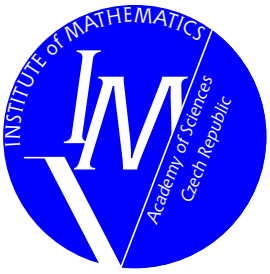
Programs and Algorithms of Numerical Mathematics 17

Dolní Maxov, June 8–13, 2014

Proceedings of Seminar

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



Institute of Mathematics
Academy of Sciences of the Czech Republic
Prague 2015

ISBN 978-80-85823-64-6
Matematický ústav AV ČR, v. v. i.
Praha 2015

Contents


HOTEL MAXOV
KRUŠOVICE
PANM 17
8.-13.6.2014
RULE
THE
EMPIRE

Preface

This volume comprises peer-reviewed papers that are based on invited lectures, survey lectures, short communications, and posters presented at the 17th seminar Programs and Algorithms of Numerical Mathematics (PANM) held in Dolní Maxov, Czech Republic, June 8–13, 2014.

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 (with only one exception) biannually in Alšovice, Bratříkov, Janov nad Nisou, Kořenov, Lázně Libverda, Dolní Maxov, and Prague in the period 1983–2012. The objective of this series of seminars is to provide a forum for presenting and discussing advanced topics in numerical analysis, single- or multi-processor applications of computational methods, and new approaches to mathematical modeling.

More than 60 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 Slovakia and the United States. We appreciate the traditional participation of a significant number of young scientists, PhD students, and also some undergraduate students at the PANM seminar. We wish to believe that also those, who took part in the seminar for the first time, have found the atmosphere of the seminar friendly and working, and will join the PANM community.

The organizing committee consisted of Jan Chleboun, Petr Přikryl, Karel Segeth, Jakub Šístek, and Tomáš Vejchodský. Ms Hana Bílková kindly prepared the electronic version of the book.

All papers have been reproduced directly from materials submitted by the authors. Naturally, an attempt has been made to unify the layout of papers. A unique feature of this volume of proceedings is a photograph of the participants in front of Maxov Hotel.

By chance, we have found an undated postcard with Maxov Hotel and vicinity. We guess from several clues that this winter photograph was taken in about 1962. We believe that the members of the PANM seminar will find this photograph interesting and we publish it in the proceedings, too.

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

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

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

STABILITY ANALYSIS OF THE SPACE-TIME DISCONTINUOUS GALERKIN METHOD FOR NONSTATIONARY NONLINEAR CONVECTION-DIFFUSION PROBLEMS

Monika Balázsová, Miloslav Feistauer, Martin Hadrava, Adam Kosík

Faculty of Mathematics and Physics, Charles University in Prague
Sokolovská 83, 186 75 Praha 8, Czech Republic
b.moncsi@gmail.com, feist@karlin.mff.cuni.cz, martin@hadrava.eu, adam.kosik@atlas.cz

Abstract

This paper is concerned with the stability analysis of the space-time discontinuous Galerkin method for the solution of nonstationary, nonlinear, convection-diffusion problems. In the formulation of the numerical scheme we use the nonsymmetric, symmetric and incomplete versions of the discretization of diffusion terms and interior and boundary penalty. Then error estimates are briefly characterized. The main attention is paid to the investigation of unconditional stability of the method. Theoretical results are demonstrated by a numerical example.

1. Introduction

One of efficient and robust techniques for the numerical solution of partial differential equations is the discontinuous Galerkin (DG) method. It is based on piecewise polynomial approximations of the sought exact solution over a partition of the computational domain without any requirement of the continuity on interfaces between neighbouring elements. Most of works on the DG method are concerned with space discretization. The numerical simulation of strongly nonstationary transient problems requires the application of numerical schemes of high order of accuracy both in space and in time. For some applications, the standard Euler schemes or θ-schemes are not sufficiently accurate in time. In computational fluid dynamics, Runge-Kutta methods are very popular ([3]). However they are conditionally stable. It appears suitable to use the discontinuous Galerkin discretization with respect to space as well as time for the construction of numerical schemes with high accuracy in space and time for the solution of nonlinear nonstationary problems. The discontinuous Galerkin time discretization was introduced and analyzed e.g. in [4] for the solution of ordinary differential equations. In [6] it was combined with conforming finite elements and applied to parabolic problems. See also the monograph [7].

The papers [2] and [5] are concerned with theoretical analysis of error estimates for the space-time DG method applied to nonlinear nonstationary convection-diffusion

problems. However, in a general case the results were obtained under a CFL-like stability condition applied in the vicinity of the boundary. There is a natural question, if this condition is really necessary for guaranteeing the stability. This was the motivation for the investigation of the stability of the space-time DG method. In this paper we present a brief description of the obtained results. The analysis is rather complicated and technical and detailed proofs will be published in [1].

2. Formulation of the continuous problem

Let $\Omega \subset \mathbb{R}^d, d = 2, 3$, be a bounded domain and $T > 0$. We consider the initial-boundary value problem to find $u : Q_T = \Omega \times (0, T) \to \mathbb{R}$ such that

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} - \operatorname{div}(\beta(u)\nabla u) = g \quad \text{in } Q_T, \tag{1}$$

$$u\,|_{\partial\Omega\times(0,T)} = u_D, \tag{2}$$

$$u(x, 0) = u^0(x), \quad x \in \Omega. \tag{3}$$

We assume, that g, u_D, u^0, f_s are given functions and $f_s \in C^1(\mathbb{R}), |f_s'| \le C$, $f_s(0) = 0$, $s = 1, \dots, d$. Moreover, let the function $\beta : \mathbb{R} \to [\beta_0, \beta_1]$, $0 < \beta_0 < \beta_1 < \infty$, be Lipschitz continuous: $|\beta(u_1) - \beta(u_2)| \le L_\beta |u_1 - u_2|$ for all $u_1, u_2 \in \mathbb{R}$.

3. Space-time discretization

In the time interval $[0, T]$ we introduce a partition formed by time instants $0 = t_0 < t_1 < \ldots < t_M = T$, and denote $I_m = (t_{m-1}, t_m)$, $\tau_m = t_m - t_{m-1}$, $m = 1 \ldots, M$. We set $\tau = \max_{m=1,\dots,M} \tau_m$. For a function φ defined in $\bigcup_{m=1}^{M} I_m$ we denote one-sided limits at t_m as $\varphi_m^{\pm} = \varphi(t_m \pm) = \lim_{t\to t_m\pm} \varphi(t)$ and the jump as $\{\varphi\}_m = \varphi(t_m+) - \varphi(t_m-)$.

For each I_m we consider a system of partitions $\{\mathcal{T}_{h,m}\}_{h\in(0,h_0)}$ with $h_0 > 0$ of $\overline{\Omega}$ into a finite number of closed triangles with mutually disjoint interiors (partitions are in general different for different m). We set $h_K = \operatorname{diam}(K)$ for $K \in \mathcal{T}_{h,m}$, $h_m = \max_{K\in\mathcal{T}_{h,m}} h_K$ and $h = \max_{m=1,\dots,M} h_m$.

By $\mathcal{F}_{h,m}$ we denote the system of all faces of all elements $K \in \mathcal{T}_{h,m}$. It consists of the set of all inner faces $\mathcal{F}_{h,m}^I$ and the set of all boundary faces $\mathcal{F}_{h,m}^B$. Each $\Gamma \in \mathcal{F}_{h,m}$ will be associated with a unit normal vector $\mathbf{n}_\Gamma$. By $K_\Gamma^{(L)}$ and $K_\Gamma^{(R)} \in \mathcal{T}_{h,m}$ we denote the elements adjacent to the face $\Gamma \in \mathcal{F}_{h,m}$. We shall use the convention, that $\mathbf{n}_\Gamma$ is the outer normal to $\partial K_\Gamma^{(L)}$. Over a triangulation $\mathcal{T}_{h,m}$, for each positive integer k, we define the broken Sobolev space $H^k(\Omega, \mathcal{T}_{h,m}) = \{v; v|_K \in H^k(K)\, \forall K \in \mathcal{T}_{h,m}\}$.

If $v \in H^1(\Omega, \mathcal{T}_{h,m})$ and $\Gamma \in \mathcal{F}_{h,m}$, then $v|_\Gamma^{(L)}, v|_\Gamma^{(R)}$ will denote the traces of v on Γ from the side of the elements $K_\Gamma^{(L)}, K_\Gamma^{(R)}$ adjacent to Γ. For $\Gamma \in \mathcal{F}_{h,m}^I$ we set

$$\langle v\rangle_\Gamma \;=\; \frac{1}{2}\left(v|_\Gamma^{(L)} + v|_\Gamma^{(R)}\right), \quad [v]_\Gamma = v|_\Gamma^{(L)} - v|_\Gamma^{(R)}.$$

We use the notation

$$h(\Gamma) = \frac{h_{K_\Gamma^{(L)}} + h_{K_\Gamma^{(R)}}}{2} \quad \text{for}\, \Gamma \in \mathcal{F}_{h,m}^I, \quad h(\Gamma) = h_{K_\Gamma^{(L)}} \quad \text{for}\, \Gamma \in \mathcal{F}_{h,m}^B.$$

If $u, \varphi \in H^2(\Omega, \mathcal{T}_{h,m})$ and $c_W > 0$, we introduce the forms

$$\begin{aligned}
a_{h,m}(u,\varphi) &= \sum_{K\in\mathcal{T}_{h,m}} \int_K \beta(u)\nabla u \cdot \nabla\varphi \, dx \\
&- \sum_{\Gamma\in\mathcal{F}_{h,m}^I} \int_\Gamma \left(\langle \beta(u)\nabla u \rangle \cdot \mathbf{n}_\Gamma \, [\varphi] + \theta \, \langle \beta(u)\nabla\varphi \rangle \cdot \mathbf{n}_\Gamma \, [u] \right) dS \\
&- \sum_{\Gamma\in\mathcal{F}_{h,m}^B} \int_\Gamma \left(\beta(u)\nabla u \cdot \mathbf{n}_\Gamma \, \varphi + \theta\beta(u)\nabla\varphi \cdot \mathbf{n}_\Gamma \, u - \theta\beta(u)\nabla\varphi \cdot \mathbf{n}_\Gamma \, u_D \right) dS, \\
J_{h,m}(u,\varphi) &= c_W \sum_{\Gamma\in\mathcal{F}_{h,m}^I} h(\Gamma)^{-1} \int_\Gamma [u]\,[\varphi] dS + c_W \sum_{\Gamma\in\mathcal{F}_{h,m}^B} h(\Gamma)^{-1} \int_\Gamma u\,\varphi \, dS, \\
b_{h,m}(u,\varphi) &= - \sum_{K\in\mathcal{T}_{h,m}} \int_K \sum_{s=1}^d f_s(u) \frac{\partial\varphi}{\partial x_s} \, dx \\
&+ \sum_{\Gamma\in\mathcal{F}_{h,m}^I} \int_\Gamma H(u_\Gamma^{(L)}, u_\Gamma^{(R)}, \mathbf{n}_\Gamma) \, [\varphi] \, dS + \sum_{\Gamma\in\mathcal{F}_{h,m}^B} \int_\Gamma H(u_\Gamma^{(L)}, u_\Gamma^{(L)}, \mathbf{n}_\Gamma) \, \varphi \, dS, \qquad (4) \\
l_{h,m}(\varphi) &= \sum_{K\in\mathcal{T}_{h,m}} \int_K g\varphi \, dx + \beta_0 \, c_W \sum_{\Gamma\in\mathcal{F}_{h,m}^B} h(\Gamma)^{-1} \int_\Gamma u_D \, \varphi \, dS.
\end{aligned}$$

Let us note that in integrals over faces we omit the subscript Γ. We consider $\theta = 1$, $\theta = 0$ and $\theta = -1$ and get the symmetric (SIPG), incomplete (IIPG) and nonsymmetric (NIPG) variants of the approximation of the diffusion terms, respectively. In (4), H is a numerical flux, which is Lipschitz-continuous, consistent and conservative.

Let $p, q \geq 1$ be integers. For each $m = 1, \ldots, M$ we define the spaces

$$\begin{aligned}
S_{h,m}^p &= \{\varphi \in L^2(\Omega);\, \varphi|_K \in P^p(K) \quad \forall K \in \mathcal{T}_{h,m}\}, \\
S_{h,\tau}^{p,q} &= \{\varphi \in L^2(Q_T);\, \varphi|_{I_m} = \sum_{i=0}^q t^i \varphi_i \quad \text{with}\, \varphi_i \in S_{h,m}^p,\, m = 1, \ldots, M\}.
\end{aligned}$$

By $(\cdot,\cdot)$ and $\|\cdot\|$ we denote the scalar product and the norm in $L^2(\Omega)$. The symbol $|\cdot|_{H^1(K)}$ denotes the seminorm in the space $H^1(K)$. The space $H^1(\Omega, \mathcal{T}_{h,m})$ will be equipped with the norm

$$\|\varphi\|_{DG,m} = \left(\sum_{K\in\mathcal{T}_{h,m}} |\varphi|_{H^1(K)}^2 + J_{h,m}(\varphi,\varphi) \right)^{1/2}.$$

Definition. *We say that U is an approximate solution of (1)-(3), if $U \in S_{h,\tau}^{p,q}$ and*

$$\int_{I_m} \left(\left(\frac{\partial U}{\partial t}, \varphi \right) + a_{h,m}(u,\varphi) + \beta_0 \, J_{h,m}(u,\varphi) + b_{h,m}(U,\varphi) \right) dt + (\{U\}_{m-1}, \varphi_{m-1}^+)$$
$$= \int_{I_m} l_{h,m}(\varphi)\, dt, \quad \forall \varphi \in S_{h,\tau}^{p,q}, \quad m = 1, \ldots, M, \tag{5}$$
$$U_0^- := L^2(\Omega)\text{-projection of } u^0 \text{ on } S_{h,1}^p.$$

4. Summary of results on error estimates

The papers [5] and [2] were devoted to the analysis of the STDG method applied to problem in the case of linear diffusion and nonlinear diffusion, respectively. Under the assumptions on the regularity of the exact solution

$$u \in H^{q+1}(0,T; H^1(\Omega)) \cap C([0,T]; H^{p+1}(\Omega)),$$
$$\|\nabla u\|_{L^\infty(\Omega)} \le c_R \quad \text{for a. e.} \quad t \in (0,T),$$

using approximation properties of the $S_{h,m}^p$- and $S_{h,\tau}^{p,q}$- interpolation operators, assumptions on the properties of the meshes, namely the shape regularity and local quasiuniformity, and the condition $\tau_m \ge c\, h_m^2$, $m = 1, \ldots, M$, error estimates in terms of h and τ were proven.

Theorem 1. *There exists a constant $c > 0$ such that*

$$\|e_m^-\|^2 + \frac{\beta_0}{2} \sum_{j=1}^{m} \int_{I_j} \|e\|_{DG,j}^2 dt \le c \left(h^{2p} |u|^2_{C([0,T];H^{p+1}(\Omega))} + \tau^{2q+\alpha} |u|^2_{H^{q+1}(0,T;H^1(\Omega))} \right),$$
$$m = 1, \ldots, M, \quad h \in (0, h_0). \tag{6}$$

Here $\alpha = 2$, if u_D is a polynomial of degree $\le q$ in t. Otherwise, under the assumption that the condition

$$\tau_m \le C_{CFL} h_{K_\Gamma^{(L)}} \tag{7}$$

with a constant C_{CFL} independent of h_K, τ_m and M is satisfied for all elements K adjacent to the boundary $\partial\Omega$, estimate (6) *holds with $\alpha = 0$.*

5. Analysis of stability

There is a natural question, if condition (7) reminding the CFL stability condition is necessary for the derivation of the error estimate (6), or it is also important for guaranteeing the stability of the STDG method (5). In what follows, we shall show that method (5) is unconditionally stable. This means that our goal is to prove that

the approximate solution U of problem (1)-(3) is bounded by the L^2-norm of g, u^0 and by the $\|\cdot\|_{DGB,m}$-norm of u_D, which is defined as

$$\|u_D\|_{DGB,m} := \left(J^B_{h,m}(u_D,u_D)\right)^{1/2} = \left(c_W \sum_{\Gamma\in\mathcal{F}^B_{h,m}} h^{-1}(\Gamma)\int_\Gamma |u_D|^2 dS\right)^{1/2}.$$

The stability analysis starts by setting $\varphi := U$ in the basic relation (5). We get

$$\begin{aligned}&\int_{I_m}\left(\left(\frac{\partial U}{\partial t},U\right)+a_{h,m}(U,U)+\beta_0\, J_{h,m}(U,U)+b_{h,m}(U,U)\right)dt \\ &+\left(\{U\}_{m-1},\varphi^+_{m-1}\right)=\int_{I_m} l_{h,m}(U)\,dt.\end{aligned} \tag{8}$$

After some manipulations we can derive the following identity

$$\int_{I_m}\left(\frac{\partial U}{\partial t},U\right)dt+\left(\{U\}_{m-1},U^+_{m-1}\right)=\frac{1}{2}\left(\|U^-_m\|^2-\|U^-_{m-1}\|^2+\|\{U\}_{m-1}\|^2\right). \tag{9}$$

For a sufficiently large constant c_W, whose lower bound is determined by β_0 and the constants from the multiplicative trace inequality, inverse inequality, local quasiuniformity of the meshes, we can prove the coercivity of the diffusion term:

$$\int_{I_m}\left(a_{h,m}(U,U)+\beta_0\, J_{h,m}(U,U)\right)\,dt\geq\frac{\beta_0}{2}\int_{I_m}\|U\|^2_{DG,m}\,dt-\frac{\beta_0}{2}\int_{I_m}\|u_D\|^2_{DGB,m}\,dt. \tag{10}$$

Furthermore, if $k_1,k_2>0$ then there exists a constant $c_b=c_b(k_1)$ such that the following inequalities for the convection term and for the right-hand side form hold:

$$\int_{I_m}|b_{h,m}(U,U)|\,dt\leq\frac{\beta_0}{k_1}\int_{I_m}\|U\|^2_{DG,m}\,dt+c_b\int_{I_m}\|U\|^2\,dt. \tag{11}$$

$$\begin{aligned}\int_{I_m}|l_{h,m}(U)|\,dt\leq\frac{1}{2}\int_{I_m}\left(\|g\|^2+\|U\|^2\right)\,dt+\beta_0 k_2\int_{I_m}\|u_D\|^2_{DGB,m}\,dt\\ +\frac{\beta_0}{k_2}\int_{I_m}\|U\|^2_{DG,m}\,dt.\end{aligned} \tag{12}$$

If we substitute estimates (9)-(12) into our basic identity (8) and set $k_1=k_2=8$, $c=\max\{2c_b+1,17\beta_0\}$, after some manipulation we get

$$\begin{aligned}&\|U^-_m\|^2-\|U^-_{m-1}\|^2+\frac{\beta_0}{2}\int_{I_m}\|U\|^2_{DG,m}dt\\ &\leq c\left(\int_{I_m}\|g\|^2\,dt+\int_{I_m}\|U\|^2\,dt+\int_{I_m}\|u_D\|^2_{DGB,m}dt\right).\end{aligned} \tag{13}$$

Now our further task is to estimate the expression $\int_{I_m} \|U\|^2\,dt$ in terms of g and u_D. The main tool is the concept of the discrete characteristic function $\zeta_y \in S_{h,\tau}^{p,q}$ to U for $y \in I_m = (t_{m-1}, t_m)$ defined by

$$\int_{I_m} (\zeta_y, \varphi)\,dt = \int_{t_{m-1}}^{y} (U, \varphi)\,dt \quad \forall \varphi \in S_{h,\tau}^{p,q-1}, \quad \zeta_y(t_{m-1}^+) = U(t_{m-1}^+).$$

The operator assigning ζ_y to U is continuous, i.e, there exists $c_q > 0$, depending on q only, such that

$$\int_{I_m} \|\zeta_y\|_{DG,m}^2\,dt \le c_q \int_{I_m} \|U\|_{DG,m}^2\,dt, \quad \int_{I_m} \|\zeta_y\|^2\,dt \le c_q \int_{I_m} \|U\|^2\,dt.$$

Then, after a technical and complicated analysis, it is possible to prove this important estimate: there exists a constant $c > 0$ such that

$$\int_{I_m} \|U\|^2\,dt \le c\,\tau_m \left(\|U_{m-1}^-\|^2 + \int_{I_m} \|g\|^2 + \|u_D\|_{DGB,m}^2\,dt \right). \tag{14}$$

Now we come to the formulation of our main result, which demonstrates the unconditional stability of the STDG method in the discrete $L^2(L^\infty)$-norm, energy DG-norm and $L^2(L^2)$-norm. (A detailed proof can be found in [1].)

Theorem 2. *There exists a constant $c > 0$ such that*

$$\|U_m^-\|^2 + \frac{\beta_0}{2} \sum_{j=1}^{m} \int_{I_j} \|U\|_{DG,j}^2\,dt \le c \left(\|U_0^-\|^2 + \sum_{j=1}^{m} \int_{I_j} (\|g\|^2 + \|u_D\|_{DGB,j}^2)\,dt \right),$$
$$m = 1, \dots, M,\ h \in (0, h_0),$$
$$\|U\|_{L^2(Q_T)}^2 \le c \left(\|U_0^-\|^2 + \sum_{m=1}^{M} \int_{I_m} (\|g\|^2 + \|u_D\|_{DGB,m}^2)\,dt \right), \quad h \in (0, h_0).$$

6. Numerical experiment

We consider the problem

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x_1} + u\frac{\partial u}{\partial x_2} = \epsilon \Delta u + g \quad \text{in} \quad (0,1)^2 \times (0, 10),$$

with $\epsilon = 0.1$ and such initial and Dirichlet boundary conditions that the exact solution has the form

$$u(x_1, x_2, t) = (1 - e^{-10t})\,\hat{u}(x_1, x_2),$$

where $\hat{u}(x_1, x_2) = 2r^\alpha x_1 x_2 (1 - x_1)(1 - x_2)$, $r = (x_1 + x_2)^{1/2}$ and $\alpha \in \mathbb{R}$ is a constant. It is possible to prove that $u \in H^{q+1}(0, T; H^\beta(\Omega))$ for all $\beta \in (0, \alpha + 3)$. (Here $H^\beta(\Omega)$ denotes the Sobolev-Slobodetskii space of functions with "noninteger derivatives".)

