in } Q_T, &(1)\\ &u\,\big|_{\partial\Omega \times (0,T)} = 0, \\ &u(x, 0) = u^0(x), \quad x \in \Omega. \end{aligned}$$

We assume $f = (f_1, \dots, f_d),\ f_s \in C^2(\mathbb{R}),\ f_s(0) = 0,\ s = 1, \dots, d$ represents convective terms, $\varepsilon \geq 0$, $g \in C([0, T]; L^2(\Omega))$ and $u^0 \in L^2(\Omega)$ is an initial condition. We assume that the weak solution of (1) is sufficiently regular, namely,

$$u \in W^{1,\infty}(0, T; H^{p+1}(\Omega)) \cap W^{2,\infty}(0, T; H^2(\Omega)), \quad u^{(3)} \in L^\infty(0, T; L^2(\Omega)), \tag{2}$$

where $u^{(k)} = \partial^k u/\partial t^k$, an integer $p \geq 1$ will denote a given degree of polynomial approximations in space.

3. Discrete problem

To simplify the expressions we use the notation $(\cdot, \cdot)$ for L^2 scalar product and $\|\cdot\|$ for L^2 norm. We employ the *symmetric interior penalty Galerkin* (SIPG) method for the space semi-discretization of (1), for details see [2]. Let $\mathcal{T}_h$ $(h > 0)$ be a partition of $\overline{\Omega}$ into a finite number of closed d-dimensional simplices K with mutually disjoint interiors. Let $S_h = \{w; w|_K \in P_p(K)\ \forall\, K \in \mathcal{T}_h\}$ denote the space of discontinuous piecewise polynomial functions of degree p on each $K \in \mathcal{T}_h$. Then we say that the function $u_h \in C^1(0, T; S_h)$ is the *semi-discrete approximate solution* of (1) if it satisfies the conditions

$$\left(\frac{\partial u_h}{\partial t}(t), w\right) + \varepsilon A_h(u_h(t), w) + b_h(u_h(t), w) = \ell_h(w)\,(t) \quad \forall\, w \in S_h,\ \forall\, t \in [0, T], \tag{3}$$

and $(u_h(0), w) = (u^0, w)\ \forall\, w \in S_h$. The bilinear form A_h represents the diffusion term with a sufficiently large interior and boundary penalty, b_h is a nonlinear form representing convective term based on the numerical fluxes well known from the finite volume method and ℓ_h represents the source term. For the exact definition of forms A_h, b_h and ℓ_h see e.g. [2]. We assume the numerical fluxes H to be Lipschitz continuous, conservative and consistent. Moreover, we assume that the numerical fluxes are E–fluxes:

$$(H(v, w, n) - f(q) \cdot n)(v - w) \geq 0, \quad \forall v, w \in \mathbb{R},\ \forall q \text{ between } v \text{ and } w, \tag{4}$$

where $n \in \mathbb{R}^d$ is an unit vector.

We find that the weak solution of (1) with property (2) satisfies the identity

$$\left(\frac{\partial u}{\partial t}(t), w\right) + \varepsilon A_h(u(t), w) + b_h(u(t), w) = \ell_h(w)\,(t) \tag{5}$$

for all $w \in S_h$ and all $t \in (0, T)$.

For simplicity we assume time partition $t_m = m\tau$, $m = 0, \ldots, r$ with the time step $\tau = T/r$. To simplify the future expressions we set the notation $v^m = v(t_m)$.

Definition 1. *We say that the set of functions $U^m \in S_h$, $m = 0, \ldots, r$ is an approximate solution of problem (1) obtained by midpoint–DGFE scheme if*

$$\begin{aligned}(U^m - U^{m-1}, w) + \frac{\tau\varepsilon}{2} A_h(U^m + U^{m-1}, w) + \tau b_h\left(\frac{U^m + U^{m-1}}{2}, w\right)& \\ = \tau \ell_h(w)(t_{m-1} + \tau/2) \quad \forall w \in S_h&, \\ (U^0, w) = (u^0, w) \quad \forall w \in S_h&.\end{aligned} \tag{6}$$

4. Error estimates

We denote the energy norm $\|\!|w|\!\|^2 := A_h(w, w)\ \forall w \in S_h$. Note that the inverse inequality takes the following form $\|\!|w|\!\| \leq Ch^{-1}\|w\|$ for $w \in S_h$. Let Π be the L^2 orthogonal projection on S_h.

We summarize the properties of the forms A_h and b_h.

Lemma 1. *Let u satisfy (2). Then*

$$A_h(v, w) \leq C\|\!|v|\!\|\,\|\!|w|\!\| \quad \forall v, w \in S_h, \tag{7}$$

$$A_h(u(t_{m-1} + s/2), w) - A_h\left(\frac{u(s) + u^{m-1}}{2}, w\right) \leq C\tau^2\|\!|w|\!\| \quad \forall w \in S_h, \quad \forall s \in [t_{m-1}, t_m], \tag{8}$$

$$A_h(\Pi u - u, w) \leq Ch^p\|\!|w|\!\| \quad \forall w \in S_h. \tag{9}$$

The proof of (7) and (9) can be done in a similar way as in [3, Lemma 9]. The proof of (8) can be done similarly as in [7, Lemma 4.3].

Lemma 2. *Let u satisfy (2). Then*

$$b_h(v,w) - b_h(\bar{v},w) \le C\|v-\bar{v}\|\, \|\!|w|\!\| \quad \forall v,\bar{v},w \in S_h \tag{10}$$

$$b_h(u(t_{m-1}+s/2),w) - b_h\left(\frac{u(s)+u^{m-1}}{2}, w\right) \le C\tau^2 \|\!|w|\!\| \quad \forall w \in S_h, \quad \forall s \in [t_{m-1},t_m], \tag{11}$$

$$b_h(v, v-\Pi u) - b_h(u, v-\Pi u) \le C\left(1+\frac{\|v-u\|_\infty^2}{h^2}\right)(h^{2p+1}+\|v-\Pi u\|^2) \quad \forall v \in S_h. \tag{12}$$

The proof of (10) can be found in [3], the proof of estimate (11) uses the regularity of arguments with respect to space and standard error estimates and (12) can be found in [6].

Our goal is to investigate the error estimates of the approximate solution U^m, $m=0,\dots,r$ obtained by the method (6). To do this we employ the strategy of continuous extension of the discrete solution mimicking to the strategy in [6].

Definition 2. *Let $s \in (0,\tau]$. We say that the function $U(t_{m-1}+s) \in S_h$ is a continuated approximate solution of problem (1) obtained by midpoint–DGFE scheme if*

$$(U(t_{m-1}+s) - U^{m-1}, w) + \frac{s\varepsilon}{2} A_h(U(t_{m-1}+s)+U^{m-1}, w) + s b_h\left(\frac{U(t_{m-1}+s)+U^{m-1}}{2}, w\right) = s\ell_h(w)(t_{m-1}+s/2) \quad \forall w \in S_h. \tag{13}$$

It is obvious that $U(t_m) = U^m$.

We denote the left–hand side and right–hand side from Definition 2

$$B_s^m(v,w) = (v-U^{m-1}, w) + \frac{s\varepsilon}{2} A_h(v+U^{m-1}, w) + s b_h\left(\frac{v+U^{m-1}}{2}, w\right), \tag{14}$$

$$L_s^m(w) = s\ell_h(w)(t_{m-1}+s/2). \tag{15}$$

We shall show that B_s^m is strongly monotone on S_h:

$$\begin{aligned} B_s^m(v,v-w) - B_s^m(w,v-w) &\ge \|v-w\|^2 + \frac{s\varepsilon}{2}\|\!|v-w|\!\|^2 - Cs\|v-w\|\, \|\!|v-w|\!\| \\ &\ge \left(1+\frac{s\varepsilon}{h^2} - \frac{Cs}{h}\right)\|v-w\|^2 = M\|v-w\|^2 \end{aligned} \tag{16}$$

for sufficiently small s respectively τ. We shall show that B_s^m is Lipschitz continuous on S_h:

$$\begin{aligned} B_s^m(v,w) - B_s^m(\bar{v},w) &\le \|v-w\|\, \|w\| + C\frac{s\varepsilon}{2}\|\!|v-\bar{v}|\!\|\, \|\!|w|\!\| + Cs\|v-\bar{v}\|\, \|\!|w|\!\| \\ &\le \left(1+\frac{Cs\varepsilon}{h^2}+\frac{Cs}{h}\right)\|v-\bar{v}\|\, \|w\| = C\|v-\bar{v}\|\, \|w\|. \end{aligned} \tag{17}$$

Since right–hand side L_s^m is evidently Lipschitz continuous, we can employ nonlinear Lax–Milgram lemma to prove the existence of the continuated discrete solution and classical discrete solution, respectively.

Now we should show that the continuated discrete solution is really continuous. Since the proof is the same at each time interval $(t_{m-1}, t_m]$, we show it for the simplicity only on the first one. Let $t, s \in (0, \tau]$. Then

$$\begin{aligned} M\|U(t) - U(s)\|^2 &\le B_t^1(U(t), U(t) - U(s)) - B_t^1(U(s), U(t) - U(s)) \qquad (18) \\ &= L_t^1(U(t) - U(s)) - L_s^1(U(t) - U(s)) \\ &\quad + B_s^1(U(s), U(t) - U(s)) - B_t^1(U(s), U(t) - U(s)). \end{aligned}$$

Since the terms on the second and third row tend to zero if $|t - s|$ tends zero we obtain continuity. Analogically we can prove the continuity at 0+. Since the exact solution u is continuous and since we have continuity on the closed interval $[0, T]$, we can see that the error $U(t) - u(t)$ is uniformly continuous.