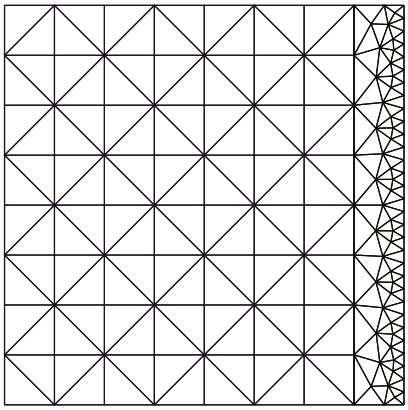


Figure 1: Coarse mesh with 235 elements

We used five special triangular meshes having $235, 333, 749$, 1622 and 2521 elements. All these meshes have refined elements along the right-hand side of the boundary. Figure 1 shows the coarsest mesh. In numerical experiments space polynomial degree $p = 1, 2, 3$ and time polynomial degree $q = 2$ were used. We choose fixed time step $\tau = 0.025$ and set $c_W = 100$ for SIPG. Tables show the computational errors in the $L^\infty(L^2(\Omega))$-norm along the time interval $[0, 10]$, and the corresponding orders of convergence (EOC). It is seen, that for a sufficiently regular exact solution (case $\alpha = 4$), for the SIPG method we have optimal order of convergence $O(h^{p+1})$ for $p = 1, 2, 3$, whereas in the case with irregular solution ($\alpha = -3/2$) the error estimates are of order $O(h^{3/2})$ for $p = 1, 2, 3$ (this result can be proven with the aid of estimates in Sobolev-Slobodetskii spaces). The presented numerical experiments demonstrate the unconditional stability of the numerical process without the CFL-like condition (7). Further numerical experiments including also the NIPG case can be found in [1].

		p=1		p=2		p=3	
Mesh	h	$\|e_h\|$	EOC	$\|e_h\|$	EOC	$\|e_h\|$	EOC
1	1.768E-01	2.167E-03	-	1.305E-04	-	6.681E-06	-
2	1.414E-01	1.488E-03	1.685	7.218E-05	2.654	2.948E-06	3.667
3	8.839E-02	6.549E-04	1.746	1.984E-05	2.748	5.019E-07	3.767
4	5.657E-02	2.914E-04	1.814	5.615E-06	2.828	9.011E-08	3.848
5	4.419E-02	1.842E-04	1.858	2.764E-06	2.872	3.440E-08	3.901

Table 1: Computational errors and the corresponding experimental orders (EOC) of convergence of the SIPG method for $\alpha = 4$

		p=1		p=2		p=3	
Mesh	h	$\|e_h\|$	EOC	$\|e_h\|$	EOC	$\|e_h\|$	EOC
1	1.768E-01	2.668E-02	-	6.038E-03	-	2.784E-03	-
2	1.414E-01	1.946E-02	1.415	4.330E-03	1.490	2.003E-03	1.475
3	8.839E-02	9.856E-03	1.447	2.149E-03	1.491	9.985E-04	1.481
4	5.657E-02	5.116E-03	1.469	1.103E-03	1.493	5.145E-04	1.486
5	4.419E-02	3.552E-03	1.478	7.629E-04	1.495	3.562E-04	1.489

Table 2: Computational errors and the corresponding experimental orders of convergence of the SIPG method for $\alpha = -3/2$

Acknowledgements

This work was supported by the grant No. 13-00522S (M. Feistauer) of the Czech Science Foundation, and by the grant SVV-2014-260106 financed by the Charles University in Prague (M. Balázsová, M. Hadrava and A. Kosík).

References

[1] Balázsová, M., Feistauer, M., Hadrava, M., and Kosík, A.: On the stability of the space-time discontinuous Galerkin method for the numerical solution of nonstationary nonlinear convection-diffusion problems. J. Numer. Math. (accepted).

[2] Česenek, J., and Feistauer, M.: Theory of the space-time discontinuous Galerkin method for nonstationary parabolic problems with nonlinear convection and diffusion. SIAM J. Numer. Anal. **50** (2012), 1181–1206.

[3] Cockburn, B., and Shu, C.-W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. **16** (2001), 173–261.

[4] Eriksson, K., Estep, D., Hansbo, P., and Johnson, C.: *Computational differential equations*. Cambridge University Press, Cambridge, 1996.

[5] Feistauer, M., Kučera, V., Najzar, K., and Prokopová, J.: Analysis of space-time discontinuous Galerkin method for nonlinear convection-diffusion problems. Numer. Math. **117** (2011), 251–288.

[6] Schötzau, D.: *hp-DGFEM for parabolic evolution problems. Applications to diffusion and viscous incompressible fluid flow*. PhD thesis, ETH No. 13041, Zürich, 1999.

[7] Thomée, V.: *Galerkin finite element methods for parabolic problems*. Springer, Berlin, 2006.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

ENVELOPE CONSTRUCTION OF TWO-PARAMETERIC SYSTEM OF CURVES IN THE TECHNOLOGICAL PRACTICE

Stanislav Bartoň, Michal Petřík

Mendel University in Brno
Zemědělská 1, 61300 Brno, Czech Republic
barton@mendelu.cz

Abstract

A two-parametric system of close planar curves is defined in the introduction of the presented article. Next a theorem stating the existence of the envelope is presented and proved. A mathematical model of the collecting mechanism of the Horal forage trailer is developed and used for practical demonstrations. The collecting mechanism is a double joint system composed of three rods. An equation describing the trajectory of a random point of the working rod is derived using Maple. The trajectories of two close points of the working rod create a planar system of close curves, of which the envelope can be computed. As this computation is extremely complex, MAPLE was used to optimize the computations. Through this optimization the computation needed less memory and the processing time was shorter. In the final part the working areas and the corresponding envelopes of all rods defining the collecting mechanism are plotted.

1. Theoretical introduction

Definition of the planar close curves: *Smooth and continuous two-parametric curves* $[x(p,q), y(p,q), z]$ *and* $[x(P,Q), y(P,Q), z]$ *are planar close curves if:*

$$(Q = q + \Delta q,\ \Delta q \to 0, \quad \text{or} \quad P = p + \Delta p,\ \Delta p \to 0) \quad \text{and} \quad z \equiv 0.$$

Theorem 1 (Condition of the envelope existence)**.** *Let* $[x(p,q), y(p,q)]$ *be a system of close planar curves. The envelope exists if and only if the parameters* p *and* q *satisfy (see [2]):*

$$\frac{\dfrac{\partial\, y(p,q)}{\partial p}}{\dfrac{\partial\, x(p,q)}{\partial p}} = \frac{\dfrac{\partial\, y(p,q)}{\partial q}}{\dfrac{\partial\, x(p,q)}{\partial q}} \iff \begin{vmatrix} \dfrac{\partial\, x(p,q)}{\partial q} & \dfrac{\partial\, y(p,q)}{\partial p} \\ \dfrac{\partial\, x(p,q)}{\partial q} & \dfrac{\partial\, y(p,q)}{\partial p} \end{vmatrix} = 0 \quad .$$

Proof: Let us consider two curves belonging to above stated system. The second curve can be written as $[x(P,Q), y(P,Q)]$ and the parameters values p, P, q, Q correspond to their intersection point. If these curves are close, the intersection point is close to the contact point of the curves with their envelope. The intersection point must satisfy:

$$x(p,q) = x(P,Q) \quad \text{and} \quad y(p,q) = y(P,Q).$$

The curves are close, thus satisfying the definition (1),

$$x(P,Q) = x(p+\Delta p, q+\Delta q) \quad \text{and} \quad y(P,Q) = y(p+\Delta p, q+\Delta q). \tag{1}$$

Expanding (1) in the Taylor series leads to:

$$\begin{aligned} x(p,q) &= x(p,q) + \sum_{i=1}^{\infty} \frac{\partial^i x(p,q)}{i!\,\partial p^i} (\Delta p)^i + \sum_{i=1}^{\infty} \frac{\partial^i x(p,q)}{i!\,\partial q^i} (\Delta q)^i \\ y(p,q) &= y(p,q) + \sum_{i=1}^{\infty} \frac{\partial^i y(p,q)}{i!\,\partial p^i} (\Delta p)^i + \sum_{i=1}^{\infty} \frac{\partial^i y(p,q)}{i!\,\partial q^i} (\Delta q)^i \end{aligned} . \tag{2}$$

If $i \geq 2$ we can neglect the powers $(\Delta p)^i, (\Delta q)^i$. Therefore the condition simplifies (2) to

$$\frac{\partial x(p,q)}{\partial p}\Delta p + \frac{\partial x(p,q)}{\partial q}\Delta q = 0 \quad \text{and} \quad \frac{\partial y(p,q)}{\partial p}\Delta p + \frac{\partial y(p,q)}{\partial q}\Delta q = 0. \tag{3}$$

From the equations (3) the binding condition follows easily.

2. Kinematics of pick up vehicle Horal collector mechanism

The Horal pick up vehicle is a drawn forage wagon for collecting grass, hay or straw. The basic constructive scheme of the collecting mechanism is depicted in the Figure 1, see [3] for the technical parameters. The axis of the drive shaft was chosen as origin of the coordinate system. The propelling handle r, which is at the point $B \equiv [B_x(t), B_y(t)]$, is linked by a joint to the two-part work rod $L2$–$L3$. These two parts form an angle f. The end of the upper part of the work rod with length $L2$ is joined at the point $C \equiv [C_x(t), C_y(t)]$ with an additional handle of length $L1$ which can freely rotate with its other end around point of coordinates $A \equiv [A_x, A_y]$. The free working point, marked with P creates the curve $[X(t), Y(t)]$. First we would like to determine the position of the additional point $D \equiv [D_x(t), D_y(t)]$ which corresponds to the position of the work point P when the two parts $L2$ and $L3$ are parallel. Then we can rotate the point D around point B by the angle f to calculate the position of work point P.

3. Determination of the trajectories of individual points of the work rod

All necessary calculations are carried out in program Maple, [1]. Considering the page restrictions for this article the output of the program is suppressed apart from a few exceptions. The specifications of the variables, important points and the

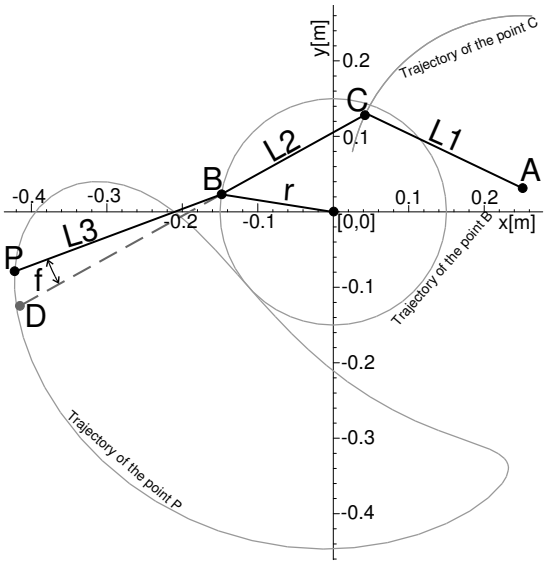


Figure 1: The constructive schema

coordinate system can be seen in Figure 1. The dimensions and the coordinates saved in the variable `Ksu` are set in meters. The driving handle rotates to left direction with speed of one revolution per second.

```
> restart; with(plots): Ksu:=[Ax=0.25,Ay=0.03,r=0.15,L1=0.23,L2=0.22,L3=0.30,
   f=55*Pi/1800,omega=-2*Pi];
> Dx:=Bx+(Bx-Cx)*L3/L2: Dy:=By+(By-Cy)*L3/L2:
> X:=(Dx-Bx)*cos(f)+(Dy-By)*sin(f)+Bx;Y:=(Bx-Dx)*sin(f)+(Dy-By)*cos(f)+By;
```

The current coordinates of the joint C are determined as the intersection point of the circles $c_1 \equiv ([B_x(t), B_y(t)], L2])$ and $c_2 \equiv ([A_x, A_y], L1)$. The point with y positive is chosen from both intersections.

```
> CB:=(x-Bx)^2+(y-By)^2=L2^2; CA:=(x-Ax)^2+(y-Ay)^2=L1^2;
> SolC:=[allvalues(solve({CB,CA},{x,y}))]: SolCf:=subs(Bx=r,By=0,Ksu,SolC);
> SolC:=zip((u,v)->`if`(subs(u,y)>0,v,NULL),SolCf,SolC)[]:
> Cx:=subs(SolC,x): Cy:=subs(SolC,y): Bx:=r*cos(omega*t); By:=r*sin(omega*t);
```

Now we choose ten points on both parts of the work rod and draw their trajectories. For the upper part $L2$ these trajectories are marked light gray, for lower part $L3$ in dark gray, see Figure 2. The parameter λ determines the positions of the points.

```
> Lambda1:=1/N1*[$1..N1-1]: Lambda2:=1/N2*[$1..N2-1]: RL3x:=Bx+(X-Bx)*lambda:
>  RL3y:=By+(Y-By)*lambda: RL2x:=Bx+(Cx-Bx)*lambda: RL2y:=By+(Cy-By)*lambda:
> G7:=plot([seq(subs(Ksu,[RL3x,RL3y,t=0..1]),lambda=Lambda1)],color=red):
> G8:=plot([seq(subs(Ksu,[RL2x,RL2y,t=0..1]),lambda=Lambda2)], color=blue):
> G9:=plot(evalf([seq(subs(Ksu,[Bx,By]),t=T)]),style=point,color=black):
> G10:=textplot(evalf([seq(subs(Ksu,[Bx,By,cat(" ",convert(evalf(t,2),string))]),
      t=T)]),color=black,align=right, font=[HELVETICA,BOLD,12]):
> display({G4,G5,G6,G7,G8,G9,G10},labels=["x [m]","y [m]"]);
```

From Figure 2 it is obvious that two envelopes really exist for both systems of curves which delimit the work area of rods $L2$ and $L3$.

In order to estimate when individual points of both rods reach contact with the envelope, the position of the joint B is highlighted in the time period of 1/10 of the work period.

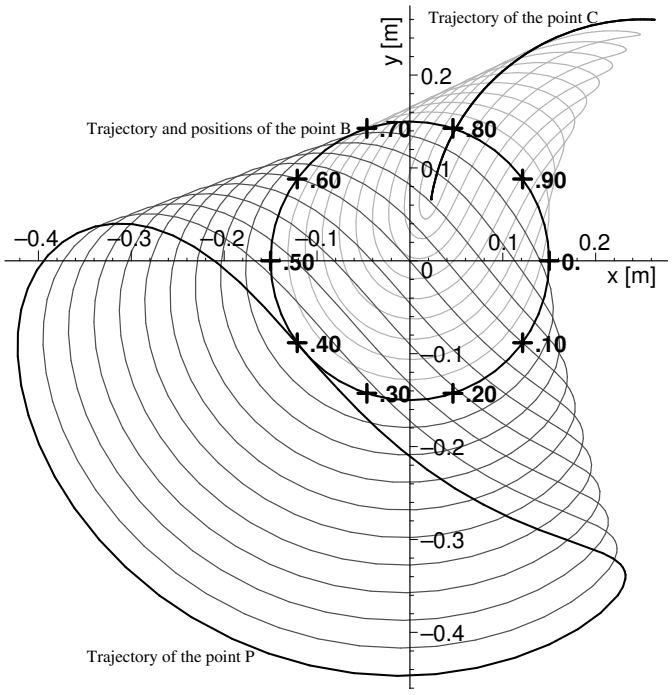


Figure 2: Trajectories of the 10 points on the rods $L2$ and $L3$

4. Envelope construction

The systems of the light and dark gray curves are two-parametric systems satisfying the definition of the planar close curves, see page 17. The first parameter is the time t, which determines the position of work rod, the second parameter λ determines position of point on the rod. By using the binding condition between both parameters we can determine the points that constitute the envelopes of both systems of curves according to equation (1). The binding conditions `O2` for rod $L2$ and `O3` for rod $L3$ are very complicated expressions and therefore it pays to use the MAPLE optimization facility.

Because the binding conditions O2 and O3 are non-linear implicit equations, it is necessary to execute the following calculations numerically.

```
> t:='t': lambda:='lambda': Digits:=24: C31:=cost(O3): C21:=cost(O2):
> PO3:=optimize(makeproc(O3,t)): PO2:=optimize(makeproc(O2,t)):
> C32:=cost(PO3): C22:=cost(PO2): Savings=C31+C21-C32-C22;
```

Savings = 55178 multiplications +16198 functions +9372 additions −239 storage −239 assignments

It is obvious that the optimization of MAPLE is very effective. Newton's iteration is used to determine numerically the time t for each point when this point reaches the envelope. The use of automatic derivatives is a great advantage for computing this step.

```
ILT:=proc(PROC,VAR) local var, dvar;
 var:=VAR; dvar:=1;
 while abs(dvar)>1e-6 do; dvar:=-PROC(var)/D(PROC)(var); var:=var+dvar; end do;
 var;
end proc;
```

From Figure 2 it is obvious that point B whose parameter $\lambda = 0$ for both rods, reaches the left envelope at time $t \approx 0.7\,\mathrm{s}$ and the right shell at time $t \approx 1\,\mathrm{s}$. These time values can be used as an input data of iterative procedure. If there is a shift from point B to a close point (a little shift corresponds to $\Delta\lambda \to 0$), then it is possible to determine the time from binding condition (1) by means of the iterative procedure `ILT` when this close point reaches the envelope. The final time t used by the previous point for reaching the envelope is used as initial value for the next iteration.

The time values t_k of reaching the envelope for individual values of λ will be saved as ordered pairs $[\lambda_k, t_k]$ which will be used to draw envelope in detail, at the end of the calculation, see Figure (3).

```
> lambda:=0; tau31:=ILT(PO3,0.7); tau32:=ILT(PO3,1.0);
> tau21:=ILT(PO2,0.7); tau22:=ILT(PO2,1.0); LT31:=[[0,tau31]];
> LT32:=[[0,tau32]]; LT21:=[[0,tau21]]; LT22:=[[0,tau22]];
> for lambda from 0.01 to 1 by 0.01 do;
   tau31:=ILT(PO3,tau31);tau32:=ILT(PO3,tau32);tau21:=ILT(PO2,tau21);
   tau22:=ILT(PO2,tau22);LT31:=[LT31[],[lambda,tau31]];
   LT32:=[LT32[],[lambda,tau32]];LT21:=[LT21[],[lambda,tau21]];
   LT22:=[LT22[],[lambda,tau22]];
> end do:
> Digits:=10; lambda:='lambda';
> O31:=map(u->evalf(subs(Ksu,lambda=u[1],t=u[2],[RL3x,RL3y])),LT31):
> O32:=map(u->evalf(subs(Ksu,lambda=u[1],t=u[2],[RL3x,RL3y])),LT32):
> O21:=map(u->evalf(subs(Ksu,lambda=u[1],t=u[2],[RL2x,RL2y])),LT21):
> O22:=map(u->evalf(subs(Ksu,lambda=u[1],t=u[2],[RL2x,RL2y])),LT22):
> G11:=plot([O31,O32,O21,O22],color=black,thickness=3):
> display({G4,G5,G6,G7,G8,G11},labels=["x [m]","y [m]"]);
```

5. Conclusion

The process described in this article can be generalized to compute areas and envelopes for other curves. This method can be used to avoid contacts of different moving parts. For example, Figure 3 demonstrates that there exists an area inside the Horal collector mechanism passed by three different components.

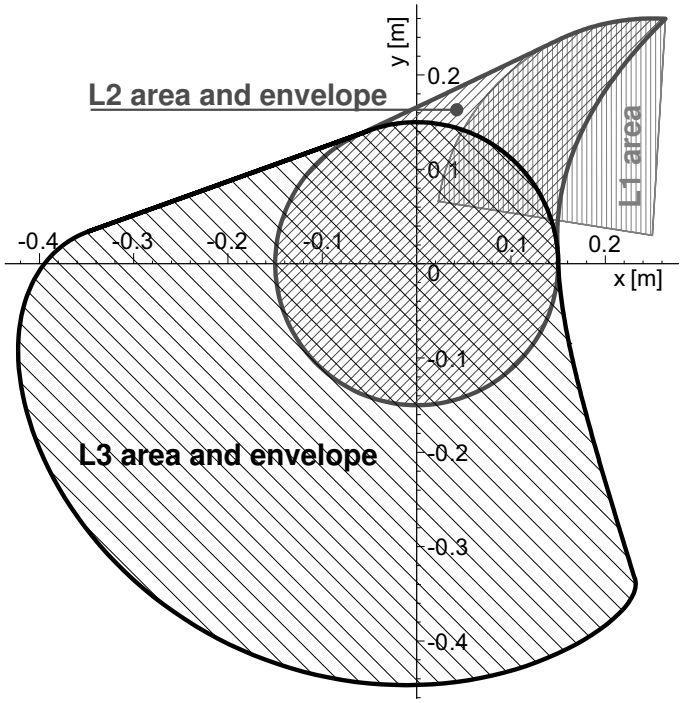


Figure 3: The envelopes and rods workspaces

It can also be used to find points of maximal deflection of the trajectory, so called *dead points of working mechanisms*. If a component passes through this point some of its points have maximum acceleration, *tangent or normal*. Taking these accelerations into account is essential for proper construction of the components.

Acknowledgments

The research has been supported by the project TP 4/2014 "Analysis of degradation processes of modern materials used in agricultural technology" financed by IGA AF MENDELU.

References

[1] *Maple User Manual.* Maplesoft, 2011, Waterloo Canada, ISBN 978-1-926902-07-4.

[2] Rutter, J. W.: *Geometry of curves.* CRC Press, 2000, ISBN 1-58488-166-6, 243–270.

[3] *Sběrací vozy Horal SP 3-341.*
`http://www.vozy-biso.cz/cs/produkty/sberaci-vozy-horal-sp-3-341/`

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

ISOGEOMETRIC ANALYSIS FOR FLUID FLOW PROBLEMS

Bohumír Bastl[1,2], Marek Brandner[1,2], Jiří Egermaier[1,2],
Kristýna Michálková[1,2], Eva Turnerová[1,2]

[1] University of West Bohemia in Pilsen
Univerzitní 8; 306 14, Pilsen, Czech Republic
[2] European Centre of Excellence – New Technologies for the Information Society
University of West Bohemia in Pilsen
Univerzitní 8; 306 14, Pilsen, Czech Republic
bastl@kma.zcu.cz, brandner@kma.zcu.cz, jirieggy@kma.zcu.cz,
kslaba@kma.zcu.cz, turnerov@kma.zcu.cz

Abstract

The article is devoted to the simulation of viscous incompressible fluid flow based on solving the Navier-Stokes equations. As a numerical model we chose isogeometrical approach. Primary goal of using isogemetric analysis is to be always geometrically exact, independently of the discretization, and to avoid a time-consuming generation of meshes of computational domains. For higher Reynolds numbers, we use stabilization techniques SUPG and PSPG. All methods mentioned in the paper are demonstrated on a standard test example – flow in a lid-driven cavity.

1. Introduction

Typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data. Primary goal of using isogeometric analysis is to be geometrically exact, independently of the discretization. Then we do not need to create any other mesh - the mesh of the so-called "NURBS elements" is acquired directly from CAD representation. Further refinement of the mesh or increasing the order of basis functions are very simple, efficient and robust.

2. NURBS Surfaces

NURBS surface of degree p, q is determined by a control net $\mathbf{P}$ (of control points $P_{i,j}$, $i = 0, \ldots, n$, $j = 0, \ldots, m$), weights $w_{i,j}$ of these control points and two knot vectors $U = (u_0, \ldots, u_{n+p+1})$, $V = (v_0, \ldots, v_{m+q+1})$ and is given by a parametrization

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} P_{i,j} N_{i,p}(u) M_{j,q}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} N_{i,p}(u) M_{j,q}(v)} = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{i,j} R_{i,j}(u,v). \qquad (1)$$

B-spline basis functions $N_{i,p}(u)$ and $M_{j,q}(v)$ are determined by knot vectors U and V and degrees p and q, respectively, by a formula (for $N_{i,p}(u)$, $M_{j,q}(v)$ is constructed by the similar way)

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & u_i \leq t < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \tag{2}$$

Knot vector is a non-decreasing sequence of real numbers which determines the distribution of a parameter on the corresponding curve/surface. B-spline basis functions (see Figure 1) of degree p are C^{p-1}-continuous in general. Knot repeated k times in the knot vector decreases the continuity of B-spline basis functions by $k-1$. Support of B-spline basis functions is local – it is nonzero only on the interval $[t_i, t_{i+p+1}]$ in the parameter space and each B-spline basis function is non-negative, i.e., $N_{i,p}(t) \geq 0, \forall t$. See [7] for more information.

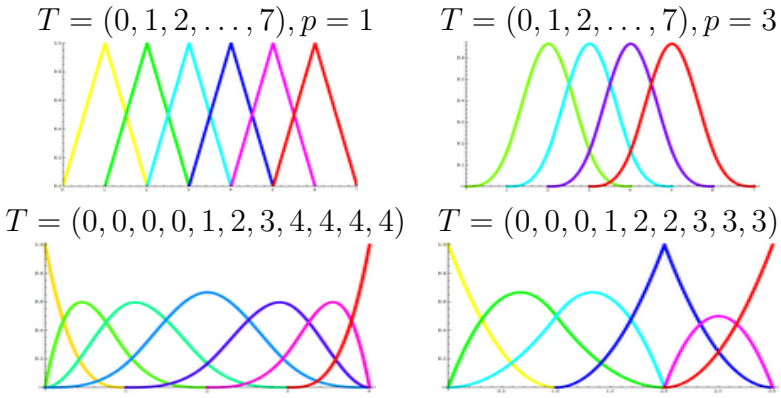


Figure 1: B-spline basis functions

3. Stationary Navier-Stokes equations

The model of viscous flow of an incompressible Newtonian fluid can be described by the Navier-Stokes equations in the common form

$$\begin{aligned} \nabla p + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} &= f, \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned} \tag{3}$$

where $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$ is the vector function describing flow velocity, $p = p(\boldsymbol{x})$ is the pressure normalized by density function, ν describes kinematic viscosity and f additional body forces acting on the fluid. The boundary value problem is considered as the system (3) together with boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{w} \quad \text{on } \partial\Omega_D \quad \text{(Dirichlet condition)} \\ \nu \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} - \boldsymbol{n} p &= \boldsymbol{0} \quad \text{on } \partial\Omega_N \quad \text{(Neumann condition).} \end{aligned} \tag{4}$$

If the velocity is specified everywhere on the boundary, then the pressure solution is only unique up to a (hydrostatic) constant.

Let V be a velocity solution space and V_0 be the corresponding space of test functions, i.e.,

$$\begin{aligned} V &= \{\boldsymbol{u} \in H^1(\Omega)^d | \boldsymbol{u} = \boldsymbol{w} \text{ on } \partial\Omega_D\} \\ V_0 &= \{\boldsymbol{v} \in H^1(\Omega)^d | \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial\Omega_D\}. \end{aligned} \tag{5}$$

Then a weak formulation of the boundary value problem is: find $\boldsymbol{u} \in V$ and $p \in L_2(\Omega)$ such that

$$\begin{aligned} \nu \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \int_\Omega (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) \boldsymbol{v} - \int_\Omega p \nabla \cdot \boldsymbol{v} &= \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \qquad \forall \boldsymbol{v} \in V_0 \\ \int_\Omega q \nabla \cdot \boldsymbol{u} &= 0 \qquad \forall q \in L_2(\Omega) \end{aligned}$$

3.1. Approximation using isogeometric analysis

We define the finite-dimensional spaces $V^h \subset V$, $V_0^h \subset V_0$, $W^h \subset L_2(\Omega)$ and their basis functions. We want to find $\boldsymbol{u}_h^{k+1} \in V^h$ and $p_h^{k+1} \in W^h$ such that for all $\boldsymbol{v}_h \in V_0^h$ and $q_h \in W^h$ it holds

$$\nu \int_\Omega \nabla \boldsymbol{u}_h^{k+1} : \nabla \boldsymbol{v}_h + \int_\Omega (\boldsymbol{u}_h^k \cdot \nabla \boldsymbol{u}_h^{k+1}) \boldsymbol{v}_h - \int_\Omega p_h^{k+1} \nabla \cdot \boldsymbol{v}_h = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}_h, \tag{6}$$

$$\int_\Omega q_h \nabla \cdot \boldsymbol{u}_h^{k+1} = 0. \tag{7}$$

This approach is based on the Picard's method (fixed point iteration). For isogeometric analysis, basis functions of V_0^h and W^h are NURBS basis functions obtained from the NURBS description of the computational domain (for velocity and pressure). We can express $\boldsymbol{u}_h^k$ and p_h^k as a linear combination of the basis functions (2) (we use the values $p = 3, q = 3$ for the velocity and $p = 2, q = 2$ for the pressure in the follow-up examples). These linear combinations are substituted to (6) and (7). Linearization is done with help of Picard's iteration and we obtain a sequence of solutions $(\boldsymbol{u}_h^k, p_h^k) \in V^h \times W^h$, which converges to the weak solution. We obtain a matrix formulation of the problem in the form

$$\begin{bmatrix} \boldsymbol{A} + \boldsymbol{N}(\boldsymbol{u}^k) & \boldsymbol{0} & -\boldsymbol{B}_1^\top \\ \boldsymbol{0} & \boldsymbol{A} + \boldsymbol{N}(\boldsymbol{u}^k) & -\boldsymbol{B}_2^\top \\ \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^{k+1} \\ \boldsymbol{u}_2^{k+1} \\ \boldsymbol{p}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 - (\boldsymbol{A}^* + \boldsymbol{N}^*(\boldsymbol{u}^k)) \cdot \boldsymbol{u}_1^* \\ \boldsymbol{f}_2 - (\boldsymbol{A}^* + \boldsymbol{N}^*(\boldsymbol{u}^k)) \cdot \boldsymbol{u}_2^* \\ -\boldsymbol{B}_1^* \cdot \boldsymbol{u}_1^* - \boldsymbol{B}_2^* \cdot \boldsymbol{u}_2^* \end{bmatrix}, \tag{8}$$

where

$$\begin{aligned} \boldsymbol{A} &= \left[A_{ij}\right]_{1 \le i \le n_d^u, 1 \le j \le n_d^u}, & \boldsymbol{A}^* &= \left[A_{ij}\right]_{1 \le i \le n_d^u, n_d^u+1 \le j \le n_v^u}, \\ \boldsymbol{N}(\boldsymbol{u}) &= \left[N_{ij}(\boldsymbol{u})\right]_{1 \le i \le n_d^u, 1 \le j \le n_d^u}, & \boldsymbol{N}^*(\boldsymbol{u}), &= \left[N_{ij}(\boldsymbol{u})\right]_{1 \le i \le n_d^u, n_d^u+1 \le j \le n_v^u}, \\ \boldsymbol{B}_k &= \left[B_{kij}\right]_{1 \le i \le n^p, 1 \le j \le n_d^u}, & \boldsymbol{B}_k^* &= \left[B_{kij}\right]_{1 \le i \le n^p, n_d^u+1 \le j \le n_v^u}, \end{aligned} \tag{9}$$

$$
\begin{aligned}
A_{ij} &= \nu \int_\Omega (\nabla R_i^u \cdot J^{-1}) \cdot (\nabla R_j^u \cdot J^{-1}) |\det J|, \\
N_{ij}(\boldsymbol{u}) &= \int_\Omega R_i^u \left[\left(\sum_{l=1}^{n_v^u} (u_{1l}, u_{2l}) R_l^u \right) \cdot (\nabla R_j^u \cdot J^{-1}) \right] |\det J|, \\
B_{kij} &= \int_\Omega R_i^p \left[(\nabla R_j^u \cdot J^{-1}) \cdot \boldsymbol{e}_k \right] |\det J|.
\end{aligned} \tag{10}
$$

Here n_d^u is the number of points where the Dirichlet boundary condition is not defined and $\boldsymbol{u}_1^*, \boldsymbol{u}_2^*$ are fixed coefficient so that the Dirichlet boundary condition is satisfied. J is the Jacobi matrix of a mapping from parametric domain to the computational domain. The initial nonlinear Navier-Stokes problem was transformed to the sequential solving of linear systems.

In the follow-up examples, we use strong imposition of Dirichlet boundary conditions. If the given function $\boldsymbol{w}$ belongs to V^h, Dirichlet boundary condition is prescribed directly on control points describing $\partial\Omega_D$. Otherwise, we have to find an approximation $\boldsymbol{w}_h$ of $\boldsymbol{w}$ in V^h and again prescribe this condition directly on control points.

3.2. LBB (Ladyženskaja-Babuška-Brezzi) condition

In general, it is not possible to use an arbitrary combination of discretizations for pressure and velocity for solving Stokes problem in order for given discretizations to be stable, it needs to satisfy the so-called LBB condition (or inf-sup condition). It can be shown that one of such suitable choices of discretizations is represented by spaces with basis function of degree p (for pressure) and degree $p+1$ (for velocity) obtained with the help of p-refinement (see [1] for more details).

4. Stabilization methods

The solving of Navier-Stokes equations leads to numerical nonstability for high Reynolds numbers. We review two methods to reduce nonphysical oscillations based on the construction of test functions in special forms (see for example [6]).

4.1. SUPG - Streamline Upwind/Petrov-Galerkin

The first equation (3) is multiplied by test function $\overline{\boldsymbol{v}}$ in the form

$$
\overline{\boldsymbol{v}} = \boldsymbol{v} + \tau_S \boldsymbol{u} \cdot \nabla \boldsymbol{v}, \tag{11}
$$

where

$$
\tau_S = \frac{h}{2 \deg(\boldsymbol{u}) \|\boldsymbol{u}\|} \left(\coth P - \frac{1}{P} \right), \tag{12}
$$

h is the element diameter in the direction of the $\boldsymbol{u}$ and $P = \frac{\|\boldsymbol{u}\| h}{2\nu}$ is the local Péclet number which determines whether the problem is locally convection dominated or diffusion dominated. Then we integrate over Ω and use Picard's linearization method.

The first equation has the form

$$\nu \underbrace{\int_\Omega \nabla \boldsymbol{u}^{k+1} : \nabla \boldsymbol{v}}_{\boldsymbol{A}} + \underbrace{\int_\Omega \boldsymbol{u}^k \cdot \nabla \boldsymbol{u}^{k+1}}_{\boldsymbol{N}(u)} - \underbrace{\int_\Omega p^{k+1} \nabla \cdot \boldsymbol{v}}_{\boldsymbol{B}} - \nu \underbrace{\int_\Omega \Delta \boldsymbol{u}^{k+1} \tau_S \boldsymbol{u}^k \cdot \nabla \boldsymbol{v}}_{SUPG} +$$

$$+ \underbrace{\int_\Omega (\boldsymbol{u}^k \cdot \nabla \boldsymbol{u}^{k+1}) \tau_S \boldsymbol{u}^k \cdot \nabla \boldsymbol{v}}_{SUPG} + \underbrace{\int_\Omega \nabla p^{k+1} \tau_S \boldsymbol{u}^k \cdot \nabla \boldsymbol{v}}_{SUPG} = \int_\Omega \boldsymbol{f} \cdot \overline{\boldsymbol{v}}. \qquad (13)$$

4.2. PSPG (Pressure Stabilized/Petrov-Galerkin)

In this case we multiply the first equation (3) by the test function in the form

$$\overline{\boldsymbol{v}} = \boldsymbol{v} + \tau_S \boldsymbol{u} \cdot \nabla \boldsymbol{v} + \tau_P \nabla q, \qquad (14)$$

where $0 \leq \tau_P \leq \tau_S$ and integrate over Ω. By application of Picard's method we have

$$\nu \underbrace{\int_\Omega \nabla \boldsymbol{u}^{k+1} : \nabla \boldsymbol{v}}_{\boldsymbol{A}} + \underbrace{\int_\Omega \boldsymbol{u}^k \cdot \nabla \boldsymbol{u}^{k+1}}_{\boldsymbol{N}(u)} - \underbrace{\int_\Omega p^{k+1} \nabla \cdot \boldsymbol{v}}_{\boldsymbol{B}} - \qquad (15)$$

$$-\nu \underbrace{\int_\Omega \Delta \boldsymbol{u}^{k+1} \tau_S \boldsymbol{u}^k \cdot \nabla \boldsymbol{v}}_{SUPG} + \underbrace{\int_\Omega (\boldsymbol{u}^k \cdot \nabla \boldsymbol{u}^{k+1}) \tau_S \boldsymbol{u}^k \cdot \nabla \boldsymbol{v}}_{SUPG} + \underbrace{\int_\Omega \nabla p^{k+1} \tau_S \boldsymbol{u}^k \cdot \nabla \boldsymbol{v}}_{SUPG} +$$

$$+ \underbrace{\int_\Omega \tau_P \nabla p^{k+1} \nabla q}_{PSPG} - \nu \underbrace{\int_\Omega \tau_P \Delta \boldsymbol{u}^{k+1} \nabla q}_{PSPG} + \underbrace{\int_\Omega (\boldsymbol{u}^k \cdot \nabla \boldsymbol{u}^{k+1}) \tau_P \nabla q}_{PSPG} = \int_\Omega \boldsymbol{f} \cdot \overline{\boldsymbol{v}}$$

If we use these stabilization techniques, the LBB condition need not be satisfied.

5. Non-stationary Navier-Stokes problem

For the simplicity we solve the homogeneous problem

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \nabla p + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} &= 0 \quad \text{in } \Omega \times (0,T) \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{v } \Omega \end{aligned} \qquad (16)$$

with the initial and boundary conditions

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},t) &= \boldsymbol{w}(\boldsymbol{x},t) \quad \text{on } \partial\Omega \times [0,T], \\ \boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{u}_0(\boldsymbol{x}) \quad \text{in } \Omega. \end{aligned} \qquad (17)$$

A method described in [4] is used. It is given $\boldsymbol{u}^0$, $\theta \in (0, \frac{1}{2})$, $\alpha \in (0,1)$, $\beta \in (0,1)$ and we search for $\boldsymbol{u}^1$,$\boldsymbol{u}^2$, ..., $\boldsymbol{u}^n$ by the following three steps:

1. step

$$\begin{aligned} \frac{\boldsymbol{u}^{n+\theta}-\boldsymbol{u}^{n}}{\theta\Delta t}+\nabla p^{n+\theta}-\alpha\nu\Delta\boldsymbol{u}^{n+\theta} &= \beta\nu\Delta\boldsymbol{u}^{n}-\boldsymbol{u}^{n}\cdot\nabla\boldsymbol{u}^{n} \\ \nabla\cdot\boldsymbol{u}^{n+\theta} &= \boldsymbol{0} \\ \boldsymbol{u}^{n+\theta} &= \boldsymbol{g}^{n+\theta} \text{ on } \partial\Omega \end{aligned} \tag{18}$$

2. step

$$\begin{aligned} \frac{\boldsymbol{u}^{n+1-\theta}-\boldsymbol{u}^{n+\theta}}{(1-2\theta)\Delta t}-\beta\nu\Delta\boldsymbol{u}^{n+1-\theta}+\boldsymbol{u}^{*}\cdot\nabla\boldsymbol{u}^{n+1-\theta} &= \alpha\nu\Delta\boldsymbol{u}^{n+\theta}-\nabla p^{n+\theta} \\ \boldsymbol{u}^{n+1-\theta} &= \boldsymbol{g}^{n+1-\theta} \text{ on } \partial\Omega \end{aligned} \tag{19}$$

3. step

$$\begin{aligned} \frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n+1-\theta}}{\theta\Delta t}+\nabla p^{n+1}-\alpha\nu\Delta\boldsymbol{u}^{n+1} &= \beta\nu\Delta\boldsymbol{u}^{n+1-\theta}-\boldsymbol{u}^{*}\cdot\nabla\boldsymbol{u}^{n+1-\theta} \\ \nabla\cdot\boldsymbol{u}^{n+1} &= \boldsymbol{0} \\ \boldsymbol{u}^{n+1} &= \boldsymbol{g}^{n+1} \text{ on } \partial\Omega \end{aligned} \tag{20}$$

This is a self-starting scheme. Choosing $\alpha=\beta=\frac{1}{2}$ or setting $\theta=1-\frac{1}{\sqrt{2}}$ with $\alpha+\beta=1$ gives second-order accuracy as $\Delta t\to 0$. In particular, setting $\theta=1-\frac{1}{\sqrt{2}}$ and $\alpha=\frac{1-2\theta}{1-\theta}, \beta=\frac{\theta}{1-\theta}$ gives a method which is second-order accurate in time, unconditionally stable and has good asymptotic properties.

6. Examples

We present test example, which is symmetric to the well-known test problem, the so-called lid-driven cavity flow in 2D. The only difference is that the moving wall is situated at the bottom of the cavity. This change has no compelling reason, the test problem is sufficient for testing the solver and comparing the results with benchmark ones.

It should be noted that the presented solver uses both presented stabilization techniques, it means that the degree of basis functions for pressure is one less than the degree of velocity basis functions and the PSPG stabilization technique is also used. Using only one technique is sufficient for the stable solution and we tested both of them as well as their combination.

6.1. Stationary flow

The first experiment is devoted to the stationary flow. So we solve stationary Navier-Stokes equations (3) with the bottom boundary moving from left to right ($\boldsymbol{u}=(u_x,0)$) and no-slip boundary condition on the other walls. Figure 2 shows the solutions with the three different Reynolds numbers and instability for higher ones. The solution of the same problem where the stabilization methods are used is illustrated on Figure 3. It is known (see for example [5]), that the solution of

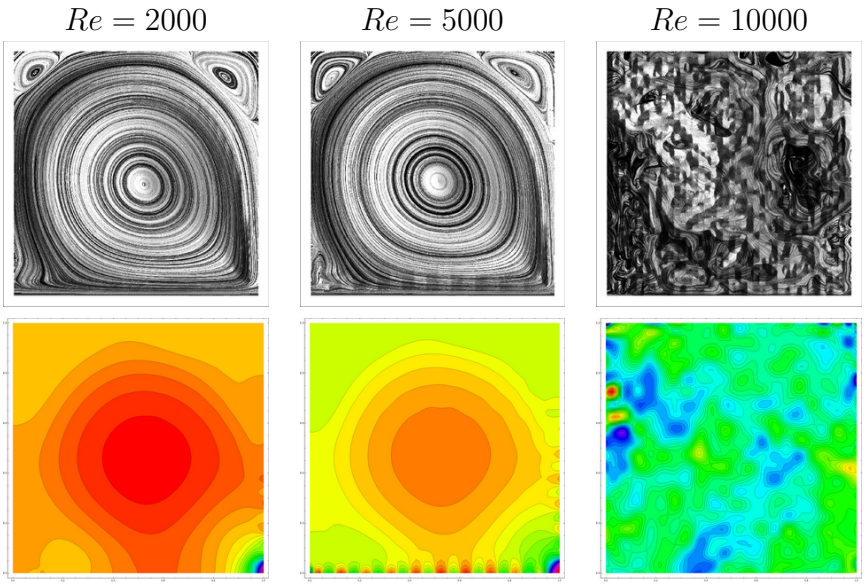


Figure 2: Stationary Navier-Stokes problem. Solution without stabilization techniques. Velocity is illustrated at the upper figures, pressure is illustrated at lower figures.