As the final step we shall derive the error estimate of the continuated solution at arbitrary time $t \in [0, T]$ which immediately imply the error estimate for the classical method.

In the sequel we use the notation $\xi(t) = U(t) - \Pi u(t)$, $\eta(t) = \Pi u(t) - u(t)$ and $e(t) = U(t) - u(t) = \xi(t) + \eta(t)$.

Lemma 3. *Let u satisfy (2). Then*

$$\|\eta(t)\| \le Ch^{p+1}, \qquad (19)$$

$$(u(t_{m-1} + s) - u^{m-1} - s\frac{\partial u}{\partial t}(t_{m-1} + s/2), w) \le Cs^3\|w\| \quad \forall w \in S_h, \forall s \qquad (20)$$

$$(\eta(t_{m-1} + s) - \eta^{m-1}, w) \le Csh^{p+1}\|w\| \quad \forall w \in S_h, \forall s \qquad (21)$$

Proof. The estimate (19) is standard estimate for L^2 projection approximation. The estimate (20) can be done similarly as in [4] and the last estimate (21) can be found in [1]. □

Lemma 4. *Let $p > d/2$. Let $s \in (0, \tau]$. If $\|e(t)\| \le h^{1+d/2}$ for $t \le t_{m-1} + s$, then*

$$\sup_{t \in [0, t_{m-1}+s]} \|e(t)\|^2 \le C_T^2(h^{2p+1} + \varepsilon h^{2p} + \tau^4), \qquad (22)$$

where the constant C_T is independent of h, τ, ε.

Proof. Multiplying (5) for $t = t_{m-1} + s/2$ by s, subtracting from (13) and adding several terms we get

$$\begin{aligned}
&\Big(\xi(s) - \xi^{m-1}, w) + \frac{s\varepsilon}{2} A_h(\xi(s) + \xi^{m-1}), w\Big) \qquad (23)\\
&\le \left(s\frac{\partial u}{\partial t}(t_{m-1} + s/2) - u(s) + u^{m-1}, w \right)\\
&+s\left(b_h(u(t_{m-1} + s/2), w) - b_h\left(\frac{u(s) + u^{m-1}}{2}, w\right)\right) + (\eta(s) - \eta^{m-1}, w)\\
&+s\left(b_h\left(\frac{u(s) + u^{m-1}}{2}, w\right) - b_h\left(\frac{U(s) + U^{m-1}}{2}, w\right)\right) - \frac{s\varepsilon}{2} A_h(\eta(s) + \eta^{m-1}), w)\\
&+s\left(A_h(u(t_{m-1} + s/2), w) - A_h\left(\frac{u(s) + u^{m-1}}{2}, w\right)\right).
\end{aligned}$$

Setting $w = \xi(s) + \xi^{m-1}$ and using Lemmas 1–3 to estimate the right–hand side we get

$$\begin{aligned}
&\|\xi(s)\|^2 - \|\xi^{m-1}\|^2\\
&\qquad \le Cs\left(1 + \frac{\|e(s) + e^{m-1}\|_\infty^2}{h^2}\right)(\varepsilon h^{2p} + h^{2p+1} + \tau^2 + \|\xi(s)\|^2 + \|\xi^{m-1}\|^2).
\end{aligned}$$

Using the assumptions we can get rid of the unpleasant term $\|e(s) + e^{m-1}\|_\infty^2/h^2$ and by standard Gronwall lemma we can finish the proof. □

We are ready to present the main result.

Theorem 5. *Let $p > 1 + d/2$. Let $h_1, \tau_1 > 0$ are such that*

$$C_T^2(h_1^{2p+1} + \varepsilon h_1^{2p} + \tau_1^4) \le \frac{1}{2} h_1^{2+d}. \qquad (24)$$

Let τ_1 is sufficiently small to guarantee the existence and continuity of the continuated discrete solution. Then for all $h \in (0, h_1)$ and $\tau \in (0, \tau_1)$ we get

$$\sup_{t\in[0,T]} \|e(t)\|^2 \le C_T^2(h^{2p+1} + \varepsilon h^{2p} + \tau^4), \qquad (25)$$

where the constant C_T is independent of h, τ, ε.

Proof. We will follow the idea of continuous mathematical induction from [6]. For time $t = 0$ it is easy to see that the error estimate holds true, because the error is in fact the error of L^2 projection in initial data, which is sufficiently small under the assumptions of the theorem. Let as assume that the error estimate holds true on the interval $[0, s]$. According to the assumption (24) we can see that the error can be estimated by $\|e(t)\| \le \frac{1}{2} h^{1+d/2}$, $t \in [0, s]$. Since the error $e(t)$ is continuous (even uniformly continuous) we know that there exists some $\delta > 0$ such that $\|e(t)\| \leq h^{1+d/2}$, $t \in [0, s+\delta]$ and we can see that it is possible to use Lemma 4 even on the interval $[0, s+\delta]$, which guarantees the error estimate on $[0, s+\delta]$. Since the error is uniformly continuous, we have fixed $\delta > 0$ during the process and using the argument repeatedly we obtain the result. □

Acknowledgements

The authors are junior researchers of the University centre for mathematical modelling, applied analysis and computational mathematics (Math MAC).

References

[1] Dolejší, V., Feistauer, M., and Hozman, J.: Analysis of semi-implicit DGFEM for nonlinear convection-diffusion problems. Comput. Methods Appl. Mech. Engrg. **196** (2007), 2813–2827. doi:10.1016/j.cma.2006.09.025.

[2] Dolejší, V., Feistauer, M., Kučera, V., and Sobotíková, V.: An optimal $L^\infty(L^2)$-error estimate for the discontinuous Galerkin approximation of a nonlinear nonstationary convection-diffusion problem. IMA J. Numer. Anal. **28** (2008), 496–521. doi:10.1093/imanum/drm023.

[3] Dolejší, V., Feistauer, M., and Sobotíková, V.: Analysis of the discontinuous Galerkin method for nonlinear convection–diffusion problems. Comput. Methods Appl. Mech. Engrg. **194** (2005), 2709–2733. doi:10.1016/j.cma.2004.07.017.

[4] Dolejší, V. and Vlasák, M.: Analysis of a BDF-DGFE scheme for nonlinear convection-diffusion problems. Numer. Math. **110** (2008), 405–447. doi: 10.1007/s00211-008-0178-2.

[5] Feistauer, M., Hájek, J., and Švadlenka, K.: Space-time discontinuous Galerkin method for solving nonstationary convection-diffusion-reaction problems. Appl. Math., Praha **52** (2007), 197–233. doi:10.1007/s10492-007-0011-8.

[6] V. Kučera. On diffusion-uniform error estimates for the DG method applied to singularly perturbed problems. The Preprint Series of the School of Mathematics, preprint No. MATHknm- 2011/3 (2011), http://www.karlin.mff.cuni.cz/ms-preprints/prep.php. (submitted)

[7] Vlasák, M., Dolejší, V., and Hájek, J.: A priori error estimates of an extrapolated space-time discontinuous Galerkin method for nonlinear convection-diffusion problems. Numer. Methods Partial Differ. Equations **27** (2011), 1456–1482. doi:10.1002/num.20591.

[8] Zhang, Q. and Shu, C.W.: Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin method for symmetrizable systems of conservation laws. SIAM J. Numer. Anal. **44** (2006), 1703–1720. doi:10.1137/040620382.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

MODIFICATIONS OF THE LIMITED-MEMORY BFGS METHOD BASED ON THE IDEA OF CONJUGATE DIRECTIONS

Jan Vlček[1], Ladislav Lukšan[1,2]

[1] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
vlcek@cs.cas.cz, luksan@cs.cas.cz
[2] Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

Abstract

Simple modifications of the limited-memory BFGS method (L-BFGS) for large scale unconstrained optimization are considered, which consist in corrections of the used difference vectors (derived from the idea of conjugate directions), utilizing information from the preceding iteration. For quadratic objective functions, the improvement of convergence is the best one in some sense and all stored difference vectors are conjugate for unit stepsizes. The algorithm is globally convergent for convex sufficiently smooth functions. Numerical experiments indicate that the new method often improves the L-BFGS method significantly.

1. Introduction

We propose some modifications of the L-BFGS method (see [5], [10]) for large scale unconstrained minimization of the differentiable function $f : \mathcal{R}^N \to \mathcal{R}$. Similarly as in the multi-step quasi-Newton methods (see e.g. [9]), we utilize information from the preceding iteration. However, while the multi-step methods derive the corrections of the difference vectors from various interpolation methods, our approach is based on the idea of conjugate directions (see e.g. [4, 11]).

The L-BFGS method belongs to the variable metric (VM) or quasi-Newton line search methods, see [4], [8]. They start with an initial point $x_0 \in \mathcal{R}^N$ and generate iterations $x_{k+1} \in \mathcal{R}^N$ by the process $x_{k+1} = x_k + t_k d_k$, $k \ge 0$, where d_k is the direction vector and $t_k > 0$ is a stepsize, usually chosen in such a way that

$$f_{k+1} - f_k \le \varepsilon_1 t_k g_k^T d_k, \qquad g_{k+1}^T d_k \ge \varepsilon_2 g_k^T d_k, \tag{1}$$

$k \ge 0$, where $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$ and $d_k = -H_k g_k$ with a symmetric positive definite matrix H_k; usually $H_0 = I$ and H_{k+1} is obtained from H_k by a VM update to satisfy the quasi-Newton condition $H_{k+1} y_k = s_k$ (see [4, 8]), where $s_k = x_{k+1} - x_k = t_k d_k$ and $y_k = g_{k+1} - g_k$, $k \ge 0$.

Among VM methods, the BFGS method belongs to the most efficient; the update formula can be written in the form (note that $b_k > 0$ for $g_k \ne 0$ by (1))

$$H_{k+1} = (1/b_k) s_k s_k^T + V_k H_k V_k^T, \quad b_k = s_k^T y_k, \quad V_k = I - (1/b_k) s_k y_k^T,$$

$k \geq 0$, see [4, 8, 11], on which the L-BFGS method – a limited-memory adaptation of the BFGS method – is based. Instead of an $N \times N$ matrix H_k, only the last $\tilde{m}+1$ couples $\{s_j, y_j\}_{j=k-\tilde{m}}^{k}$ are stored, where $\tilde{m} = \min(k, m-1)$ and $m \geq 1$ is a given parameter. The direction vector is computed by the Strang recurrences, see [10], and still satisfies $d_{k+1} = -H_{k+1}g_{k+1}$, $k \geq 0$, but matrix H_{k+1} is not formed explicitly.

Here we will investigate such corrections of vectors s_k, y_k which provide conjugacy of consecutive corrected vectors. Thus we will define corrected quantities $\bar{s}_k$, $\bar{y}_k$, $\bar{b}_k$ and $\bar{V}_k$, $k \geq 0$, by $\bar{s}_0 = s_0$, $\bar{y}_0 = y_0$, $\bar{b}_0 = b_0$, $\bar{V}_0 = V_0$ and

$$\bar{s}_k = s_k - \alpha_k \bar{s}_{k-1}, \quad \bar{y}_k = y_k - \beta_k \bar{y}_{k-1}, \quad \bar{b}_k = \bar{s}_k^T \bar{y}_k, \quad \bar{V}_k = I - (1/\bar{b}_k)\bar{s}_k \bar{y}_k^T, \tag{2}$$

$k > 0$, with such α_k, $\beta_k \in \mathcal{R}$ that $\bar{b}_k > 0$. Correspondingly, we will use a direction vector $d_k = -\bar{H}_k g_k$, $k \geq 0$, where $\bar{H}_0 = I$ and symmetric positive definite matrix

$$\begin{aligned} \bar{H}_{k+1} \;=\; & (s_k^T y_k/|y_k|^2)\, \bar{V}_k \cdots \bar{V}_{k-\tilde{m}}\; \bar{V}_{k-\tilde{m}}^T \cdots \bar{V}_k^T \\ & +\; (1/\bar{b}_{k-\tilde{m}})\, \bar{V}_k \cdots \bar{V}_{k-\tilde{m}+1}\; \bar{s}_{k-\tilde{m}} \bar{s}_{k-\tilde{m}}^T\; \bar{V}_{k-\tilde{m}+1}^T \cdots \bar{V}_k^T \\ & +\; \cdots \;+\; (1/\bar{b}_{k-1})\, \bar{V}_k\; \bar{s}_{k-1}\bar{s}_{k-1}^T\; \bar{V}_k^T \;+\; (1/\bar{b}_k)\, \bar{s}_k \bar{s}_k^T\,,\; k \geq 0, \end{aligned} \tag{3}$$

satisfies the quasi-Newton condition $\bar{H}_{k+1}\bar{y}_k = \bar{s}_k$ and is obtained by the repeated BFGS update of $(s_k^T y_k/|y_k|^2)I$ with corrected vectors. We denote $\bar{B}_k = \bar{H}_k^{-1}$, $k \geq 0$.

In Section 2 we investigate the standard BFGS update with corrected vectors

$$\bar{H}_+ = (1/\bar{b})\bar{s}\bar{s}^T + \bar{V}\bar{H}\bar{V}^T, \quad \bar{b} = \bar{s}^T\bar{y}, \quad \bar{V} = I - (1/\bar{b})\bar{s}\bar{y}^T, \tag{4}$$

(in the simplified form) of any symmetric positive definite matrix $\bar{H}$ with corrected difference vectors $\bar{s} = s - \alpha\bar{s}_-$, $\bar{y} = y - \beta\bar{y}_-$ and discuss the choice of parameters α, β. In Section 3 we focus on quadratic functions and show optimality of our choice of parameters and conjugacy and other properties for unit stepsizes. Application to limited-memory methods and the corresponding algorithm are described in Section 4, global convergence of the algorithm is established in Section 5 and numerical results are reported in Section 6. Details and proofs of assertions can be found in [13].

2. The BFGS update with corrected vectors

The following lemma enables us to distinguish roles of products $\bar{s}^T\bar{y}_-$, $\bar{s}_-^T\bar{y}$ and shows that, under some assumptions, the conjugacy of difference vectors $\bar{s}$, $\bar{s}_-$ with respect to matrices $\bar{B} = \bar{H}^{-1}$, $\bar{B}_+ = \bar{H}_+^{-1}$ is equivalent to the satisfaction of condition $\bar{H}_+\bar{y}_- = \bar{s}_-$. Note that condition $\bar{H}\bar{y}_- = \bar{s}_-$ represents the quasi-Newton condition from the preceding update, which is satisfied for $m > 1$, see [13].

Lemma 1. *Let $\bar{H}$ be any symmetric positive definite matrix with $\bar{H}\bar{y}_- = \bar{s}_-$, matrix $\bar{H}_+$ be given by (4) with $\bar{b} > 0$ and $\Delta_1 = (\bar{H}_+\bar{y}_- - \bar{s}_-)^T\bar{B}_+(\bar{H}_+\bar{y}_- - \bar{s}_-)$. Then*

$$\Delta_1 = \left[(\bar{s}_-^T\bar{y} - \bar{s}^T\bar{y}_-)^2 + \omega(\bar{s}^T\bar{y}_-)^2\right]/\bar{b}\,, \tag{5}$$

where $\omega \geq 0$, with $\omega = 0$ only in case of dependency of vectors $\bar{s}$, $\bar{H}\bar{y}$. If vectors $\bar{s}$, $\bar{H}\bar{y}$ are linearly independent then $\bar{H}_+$ satisfies $\bar{H}_+\bar{y}_- = \bar{s}_-$ if and only if vectors $\bar{s}$, $\bar{s}_-$ are conjugate with respect to matrices $\bar{B}$, $\bar{B}_+$.

Since value ω could be large, we can see from relation (5) that mainly value $\bar{s}^T\bar{y}_-$ should be close to zero, to have Δ_1 small. Therefore we prefer the choice $\alpha = s^T\bar{y}_-/\bar{b}_-$, for which $\bar{s}^T\bar{y}_- = 0$. Similarly, the basic choice of β is $\beta_Z = \bar{s}_-^T y/\bar{b}_-$, which yields $\bar{s}_-^T\bar{y} = 0$ (thus $\bar{H}_+\bar{y}_- = \bar{s}_-$ by $\Delta_1 = 0$) and has some interesting properties.

Theorem 2. *Let $\bar{H}$ be any symmetric positive definite matrix with $\bar{H}\bar{y}_- = \bar{s}_-$ and matrix $\bar{H}_+$ be given by (4) with $\bar{b} > 0$. If $\alpha = s^T\bar{y}_-/\bar{b}_-$ then $\bar{s}^T\bar{y}_- = 0$, $\bar{b} = b - \alpha\, \bar{s}_-^T y$ and both value $\bar{a}$ and the condition number of matrix $\bar{H}^{1/2}\bar{B}_+\bar{H}^{1/2}$ as functions of β are minimized by the choice $\beta = \bar{s}_-^T y/\bar{b}_-$.*

Satisfaction of condition $\bar{H}_+\bar{y}_- = \bar{s}_-$ also guarantees that matrix $\bar{H}_+$ is closer to $\bar{H}$ than to $\bar{H}_-$ in some sense, as we can see from Theorem 3 with $\bar{H}_-$, $\bar{H}$, $\bar{s}_-$, $\bar{y}_-$ instead of $\bar{H}$, $\bar{H}_+$, $\bar{s}$, $\bar{y}$ and $\tilde{G} = \bar{H}_+^{-1}$ ($\|.\|_F$ denotes the Frobenius matrix norm).

Theorem 3. *Let $\bar{H}$ be any symmetric positive definite matrix, matrix $\bar{H}_+$ be given by (4) with $\bar{b} > 0$, $\tilde{G}$ be any symmetric positive definite matrix satisfying $\tilde{G}\bar{s} = \bar{y}$, $W_+ = \tilde{G}^{1/2}\bar{H}_+\tilde{G}^{1/2}$ and $W = \tilde{G}^{1/2}\bar{H}\tilde{G}^{1/2}$. Then*

$$\|I - W_+\|_F^2 - \|I - W\|_F^2 = -\|W_+ - W\|_F^2 \leq -\left(\bar{a}/\bar{b} - 1\right)^2. \tag{6}$$

The following lemma indicates that β should also be near to α, to have $|\bar{H}_+ y - s|$ small. E.g. the choice $\beta = \pm\sqrt{\beta_Z \alpha}$ has interesting properties.