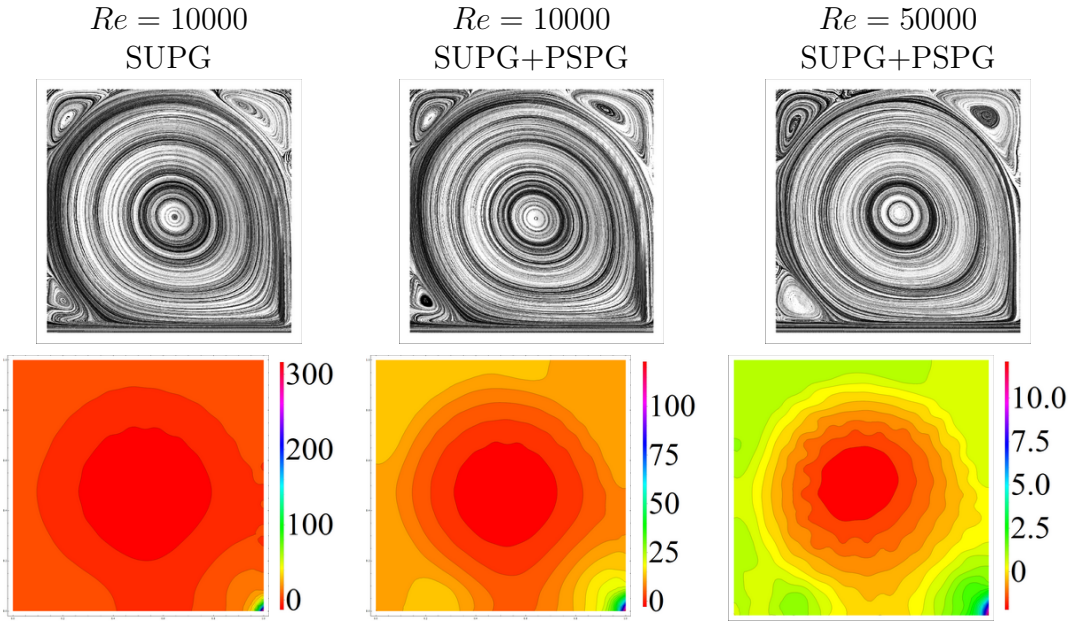


Figure 3: Stationary Navier-Stokes problem. Solution with stabilization techniques. Velocity is illustrated at the upper figures, pressure is illustrated at lower figures.

this test example has a stable solution only for much smaller Reynolds numbers than presented $Re = 50000$. So the result for the $Re = 50000$ is not very physically meaningful, it is rather an example of the used stabilization techniques. It should be also noted, that the NURBS discretization uses fewer elements than the finite element discretization in general. This coarse discretization causes more artificial viscosity.

6.2. Non-stationary flow

The second example is devoted to the non-stationary flow. We solve non-stationary Navier-Stokes equations (16) with the same boundary conditions as in the first example. Initial condition is described by the zero velocity inside the cavity ($\boldsymbol{u} = \boldsymbol{0}$). Solution with Reynolds number Re $= 1000$ is illustrated at Figure 4.

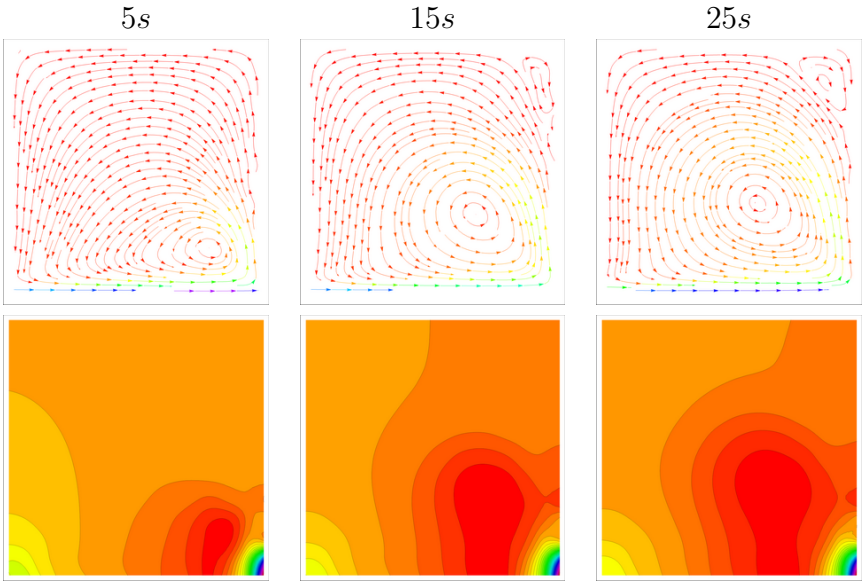


Figure 4: Non-stationary Navier-Stokes problem. Solution with stabilization techniques. Velocity is illustrated at the upper figures, pressure is illustrated at lower figures.

7. Conclusion

We developed and tested an isogeometric analysis based solver for solving stationary and nonstationary flow based on Navier-Stokes equations. The presented results show that the isogeometric analysis is a suitable tool for solving such complex problems. Iterative solution of stationary Navier-Stokes equations converges only for relatively low Reynolds numbers. Therefore, it is necessary to use stabilization methods (e.g. SUPG, PSPG, see [2]). The problems with oscillations can be solved by the SOLD methods [6]. The presented scheme for solving non-stationary

Navier-Stokes equations is currently enlarged by turbulence model. The turbulence is included by the RANS equations using $k - \omega$ model [3].

Acknowledgements

This work has been supported by Technology agency of the Czech Republic through the project TA03011157 "Innovative techniques for improving utility qualities of water turbines with the help of shape optimization based on modern methods of geometric modeling."

References

[1] Bressan, A. and Sangalli, G.: Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique. IMA J. Numer. Anal., 2012.

[2] Brooks, A. N. and Hughes, T. J. R.: Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput. Methods Appl. Mech. Engrg. **32** (1982), 199–259.

[3] Davidson, L.: *Fluid mechanics, turbulent flow and turbulence modeling, Lecture Notes in MSc courses*. Chalmers University of Technology, Sweden, 2013.

[4] Elman, H. C., Silvester, D. J., and Wathen, A. J.: Iterative methods for problems in computational fluid dynamics. Report CS-TR-3675, UMIACS-TR-96-58, 1996.

[5] Elman, H. C., Silvester, D. J., and Wathen, A.J.: *Finite elements and fast iterative solvers with applications in incompressible fluid dynamics*. Oxford University Press, 2005.

[6] John, V. and Knobloch, P.: On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I: A review. Comput. Methods Appl. Mech. Engrg. **196** (2007), 2197–2215.

[7] Piegl, L. and Tiller, W.: *The NURBS Book*. Springer, 1996.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

AN A POSTERIORI ERROR ESTIMATE FOR THE STOKES-BRINKMAN PROBLEM IN A POLYGONAL DOMAIN

Pavel Burda[1], Martin Hasal[2]

[1] Czech University of Technology, Faculty of Mechanical Engineering,
Department of Mathematics
Karlovo nám. 13, CZ-121 35 Praha 2, Czech Republic
pavel.burda@fs.cvut.cz
[2] VŠB TU Ostrava
17. listopadu 15, 708 03 Ostrava-Poruba, Czech Republic
martin.hasal@vsb.cz

Abstract

We derive a residual based a posteriori error estimate for the Stokes-Brinkman problem on a two-dimensional polygonal domain. We use Taylor-Hood triangular elements. The link to the possible information on the regularity of the problem is discussed.

1. Introduction

In the paper we try to contribute to the technique of a posteriori error estimates for the finite element solution of linearized flow problems. In this respect we note that important results have already been obtained: concerning linear elliptic equations let us mention I. Babuška, W. C. Rheinboldt [2], I. Babuška, R. Durán, R. Rodríguez [3], concerning the Stokes problem e.g. M. Ainsworth, J. T. Oden [1], R. E. Bank, D. Welfert [5], C. Carstensen, S. Jansche [7], C. Johnson, R. Rannacher, M. Boman [12], R. Verfürth [15].

The goal of this paper is to link the problem of a posteriori error estimates as much as possible to the information on the regularity of the solution.

Let us illustrate it first on the Dirichlet problem for the Poisson equation

$$\begin{aligned} -\Delta u &= f \ \text{ in } \ \Omega, \\ u &= 0 \ \text{ on } \ \partial\Omega, \end{aligned} \tag{1}$$

where Ω is a polygonal domain in R^2. Let u_h be the finite element solution of (1), with linear triangular elements. Let us denote

$$e = u - u_h,$$

the approximation error, and

$$R(u_h) = f + \Delta u_h,$$

the residual. Following the technique of K. Eriksson et al. [10], we first express the error by means of product of residual and solution of the dual problem, then use the Galerkin orthogonality and get the estimate of the error, in the L_2-norm:

$$\|e\|_0^2 \leq \sum_{K\in\mathcal{T}_h} \Big\{ \|R(u_h)\|_{0,K} \|\varphi - \pi_h\varphi\|_{0,K} + \sum_{l\in\partial K} \left\| \frac{1}{2} \left[\!\left[\frac{\partial u_h}{\partial \boldsymbol{n}} \right]\!\right]_l \right\|_{0,l} \|\varphi - \pi_h\varphi\|_{0,l} \Big\}, \qquad (2)$$

where φ is the solution of the dual problem

$$\begin{aligned} -\Delta\varphi &= e \ \text{ in } \ \Omega, \\ \varphi &= 0 \ \text{ on } \ \partial\Omega, \end{aligned} \qquad (3)$$

$\pi_h\varphi$ means the interpolant of φ. The sum in (2) is taken over all triangles in the triangulation $\mathcal{T}_h$, the symbol $\left[\!\left[\frac{\partial u_h}{\partial \boldsymbol{n}}\right]\!\right]_l$ means the jump of the normal derivative $\frac{\partial u_h}{\partial \boldsymbol{n}}$ over the edge l of the triangle K.
Let us now distinguish 3 cases:

A) *General polygonal domain* Ω:
Let h_K be the largest side of the triangle K. The interpolation property together with the (low) regularity of the dual problem (3) yield

$$\|\varphi - \pi_h\varphi\|_{0,K} \leq C_I h_K \|\varphi\|_1 \leq C_I C_R h_K \|e\|_0.$$

Combining this with (2), we come to the a posteriori error estimate

$$\|e\|_0 \leq C_I C_R \sum_{K\in\mathcal{T}_h} h_K \Big\{ \|R(u_h)\|_{0,K} \ + h_K^{-\frac{1}{2}} \sum_{l\in\partial K} \left\| \frac{1}{2} \left[\!\left[\frac{\partial u_h}{\partial \boldsymbol{n}} \right]\!\right]_l \right\|_{0,l} \Big\}. \qquad (4)$$

B) *Convex polygon* Ω:
Now the regularity of the dual problem (3) is higher, cf. R. B. Kellogg, J. E. Osborn [13], and together with the interpolation property it gives

$$\|\varphi - \pi_h\varphi\|_{0,K} \leq C_I h_K^2 \|\varphi\|_2 \leq C_I C_R h_K^2 \|e\|_0.$$

Combining this with (2), we come to the more precise a posteriori estimate

$$\|e\|_0 \leq C_I C_R \sum_{K\in\mathcal{T}_h} h_K^2 \Big\{ \|R(u_h)\|_{0,K} \ + h_K^{-\frac{1}{2}} \sum_{l\in\partial K} \left\| \frac{1}{2} \left[\!\left[\frac{\partial u_h}{\partial \boldsymbol{n}} \right]\!\right]_l \right\|_{0,l} \Big\}. \qquad (5)$$

C) *Nonconvex polygon* Ω *with known singularity*:
It is well-known that the solution near the nonconvex corner, in the local spherical coordinates, has the form

$$u(r,\vartheta) = r^\gamma w(\vartheta),$$

where r is the distance from the corner, $\gamma \in (0,1)$. For instance, the case of the L-shaped domain with the interior angle $\omega = \frac{3}{2}\pi$ gives $\gamma = \frac{2}{3}$, cf. also [6]. Now the interpolation together with the above regularity gives

$$\|\varphi - \pi_h\varphi\|_{0,K} \leq C_I h_K^{1+\gamma-\varepsilon} \|\varphi\|_{H^{1+\gamma-\varepsilon}} \leq C_I C_R h_K^{1+\gamma-\varepsilon} \|e\|_0, \ \forall \varepsilon > 0,$$

which, combined with (2), finally leads to the a posteriori estimate

$$\|e\|_0 \leq C_I C_R \sum_{K\in\mathcal{T}_h} h_K^{1+\gamma-\varepsilon} \left\{ \|R(u_h)\|_{0,K} + h_K^{-\frac{1}{2}} \sum_{l\in\partial K} \left\| \frac{1}{2} \left[\!\left[\frac{\partial u_h}{\partial \boldsymbol{n}} \right]\!\right]_l \right\|_{0,l} \right\}, \tag{6}$$

valid $\forall \varepsilon > 0$. Of course, in (6) the parameter γ applies only in the nearest neighborhood of the corner.

Comparing the estimates (4), (5), (6) we see that the a posteriori error estimate depends significantly on the regularity of the problem. Having this in mind, we try to derive the a posteriori error estimate for the Stokes-Brinkman problem.

2. The Stokes-Brinkman model

Let Ω be a bounded Lipschitzian domain, $\Omega \subset R^2$, which consists of two parts: porous part Ω_p and fluid part Ω_f, $\bar{\Omega} = \bar{\Omega}_p \cup \bar{\Omega}_f$. The Stokes-Brinkman equation representing a mathematical model of a single phase flow in a porous/free flow media has the following form

$$\nu \mathbf{K}^{-1}\mathbf{v} + \nabla p - \nu^* \Delta \mathbf{v} = \mathbf{f} \quad \text{in } \Omega, \tag{7}$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega, \tag{8}$$

$$\mathbf{v} = \mathbf{w} \quad \text{on } \partial\Omega_D, \qquad \frac{\partial \mathbf{v}}{\partial n} - \mathbf{n}p = \mathbf{s} \quad \text{on } \partial\Omega_N, \tag{9}$$

where $\mathbf{v}$ is the vector of velocity, P is the pressure, $\mathbf{f}$ is the vector of external force, $\mathbf{n}$ is the outward-pointing normal to the boundary, ν^* is the effective viscosity and ν - the physical viscosity - is a uniform constant in the entire domain Ω. $\mathbf{K}$ is a symmetric permeability tensor, which in Ω_p is equal to the Darcy permeability of the porous media. Note that with the choice $\nu^* = 0$ in the vugular region Ω_p, the equation (7) reduces to the problem of Darcy's law. On the other hand by choosing $k_{ij} \to \infty$ (or very large) in fluid domain Ω_f, the equation (7) reduces to the problem of Stokes flow (here ν^* is taken equal to the physical fluid viscosity ν). Thus, the Stokes or Darcy's equations can be obtained by suitable choices of the parameters ν^* and $\mathbf{K}$ by defining them in vugular and rock matrix regions, respectively.

In the porous region ($\mathbf{K} < \infty$) it is known [14], that for moderately small permeabilities and pore fractions, the diffusive term $\nu^*\Delta\mathbf{v}$, where ν^* takes values close to the fluid viscosity ν, intoroduces only a small perturbation of the velocity and pressure fields in comparison with a pure Darcy law with $\nu^* = 0$. In [14] it is shown that Stokes-Brinkman equation with the choice $\nu^* = \nu$ in the porous region is very close to the solution of coupled Stokes and Darcy's equations.

The advantage of Stokes-Brinkman model is usage of uniform equations for porous and free flow domains. Boundary conditions between these two domains are represented by $\mathbf{K}$. This approach makes it possible to model heterogeneous material. Moreover, by a numerical point of view, it is easier to solve a monolithic system such as Stokes-Brinkman, in contrast to a coupled Darcy-Stokes system which requires an additional iterative scheme. Also, near the interface, Stokes-Brinkman equations allow us to avoid the typical grid refinement issues necessary for solving the interface between Darcy and Stokes region. On the other hand usage of Taylor-Hood elements for the whole domain requires big load of memory.

3. Weak formulation of Stokes-Brinkman equations

In what follows we denote $G = \mathbf{K}^{-1}$ and assume G is symmetric.

For the weak formulation we denote

$$\mathbf{H}^1_E := \{\mathbf{u} \in H^1(\Omega)^2 | \mathbf{u} = \mathbf{w} \text{ na } \partial\Omega_D\}, \tag{10}$$

$$\mathbf{H}^1_{E_0} := \{\mathbf{v} \in H^1(\Omega)^2 | \mathbf{v} = \mathbf{0} \text{ na } \partial\Omega_D\}. \tag{11}$$

Now the weak form of the Stokes-Brinkman problem reads:
Find $\boldsymbol{v} \in \mathbf{H}^1_{E_0}$ and $p \in L^2_0(\Omega)$ such that

$$\nu^* \int_\Omega \nabla \boldsymbol{v} : \nabla \boldsymbol{v^*} + \nu \int_\Omega \boldsymbol{v^T} G \boldsymbol{v^*} - \int_\Omega p \nabla \cdot \boldsymbol{v^*} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v^*} \qquad \forall \boldsymbol{v^*} \in \mathbf{H}^1_{E_0}, \tag{12}$$

$$\int_\Omega q \nabla \cdot \boldsymbol{v} = 0 \qquad \forall q \in L^2_0(\Omega). \tag{13}$$

Here $L^2_0(\Omega)$ is the space of L^2 functions having mean value zero.

On the space $V = \Big(H^1_0(\Omega)^2 \times L^2_0(\Omega)\Big)$ we define the bilinear form

$$\mathcal{A}\Big(\{\boldsymbol{v}, p\}, \{\boldsymbol{v}^*, p^*\}\Big) = \nu^* \int_\Omega \nabla \boldsymbol{v} : \nabla \boldsymbol{v}^* + \nu \int_\Omega \boldsymbol{v^T} G \boldsymbol{v}^* - \int_\Omega p \nabla \cdot \boldsymbol{v}^* - \int_\Omega p^* \nabla \cdot \boldsymbol{v} \tag{14}$$

where $(.,.)_0$ means the scalar product in L^2.