Lemma 4. *Let $\bar{H}$ be any symmetric positive definite matrix with $\bar{H}\bar{y}_- = \bar{s}_-$ and matrix $\bar{H}_+$ be given by (4) with $\bar{b} > 0$. If $\alpha = s^T\bar{y}_-/\bar{b}_-$ then $\Delta_1 = (\bar{s}_-^T\bar{y})^2/\bar{b}$ and*

$$(\bar{H}_+ y - s)^T\bar{B}_+(\bar{H}_+ y - s) = \bar{b}_-[(\beta - \alpha)^2 + (\beta - \beta_Z)^2(s^T\bar{y}_-)^2/(\bar{b}\,\bar{b}_-)]. \tag{7}$$

Moreover, if $\beta^2 = s^T\bar{y}_-\,\bar{s}_-^T y/\bar{b}_-^2$, then $y^T(\bar{H}_+ y - s) = 0$.

3. Results for quadratic functions

In this section we suppose that f is a quadratic function with a symmetric positive definite matrix G and that $\beta = \alpha$, which is a natural choice, if we want to have $\bar{y} = G\bar{s}$, similarly as for non-corrected vectors. Here we consider only the G-conjugacy of vectors.

The conjugacy of $\bar{s}$, $\bar{s}_-$ can be achieved by the choice $\alpha = s^T\bar{y}_-/\bar{b}_- = \bar{s}_-^T y/\bar{b}_-$ by (2). The following theorem shows that this choice is the best in some sense.

Theorem 5. *Let $\hat{\alpha} = s^T\bar{y}_-/\bar{b}_- = \bar{s}_-^T y/\bar{b}_-$, $\bar{H}$ be any symmetric positive definite matrix with $\bar{H}\bar{y}_- = \bar{s}_-$, $\bar{H}_+$ be given by (4) with $\beta = \alpha$ and let f be a quadratic function $f(x) = \frac{1}{2}(x - x^*)^T G(x - x^*)$, $x^* \in \mathcal{R}^N$, with a symmetric positive definite matrix G. If vectors s, $\bar{s}_-$ are linearly independent, then $\bar{b} > 0$ and the choice $\alpha = \hat{\alpha}$ implies $\bar{H}_+ y = s$ and minimizes the values $\bar{b}$, $\|G^{1/2}\bar{H}_+G^{1/2} - I\|_F$ as functions of α.*

The L-BFGS method with exact line searches generates conjugate directions vectors and preserves $\tilde{m}$ previous quasi-Newton conditions, see e.g. [10]. Similarly for update (4) with unit stepsizes we get that all stored vectors $\bar{s}_k$ are conjugate and $\tilde{m}$ previous quasi-Newton conditions are preserved, if every stepsize is unit.

Theorem 6. *Let $x_0 \in \mathcal{R}^N$, $x^* \in \mathcal{R}^N$, $\bar{k} > 0$, $m \geq 1$, f be the quadratic function $f(x) = \frac{1}{2}(x - x^*)^T G(x - x^*)$ with a symmetric positive definite matrix G, and let for $0 \leq k \leq \bar{k}$ iterations $x_{k+1} = x_k + s_k$ be generated by $s_k = -t_k \bar{H}_k g_k$, $g_k = \nabla f(x_k)$, $t_k > 0$, with matrices $\bar{H}_k$ defined in the following way: $\bar{H}_0 = I$ and matrices $\bar{H}_{k+1}$, $0 \leq k < \bar{k}$, are given by (3), where $\tilde{m} = \min(k, m{-}1)$, $y_k = g_{k+1} - g_k$, and quantities $\bar{s}_j$, $\bar{y}_j$, $\bar{V}_j$ and $\bar{b}_j$, $j \geq 0$, are formally defined by $\bar{s}_0 = s_0$, $\bar{y}_0 = y_0$, $\bar{s}_{j+1} = s_{j+1} - \alpha_{j+1}\bar{s}_j$, $\bar{y}_{j+1} = y_{j+1} - \alpha_{j+1}\bar{y}_j$, $\alpha_{j+1} = s_{j+1}^T \bar{y}_j / \bar{b}_j$, $\bar{V}_j = I - (1/\bar{b}_j)\bar{s}_j \bar{y}_j^T$, $\bar{b}_j = \bar{s}_j^T \bar{y}_j$.*

If every generated vector s_k is linearly independent of $\bar{s}_{k-1}$, $0 < k \leq \bar{k}$, then the method is well defined. Moreover, if $t_{k+1} = 1$ for some k, $0 \leq k < \bar{k}$, it holds

$$\bar{H}_{k+i}\bar{y}_k = \bar{s}_k, \quad \bar{s}_k^T G \bar{s}_{k+i} = 0, \quad \bar{s}_k^T g_{k+i+1} = 0, \;\; 1 \leq i \leq \min(\tilde{m}{+}1, \bar{k}{-}k). \tag{8}$$

4. Application to limited-memory methods

From the theory in Section 3 we can deduce that we should use the corrected difference vectors whenever objective function is close to a quadratic function. As measure of deviation from a quadratic function at points x_{k-1}, x_k, x_{k+1}, e.g. value $|s_k^T y_{k-1} - s_{k-1}^T y_k|$ could serve (zero for quadratic functions), $k > 0$; we use value $|s_k^T \bar{y}_{k-1} - \bar{s}_{k-1}^T y_k| = \bar{b}_{k-1}|\alpha_k - \beta_k|$, which gives very similar results. We do not correct if it is greater than $\bar{b}_{k-1}^2 / b_k$, if $(s_k^T \bar{y}_{k-1}).(\bar{s}_{k-1}^T y_k) \leq 0$ or if $\bar{b}_k \leq 10^{-6} b_k$.

Value $\beta_k = \mathrm{sgn}(\alpha_k)\sqrt{\theta_k / \bar{b}_{k-1}}$, corresponding to the choice in Lemma 4, appears to be suitable if value $\bar{b}_k$ is sufficiently large with respect to b_k (we use condition $\bar{b}_k > 10^{-2} b_k$). This choice satisfies $|\beta_k| < \sqrt{b_k / \bar{b}_{k-1}}$; it is a reason why we use this value β_k also in case that $|\bar{s}_{k-1}^T y_k / \bar{b}_{k-1}| > 2\sqrt{b_k / \bar{b}_{k-1}}$ to prove global convergence.

Global convergence can be easily established (in a similar way as for the L-BFGS method, see [5]), if $|\bar{s}_k|/|s_k| \leq \Delta$ and $|\bar{y}_k|/|y_k| \leq \Delta$, $k > 0$, where $\Delta > 1$ is a constant. If this condition is not satisfied, it suffices to replace the oldest saved vectors $\bar{s}_{k-\tilde{m}}$, $\bar{y}_{k-\tilde{m}}$ e.g. by s_k, y_k. Note that in our numerical experiments with $N = 1000$, value $|\bar{y}_k|/|y_k|$ was rarely greater than 10 and value $|\bar{s}_k|/|s_k|$ greater than 50.

We now state the method in details. For simplicity, we omit stopping criteria.

Algorithm 4.1

Data: The number $m \geq 1$ of VM updates per iteration, line search parameters ε_1, ε_2, $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$, and correction parameter $\Delta > 1$.

Step 0: Initiation. Choose starting point $x_0 \in \mathcal{R}^N$, define starting matrix $\bar{H}_0^0 = I$ and direction vector $d_0 = -\nabla f(x_0)$ and initiate iteration counter k to zero.

Step 1: Line search. Compute $x_{k+1} = x_k + t_k d_k$, where t_k satisfies (1), $s_k = x_{k+1} - x_k$, $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ and $b_k = s_k^T y_k$. If $k = 0$ set $\bar{s}_k = s_k$, $\bar{y}_k = y_k$ and go to Step 4.

Step 2: Correction preparation. Set $\alpha_k = s_k^T \bar{y}_{k-1}/\bar{b}_{k-1}$, $\beta_k = \bar{s}_{k-1}^T y_k/\bar{b}_{k-1}$. If $\alpha_k\beta_k \le 0$ or $\bar{b}_k \le 10^{-6} b_k$ or $|\alpha_k - \beta_k| \ge \bar{b}_{k-1}/b_k$, set $\alpha_k = \beta_k = 0$ and go to Step 3. If $|\beta_k| > 2\sqrt{b_k/\bar{b}_{k-1}}$ or $\bar{b}_k > 10^{-2} b_k$, replace β_k by $\beta_k\sqrt{\alpha_k/\beta_k}$.

Step 3: Correction. Set $\bar{s}_k = s_k - \alpha_k \bar{s}_{k-1}$, $\bar{y}_k = y_k - \beta_k \bar{y}_{k-1}$.

Step 4: Update definition. Set $\tilde{m} = \min(k, m-1)$, $\bar{b}_k = \bar{s}_k^T \bar{y}_k$ and define $\bar{V}_k = I - (1/\bar{b}_k)\bar{s}_k \bar{y}_k^T$. If $|\bar{s}_{k-\tilde{m}}|/|s_{k-\tilde{m}}| > \Delta$ or $|\bar{y}_{k-\tilde{m}}|/|y_{k-\tilde{m}}| > \Delta$, set $\bar{s}_{k-\tilde{m}} = s_k$, $\bar{y}_{k-\tilde{m}} = y_k$ and $\bar{b}_{k-\tilde{m}} = b_k$. Define $\bar{H}_{k+1}$ by (3).

Step 5: Direction vector. Compute $d_{k+1} = -\bar{H}_{k+1} g_{k+1}$ by the Strang recurrences, set $k := k+1$ and go to Step 1.

5. Global convergence

Assumption 7. *The objective function $f : \mathcal{R}^N \to \mathcal{R}$ is bounded from below and uniformly convex with bounded second-order derivatives (i.e. $0 < \underline{G} \le \underline{\lambda}(G(x)) \le \overline{\lambda}(G(x)) \le \overline{G} < \infty$, $x \in \mathcal{R}^N$, where $\underline{\lambda}(G(x))$ and $\overline{\lambda}(G(x))$ are the lowest and the greatest eigenvalues of the Hessian matrix $G(x)$).*

Theorem 8. *If objective function f satisfies Assumption 7, Algorithm 4.1 generates a sequence $\{g_k\}$ that either satisfies $\lim_{k\to\infty} |g_k| = 0$ or terminates with $g_k = 0$ for some k.*

6. Numerical results

In this section, we demonstrate the influence of vector corrections on the number of evaluations (NFE) and computational time, using the following collections of test problems: Test 11 from [7] (55 chosen problems), which are modified problems from CUTE collection [2] with N ranging from 1000 to 5000, test from [1], termed Test 12 here, 73 problems, $N = 5000$, Test 25 from [6] (68 chosen problems), $N = 10000$.

Table 1 contains results for the following limited-memory methods: L-BFGS, see [10], method from [12] (Algorithm 3.1 with $\sigma = 0.4$) and new Algorithm 4.1. We have used $m = 5$, $\Delta = 100$, the final precision $\|g(x^\star)\|_\infty \le 10^{-6}$, $\varepsilon_1 = 10^{-4}$ and $\varepsilon_2 = 0.8$.

Method	Test 11		Test 12		Test 25	
	NFE	Time	NFE	Time	NFE	Time
L-BFGS	80539	32.50	43648	46.17	462104	519.40
Alg. 3.1 in [12]	80328	34.52	43182	56.67	512880	649.15
Algorithm 4.1	64395	30.20	34472	37.57	296321	381.08

Table 1. Comparison of the selected methods.

For a better demonstration of both the efficiency and the reliability, we compare selected optimization methods for Test 25 by using performance profiles introduced in [3]. The value of $\pi_M(\tau)$ at $\tau = 0$ gives the percentage of test problems for which the method M is the best and the value for τ large enough is the percentage of test problems that method M can solve. The relative efficiency and reliability of each method can be directly seen from the performance profiles: the higher is the particular curve the better is the corresponding method.

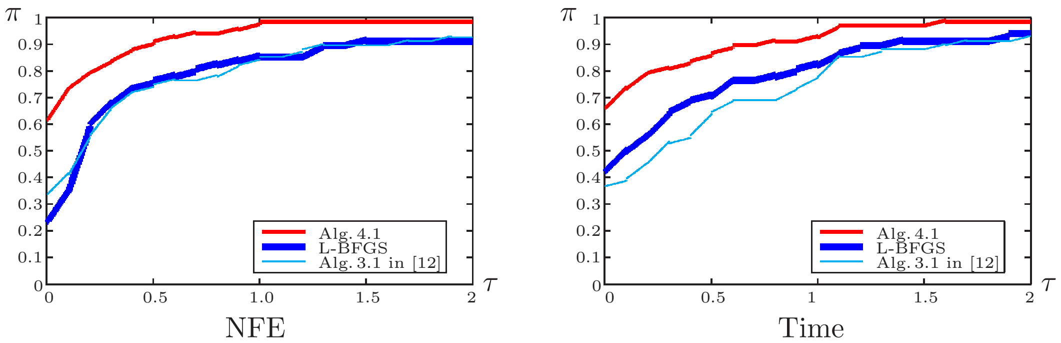


Figure 1: (Test 25, $m = 5$, 68 problems, N=10 000)

Acknowledgements

This work was supported by RVO: 67985807.

References

[1] Andrei, N.: An unconstrained optimization test functions collection. Advanced Modeling and Optimization **10** (2008), 147–161.

[2] Bongartz, I., Conn, A. R., Gould, N., and Toint, P. L.: CUTE: constrained and unconstrained testing environment. ACM Trans. Math. Software **21** (1995), 123–160.

[3] Dolan, E. D. and Moré, J. J.: Benchmarking optimization software with performance profiles. Math. Prog. **91** (2002) 201–213.

[4] Fletcher, R.: *Practical Methods of Optimization*. John Wiley & Sons, Chichester, 1987.

[5] Liu, D. C. and Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Prog. **45** (1989) 503–528.

[6] Lukšan, L., Matonoha, C., and Vlček, J.: Sparse test problems for unconstrained optimization. Report V-1064, ICS AS CR, Prague, 2010.

[7] Lukšan L., Matonoha C., Vlček J.: Modified CUTE problems for sparse unconstrained optimization. Report V-1081, ICS AS CR, Prague, 2010.

[8] Lukšan, L. and Spedicato, E.: Variable metric methods for unconstrained optimization and nonlinear least squares. J. Comput. Appl. Math. **124** (2000) 61–95.

[9] Moughrabi, I. A.: New implicit multistep quasi-Newton methods. Numerical Analysis and Applications **2** (2009) 154–164.

[10] Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comp. **35** (1980) 773–782.

[11] Nocedal, J. and Wright, S. J.: *Numerical optimization*. Springer-Verlag, New York, 1999.

[12] Vlček, J., Lukšan, L.: Limited-memory variable metric methods that use quantities from the preceding iteration. Proc. of Semin. PANM 15, D. Maxov, 2010.

[13] Vlček, J. and Lukšan, L.: A conjugate directions approach to improve the limited-memory BFGS method. Appl. Math. Comput. **219** (2012), 800–809.

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

CALCULATION OF THE GREATEST COMMON DIVISOR OF PERTURBED POLYNOMIALS

Jan Zítko, Ján Eliaš

Department of Numerical Mathematics,
Faculty of Mathematics and Physics, Charles University
Sokolovská 83, Prague 8, Czech Republic
zitko@karlin.mff.cuni.cz, janelias@ymail.com

Abstract

The coefficients of the greatest common divisor of two polynomials f and g $(\mathrm{GCD}(f,g))$ can be obtained from the Sylvester subresultant matrix $S_j(f,g)$ transformed to lower triangular form, where $1 \le j \le d$ and $d = \deg(\mathrm{GCD}(f,g))$ needs to be computed. Firstly, it is supposed that the coefficients of polynomials are given exactly. Transformations of $S_j(f,g)$ for an arbitrary allowable j are in details described and an algorithm for the calculation of the $\mathrm{GCD}(f,g)$ is formulated. If inexact polynomials are given, then an approximate greatest common divisor (AGCD) is introduced. The considered techniques for an AGCD computations are shortly discussed and numerically compared in the presented paper.

1. Introduction

Consider the polynomials f and g,

$$f(x) = a_0x^m + a_1x^{m-1} + \cdots + a_{m-1}x + a_m, \quad a_0 \times a_m \neq 0, \tag{1}$$
$$g(x) = b_0x^n + b_1x^{n-1} + \cdots + b_{n-1}x + b_n, \quad b_0 \times b_n \neq 0. \tag{2}$$

In the first part of this paper it is assumed that the coefficients are given exactly, all calculations are performed symbolically and $m \ge n$. Let us put $f_0 := f$, $f_1 := g$. The polynomials

$$\begin{aligned}&f_r(x) = q_r(x)f_{r+1}(x) + f_{r+2}(x), \quad \deg(f_{r+2}) < \deg(f_{r+1}),\\ &\text{for } r = 0,1,2,\ldots, \quad f_r \neq 0\,\forall r \le k\end{aligned}$$

in the successive divisions of Euclid's algorithm are well defined, [1, 7, 15]. If $f_{k+1} = 0$ then f_k is the GCD of f_0 and f_1, which is written as $f_k = \mathrm{GCD}(f_0, f_1) = \mathrm{GCD}(f, g)$.