In what follows we assume $\mathbf{w} = 0$, i. e. only zero Dirichlet condition on the whole boundary $\partial\Omega$. Problem (12), (13) can be written as follows: find $\{\boldsymbol{v}, p\} \in V$, such that

$$\mathcal{A}\Big(\{\boldsymbol{v}, p\}, \{\boldsymbol{v}^*, p^*\}\Big) = (\boldsymbol{f}, \boldsymbol{v}^*)_0, \quad \forall \{\boldsymbol{v}^*, p^*\} \in V. \tag{15}$$

4. Finite element approximation

We suppose Ω to be a polygon, for simplicity. Let $\mathcal{T}_h$ be regular [11] triangulations of Ω. Let X^h, M^h be the finite element spaces of Taylor-Hood elements (cf. e.g. F. Brezzi, M. Fortin [4]), i.e.

$$X^h = \{\boldsymbol{v} \in H^1_0(\Omega)^2, \boldsymbol{v}/_T \in P^2(T)^2, T \in \mathcal{T}_h\},$$
$$M^h = \{p \in L^2_0(\Omega), p/_T \in P^1(T), T \in \mathcal{T}_h\}.$$

These satisfy the Babuška-Brezzi condition [4]. The finite element approximation of the Stokes-Brinkman problem consists in finding $\{\boldsymbol{v}_h, p_h\} \in X^h \times M^h$ such that

$$\mathcal{A}\Big(\{\boldsymbol{v}_h, p_h\}, \{\boldsymbol{v}_h^*, p_h^*\}\Big) = (\boldsymbol{f}, \boldsymbol{v}_h^*)_0 \ , \quad \forall \{\boldsymbol{v}_h^*, p_h^*\} \in X^h \times M^h. \tag{16}$$

5. A posteriori error estimate

We follow the idea of K. Eriksson et al. [10] who proved the a posteriori error estimate for the Poisson equation. We define the residual components by the relations

$$\boldsymbol{R_1}\{\boldsymbol{v}_h, p_h\} = \boldsymbol{f} + \nu^* \Delta \boldsymbol{v}_h - \nu G \boldsymbol{v}_h - \nabla p_h, \quad R_2\{\boldsymbol{v}_h, p_h\} = \text{div } \boldsymbol{v}_h. \tag{17}$$

Next we study the properties of the errors

$$\boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}_h \ , \ e_p = p - p_h \ ,$$

where $\{\boldsymbol{v}, p\}$ is the exact solution of (15), $\{\boldsymbol{v}_h, p_h\}$ is the approximate solution defined in (16). The V norm of $\{\boldsymbol{e}_v, e_p\}$ is

$$\|\{\boldsymbol{e}_v, e_p\}\|_V^2 = (\boldsymbol{e}_v, \boldsymbol{e}_v)_1 + (e_p, e_p)_0 = \int_\Omega (\boldsymbol{e}_v \cdot \boldsymbol{e}_v + \nabla \boldsymbol{e}_v : \nabla \boldsymbol{e}_v) + \int_\Omega e_p e_p.$$

By the Poincaré-Friedrichs inequality, cf. [9], as $\boldsymbol{e}_v \in H_0^1(\Omega)^2$

$$(\boldsymbol{e}_v, \boldsymbol{e}_v)_1 \leq C_P \int_\Omega \nabla \boldsymbol{e}_v : \nabla \boldsymbol{e}_v \tag{18}$$

5.1. Dual Stokes-Brinkman problem

To study the above norms we introduce the dual Brinkman-Stokes problem by

$$\begin{aligned} -\nu^* \Delta \boldsymbol{\varphi_v} + \nu G \boldsymbol{\varphi_v} + \nabla \varphi_p &= -\Delta \boldsymbol{e}_v \quad \text{in } \Omega, \text{ here } \Delta \boldsymbol{e}_v \in H^{-1}(\Omega) \\ -\text{div } \boldsymbol{\varphi_v} &= e_p \quad \text{in } \Omega, \\ \boldsymbol{\varphi_v} &= \boldsymbol{0} \quad \text{on } \partial\Omega, \end{aligned} \tag{19}$$

which in a weak form is: find $\boldsymbol{\varphi_v} \in H^1(\Omega)^2$ and $\varphi_p \in L_0^2(\Omega)$ such that

$$\begin{aligned} (\nu^* \nabla \boldsymbol{\varphi_v}, \nabla \boldsymbol{v}^*)_0 + \nu((G\boldsymbol{\varphi_v}), \boldsymbol{v}^*) - (\varphi_p, \nabla \boldsymbol{v}^*)_0 = (\nabla \boldsymbol{e}_v, \nabla \boldsymbol{v}^*)_0, \ \forall \boldsymbol{v}^* \in H_0^1(\Omega)^2, \\ (-\text{div } \boldsymbol{\varphi_v}, p^*)_0 = (e_p, p^*)_0, \ \forall p^* \in L_0^2(\Omega), \end{aligned} \tag{20}$$

or, using the notation (14)

$$\mathcal{A}(\{\boldsymbol{\varphi_v}, \varphi_p\}, \{\boldsymbol{v}^*, p^*\}) = (\nabla \boldsymbol{e}_v, \nabla \boldsymbol{v}^*)_0 + (e_p, p^*)_0 \ , \ \forall \{\boldsymbol{v}^*, p^*\} \in V \ . \tag{21}$$

By (18) and (20) where we put $\boldsymbol{v}^* = \boldsymbol{e}_v, \ p^* = e_p$, we get

$$\begin{aligned} \frac{1}{C_P}(\boldsymbol{e}_v, \boldsymbol{e_v})_1 \leq \ & (\nabla \boldsymbol{e_v}, \nabla \boldsymbol{e_v})_0 = \nu^*(\nabla \boldsymbol{\varphi_v}, \nabla \boldsymbol{e_v})_0 + \nu((G\boldsymbol{\varphi_v}), \boldsymbol{e_v}) - (\varphi_p \nabla, \boldsymbol{e_v})_0 \\ &= \nu^*(\nabla \boldsymbol{\varphi_v}, \nabla \boldsymbol{v})_0 + \nu((G\boldsymbol{\varphi_v})\boldsymbol{v}) - (\varphi_p \nabla, \boldsymbol{v})_0 - \nu^*(\nabla \boldsymbol{\varphi_v}, \nabla \boldsymbol{v_h})_0 \\ &\quad - \nu((G\boldsymbol{\varphi_v})\boldsymbol{v_h}) + (\varphi_p \nabla, \boldsymbol{v_h})_0, \end{aligned} \tag{22}$$

$$(e_p, e_p)_0 = (e_p, -\text{div } \boldsymbol{\varphi_v})_0 = -(p\nabla, \boldsymbol{\varphi_v})_0 + (p_h \nabla, \boldsymbol{\varphi_v})_0. \tag{23}$$

5.2. Estimation of the error by means of the residual and solution of the dual problem

Combining (22), (23), and (19) we get (as $C_P \geq 1$)

$$
\begin{aligned}
&\frac{1}{C_P}\Big\{(\boldsymbol{e}_v, \boldsymbol{e}_v)_1 + (e_p, e_p)_0\Big\} \\
&\leq \nu^*(\nabla \boldsymbol{v}, \nabla \boldsymbol{\varphi_v})_0 + \nu((G\boldsymbol{v}\varphi_v)) - (p, \nabla \boldsymbol{\varphi_v})_0 - (\nabla \boldsymbol{v}, \varphi_p)_0 \\
&\quad + \sum_{K\in\mathcal{T}_h} \{-\nu^*(\nabla \boldsymbol{\varphi_v}, \nabla \boldsymbol{v}_h)_{0,K} - \nu((G\boldsymbol{v_h}\varphi_v)) + (p_h, \nabla \boldsymbol{\varphi_v})_{0,K} + (\varphi_p, \nabla \boldsymbol{v}_h)_{0,K}\} \\
&= (\boldsymbol{f}, \boldsymbol{\varphi_v})_0 + \sum_{K\in\mathcal{T}_h} \left\{(\nu^*\Delta \boldsymbol{v}_h, \boldsymbol{\varphi_v})_{0,K} - \int_{\partial K} \nu^* \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} \boldsymbol{\varphi_v} ds\right\} - \nu((G\boldsymbol{v_h}\varphi_v)) \qquad (24) \\
&\quad - \sum_{K\in\mathcal{T}_h} \left\{(\nabla p_h, \boldsymbol{\varphi_v})_{0,K} + \int_{\partial K} p_h \boldsymbol{\varphi_v} \cdot \boldsymbol{n} ds \ + (\text{div } \boldsymbol{v}_h, \varphi_p)_{0,K}\right\} \\
&= \sum_{K\in\mathcal{T}_h} (\boldsymbol{f} + \nu^*\Delta \boldsymbol{v}_h - \nu((G\boldsymbol{v_h}\varphi_v)) - \nabla p_h, \boldsymbol{\varphi_v})_{0,K} + \sum_{K\in\mathcal{T}_h} (\text{div } \boldsymbol{v}_h, \varphi_p)_{0,K} \\
&\quad - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \nu^* \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} \boldsymbol{\varphi_v} ds + \sum_{K\in\mathcal{T}_h} \int_{\partial K} p_h \boldsymbol{\varphi_v} \cdot \boldsymbol{n} ds
\end{aligned}
$$

In view of (16) we also have

$$
\begin{aligned}
&\sum_{K\in\mathcal{T}_h} (\boldsymbol{f} + \nu^*\Delta \boldsymbol{v}_h - \nu G\boldsymbol{v}_h - \nabla p_h, \boldsymbol{v}_h^*)_{0,K} + \ (\text{div } \boldsymbol{v}_h, p_h^*)_0 \\
&= (\boldsymbol{f}, \boldsymbol{v}_h^*)_0 + \sum_{K\in\mathcal{T}_h} \left\{(-\nu^*\nabla \boldsymbol{v}_h, \nabla \boldsymbol{v}_h^*)_{0,K} - \nu(G\boldsymbol{v}_h, \boldsymbol{v}_h^*) + \int_{\partial K} \nu^* \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} \boldsymbol{v}_h^* ds\right\} \\
&\quad + (\nabla p_h, \boldsymbol{v}_h^*)_0 - \sum_{K\in\mathcal{T}_h} \int_{\partial K} p_h \boldsymbol{v}_h^* \cdot \boldsymbol{n} ds + (\text{div } \boldsymbol{v}_h, p_h^*)_0 \qquad (25) \\
&= 0 + \sum_{K\in\mathcal{T}_h} \int_{\partial K} \nu \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} \boldsymbol{v}_h^* ds - \sum_{K\in\mathcal{T}_h} \int_{\partial K} p_h \boldsymbol{v}_h^* \cdot \boldsymbol{n} ds, \ \forall \{\boldsymbol{v}_h^*, p_h^*\} \in X^h \times M^h.
\end{aligned}
$$

This implies, taking $\boldsymbol{v}_h^* = \pi_h \boldsymbol{\varphi_v}$, $p_h^* = \pi_h \varphi_p$, the Clement interpolants, (cf. e.g. [8], p. 146) that

$$
\begin{aligned}
&\sum_{K\in\mathcal{T}_h} (\boldsymbol{f} + \nu^*\Delta \boldsymbol{v}_h - \nu G\boldsymbol{v}_h - \nabla p_h, \pi_h \boldsymbol{\varphi_v}) + \ (\text{div } \boldsymbol{v}_h, \pi_h \varphi_p)_0 \\
&\qquad\qquad - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \nu^* \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} \pi_h \boldsymbol{\varphi_v} ds - \sum_{K\in\mathcal{T}_h} \int_{\partial K} p_h \pi_h \boldsymbol{\varphi_v} \cdot \boldsymbol{n} ds = 0 \qquad (26)
\end{aligned}
$$

Now subtracting zero in (26) from (24) we get

$$
\begin{aligned}
&\frac{1}{C_P}\Big\{(\boldsymbol{e}_v,\boldsymbol{e}_v)_1+(e_p,e_p)_0\Big\}\\
&\le \sum_{K\in\mathcal{T}_h}(\boldsymbol{f}+\nu^*\Delta\boldsymbol{v}_h-\nu G\boldsymbol{v}_h-\nabla p_h,\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v})_{0,K}+\ (\text{div }\boldsymbol{v}_h,\varphi_p-\pi_h\varphi_p)_0\\
&\quad-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}(\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v})ds+\sum_{K\in\mathcal{T}_h}\int_{\partial K}p_h(\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v})\cdot\boldsymbol{n}ds \qquad (27)\\
&=\sum_{K\in\mathcal{T}_h}(\boldsymbol{f}+\nu^*\Delta\boldsymbol{v}_h-\nu G\boldsymbol{v}_h-\nabla p_h,\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v})_{0,K}+\ (\text{div }\boldsymbol{v}_h,\varphi_p-\pi_h\varphi_p)_0\\
&\quad-\sum_{K\in\mathcal{T}_h}\sum_{l\in\partial K}\int_l\left(\frac{1}{2}\left[\!\!\left[\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right]\!\!\right]_l\right)(\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v})ds,
\end{aligned}
$$

where we denoted

$$
\left[\!\!\left[\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right]\!\!\right]_l=\left(\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right)\Big|_{l+}-\left(\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right)\Big|_{l-}
$$

the jump along the common side l of two adjacent triangles. Then, using in turn the Schwarz inequality, the interpolation properties of $X^h,\ M^h$ (cf. e.g. [4]), and the estimate of the solution of the dual problem (19) (cf. [4]), we get the inequalities

$$
\begin{aligned}
&\|\boldsymbol{e}_v\|_1^2+\|e_p\|_0^2\\
&\le C_P\sum_{K\in\mathcal{T}_h}\Big\{\|\boldsymbol{R_1}\{\boldsymbol{v}_h,p_h\}\|_{0,K}\|\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v}\|_{0,K}+\|R_2\{\boldsymbol{v}_h,p_h\}\|_{0,K}\|\varphi_p-\pi_h\varphi_p\|_{0,K}\Big\}\\
&\quad+C_P\sum_{K\in\mathcal{T}_h}\sum_{l\in\partial K}\left\|\frac{1}{2}\left[\!\!\left[\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right]\!\!\right]_l\right\|_{0,l}\|\boldsymbol{\varphi_v}-\pi_h\boldsymbol{\varphi_v}\|_{0,l} \qquad (28)\\
&\le C_PC_I\sum_{K\in\mathcal{T}_h}\Big\{h_K\ \|\boldsymbol{R_1}\{\boldsymbol{v}_h,p_h\}\|_{0,K}\ \|\boldsymbol{\varphi_v}\|_1+\|R_2\{\boldsymbol{v}_h,p_h\}\|_{0,K}\ \|\varphi_p\|_0\Big\}\\
&\quad+C_PC_I\sum_{K\in\mathcal{T}_h}(h_K)^{\frac{1}{2}}\sum_{l\in\partial K}\left\|\frac{1}{2}\left[\!\!\left[\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right]\!\!\right]_l\right\|_{0,l}\|\boldsymbol{\varphi_v}\|_1\\
&\le C_PC_IC_R\sum_{K\in\mathcal{T}_h}\Bigg\{h_K\ \|\boldsymbol{R}_1\{\boldsymbol{v}_h,p_h\}\|_{0,K}\ +\|R_2\{\boldsymbol{v}_h,p_h\}\|_{0,K}\\
&\quad+\sum_{l\in\partial K}(h_K)^{\frac{1}{2}}\ \left\|\frac{1}{2}\left[\!\!\left[\nu\frac{\partial\boldsymbol{v}_h}{\partial\boldsymbol{n}}-p_h\boldsymbol{n}\right]\!\!\right]_l\right\|_{0,l}\Bigg\}\cdot\big\{\|\Delta\boldsymbol{e}_v\|_{-1}+\|e_p\|_0\big\}.
\end{aligned}
$$

Using then the relations

$$
\begin{aligned}
\|\Delta\boldsymbol{e}_v\|_{-1}&\equiv\sup_{\boldsymbol{v}^*\in H_0^1,\boldsymbol{v}^*\neq0}\frac{|(\Delta\boldsymbol{e}_v,\boldsymbol{v}^*)_0|}{\|\boldsymbol{v}^*\|_1}=\sup_{\boldsymbol{v}^*\in H_0^1,\boldsymbol{v}^*\neq0}\frac{|(\nabla\boldsymbol{e}_v,\nabla\boldsymbol{v}^*)_0|}{\|\boldsymbol{v}^*\|_1}\\
&\le\sup_{\boldsymbol{v}^*\in H_0^1,\boldsymbol{v}^*\neq0}\frac{\|\nabla\boldsymbol{e}_v\|_0\ \|\nabla\boldsymbol{v}^*\|_0}{\|\boldsymbol{v}^*\|_1}\le\|\nabla\boldsymbol{e}_v\|_0\le\|\boldsymbol{e}_v\|_1
\end{aligned}
$$

we get, by (28)

$$\left\{\|\boldsymbol{e}_v\|_1 + \|e_p\|_0\right\}^2 \leq 2\left\{\|\boldsymbol{e}_v\|_1^2 + \|e_p\|_0^2\right\} \leq 2C_P C_I C_R \sum_{K\in\mathcal{T}_h} \left\{ h_K \|\boldsymbol{R_1}\{\boldsymbol{v}_h, p_h\}\|_{0,K} \right.$$
$$\left. + \|R_2\{\boldsymbol{v}_h, p_h\}\|_{0,K} + h_K^{\frac{1}{2}} \sum_{l\in\partial K} \left\| \frac{1}{2} \left[\!\left[\nu \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} - p_h \boldsymbol{n} \right]\!\right]_l \right\|_{0,l} \right\} \cdot \left\{\|\boldsymbol{e}_v\|_1 + \|e_p\|_0\right\}. \quad (29)$$

Upon cancelling $\left\{\|\boldsymbol{e}_v\|_1 + \|e_p\|_0\right\}$ in (29) we finally get the following theorem:

Theorem 1. *Let Ω be a polygon in R^2. Let $\mathcal{T}_h$ be a family of regular triangulations of Ω. Let $\{\boldsymbol{v_h}, p_h\}$ be the Taylor-Hood approximation of the solution $\{\boldsymbol{v}, p\}$ of the Stokes-Brinkman problem. Then the error $\{\boldsymbol{e}_v, e_p\}$ satisfies the following a posteriori estimate*

$$\|\boldsymbol{e}_v\|_1 + \|e_p\|_0 \leq 2C_P C_I C_R \sum_{K\in\mathcal{T}_h} \left\{ h_K \|\boldsymbol{R}_1\{\boldsymbol{v}_h, p_h\}\|_{0,K} + \|R_2\{\boldsymbol{v}_h, p_h\}\|_{0,K} \right.$$
$$\left. + h_K^{\frac{1}{2}} \sum_{l\in\partial K} \left\| \frac{1}{2} \left[\!\left[\nu \frac{\partial \boldsymbol{v}_h}{\partial \boldsymbol{n}} - p_h \boldsymbol{n} \right]\!\right]_l \right\|_{0,l} \right\}. \quad (30)$$

where C_P, C_I, C_R are positive constants, residuals $\boldsymbol{R}_1$ and R_2 are defined in (17) .

Conclusions

The estimate in Theorem 1 applies to more general class of elements. Of course, for Taylor-Hood elements with continuous pressure the jumps of p_h along the common sides disappear.

Let us note that for convex domains stronger regularity applies to the Stokes problem, cf. [13], and better a posteriori error estimate may be expected.

For nonconvex domains with corners we do not obtain so strong regularity as in [13], cf. e.g. [6], but still the a posteriori error estimate should be better than in (30), as it was for the Poisson equation in (2).

Acknowledgments

This work was supported by the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070, and by the grant Kontakt II number LH11004.

References

[1] Ainsworth, M. and Oden, J. T.: Aposteriori error estimators for the Stokes and Oseen problems. SIAM J. Numer. Anal. **34** (1997), 228–245.

[2] Babuška, I. and Rheinboldt, W. C.: A posteriori error estimates for the finite element method. Int. J. Numer. Meth. Eng. **12** (1978), 1597–1615.

[3] Babuška, I., Durán, R., and Rodríguez, R.: Analysis of the efficiency of an a posteriori error estimator for linear triangular finite elements. SIAM J. Numer. Anal. **29** (1992), 947–964.

[4] Brezzi, F. and Fortin, M., *Mixed and hybrid finite element methods*. Springer, Berlin, 1991.

[5] Bank, R. E. and Welfert, D.: A posteriori error estimates for the Stokes equations: A comparison. Comp. Meth. Appl. Mech. Eng. **82** (1990), 323–340.

[6] Burda, P.: On the F.E.M. for the Navier-Stokes equations in domains with corner singularities. In: M. Křížek, P. Neittaanmäki, and R. Stenberg (Eds.), *Finite Element Methods, Supeconvergence, Post-Processing and A Posteriori Estimates*, pp. 41–52. Marcel Dekker, New York, 1998.

[7] Carstensen, C. and Jansche, S.: A posteriori error estimates for the finite element discretizations of the Stokes problem, Berichtsreihe Math. Sem. Kiel. Techn. Report 97–9, 1997.

[8] Ciarlet, P. G.: *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1980.

[9] Elman, H. C., Silvester, D., and Wathen A. J.: *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, New York, 1994.

[10] Eriksson, K., Estep, D., Hansbo, P., and Johnson, C.: *Introduction to adaptive methods for differential equations*. Acta Numerica, CUP (1995), 105–158.

[11] Girault, V. and Raviart, P. G.: *Finite element method for Navier-Stokes equations*. Springer, Berlin, 1986.

[12] Johnson, C., Rannacher, R., and Boman, M.: Numerics and hydrodynamic stability: towards error control in computational fluid dynamics. SIAM J. Numer. Anal. **32** (1995), 1058–1079.

[13] Kellogg, R. B. and Osborn, J. E.: A regularity result for the Stokes problem in a convex polygon. J. Funct. Anal. **21** (1976), 397–431.

[14] Popov, P., Efendiev, L. B. Y., Erwing, R. E., and Quin, G.: Multi-physics and multi-scale methods for modeling fluid flow through naturally-fractured vuggy carbonate reservoirs. *SPE International*. SPE 105378

[15] Verfürth, R.: *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. Wiley and Teubner, Chichester, 1996.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

QUANTITATIVE PROPERTIES OF QUADRATIC SPLINE WAVELET BASES IN HIGHER DIMENSIONS

Dana Černá[1], Václav Finěk[1], Martina Šimůnková [2]

[1] KMD FP TU Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
Vaclav.Finek@tul.cz, Dana.Cerna@tul.cz
[2] KAP FP TU Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
martina.simunkova@tul.cz

Abstract

To use wavelets efficiently to solve numerically partial differential equations in higher dimensions, it is necessary to have at one's disposal suitable wavelet bases. Ideal 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 our contribution, we compare condition numbers of different quadratic spline wavelet bases in dimensions $d = 1, 2$ and 3 on tensor product domains $(0, 1)^d$.

1. Introduction

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 [1]. 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, dual wavelets are also compactly supported. Here, we use recently proposed wavelets based on quadratic splines [2, 3, 4, 5] and propose one other modification. These wavelets have shorter supports, are better conditioned but dual wavelets are not compactly supported. Due to their properties these wavelet bases can be used in the wavelet Galerkin method as well as in adaptive wavelet methods for solving second-order elliptic problems with homogeneous Dirichlet boundary conditions.

2. Wavelet bases

We consider here families $\Psi = \{\psi_\lambda, \lambda \in \mathcal{J}\} \subset L_2(0, 1)$ of functions (wavelets). Let $\mathcal{J}$ be an infinite index set and $\mathcal{J} = \mathcal{J}_\Phi \cup \mathcal{J}_\Psi$, where $\mathcal{J}_\Phi$ is a finite set representing scaling functions living on the coarsest scale. Any index $\lambda \in \mathcal{J}$ is of the form $\lambda = (j, k)$, where $|\lambda| = j$ denotes a scale and k denotes spatial location. At last,

for $s \geq 0$ the space H^s will denote a closed subspace of the Sobolev space $H^s(0,1)$, defined e.g. by imposing homogeneous boundary conditions at one or both endpoints, and for $s < 0$ the space H^s will denote the dual space $H^s := (H^{-s})'$. $\|.\|_{H^s}$ will denote the corresponding norm. Further $l_2(\mathcal{J})$ will denote the space consisting of the power summable sequences and $\|.\|_{l_2(\mathcal{J})}$ will denote the corresponding norm.

A family $\Psi = \{\psi_\lambda, \lambda \in \mathcal{J}\} \subset L_2(0,1)$ is called a *wavelet basis* of H^s for some $\gamma, \widetilde{\gamma} > 0$ and $s \in (-\widetilde{\gamma}, \gamma)$, if

- Ψ is a Riesz basis of H^s, that means Ψ forms a basis of H^s and there exist constants $c_s, C_s > 0$ such that for all $\mathbf{b} = \{b_\lambda\}_{\lambda \in \mathcal{J}} \in l_2(\mathcal{J})$ holds

$$c_s \|\mathbf{b}\|_{l_2(\mathcal{J})} \leq \left\|\mathbf{b}^T \Psi\right\|_{H^s} \leq C_s \|\mathbf{b}\|_{l_2(\mathcal{J})},$$

  where $\sup c_s$, $\inf C_s$ are called Riesz bounds and $\mathrm{cond}\Psi := \dfrac{\inf C_s}{\sup c_s}$ is called the condition number of Ψ.

- Functions are local in the sense that $\mathrm{diam}(\mathrm{supp}\,\psi_\lambda) \leq C 2^{-|\lambda|}$ for all $\lambda \in \mathcal{J}$, where C is a constant independent of λ.

- Functions ψ_λ, $\lambda \in \mathcal{J}_\Psi$, have cancellation properties of order m, i.e.

$$\left| \int_0^1 v(x)\, \psi_\lambda(x)\, dx \right| \leq 2^{-m|\lambda|} |v|_{H^m(0,1)}, \qquad \forall v \in H^m(0,1).$$

  It means that integration against wavelets eliminates smooth parts of functions and it is equivalent with vanishing wavelet moments of order m.

3. Scaling functions

As inner scaling functions, we use quadratic B-splines, because they have short support and can be easily adapted to a bounded interval by employing multiple knots at the endpoints. In the case of quadratic basis, there is necessary to add only one boundary scaling function at each boundary to preserve polynomial exactness and homogeneous boundary conditions. Specifically, the quadratic spline function $\phi(x)$ is given by

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

and the left boundary function $\phi_b(x)$ is given by

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

Then a scaling basis satisfying homogeneous Dirichlet boundary conditions is determined by

$$\Phi_j = \left\{ \phi_{j,k} / \left|\phi_{j,k}\right|_{H_0^1(0,1)}, k = 1, \ldots, 2^j \right\},$$

where for $j \geq 2$ and $x \in [0,1]$ we define

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

4. Wavelets and wavelet bases

Constructed wavelets should have small support and vanishing moments and the corresponding wavelet basis should be well-conditioned. Unlike biorthogonal wavelets [1, 6], where primal wavelets have significantly larger support than scaling functions and dual wavelets are local, we focus here on primal wavelets which have the same length of support as scaling functions or shorter. Let us denote the space spanned by the set Φ_j by V_j. Then, we define complement spaces W_j by $V_{j+1} = V_j \oplus W_j$ and a wavelet basis is constructed to be the basis of W_j.

First, we look at inner wavelets with minimal support and with the number of vanishing moments that equals to the degree of used B-splines. The quadratic spline-wavelet is then given by

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

In [2], the boundary wavelet was constructed by prescribing the number of vanishing moments, the support in the interval $[0, 5/2]$, homogeneous Dirichlet boundary conditions and finally, it should be from the space spanned by $\{\phi_b(2x),\ \phi(2x-k) :\ k \in \mathbb{N}_0\}$. This boundary wavelet is determined by

$$\psi_b^1(x) = -\frac{5}{2}\phi_b(2x) + \frac{47}{12}\phi(2x) - \frac{13}{4}\phi(2x-1) + \phi(2x-2).$$

In [3], we constructed a new boundary wavelet by allowing its support to be in the interval $[0, 3]$ and prescribing three vanishing moments, homogeneous Dirichlet boundary conditions and finally, it again should be from the space spanned by $\{\phi_b(2x),\ \phi(2x-k) :\ k \in \mathbb{N}_0\}$. Then, there are infinitely many solutions and we selected one that has zero wavelet coefficient corresponding to the scaling function $\phi(2x-2)$. Consequently, the arising system has exactly one solution up to multiplication by a constant. This boundary wavelet is given by

$$\psi_b^2(x) = -\frac{15}{2}\phi_b(2x) + \frac{43}{4}\phi(2x) - \frac{27}{4}\phi(2x-1) + \phi(2x-3).$$

In [4], we propose the boundary wavelet prescribing the same properties as above. But we use the free parameter identified in [3] to ensure the orthogonality of constructed boundary wavelet with the nearest inner wavelet:

$$\psi_b^3(x) = -\frac{920}{209}\phi_b(2x) + \frac{3697}{627}\phi(2x) - \frac{569}{209}\phi(2x-1) - \frac{259}{209}\phi(2x-2) + \phi(2x-3).$$

Here, we use also the above mentioned free parameter to ensure the orthogonality of the first derivative of the constructed boundary wavelet with the first derivative of the nearest inner wavelet:

$$\psi_b^4(x) = -\frac{40}{13}\phi_b(2x) + \frac{149}{39}\phi(2x) - \phi(2x-1) - \frac{23}{13}\phi(2x-2) + \phi(2x-3).$$

Further, we look at wavelets with one vanishing moment. An inner wavelet ψ with $\operatorname{supp}\psi = [0.5, 2.5]$ is then given by

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

And a boundary wavelet ψ_b with $\operatorname{supp}\psi_b = [0, 1.5]$ and with one vanishing wavelet moment is defined by:

$$\psi_b(x) = \frac{\phi_b(2x)}{2} - \frac{\phi(2x)}{3}.$$

For $j \geq 2$ and $x \in [0,1]$ we define

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

We denote

$$\Psi_j = \left\{ \psi_{j,k} / \left|\psi_{j,k}\right|_{H_0^1(0,1)}, k = 1, \ldots, 2^j \right\}.$$

Then the sets

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

are a multiscale wavelet basis and a wavelet basis of the space $H_0^1(0,1)$, respectively. The proof can be found in [5]. Multiscale wavelet bases and wavelet bases for other construction can be defined in a similar way.

To define wavelets in higher dimensions we use the tensor product. The tensor product of functions u and v is defined by $(u \otimes v)(x_1, x_2) := u(x_1)\, v(x_2)$. We show an example of such wavelet basis in dimension $d = 2$. We set

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

where $\Omega = (0,1)^2$. Then the sets defined by

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

are a wavelet basis and a multiscale wavelet basis of the space $H_0^1(\Omega)$.

5. Condition numbers

We compute condition numbers of stiffness matrices corresponding to the following Dirichlet problem

$$-\sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2} = f \quad \text{in} \quad \Omega = (0,1)^d \qquad \text{with} \quad u = 0 \quad \text{on} \quad \partial\Omega$$

discretized using above mentioned wavelet bases. These condition numbers are closely related to the condition number of wavelet basis of the space $H_0^1(\Omega)$. In all tables, $B3^i$ will denote the wavelet basis with three vanishing wavelet moments and with boundary wavelet ψ_b^i, $B1$ will denote the wavelet basis with one vanishing wavelet moment and finally n will denote a number of used basis functions.

n	$B3^1$	$B3^2$	$B3^3$	$B3^4$	$B1$
8	8.9	5.4	3.9	3.5	2.8
16	10.1	6.1	4.6	4.4	2.8
32	10.6	6.5	4.9	4.7	2.8
64	10.8	6.6	5.0	4.8	2.8
128	10.9	6.7	5.1	4.9	2.8
256	10.9	6.8	5.2	4.9	2.8
512	10.9	6.8	5.2	4.9	2.8
1024	11.0	6.8	5.2	4.9	2.8
2048	11.0	6.9	5.2	4.9	2.8
4096	11.0	6.9	5.2	4.9	2.8

Table 1: Condition numbers for $d = 1$

n	$B3^1$	$B3^2$	$B3^3$	$B3^4$	$B1$
64	56.6	29.3	18.7	21.9	7.5
256	83.6	42.1	29.9	32.6	11.0
1096	98.7	50.2	36.7	38.0	13.6
4096	107.5	55.4	40.6	40.8	15.3
16384	113.1	58.7	43.1	42.5	16.5
65536	116.8	61.0	44.6	43.7	17.3
262144	119.4	62.5	45.7	44.4	17.9
1048576	121.3	63.7	46.5	44.9	18.3

Table 2: Condition numbers for $d = 2$

n	$B3^1$	$B3^2$	$B3^3$	$B3^4$	$B1$
512	470.8	227.7	169.3	208.5	47.4
4096	815,9	402.2	313.2	362.6	85.0
32768	1027,1	500.9	389.1	429.5	113.8
262144	1153,6	552.9	425.0	459.0	132.9
2097152	1230,7	581.8	443.5	474.1	145.3

Table 3: Condition numbers for $d = 3$

Acknowledgements

This work has been supported by the SGS project "Modern numerical methods II" financed by Technical University of Liberec and special thanks belongs to T. Šimková for her help with numerical experiments.

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] Černá, D., Finěk, V. and Šimůnková, M.: A quadratic spline-wavelet basis on the interval. In: J. Chleboun, K. Segeth, J. Šístek, and T. Vejchodský (Eds.), *Programs and Algorithms of Numerical Matematics 16*. Institute of Mathematics AS CR, Prague 2012.

[3] Černá, D. and Finěk, V.: Quadratic wavelets with short support on the interval. In: G. Venkov, R. Kovacheva, and V. Pasheva (Eds.), *AMEE – Applications of Mathematics in Engineering and Economics*. American Institute of Physics, New York, 2012.

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

[5] Černá, D. and Finěk, V.: Quadratic spline wavelets with short support. In preparation.

[6] Dahmen, W., Kunoth, A., and Urban, K.: Biorthogonal spline wavelets on the interval – stability and moment conditions. Appl. Comp. Harm. Anal. **6** (1999), 132–196.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

DIFFERENT APPROACHES TO INTERFACE WEIGHTS IN THE BDDC METHOD IN 3D

Marta Čertíková[1], Jakub Šístek[2], 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

In this paper, we discuss the choice of weights in averaging of local (subdomain) solutions on the interface for the BDDC method (Balancing Domain Decomposition by Constraints). We try to find relations among different choices of the interface weights and compare them numerically on model problems of the Poisson equation and linear elasticity in 3D. Problems with jumps in coefficients of material properties are considered and both regular and irregular interfaces between subdomains are tested.

1. Introduction

An important ingredient of many domain decomposition methods is a technique used for determining a continuous approximation of solution at interface from discontinuous local solutions from adjacent subdomains. A standard approach described already in [3] is to compute global value of any given interface unknown as some weighted average of local (subdomain) values of the corresponding unknown only. Very often just arithmetic average is used, based simply on counting number of subdomains to which the interface unknown belongs. More sophisticated method is to derive the weights from diagonal stiffness of subdomain Schur complements with respect to the interface. As these complements are typically not computed explicitly in efficient implementations, the diagonal of the Schur complement is sometimes approximated by the diagonal of the original matrix (also known as the *diagonal stiffness scaling*). Another method is the so called *ρ-scaling* (see e.g. [5] for theoretical analysis). However, it is limited to the case of material coefficients constant on each subdomain, which is often too restrictive requirement for applications, and it is also not preserved in our examples. Nevertheless, we tried its modification, using material coefficients on individual elements instead. In any case, this approach requires access to material coefficients from the equations solver in an implementation.

A different recent method of evaluation of global values, called *deluxe scaling*, represents solving local problems containing two or more adjacent subdomains (see e.g. [4]). Implementation of this method is quite demanding, and it is not covered by our numerical experiments, although we involve it in our theoretical considerations.

Our method of *averaged unit jump* was originally derived by approximate minimization of the upper bound on the condition number of the preconditioned operator (see e.g. [1]).

In this paper, we analyze theoretically relationships among methods mentioned above. We use an abstract formulation of the BDDC preconditioner presented in [6] in order to obtain a clearer form of our results formulated in Lemma 1 that seems to be new. We also test the methods numerically on 3D Poisson and linear elasticity problems, together with two heuristic methods (called *unit jump* and *unit load*; *unit load* was proposed and tested on 2D Poisson problem in [2]).

2. Notation

Consider a system of linear equations $\mathbf{Ku} = \mathbf{f}$ obtained by discretization of boundary value problem with a self-adjoint operator defined on a domain Ω. Let us decompose the domain Ω into N non-overlapping subdomains Ω_i, $i = 1, \ldots, N$. Unknowns common to at least two subdomains are called *interface unknowns*, and the union of all interface unknowns form the *interface*. The first step is the reduction of the problem to the interface. We thus arrive at the *Schur complement problem* for the interface unknowns $\widehat{\mathbf{S}}\,\widehat{\mathbf{u}} = \widehat{\mathbf{g}}$, where $\widehat{\mathbf{S}}$ is a symmetric positive definite (SPD) matrix. This system is solved by the preconditioned conjugate gradient method (PCG). More detailed description of this reduction to the interface can be found in [2].

According to [6], let us interpret the matrix $\widehat{\mathbf{S}}$ as an operator $\widehat{S} : \widehat{W} \to \widehat{W}'$, where $\widehat{W}$ is a finite dimensional linear space. Let us introduce another space $\widetilde{W}$ such that $\widehat{W} \subset \widetilde{W}$ (in terms of subdomains, the space $\widetilde{W}$ represents functions which can be discontinuous across parts of the interface, it means they can have different values of interface unknowns on individual subdomains – there can be a jump across the interface). Let $\widetilde{S}$ be an extension of $\widehat{S}$ to $\widetilde{W}$. Denote R the natural injection from $\widehat{W}$ to $\widetilde{W}$, then we have $\widehat{S} = R^T \widetilde{S} R$. The space $\widetilde{W}$ has to be chosen so that the extended operator $\widetilde{S}$ is positive definite and its inversion can be applied efficiently. Then the BDDC preconditioner can be expressed as

$$M = E\widetilde{S}^{-1}E^T, \tag{1}$$

where operator E^T represents splitting of the residual to subdomains, $\widetilde{S}^{-1}$ stands for solution on subdomains and the coarse level, and E represents averaging of subdomain solutions back to the global problem. The condition number κ of the preconditioned operator $M\widehat{S}$ is bounded by

$$\kappa \leq ||RE||^2_{\widetilde{S}}\,, \tag{2}$$

where the energetic norm on the right-hand side is defined by the scalar product as $||u||^2_{\widetilde{S}} = \langle \widetilde{S}u, u\rangle$. The relationship (2) was proved in [6] 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.

3. Theoretical background of the averaging methods

In every step of PCG, for the given residual $r \in \widehat{W}'$, an approximation Mr of its preimage $e = \widehat{S}^{-1}r$ is to be computed. Our goal is to construct the averaging operator E so that good convergence of the PCG method is achieved, and its action is not too expensive. There is an upper bound on the distance of e from Mr:

Lemma 1. *Assume $ER = I$ and use the notation from Section 2. Denote $w = \widetilde{S}^{-1}E^{\mathrm{T}}r$. Then the following estimation holds:*

$$||e - Mr\ ||_{\widehat{S}} \leq ||R\,e - w\,||_{\widetilde{S}} + ||(I - RE)w\,||_{\widetilde{S}}\ , \tag{3}$$

where $||.||_{\widehat{S}}$ and $||.||_{\widetilde{S}}$ are the energetic norms in $\widehat{W}$ and $\widetilde{W}$, respectively.
Square of the first term on the right-hand side can be expressed as

$$||R\,e - w\ ||^2_{\widetilde{S}} = \langle Mr, r\rangle - ||e||^2_{\widehat{S}}\ . \tag{4}$$

Proof. The inequality (3) is obtained from triangle inequality and the fact, that the energetic norms in $\widehat{W}$ and $\widetilde{W}$ are equal:

$$||e - Mr\ ||_{\widehat{S}} = ||e - Ew\ ||_{\widehat{S}} = ||R\,e - REw\ ||_{\widetilde{S}} \leq ||R\,e - w\ ||_{\widetilde{S}} + ||w - REw\ ||_{\widetilde{S}}\ .$$

Rewriting square of the first term on the right-hand side of the inequality (3):

$$\begin{aligned}
||Re - w||^2_{\widetilde{S}} &= \langle Re - w, \widetilde{S}(Re - w)\rangle = \langle Re - \widetilde{S}^{-1}E^{\mathrm{T}}r, \widetilde{S}Re - \widetilde{S}\widetilde{S}^{-1}E^{\mathrm{T}}r\rangle \\
&= \langle Re, \widetilde{S}Re\rangle - \langle Re, E^{\mathrm{T}}r\rangle - \langle \widetilde{S}^{-1}E^{\mathrm{T}}r, \widetilde{S}Re\rangle + \langle \widetilde{S}^{-1}E^{\mathrm{T}}r, E^{\mathrm{T}}r\rangle \\
&= \langle e, R^{\mathrm{T}}\widetilde{S}Re\rangle - \langle ERe, r\rangle - \langle r, ERe\rangle + \langle E\widetilde{S}^{-1}E^{\mathrm{T}}r, r\rangle \\
&= \langle e, \widehat{S}e\rangle - 2\langle e, r\rangle + \langle Mr, r\rangle = \langle Mr, r\rangle - \langle e, r\rangle = \langle Mr, r\rangle - ||e||^2_{\widehat{S}}\ .
\end{aligned}$$

□

It seems that nearly all methods used so far for averaging have some connection with minimizing the second term $||(I - RE)w\,||_{\widetilde{S}}$ on the right-hand side of (3), as we show bellow. Moreover, there is also a connection with the upper bound (2), because norms of the complementary projections are equal: $||RE||_{\widetilde{S}} = ||I - RE||_{\widetilde{S}}$.

Our approach in [1] to find proper weights for averaging (in other words, to find the so called weight matrix) is to minimize the term $||(I - RE)u\,||_{\widetilde{S}}$ for some given $u \in \widetilde{W}$ under the standard assumption that the global value of any given interface unknown is computed as weighted average of subdomain values of the corresponding unknown only – it means that the weight matrix is supposed to be diagonal. For two

adjacent subdomains (with no coarse space) it leads to a system of linear equations for unknown diagonal of the weight matrix $\mathbf{A}$:

$$\mathbf{Ad} = \widehat{\mathbf{S}}^{-1}\mathbf{S_1 d} = (\mathbf{S_1} + \mathbf{S_2})^{-1}\mathbf{S_1 d}, \tag{5}$$

where $\mathbf{S_i}$ is the local Schur complement of the i-th subdomain, and the vector $\mathbf{d}$ represents a jump across the interface in some given test vector $\mathbf{u}$. For more details see [1]. This relationship can be interpreted as: for a given jump $\mathbf{d}$, find a diagonal representation $\mathbf{A}$ (dependent on $\mathbf{d}$) of some general matrix $(\mathbf{S_1} + \mathbf{S_2})^{-1}\mathbf{S_1}$ that is independent of $\mathbf{d}$. And so maybe we can use this general matrix for a construction of the averaging operator E independent of $\mathbf{d}$ and use the (full) matrix $(\mathbf{S_1} + \mathbf{S_2})^{-1}\mathbf{S_1}$ instead of the diagonal matrix $\mathbf{A}$. By this approach we arrive to *deluxe* scaling proposed in [4]. In this method, the system $(\mathbf{S_1} + \mathbf{S_2})\mathbf{v} = \mathbf{S_1 r}$ is solved for every pair (or, in some cases, group) of adjacent subdomains in every step of the PCG method, where $\mathbf{r}$ is a local residual on the appropriate part of the interface.

As we would like to avoid solving that large number of local systems, we look for some simplification of system (5). One option is to omit all off-diagonal entries of matrices $\mathbf{S_i}$, which leads to the choice of weights as ratios of diagonals of local and global Schur complements. Another option is to assume that all weights on the local interface are equal (as in the case of arithmetic average) and choose some suitable test jump $\mathbf{d}$. We have chosen a unit jump for numerical tests and call the method *averaged unit jump* method.

Note: The expression (4) has led us to the idea of trying to minimize the term $\langle Mr, r\rangle$. For the case of two adjacent subdomains, using the same assumption of diagonal weight matrix and using the same process of minimization as in [1], we arrive at equations

$$\mathbf{Ar} = (\mathbf{S}_1^{-1} + \mathbf{S}_2^{-1})^{-1}\mathbf{S}_2^{-1}\mathbf{r}, \tag{6}$$

where the vector $\mathbf{r}$ represents a local residual on the appropriate part of the interface. Nevertheless, this result does not seem to bring any practical advantage compared to system (5). Again, omitting all off-diagonal entries of matrices $\mathbf{S_i}$ leads to the well-known choice of weights as ratios of diagonals of local and global Schur complements.

4. Approaches for computation of weights at interface nodes

For the sake of clarity, formulas are presented again for 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 generalise these methods to more than two adjacent subdomains (on edges) or more degrees of freedom at a node. Notation:

j ... number of a node in numbering with regard to the interface

i ... global number of the j-th interface node with regard to the global numbering

w_j^1 ... weight at the j-th node at the interface corresponding to the first subdomain (the weight w_j^2 for the second subdomain is then $w_j^2 = 1 - w_j^1$)

• *aa* ... arithmetic average: $w_j^1 = \frac{1}{2}$

• *dk* ... fractions of diagonal entries of the system matrix $\mathbf{K}$: $w_j^1 = \frac{k_{qq}^1}{k_{ii}}$, where k_{ii} is a diagonal entry of the (global) system matrix $\mathbf{K}$, k_{qq}^1 is the corresponding diagonal entry of the local matrix for the first subdomain; q is a local number (at the first subdomain) of the i-th node (in global numbering)

• *rho* ... element-wise ρ-scaling: $w_j^1 = \frac{\alpha_j^1}{\alpha_j^1+\alpha_j^2}$, where α_j^k is a local material coefficient computed as an arithmetic average of material coefficients given on elements containing the j-th node and belonging to k-th subdomain

• *auj* ... averaged unit jump method: $w_j^1 = \frac{\mathbf{d}^T\mathbf{S}_1\mathbf{d}}{\mathbf{d}^T(\mathbf{S}_1+\mathbf{S}_2)\mathbf{d}}$, where $\mathbf{d}$ stands for test vector equal to ones at the common face of the two subdomains and zeros otherwise (representing jump at that face), and $\mathbf{S}_k$ is the local Schur complement for the k-th subdomain

• *uj* ... unit jump method: $w_j^1 = \frac{g_j^2}{g_j^1+g_j^2}$, where $\mathbf{g}^k = (g_1^k, g_2^k, \dots)^T$ is the local vector of reaction on the k-th subdomain induced by a unit jump: $\mathbf{g}^k = \mathbf{S}_k\,\mathbf{d}$

• *ul* ... unit load method: $w_j^1 = \frac{v_j^1}{v_j^1+v_j^2}$, where $\mathbf{v}^k = (v_1^k, v_2^k, \dots)^T$ is the vector of the local solution on the k-th subdomain under unit load, i.e. with the right-hand side equal to ones: $\mathbf{S}_k\,\mathbf{v}^k = \mathbf{d}$

5. Numerical results

Our aim is to numerically compare the robustness of the approaches to averaging listed in Section 4 with respect to two model aspects known to cause issues to domain decomposition methods, namely (i) roughness of the interface among subdomains, (ii) jumps in material coefficients inside the domain.