The Sylvester matrix $S(f,g) \in \mathbb{R}^{(m+n)\times(m+n)}$, [1, 3, 4, 7, 12, 13, 15], is the matrix

$$S(f,g) \;=\; \begin{array}{c}\left[\begin{array}{cccccccc} a_0 & & & & b_0 & & & \\ a_1 & a_0 & & & b_1 & b_0 & & \\ \cdot & a_1 & \cdot & & \cdot & b_1 & \cdot & \\ \cdot & \cdot & \cdot & a_0 & \cdot & \cdot & \cdot & b_0 \\ a_m & \cdot & \cdot & a_1 & b_n & \cdot & \cdot & b_1 \\ & a_m & \cdot & \cdot & & b_n & \cdot & \cdot \\ & & \cdot & \cdot & & & \cdot & \cdot \\ & & & a_m & & & & b_n \end{array}\right] \\ \underbrace{\hphantom{a_0\quad a_0\quad a_0\quad a_0}}_{n \text{ columns}} \; \underbrace{\hphantom{b_0\quad b_0\quad b_0\quad b_0}}_{m \text{ columns}} \end{array}.$$

Let j be an integer, $1 \le j \le n$. The jth Sylvester subresultant matrix $S_j(f,g) \in \mathbb{R}^{(m+n-j+1)\times(m+n-2j+2)}$ is formed by deleting the last $(j-1)$ rows, and the last $(j-1)$ columns of the coefficients of f and g of $S(f,g)$. The vector e_i denotes the ith column of the identity $r \times r$ matrix I_r, and the matrix $E_{i,j}(\sigma) = I_r - \sigma e_i e_j^T$, where $\sigma \in \mathbb{R}$, is the elementary triangular matrix. It is lower and upper triangular matrix for $i \ge j$ and $i \le j$, respectively.

Transformations of the Sylvester subresultant matrix $S_j(f,g)$ that correspond to the first stage of Euclid's algorithm can be expressed by multiplying $S_j(f,g)$ by the elementary triangular matrices. The polynomial f_2 arises from the first stage. For illustration, let us consider the Sylvester resultant matrix $S_2 := S_2(f,g)$ for the polynomials f and g of degrees $m = 6$ and $n = 3$.

The first step in the transformation of S_2 consists of the subtraction of the third and fourth column, multiplied by $\sigma_1 = a_0/b_0$, from the first and second column, respectively. This is implemented in such a way that the matrix S_2 is multiplied successively by the matrices $E_{3,1}(\sigma_1)$ and $E_{4,2}(\sigma_1)$ yielding $S_2^{(1)} := S_2 E_{3,1}(\sigma_1) E_{4,2}(\sigma_1)$,

$$S_2^{(1)} = \left[\begin{array}{ccccccc} 0 & & b_0 & & & & \\ a_1^{(1)} & 0 & b_1 & b_0 & & & \\ a_2^{(1)} & a_1^{(1)} & b_2 & b_1 & b_0 & & \\ a_3^{(1)} & a_2^{(1)} & b_3 & b_2 & b_1 & b_0 & \\ a_4^{(1)} & a_3^{(1)} & & b_3 & b_2 & b_1 & b_0 \\ a_5^{(1)} & a_4^{(1)} & & & b_3 & b_2 & b_1 \\ a_6^{(1)} & a_5^{(1)} & & & & b_3 & b_2 \\ & a_6^{(1)} & & & & & b_3 \end{array}\right],$$

where

$$a_i^{(1)} = \begin{cases} a_i - \underbrace{(a_0/b_0)}_{\sigma_1} b_i & i = 1,2,3 \\ a_i & i = 4,5,6. \end{cases}$$

Analogously, the numbers σ_2, σ_3 and σ_4 can be constructed such that the firs two columns of the matrix $S_2^{(4)}$, where successively

$$S_2^{(2)} = S_2^{(1)} E_{4,1}(\sigma_2) E_{5,2}(\sigma_2), \quad S_2^{(3)} = S_2^{(2)} E_{5,1}(\sigma_3) E_{6,2}(\sigma_3), \quad S_2^{(4)} = S_2^{(3)} E_{6,1}(\sigma_4) E_{7,2}(\sigma_4),$$

contain the elements $0, 0, 0, 0, a_4^{(4)}, a_5^{(4)}, a_6^{(4)}$ [1] at the locations of $0, a_1^{(1)}, a_2^{(1)}, a_3^{(1)}, a_4^{(1)}, a_5^{(1)}, a_6^{(1)}$ of $S_2^{(1)}$.

[1] The upper index, e.g. $a_4^{(4)}$, specifies that the coefficients belong to the matrix $S_2^{(4)}$.

Then the permutation matrix $P = [e_3, e_4, e_5, e_6, e_7, e_1, e_2] \in \mathbb{R}^{7\times 7}$ applied to $S_2^{(4)}$ gives

$$S_2^{(4)}P = \left[\begin{array}{cccc|ccc} b_0 & & & & 0 & 0 & 0 \\ b_1 & b_0 & & & 0 & 0 & 0 \\ b_2 & b_1 & b_0 & & 0 & 0 & 0 \\ b_3 & b_2 & b_1 & b_0 & 0 & 0 & 0 \\ \hline 0 & b_3 & b_2 & b_1 & b_0 & a_4^{(4)} & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & a_5^{(4)} & a_4^{(4)} \\ 0 & 0 & 0 & b_3 & b_2 & a_6^{(4)} & a_5^{(4)} \\ 0 & 0 & 0 & 0 & b_3 & 0 & a_6^{(4)} \end{array}\right] = \left[\begin{array}{c|c} L_{1,1} & 0 \\ \hline L_{2,1} & L_{2,2} \end{array}\right]$$

where $L_{2,2} = S_2(g, f_2)$ and $f_2(x) = a_4^{(4)}x^2 + a_5^{(4)}x + a_6^{(4)}$ is the first nonzero polynomial produced by Euclid's algorithm if $f_2 \neq 0$. In this case the matrix $L_{1,1}$ is square, lower triangular and nonsingular.

The following four cases may happen:

1. $f_2 = 0$, i.e. $a_4^{(4)} = a_5^{(4)} = a_6^{(4)} = 0$. Then g divides f and the matrix $S_2^{(4)}P$ without any block structure is lower triangular matrix having two last zero columns.

2. $a_4^{(4)} \neq 0$ and f_2 divides g. Then elementary matrices applied to $L_{2,2}$ transform $L_{2,2}$ to the matrix $S_{2,\star}^{(4)}$. Hence, the matrices $S_2^{(4)}$ and S_2 are rank deficient of order 1. In this case $n_2 := \deg(\mathrm{GCD}(f, g)) = 2$.

$$S_{2,\star}^{(4)} = \begin{bmatrix} a_4^{(4)} & 0 & 0 \\ a_5^{(4)} & a_4^{(4)} & 0 \\ a_6^{(4)} & a_5^{(4)} & 0 \\ 0 & a_6^{(4)} & 0 \end{bmatrix}$$

3. $a_4^{(4)} \neq 0$ and f_2 does not divide g. Then elementary matrices applied to $L_{2,2}$ transform $L_{2,2}$ to the lower triangular matrix having linearly independent columns..

4. $a_4^{(4)} = 0$ but $f_2 \neq 0$. Then the matrix $S_2^{(4)}(f, g)$ can be transformed into the form

$$\tilde{S}_2^{(4)} = \left[\begin{array}{ccccc|cc} b_0 & & & & & 0 & 0 \\ b_1 & b_0 & & & & 0 & 0 \\ b_2 & b_1 & b_0 & & & 0 & 0 \\ b_3 & b_2 & b_1 & b_0 & & 0 & 0 \\ \hline 0 & b_3 & b_2 & b_1 & b_0 & 0 & 0 \\ 0 & 0 & b_3 & b_2 & b_1 & a_5^{(4)} & 0 \\ 0 & 0 & 0 & b_3 & b_2 & a_6^{(4)} & a_5^{(4)} \\ 0 & 0 & 0 & 0 & b_3 & 0 & a_6^{(4)} \end{array}\right]$$

and no other polynomials can be calculated in Euclid's algorithm in the last two cases. The matrices $S_2^{(4)}(f, g)$ and S_2 have full column rank.

In general, if the Sylvester subresultant $S_j(f,g)$ has full column rank, we have to go back to $S_{j-1}(f,g), S_{j-2}(f,g), \ldots$ as long as the rank deficient matrix appears. If $S_1(f,g) = S(f,g)$ has full column rank, then f and g are coprime.
Just presented example is generalized in the following section. The results are original.

2. Matrix formulation for the transformation of the Sylvester subresultant matrix

Let us denote $f_0 := f$ and $f_1 := g$, where f and g are defined in (1) and (2), respectively. Denote $n_0 := m = \deg(f_0)$, $n_1 := n = \deg(f_1)$.

Let us assume that the matrices $S_j(f_0, f_1)$, $S_j(f_1, f_2), \ldots$ can be constructed by Euclid's algorithm for an index j. According to our previous example, the following theorem can be easily seen. Let us write shortly $S_j := S_j(f_0, f_1)$.