Following our preliminary 2D numerical tests of some of the methods for averaging (see [2]), we choose 3D **Poisson** and **linear elasticity** problems on a unit cube domain for testing (solutions of the problems are illustrated in Figure 3). The Poisson equation has unit right-hand side and homogeneous Dirichlet boundary conditions on the surface of the cube. For linear elasticity problem, the cube is mounted at a vertical face and loaded by its own weight.

Problems are discretized by the finite element method using trilinear cubic elements, all of them of the same size h. The domain was divided into 4×4×4 cubic subdomains of size H, and we test different numbers of elements per subdomain edge, H/h = 4, 8, 16, 32, and 60. The interface is either **regular**, i.e. consisting of plane sections only, or **jagged** (see Figure 1). Both homogeneous and nonhomogeneous materials are considered. For the Poisson problem, the low material constant is chosen as 1 and the high one as 10^6, for elasticity the low value of Young modulus is 10^5 and the high one is $2.1{\cdot}10^{11}$. Three nonhomogeneous material arrangements are designed (see Figure 2):

- Material 1 – *Random elements*: For each element, the value of the material

coefficient is chosen randomly with a uniform distribution between the low and high values.

- Material 2 – *Slices* along the interface: Only the low and high values of the material coefficient are used as depicted in Figure 2. The solution to the Poisson problem defined on this domain is in Figure 3.

- Material 3 – *Stiff rods* of material with the high coefficient arranged in a 4×4 lattice inside the material with the low coefficient.

These arrangements have been chosen to model several situations encountered in engineering, such as rapidly oscillating coefficients, layered structures, or reinforced composite structures.

Coarse nodes at all crosspoints and coarse averages are used. Quality of the preconditioner is measured by the number of iterations of PCG needed to reduce the relative residual below 10^{-6}.

For computations, the *BDDCML* library – a massively parallel implementation of the Adaptive-Multilevel BDDC method – is used. The Schur complements are not computed explicitly in this implementation, so the averaging by values on diagonals of the complements is only approximated by diagonals of the subdomain matrices.

For **homogeneous material** and **regular** interface, the same results have been obtained by all methods of averaging, only the *ul* method has performed little worse. For Poisson problems the number of iterations are depicted on Figure 4 (left), for linear elasticity the results are very similar and they are not reported.

For **jagged** interface, the results are summarised in Table 1, and for Poisson problems, they are also plotted in Figure 4 (right). The behaviour of the methods is again very similar for both Poisson and linear elasticity problems, the main difference is worse convergence for elasticity. The interesting observation is that the rate of worsening of the convergence with growing ratio of H/h is different for different methods: for instance, the *auj* method is much more stable than *dk* method, and although *auj* is the worst method for 4 elements per subdomain edge, it belongs to the best for the 32 and 60 elements per edge.

	Poisson problem					linear elasticity			
H/h	4	8	16	32	60	4	8	16	32
aa	11	14	15	16	18	28	35	37	39
dk	6	8	11	17	24	13	18	28	44
rho	11	14	15	16	18	28	35	37	39
auj	12	14	15	16	18	32	40	41	42
uj	7	9	14	22	32	19	30	41	68
ul	10	13	17	22	27	21	37	51	61

Table 1: Number of iterations: homogeneous material, jagged interface

The results for **nonhomogeneous materials** are illustrated by graphs only. For Material 1 (random elements), the behavior is again very similar for both Poisson and linear elasticity problems, so only results for the Poisson problem are depicted in Figure 5 for regular (left) and jagged (right) interface. Note the different scale of the vertical axes. We can see that the jagged interface worsens dramatically the behaviour of both aa and auj methods, which do not adapt locally to the jumps along the interface and use a single weight for the whole part of the interface.

In the case of Material 2 (slices), for regular interface, all methods perform equally well with the exception of aa. For the Poisson problem, see Figure 6 (left). For jagged interface (Figure 8), methods aa and uj did not converge in 1000 and for others, there is a difference between Poisson problems (left) and linear elasticity (right): for the former, the dk method worsens quite rapidly with growing H/h, for the latter dk it is the best even for $H/h = 32$.

Material 3 (stiff rods) leads to quite challenging problems: the results are diverse and difficult to predict. Methods behave differently for Poisson problems and linear elasticity even for regular interface (see Figure 9). For jagged interface, convergence was achieved only for Poisson problems (Figure 7 right).

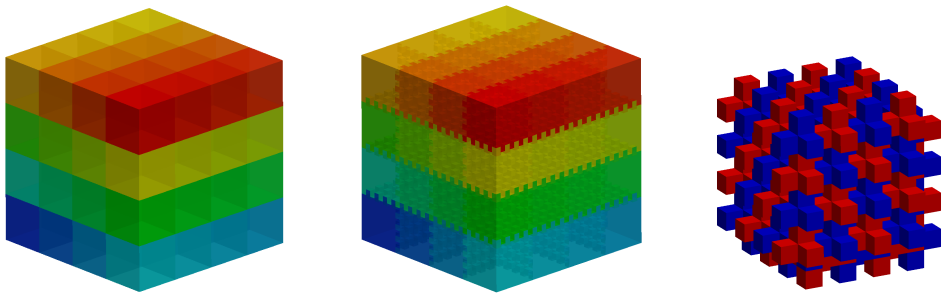


Figure 1: Regular (left) and jagged (centre) interface, and a detail of an interior jagged subdomain (right)

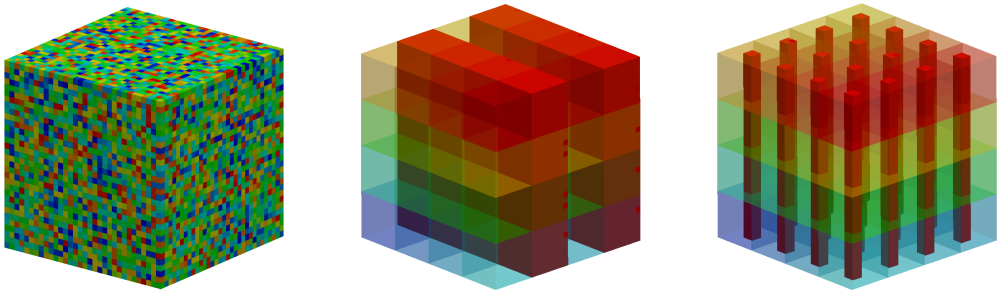


Figure 2: Material 1 – random elements (left), Material 2 – slices (centre), and Material 3 – stiff rods (right)

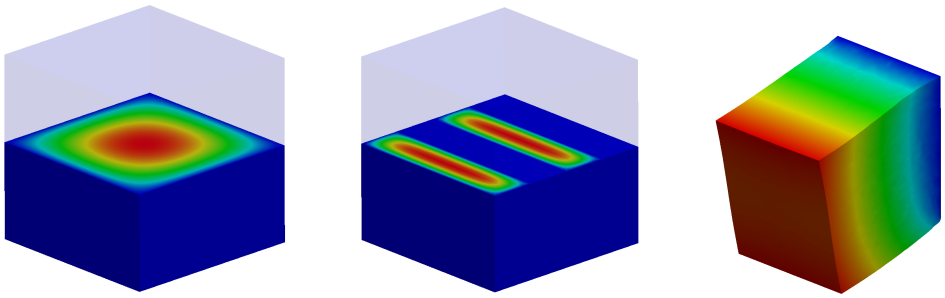


Figure 3: Solution to the Poisson problem with homogeneous material (left), for the Poisson problem with Material 2 – slices (centre), and magnified displacement of the linear elasticity problem with homogeneous material (right)

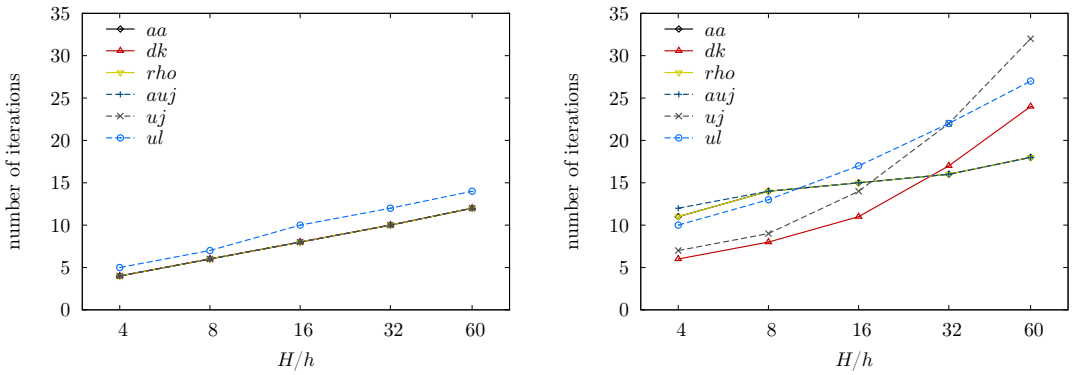


Figure 4: Homogeneous material, Poisson problem, regular (left) and jagged (right) interface

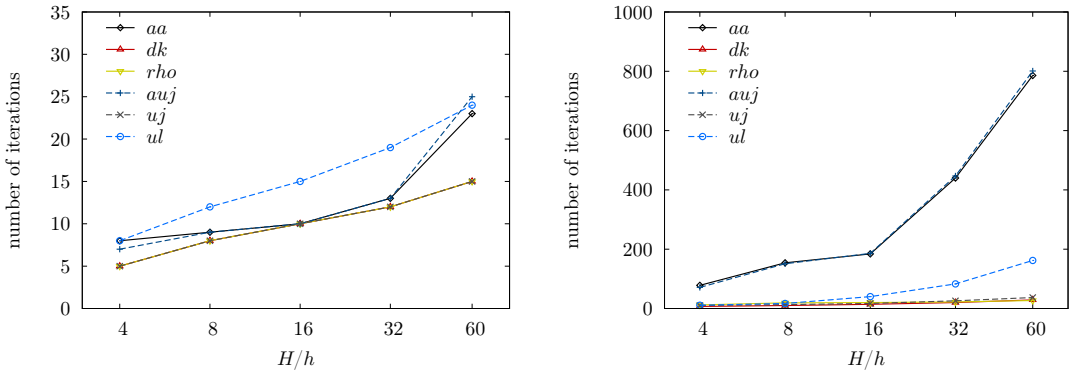


Figure 5: Material 1 (random elements), Poisson problem, regular (left) and jagged (right) interface. Note the different scale on the vertical axes

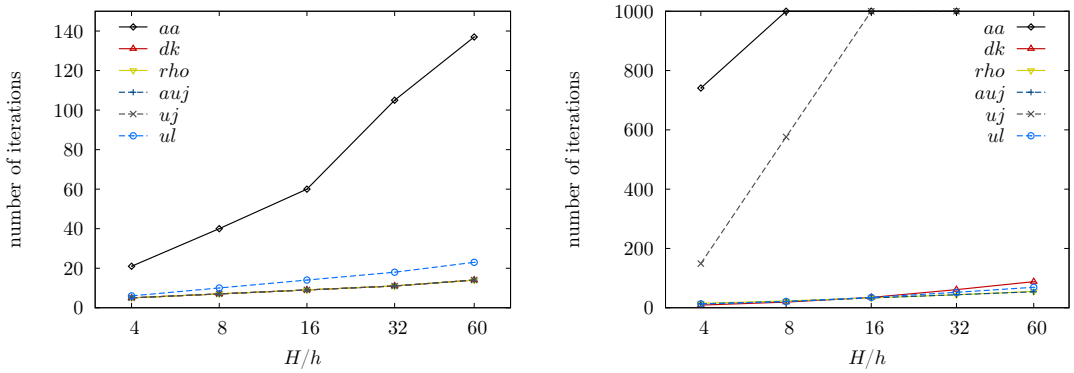


Figure 6: Material 2 (slices), Poisson problem, regular (left) and jagged (right) interface

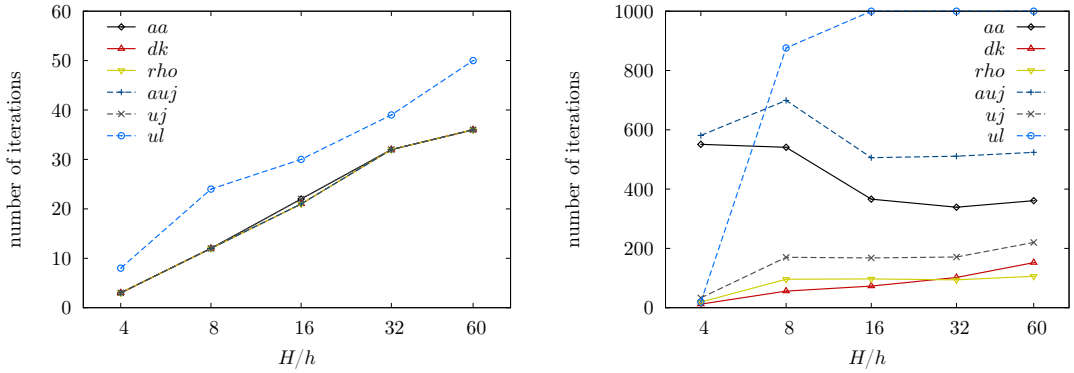


Figure 7: Material 3 (stiff rods), Poisson problem, regular (left) and jagged (right) interface. Note the different scale on the vertical axes.

6. Conclusions

Three new forms of the averaging operator (auj, uj, ul) have been numerically compared with three standard ones (aa, dk, rho) on several challenging test problems. We have found that the choice of the method of averaging has a significant influence not only on the convergence of the BDDC method, but also on the rate of worsening of the convergence with growing ratio of H/h. The main conclusion one can draw from our numerical results is that there is no single universal method for averaging that would perform well for all cases; the performance of the methods depends on the problem, on the H/h ratio as well as on the profile of the interface (regular or jagged). Moreover, it is usually not clear in advance, which method would be the best one for the given problem. It seems that a robust and efficient implementation of the BDDC method should offer a flexible choice from several different averaging methods, and it is worth trying several of them before a production computations are performed.

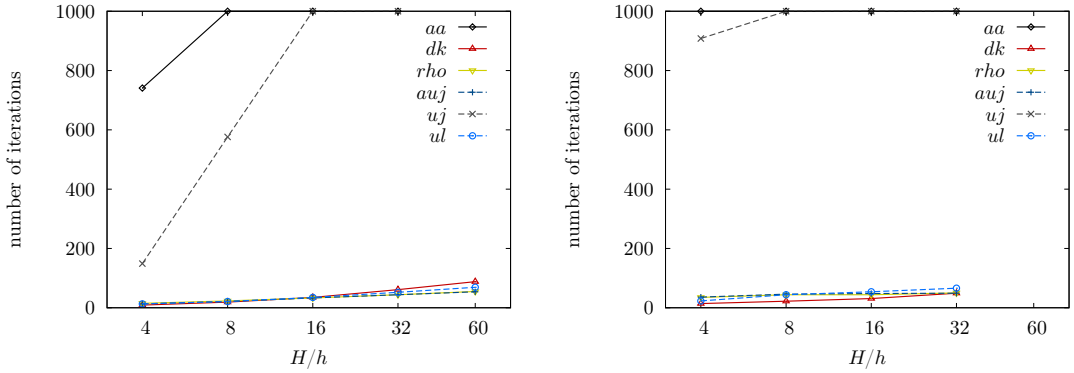


Figure 8: Material 2 (slices), jagged interface, Poisson (left) and linear elasticity (right) problems. Note the different scale on the vertical axes.

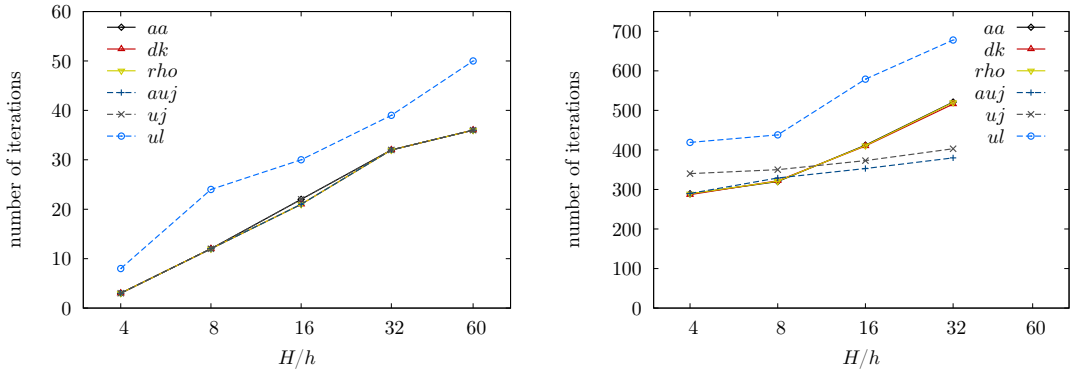


Figure 9: Material 3 (stiff rods), regular interface, Poisson (left) and linear elasticity (right) problems. Note the different scale on the vertical axes.

Nevertheless, some recommendations based on our results can still be made: For homogeneous problems, the simplest method *aa* is sufficient for both regular and irregular interface. It seems also less sensitive to growing H/h ratio than other methods. However *aa* should not be applied to nonhomogeneous materials for which the convergence can be disastrous. For nonhomogeneous problems, *dk* or, if the solver has access to material data, the *rho* scaling perform well. Moreover, *rho* seems more reliable, as convergence for *dk* deteriorates more rapidly with growing H/h ratio, especially for jagged interface. For several complicated cases combining jumps and irregular interface, the newly developed methods, *auj*, *uj*, and *ul* noticeably superseded the standard approaches, especially for linear elasticity problems.

In Lemma 1, some relationships between preconditioned residual and its preimage in both the spaces $\widehat{W}$ and $\widetilde{W}$ for the BDDC preconditioner have been presented. However, they have not led us to any new practical method for averaging so far.

Acknowledgements

This work was supported by the Czech Ministry of Education, Youth and Sports of the Czech Republic under research project LH11004, by the Czech Science Foundation under project 14-02067S, and by the Academy of Sciences of the Czech Republic through RVO: 67985840. The work was also supported by the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), funded by the European Regional Development Fund and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, as well as Czech Ministry of Education, Youth and Sports via the project Large Research, Development and Innovations Infrastructures (LM2011033). We are grateful to Jan Mandel for fruitful discussions on the topic.

References

[1] Čertíková, M., Burda, P., Novotný, J., and Šístek, J.: Some remarks on averaging in the BDDC method. In: T. Vejchodský et al. (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 15, Dolní Maxov, Czech Republic, June 6–11, 2010*, pp. 28–34. Institute of Mathematics AS CR, 2010.

[2] Čertíková, M., Šístek, J., and Burda, P.: On selection of interface weights in domain decomposition methods. In: J. Chleboun et al. (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 16, Dolní Maxov, Czech Republic, June 3–8, 2012*, pp. 35–44. Institute of Mathematics AS CR, 2013.

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

[4] Dohrmann, C. R. and Widlund, O. B.: Some recent tools and a BDDC algorithm for 3D problems in H(curl). In: R. Bank et al. (Eds.), *Domain Decomposition Methods in Science and Engineering XX*, *Lecture Notes in Computational Science and Engineering*, vol. 91, pp. 15–25. Springer, 2013.

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

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

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

IDENTIFICATION OF PARAMETERS IN INITIAL VALUE PROBLEMS FOR ORDINARY DIFFERENTIAL EQUATIONS

Jan Chleboun, Karel Mikeš

Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
chleboun@mat.fsv.cvut.cz, karel.mikes.1@fsv.cvut.cz

Abstract

Scalar parameter values as well as initial condition values are to be identified in initial value problems for ordinary differential equations (ODE). To achieve this goal, computer algebra tools are combined with numerical tools in the MATLAB® environment. The best fit is obtained through the minimization of the summed squares of the difference between measured data and ODE solution. The minimization is based on a gradient algorithm where the gradient of the summed squares is calculated either numerically or via auxiliary initial value problems. In the latter case, the MATLAB® Symbolic Math Toolbox™ is used to derive the expressions that define the auxiliary problems and to transform them into MATLAB® routines.

1. Introduction

This work was initiated by [3], where parameter identification is performed by an artificial neural network algorithm. A question arose, whether a more traditional method could be effective in solving the identification problem. By a more traditional method, we mean the minimization of a relevant cost function by a gradient-based minimization algorithm.

Parameter identification is a common task in chemistry, biology, and engineering. If the underlying problem is not ill-posed, parameters can be identified by a straightforward method, see, for instance, [5], a short report providing the reader with an easy introduction to the subject, or a more advanced applications [1, 6]. Let us emphasize that we do not consider data polluted by noise, though it is a common difficulty in practice, see [4].

2. Identification problems

Cement hydration. The cement hydration process is modeled by the following initial value problem (IVP) presented in [3]

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t}(t) = B_1 \left(\frac{B_2}{\alpha_\infty} + \alpha(t) \right) (\alpha_\infty - \alpha(t)) \exp\left(\frac{\eta}{\alpha_\infty} \alpha(t) \right) C, \tag{1}$$

$$\alpha(0) = 0, \tag{2}$$

where α is the time dependent degree of hydration and α_∞ stands for its limit value, B_1 and B_2 are coefficients dependent on the cement chemical composition, η represents the microdiffusion of free water, and $C \approx 2 \times 10^{-7}$ is a known constant, see [3].

It is assumed that

$$(\alpha_\infty, B_1, B_2, \eta) \in I_\alpha = [0.7, 1.0] \times [10^6, 10^7] \times [10^{-6}, 10^{-3}] \times [-12, -2]. \tag{3}$$

Generalized Van der Pol oscillator. Let us consider the following nonlinear IVP

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = (c_1 - c_2 y^2) \frac{\mathrm{d}y}{\mathrm{d}t} - c_3 y, \tag{4}$$

$$y(0) = c_4, \quad \frac{\mathrm{d}y}{\mathrm{d}t}(0) = c_5, \tag{5}$$

where c_1, c_2, and c_3 are positive parameters, and c_4, c_5 are real parameters. If $c_1 = c_2 = c_3 = 1$, then we get the Van der Pol oscillator. It is assumed

$$(c_1, c_2, c_3, c_4, c_5) \in I_C = [0.5, 3]^3 \times [1, 3] \times [-1, 1]. \tag{6}$$

In both IVPs, the values of parameters are to be identified through m_i, that is, the measurements of either the hydration at time points $t_i \in [0, T_\alpha]$, $i = 1, 2, \ldots, n_\alpha$, or the measurements of the position $y(t_i)$ at $t_i \in [0, T_C]$, $i = 1, 2, \ldots, n_C$.

The identification problem: Find $\widehat{p} \in I$ such that

$$\widehat{p} = \arg\min_{p \in I} \Psi(p), \tag{7}$$

where

$$\Psi(p) = \sum_{i=1}^{n} w_i (m_i - u(t_i))^2 \tag{8}$$

and either $I \equiv I_\alpha$, $n \equiv n_\alpha$, and $u \equiv \alpha$ solves (1)–(2), or $I \equiv I_C$, $n \equiv n_C$, and $u \equiv y$ solves (4)–(5). The positive weighting factors w_i are also problem dependent and enable to increase or decrease the importance of some measurements.

3. Sensitivity analysis

To employ a gradient method for the minimization of Ψ, the gradient of Ψ with respect to the components of $p \in I$ is necessary. The partial derivatives of Ψ can be approximated by the numerical differentiation of Ψ, or by solving auxiliary problems. In the latter case, since

$$\frac{\partial \Psi}{\partial p_j} = 2 \sum_{i=1}^{n} w_i \left(m_i - u(t_i)\right) u'_{p_j}(t_i), \tag{9}$$

where p_j is a component of p and u'_{p_j} stands for the derivative of the state solution with respect to p_j, we have to find functions u'_{p_j} as the solutions of auxiliary IVPs.

According to [2], the derivatives of the state solution α with respect to α_∞, B_1, B_2, and η exist and they are solutions of

$$\frac{\mathrm{d}v}{\mathrm{d}t}(t) = g(t)v(t) + q(t), \tag{10}$$

$$v(0) = 0, \tag{11}$$

where the function g originates from the right-hand side of the state equation (1) differentiated w.r.t. the symbol α, whereas the derivative w.r.t. a parameter from the set $\{\alpha_\infty, B_1, B_2, \eta\}$ results in the function q.

To arrive at an IVP analogous to (10)–(11), we rewrite (4)–(5) into a system of first order equations

$$\frac{\mathrm{d}y_1}{\mathrm{d}t} = y_2, \tag{12}$$

$$\frac{\mathrm{d}y_2}{\mathrm{d}t} = (c_1 - c_2 y_1^2) y_2 - c_3 y_1, \tag{13}$$

$$y_1(0) = c_4, \quad y_2(0) = c_5. \tag{14}$$

By differentiating (12)–(14) w.r.t. y_1, y_2 and the parameters, we obtain the following parallel to (10)–(11)

$$\begin{pmatrix} \mathrm{d}v_1/\mathrm{d}t \\ \mathrm{d}v_2/\mathrm{d}t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2c_2 y_1 y_2 - c_3 & c_1 - c_2 y_1^2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \omega \end{pmatrix}, \tag{15}$$

$$v_1(0) = \theta_1, \quad v_2(0) = \theta_2, \tag{16}$$

where $\theta_1 = 0 = \theta_2$ and $\omega = y_2$ if the derivative of the state solution $y \equiv y_1$ w.r.t. c_1 is to be calculated, $\omega = -y_1^2 y_2$ and $\omega = -y_1$ if we differentiate w.r.t. c_2 and c_3, respectively. If the derivative of y with respect to the initial conditions is sought, then $\omega = 0$ in (15) and $\theta_1 = 1$, $\theta_2 = 0$ in (16) if we differentiate w.r.t. c_4, or $\theta_1 = 0$, $\theta_2 = 1$ if we are interested in the sensitivity to c_5. Details in [2, Chapter 13 and 14].

To summarize, let us recall that for each parameter $\alpha_\infty, B_1, B_2, \eta$ (or $c_1, \ldots, c_5$), we infer and solve (10)–(11) (or (15)–(16)). After substituting v (or v_1) for u'_{p_j} in (9), we obtain one component of the gradient of Ψ.

The derivation of the expressions appearing on the right-hand side of (10) is easy for the generalized Van der Pol equation, see (15), but more laborious for the hydration problem. Nevertheless, it is effortlessly performed by the MATLAB® Symbolic Math Toolbox™ (we used its R2012b version), namely by its functions `diff` and `matlabFunction`. The latter converts symbolic expressions to MATLAB® functions.

4. Minimization

To minimize (8), the MATLAB® R2012b Optimization Toolbox™ `fmincon` function was used. It is designed for constrained minimization, see (3) and (6). The cost function gradient can be calculated by a black-box numerical differentiation, or by a code delivered by the user. We tried both approaches, and applied the sensitivity analysis approach explained in Section 3 in the latter.

Since parameter identification is a global minimization problem and `fmincon` is a tool for local minimization, optimization runs starting from different initial points belonging to I_α or I_c, see (3) and (6), were necessary to increase the chance of finding a global minimum.

5. Results, observations, and conclusions

In both problems, the weights w_i were chosen as equal.

Cement hydration. Figure 1 shows the graphs of the derivatives of the state solution α determined by $(0.7, 5 \times 10^6, 5 \times 10^{-4}, -2.5) \in I_\alpha$ with respect to the parameters. We observe a high sensitivity to B_2 and a low sensitivity to B_1. Moreover, the peak sensitivity occurs in a neighborhood of $t = 25$ and is, except for the case of α_∞, strongly localized. We deduce that the most important measurements are those made in between, say, $t = 5$ and $t = 50$ or $t = 100$. The state solution rapidly increases in $[0, 50]$, see Figure 2 (left), where the best fit to a set of 23719 real-world measurements of the cement hydration process (1)–(2) is depicted (time, t, in hours).

Generalized Van der Pol oscillator. Examples of the derivatives of $y \equiv y_1$, see (4) and (12), at $c_1 = c_2 = c_3 = 1$, $c_4 = 2$, and $c_5 = 0$ are depicted in Figure 3. The state solution y as well as some of its derivatives are periodic for a range of parameters, but the amplitude of the other derivatives is increasing, which might decrease the accuracy of the approximate expansion of y (w.r.t. the parameters) at times far from the initial time. Figure 2 (right) shows the initial solution y for the above values $c_1, \ldots, c_5$, also seven points obtained via "measurements" derived from the state solution determined by parameters that are to be re-identified, and the state solution determined by the identified parameters.

Let us present a few observations and conclusions. The coupling of symbolic and numerical computation substantially reduces the amount of problem-dependent user-written code. Although the derivatives of the state solution w.r.t. the parameters reveal the sensitivity of the state solution to the perturbation of the parameters and are beneficial in the evaluation of (9) and, if possible, in the placement of the times of

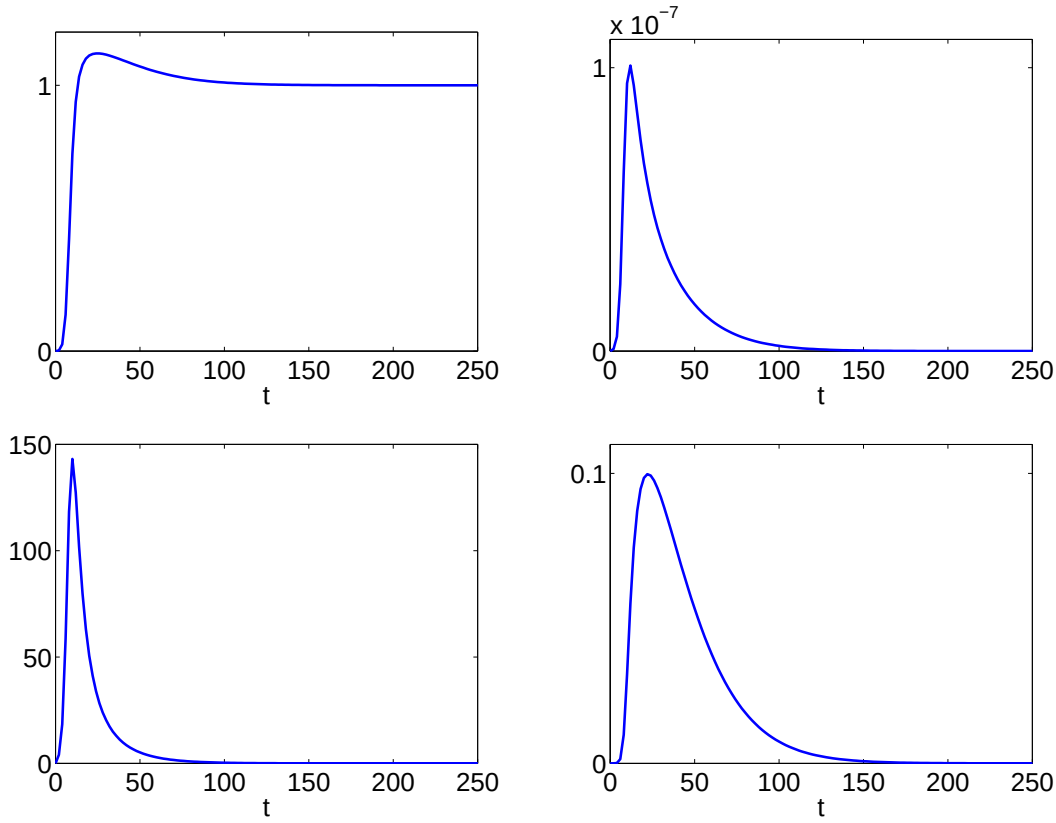


Figure 1: The derivative of α w.r.t. α_∞ (top left), B_1 (top right), B_2 (bottom left), η (bottom right).

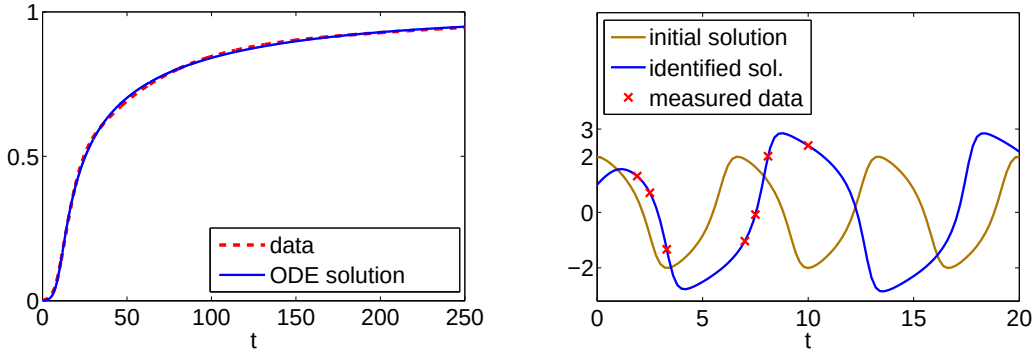


Figure 2: Identified solution to (1)–(2) (left), and to (4)–(5) (right).

measurements, their calculation slows the minimization process. Indeed, numerical differentiation turned out to be quite fast and accurate and might be considered the method of first choice in `fmincon` if the parameter identification is the only goal of calculation. In any case, however, the adjoint equation technique is worth considering to speed up the minimization process.

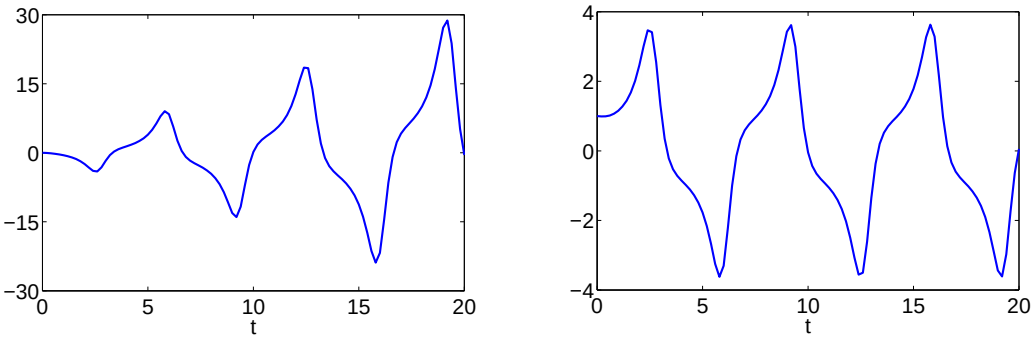


Figure 3: The derivative of y w.r.t. c_3 (left) and c_4 (right).

Acknowledgements

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/003/OHK1/1T/11. The authors also wish to thank Karel Hájek and Ondřej Petlík for their assistance in the project and an anonymous referee for valuable comments.

References

[1] Babadzanjanz, L. K., Boyle, J. A., Sarkissian, D. R., Zhu, J.: Parameter identification for oscillating chemical reactions modelled by systems of ordinary differential equations. J. Comp. Methods Sci. Eng. **3** (2003), 223–232.

[2] Kurzweil, J.: *Ordinary differential equations: introduction to the theory of ordinary differential equations in the real domain*. Studies in Applied Mechanics, vol. 13, Elsevier Science Ltd, Amsterdam, 1986.

[3] Mareš, T.: Artificial neural networks in calibration of nonlinear models. Report, Dept. of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, 2012.

[4] Müller, T. G., Noykova, N., Gyllenberg, M., Timmer, J.: Parameter identification in dynamical models of anaerobic waste water treatment. Math. Biosci. **177/178** (2002), 147–160.

[5] Munster, D.: Parameter identification: a comparison of methods. Report, Dept. of Mathematics, College of Science, Virginia Tech, 2009.

[6] Wang, J., Ye, J., Yin, H., Feng, E., Wang, L.: Sensitivity analysis and identification of kinetic parameters in batch fermentation of glycerol. J. Comput. Appl. Math. **236** (2012), 2268–2276.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

COMPARISON OF ALGORITHMS FOR CALCULATION OF THE GREATEST COMMON DIVISOR OF SEVERAL POLYNOMIALS

Jiří Eckstein, Jan Zítko

Department of Numerical Mathematics,
Faculty of Mathematics and Physics, Charles University
Sokolovská 83, Prague 8, Czech Republic
jiri.eckstein@gmail.com, jan_zitko@centrum.cz

Abstract

The computation of the greatest common divisor (GCD) has many applications in several disciplines including computer graphics, image deblurring problem or computing multiple roots of inexact polynomials. In this paper, Sylvester and Bézout matrices are considered for this purpose. The computation is divided into three stages. A rank revealing method is shortly mentioned in the first one and then the algorithms for calculation of an approximation of GCD are formulated. In the final stage the coefficients are improved using Gauss-Newton method. Numerical results show the efficiency of proposed last two stages.

1. Introduction

Sylvester matrices (see [1, 3, 5, 6, 10, 11, 14, 15, 16, 17]) or Bézout matrices (see [3, 7, 8, 12]) can be used for the calculation of GCD. We start with Sylvester matrix. The coefficients of GCD of two polynomials f_1 and f_2 can be obtained from a Sylvester subresultant $S_k(f_1, f_2)$ which is formed from the Sylvester matrix $S(f_1, f_2)$ by deleting the last $k-1$ rows, the last $k-1$ columns of the coefficients of f_1 and the last $k-1$ columns of the coefficients of f_2. If $n_i = \deg(f_i)$ for $i = 1, 2$, $n_1 \geq n_2$, and if for a positive integer $d \leq n_2$ the subresultant $S_d(f_1, f_2)$ is the first rank deficient matrix in the sequence

$$S_{n_2}(f_1, f_2), S_{n_2-1}(f_1, f_2), \ldots, S_1(f_1, f_2), \tag{1}$$

then $d = \deg(\mathrm{GCD}(f_1, f_2))$. There are two well-known procedures for calculation of the rank (or rank deficiency) of a matrix. For small dimension the usage of SVD is sufficient (see for example [2] or [9]). The numerical rank revealing algorithm with many robust examples is in detail described in the papers [11, 17]. The whole process is in details, together with the calculation of GCD and rank determination, explained

in the papers [17], [11], [14] and therefore, in the following, we are considering the calculation for m polynomials.

We now consider m real polynomials $f_1,\ f_2,\ldots f_m$. Let n_i =deg(f_i). Denote g =GCD$(f_1, f_2, \ldots, f_m)$. It is assumed that d=deg$(g) > 0$. The objective is to find polynomials $w_1,\ w_2,\ \ldots,\ w_m$ of degrees $n_1 - d,\ n_2 - d, \ldots,\ n_m - d$ respectively, such that $f_i = w_i\, g$ for all considered i, which can be expressed in the form (see [4, 16, 17])

$$C_d(w_i)\vec{g} = \vec{f_i}, \qquad \text{for} \quad i \in \{1, 2, \ldots, m\}, \tag{2}$$

where $C_d(w_i)$ is the Cauchy matrix for the polynomial w_i with $d+1$ columns, i.e., $C_d(w_i) \in \mathbb{R}^{(n_i+1)\times(d+1)}$. The symbol $\vec{g}$ denotes the vector of coefficients of g and the symbols $\vec{f_i}$ and $\vec{w_i}$ have an analogous meaning. The system (2) can be rewritten in the form

$$F(\mathbf{x}) = \mathbf{b}, \text{ where } \quad \mathbf{x} = \begin{bmatrix} \vec{g} \\ \vec{w_1} \\ \vec{w_2} \\ \cdot \\ \cdot \\ \cdot \\ \vec{w_m} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ \vec{f_1} \\ \vec{f_2} \\ \cdot \\ \cdot \\ \cdot \\ \vec{f_m} \end{bmatrix}, \quad F(\mathbf{x}) = \begin{bmatrix} (\vec{r})^T\vec{g} \\ C_d(w_1)\vec{g} \\ C_d(w_2)\vec{g} \\ \cdot \\ \cdot \\ \cdot \\ C_d(w_m)\vec{g} \end{bmatrix} \tag{3}$$

and $\vec{r}$ is a scaling vector (see [16]). Let us remark that $\vec{r}, \vec{g} \in \mathbb{R}^{d+1}$, and $\vec{w_j} \in \mathbb{R}^{n_j-d+1}$. The system (3) represents $\left(\sum_{j=1}^{m} n_j\right)+m+1$ equations with $\left(\sum_{j=1}^{m} n_j\right)+m+1-(m-1)\,d$ unknowns and the least square solution (see [9]) is applied. According to the well known theory (see [2]) we have

$$\text{grad}\left[\frac{1}{2}\,\|F(\mathbf{x}) - \mathbf{b}\,\|^2\right] = (J(\mathbf{x}))^T\,[F(\mathbf{x}) - \mathbf{b}]\,, \tag{4}$$

where $J(\mathbf{x})$ is the Jacobian of F and can be easily calculated as a Gateaux derivative of F. The problem of location of minimum leads to the solution of the system

$$(J(\mathbf{x}))^T[F(\mathbf{x}) - \mathbf{b}] = 0. \tag{5}$$

Let us mention the result formulated in [16]: for every scaling vector $\vec{r}$ satisfying $\vec{r}^T\vec{g} \neq 0$, if GCD$(w_1, w_2, \ldots, w_m) = 1$, then the Jacobi matrix has a full column rank and therefore $F(\mathbf{x}) = \mathbf{b}$. However all these investigations depend on the basic question how to find the rank d. This is well known for $m = 2$ and it is shortly analysed in Section 2. In the next section the algorithm using Sylvester matrices for $m \geq 3$ is discussed. In Section 3, the calculation of the greatest common divisor of several univariate polynomials through Bézout-like matrices is considered. Both strategies are numerically tested in the last section.

2. Calculation of GCD through Sylvester matrices

At the beginning consider the polynomials f_1 and f_2 of degrees n_1 and n_2 respectively, where $n_1 \geq n_2$. According to the previous section we determine an integer d such that $S_d(f_1, f_2)$ is the first rank deficient matrix in (1) and denote the right singular vector of the matrix $S_d(f_1, f_2) = [C_{n_2-d}(f_1), C_{n_1-d}(f_2)]$ corresponding to the smallest singular value $\sigma_{min}\left(S_d(f_1, f_2)\right)$, which is theoretically equal to zero, by $[(\vec{w}_2)^T, -(\vec{w}_1)^T]^T$. We have denoted g =GCD(f_1, f_2). The coefficients of $\vec{g}$ are calculated as the least square solution of the equation

$$C_d(w_2)\vec{g} = \vec{f_2} \qquad \text{or} \qquad C_d(w_1)\vec{g} = \vec{f_1}. \tag{6}$$

One of these equations (usually the first one) is solved and the second one is used for improvement of the result if it is necessary.

However, for three or more polynomials it is impossible to