Theorem 1. *Let f_0 and f_1 be polynomials of degrees n_0 and n_1, respectively, $n_0 \geq n_1 \geq 1$. It is assumed that Euclid's algorithm yields the polynomials f_2, $f_3, \ldots, f_k$, $f_{k+1} = 0$ of degrees $n_2, n_3, \ldots, n_k$. Therefore $f_k = GCD(f_0, f_1)$. Denote $d := n_k$ and $f_k(x) = v_0x^d + v_1x^{d-1} + \cdots + v_{d-1}x + v_d$. Consider an integer $j \in \{1, 2, \ldots, n\}$. Then the following statements hold:*

1) *There exists a nonsingular matrix Q_j of order $n_0 + n_1 - 2j + 2$ such that the matrix S_jQ_j has the following block structure. We distinguish two cases:*

a) *If $j \leq d$, then*

$$S_jQ_j = \left[\begin{array}{ccc} L_{1,1} & | & 0 \\ - & + & - \\ L_{2,1} & | & L_{2,2} \end{array}\right],$$

where $L_{1,1}$ is a square lower triangular matrix with non-zero diagonal elements and $L_{2,2}$ is a rectangular matrix with $(n_{k-1} + n_k - 2j + 2)$ columns if $f_2 \neq 0$. Contrariwise if $f_2 = 0$ then g divides f and the matrix S_jQ_j is lower triangular matrix having last $n_1 - j + 1$ zero columns. In the following let $f_2 \neq 0$. Then the matrix $L_{2,2}$ has the following form:

(i) case when $j \leq d$

$$L_{2,2} = \left[\begin{array}{cccc|ccc} v_0 & & & & 0 & . & 0 \\ v_1 & v_0 & & & 0 & . & 0 \\ . & v_1 & . & & 0 & . & 0 \\ v_d & . & . & v_0 & 0 & . & 0 \\ & v_d & . & v_1 & 0 & . & 0 \\ & & . & . & 0 & . & 0 \\ & & & v_d & 0 & . & 0 \end{array}\right]$$
$$\underbrace{\qquad\qquad\qquad}_{n_{k-1}-j+1} \quad \underbrace{\qquad\quad}_{n_k-j+1}$$

(ii) special case when $j = d$

$$L_{2,2} = \left[\begin{array}{cccc|c} v_0 & & & & 0 \\ v_1 & v_0 & & & 0 \\ . & v_1 & . & & 0 \\ v_d & . & . & v_0 & 0 \\ & v_d & . & v_1 & 0 \\ & & . & . & 0 \\ & & & v_d & 0 \end{array}\right]$$
$$\underbrace{\qquad\qquad\qquad}_{n_{k-1}-n_k+1} \quad \underbrace{\quad}_{1}$$

Moreover, the presented scheme of matrices (i) and (ii) shows that

$$\begin{aligned} \mathit{rank}(S_j) = \mathit{rank}(Q_jS_j) &= n_0 + n_1 - 2(j-1) - (n_k - j + 1) \\ &= n_0 + n_1 - j - n_k + 1 \end{aligned}$$

and the nonzero columns of the matrix $L_{2,2}$ *contain the coefficients of the polynomial* f_k*. In case* $j = d = n_k$*, the matrix* S_d *is rank deficient of order 1.*

b) *If* $j > d$*, then* S_jQ_j *is a lower triangular matrix with linearly independent columns. Hence,* S_jQ_j *and therefore* S_j *has full column rank.*

2) *If* $n_k = 0$*, then the matrix* $S_1(f_0, f_1)$ *having full rank* $n_0 + n_1$ *is only considered,* $f_k = v_0 \neq 0$ *and* $L_{2,2} = v_0 I_{n_{k-1}}$*.*

3) *The next equivalences follow from the statements formulated above:*

$$\begin{aligned} &\mathrm{rank}(S_d(f_0, f_1)) = n_0 + n_1 - 2d + 1 \Leftrightarrow \deg\,(\mathrm{GCD}(f_0, f_1)) = d, \\ &\mathrm{rank}(S_j(f_0, f_1)) < n_0 + n_1 - 2j + 1 \Leftrightarrow \deg(\mathrm{GCD}(f_0, f_1)) > j. \end{aligned}$$

Just presented overview shows the relation between the rank(S_j) and the degree of GCD(f_0, f_1). Hence if the polynomials f_0 and f_1 are known exactly and the computations are performed symbolically, then the transformation of the Sylvester subresultant matrix $S_j(f_0, f_1)$, $j \leq d$, to the lower triangular form with the resultant matrix $L_{2,2}$ yields the coefficients of the GCD(f_0, f_1).

3. Calculation of GCD

Consider the polynomials f and g in (1) and (2) of degrees $m = \deg(f_0)$ and $n = \deg(f_1)$, and put $f_0 = f$ and $f_1 = g$. Let h be the exact GCD(f_0, f_1) with $d = \deg(h)$. There exist two polynomials w_0 and w_1 so that

$$f_i = hw_i \text{ for } i = 0, 1, \quad \text{where} \quad \deg(w_0) = m - d, \quad \deg(w_1) = n - d.$$

Hence $h = f_0/w_0 = f_1/w_1 \Rightarrow f_0w_1 - f_1w_0 = 0$. Using Cauchy matrices, we can rewrite the last equality in the form

$$C_{n-d+1}(f_0)\vec{w}_1 - C_{m-d+1}(f_1)\vec{w}_0 = \underbrace{[C_{n-d+1}(f_0), C_{m-d+1}(f_1)]}_{S_d} \begin{bmatrix} \vec{w}_1 \\ -\vec{w}_0 \end{bmatrix} = \vec{0}, \qquad (3)$$

where the vectors of coefficients of the polynomials w_1, w_0 are denoted by $\vec{w}_1$ and $\vec{w}_0$. The matrix $S_d = [C_{n-d+1}, C_{m-d+1}] \in \mathbb{R}^{(m+n-d+1)\times(m+n-2d+2)}$ is rank deficient of order 1. The solution of (3) is the right singular vector corresponding to $\sigma_{min}(S_d(f_0, f_1))$ and can be computed by the Gauss-Newton iteration, see for example [2, 3, 8]. The coefficients of h are calculated as the least square solution of the equation

$$C_{d+1}(w_1)\vec{h} = \vec{f_1} \quad \text{or} \quad C_{d+1}(w_0)\vec{h} = \vec{f_0}.$$

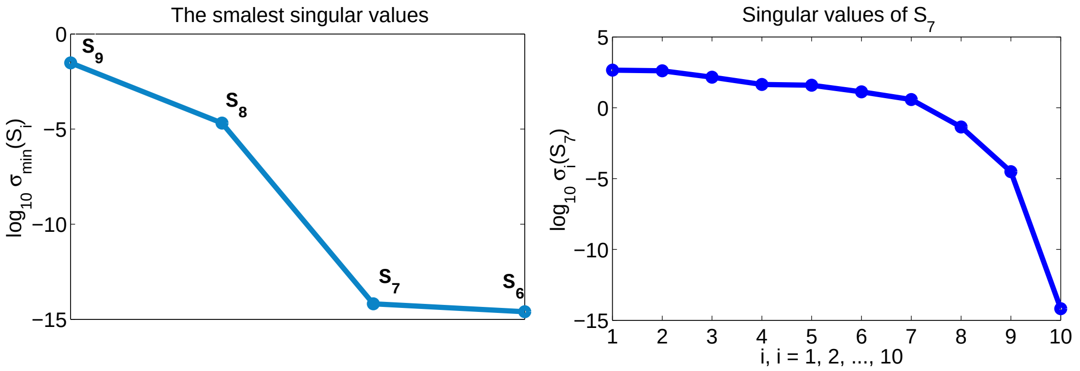


Figure 1: In the following graphs the smallest singular values of the Sylvester subresultant matrices S_9, S_8, S_7 and S_6, left-hand side, and the singular values of S_7, right-hand side, are drawn.

Let us demonstrate the mentioned theory on the following polynomials

$$f_0(x) = (x-1.2)^4(x+2)^5(x-0.5)^4, \quad f_1(x) = (x-1.4)^2(x+2)^3(x-0.5)^4 \qquad (4)$$

of degrees $\deg(f_0) = 13$ and $\deg(f_1) = 9$. Their GCD is the polynomial $\text{GCD}(f_0, f_1) = h(x) = (x+2)^3(x-0.5)^4 = x^7 + 4x^6 + 1.5x^5 - 7.5x^4 - 0.9375x^3 + 6.375x^2 - 3.25x + 0.5$ of degree $\deg(h) = d = 7$. Theorem 1 says that S_7 is the first rank deficient matrix in the sequence S_9, S_8, S_7. For illustration see Figure 1.

The matrix S_7 is the first rank deficient matrix with the smallest singular value 7.1678_{10}^{-14} and the corresponding right singular vector

$$[-0.1090, 0.3051, -0.2135, 0.1090, -0.0872, -0.7147, 0.9204, 0.9790, -2.1086, 0.9037]^T.$$

The LS solution of $C_8(\vec{w_1})\vec{h} = \vec{f_1}$ yields the coefficients of the $\text{GCD}(f_0, f_1) = \vec{h} = [1, 4, 1.5, -7, 5, -0.9375, 6.375, -3.25, 0.5]^T$. The LS solution of the system $C_8(-\vec{w}_0)\vec{h} = \vec{f_0}$ yields the same vector $\vec{h} = [1, 4, 1.5, -7, 5, -0.9375, 6.375, -3.25, 0.5]^T$.

4. Approximate greatest common divisor

It was assumed that the coefficients of polynomials are given exactly and the calculations are performed symbolically. But the calculation of the GCD is unstable in a computer environment and cannot be almost used. Moreover, numerical computation of the GCD is an ill-posed problem. Therefore the concept of an approximate greatest common divisor (AGCD) was introduced [3, 6, 13, 14].

Definition. Let f and g be two polynomials of degrees m and n, respectively, and let $0 < \theta << 1$ be a positive number. The degree of an approximate greatest common divisor with respect to θ is the maximum integer $j \leq \min(m, n)$ for which there exist polynomials δf and δg with $\max(\|\delta f\|, \|\delta g\|) \leq \theta$ and $\deg(\text{GCD}(f + \delta f, g + \delta g) = j$. The approximate greatest common divisor denoted by $\text{AGCD}(f, g)$ is defined by $\text{AGCD}(f, g) = \text{GCD}(f + \delta f, g + \delta g)$.

Algorithms for the calculation of δf and δg are well known. However they are out of scope of this paper and cannot be analysed in this paper. Let us only mention the Structured Total Least Norm (STLN) method (see, for example, [10, 5, 13]) for the construction of a structured low rank approximation of the full rank Sylvester matrix in the AGCD approach.

For demonstration, let us again consider the polynomials from Section 3 and let us denote them by $\hat{f}$ and $\hat{g}$. Their exact GCD is the polynomial

$$\mathrm{GCD}(\hat{f},\hat{g}) = x^7 + 4x^6 + 1.5x^5 + 7.5x^4 - 0.9375x^3 + 6.375x^2 - 3.25x + 0.5.$$

Let f and g be inexact forms of $\hat{f}$ and $\hat{g}$, i.e. the polynomials $\hat{f}$ and $\hat{g}$ with a noise expressed by a signal-to-noise ratio equal to 10^6 added to their coefficients. The polynomials that arise from the application of the STLN method are denoted by $\tilde{f}$ and $\tilde{g}$. The schema of this process is as follows.

$$\left\{\begin{array}{c}\hat{f}(x)\\ \hat{g}(x)\end{array}\right\} \overset{\text{perturbation}}{- - - - \rightarrow} \left\{\begin{array}{c}f(x)\\ g(x)\end{array}\right\} \overset{\text{STLN}}{- - - - \rightarrow} \left\{\begin{array}{c}\tilde{f}(x)\\ \tilde{g}(x)\end{array}\right\}$$

The polynomials f and g are theoretically coprime and the procedure that follows from Theorem 1 fails in the presence of greater noise. However, we can see from the table below that the coefficients of $\mathrm{GCD}(\hat{f},\hat{g})$ and $\mathrm{GCD}(\tilde{f},\tilde{g})$ of the polynomials computed by STLN are almost identical.

	$\mathrm{GCD}(\hat{f},\hat{g})$	$\mathrm{GCD}(\tilde{f},\tilde{g})$
x^7	1	1
x^6	4	3.999978
x^5	1.5	1.499947
x^4	-7.5	-7.500006
x^3	-0.9375	-0.937463
x^2	6.375	6.375001
x^1	-3.25	-3.250011
x^0	0.5	0.499999

Acknowledgements

This work was supported by the grant prvouk p47. The authors thank for this support.

References

[1] Barnett, S.: *Polynomials and linear control systems.* Marcel Dekker, INC., New York and Basel, 1983.

[2] Björk, Å.: *Numerical method for least square problems.* SIAM, Philadelphia, 1996.

[3] Corless, R. M., Gianni, P. M., Trager, B. M., and Watt, S. M.: The singular value decomposition for polynomial systems. In: *Proc. ISSAC 95*, pp. 195–20. ACM Press 1995.

[4] Eliaš, J.: *Problémy spojené s výpočtem největšího společného dělitele.* Bachelor thesis, Charles University, Faculty of Mathematics and Physics, 2009.

[5] Kaltofen, E., Yang, Z., and Zhi, L.: Structured low rank approximation of Sylvester matrix. Preprint, 2005.

[6] Kuřátko, J.: *Analysis of computing the greatest common divisors of polynomials.* Master thesis, Charles University, Faculty of Mathematics and Physics, 2012.

[7] Leidacker, M. A.: Another theorem relating Sylvester's matrix and the greatest common divisor. Mathematics Magazine **42**, No 3, (1969), pp. 126-128.

[8] Li, T. Y. and Zeng, Z.: A rank-revealing method with updating, downdating and applications. SIAM J. Matrix Anal. Appl. **26** (4) (2005), 918–946.

[9] Lee, T. L., Li, T. Y., and Zeng, Z.: A rank-revealing method with updating, downdating and applications. Part II. SIAM J. Matrix Anal. Appl. **31** (2) (2009), 503–525.

[10] Rosen, J. B., Park, H., and Glick, J.: Total least norm formulation and solution for structured problems. SIAM J. on Matrix Anal. Appl. **17** (1) (1996), 110–128.

[11] Saad, Y.: *Numerical methods for large eigenvalue problems.* Halstead Press, New York, 1992.

[12] Winkler, J. R. and Zítko, J.: Some questions associated with the calculation of the GCD of two univariate polynomials. In: *Winter School and SNA'07*, pp. 130–137. Ostrava, 2007.

[13] Winkler, J. R., and Allan, J. D.: Structured total least norm and approximate GCDs of inexact polynomials. J. Comput. Appl. Math. **215** (2006), 1–13.

[14] Zeng, Z.: The approximate GCD od inexact polynomials, Part I: univariate algorithm. Preprint, 2004.

[15] Zítko, J. and Eliaš, J.: Application of the rank revealing algorithm for the calculation of the GCD. In: *Winter School and SNA'12*, pp. 175–180. Liberec, 2012.

List of participants

Stanislav Bartoň, barton@mendelu.cz
Ústav techniky a automobilové dopravy, Agronomická fakulta, Mendelova Univerzita v Brně

Daniela Bímová, daniela.bimova@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Marek Brandner, brandner@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Pavel Burda, pavel.burda@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Marta Čertíková, marta.certikova@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Josef Dalík, dalik.j@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně
Josef Dalík suddenly passed away on January 20, 2013.

Tomáš Dohnal, tomas.dohnal@kit.edu
Institute of Applied and Numerical Mathematics, Karlsruhe Institute of Technology, Karlsruhe, Germany

Jiří Egermaier, jirieggy@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Ján Eliaš, janelias@ymail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Miloslav Feistauer, feist@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Václav Finěk, vaclav.finek@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Cyril Fischer, fischerc@itam.cas.cz
Ústav teoretické a aplikované mechaniky AV ČR, v. v. i., Praha

Milan Hanuš, mhanus@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Martin Horák, Martin.Horak@fsv.cvut.cz
Katedra mechaniky, Fakulta stavební ČVUT v Praze

Jiří Hozman, jiri.hozman@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Jan Chleboun, chleboun@mat.fsv.cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Pavol Chocholatý, chocholaty@fmph.uniba.sk
Katedra matematickej analýzy a numerickej matematiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava, Slovenská republika

Radka Keslerová, keslerov@marian.fsik.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Hana Kopincová, kopincov@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Jiří Krček, jiri.krcek@vsb.cz
Katedra matematiky a deskriptivní geometrie, Vysoká škola báňská – Technická univerzita Ostrava

Tomáš Krumpholc, xkrumpho@node.mendelu.cz
Ústav techniky a automobilové dopravy, Agronomická fakulta, Mendelova Univerzita v Brně

Lukáš Krupička, LukasKrupicka@seznam.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Václav Kučera, vaclav.kucera@email.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Vojtěch Kumbár, vojtech.kumbar@mendelu.cz
Ústav techniky a automobilové dopravy, Agronomická fakulta, Mendelova Univerzita v Brně

Pavel Kůs, pavel.kus@gmail.com
Ústav termomechaniky AV ČR, v. v. i., Praha

Ladislav Lukšan, luksan@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

František Mach, fmach@kte.zcu.cz
Katedra teoretické elektrotechniky, Fakulta elektrotechnická, Západočeská univerzita v Plzni

Ivo Marek, marek@mbox.ms.mff.cuni.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Ctirad Matonoha, matonoha@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Petr Mayer, pmayer@mat.fsv.cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Jaroslav Mlýnek, jaroslav.mlynek@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Vratislava Mošová, vratislava.mosova@mvso.cz
Ústav exaktních věd, Moravská vysoká škola Olomouc

Štěpán Papáček, spapacek@frov.jcu.cz
Škola komplexních systémů, Fakulta rybářství a ochrany vod, Jihočeská univerzita v Českých Budějovicích

Martin Plešinger, martin.plesinger@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Lukáš Pospíšil, lukas.pospisil@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, VŠB - Technická univerzita Ostrava

Oto Přibyl, pribyl.o@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Jan Přikryl, prikryl@utia.cas.cz
Ústav teorie informace a automatizace AV ČR, v. v. i., Praha

Vladimír Prokop, prouki@seznam.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Ivana Pultarová, ivana@mat.fsv.cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Stefan Ratschan, stefan.ratschan@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Petr Salač, petr.salac@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Karel Segeth, segeth@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Martina Šimůnková, martina.simunkova@tul.cz
Katedra aplikované matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Jakub Šístek, sistek@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Pavel Šolín, solin.pavel@gmail.com
Ústav termomechaniky AV ČR, v. v. i., Praha

Jindřich Soukup, soukup@frov.jcu.cz
Škola komplexních systémů, Fakulta rybářství a ochrany vod, Jihočeská univerzita v Českých Budějovicích

Petr Sváček, Petr.Svacek@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Jiří Taufer, taufer@fd.cvut.cz
Ústav aplikované matematiky, Fakulta dopravní ČVUT v Praze

Petr Tichý, tichy@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Jiří Vala, Vala.J@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Tomáš Vejchodský, vejchod@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Miloslav Vlasák, vlasak@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Jan Vlček, vlcek@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Jan Zeman, zemanj@cml.fsv.cvut.cz
Katedra mechaniky, Fakulta stavební ČVUT v Praze

Jan Zítko, zitko@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha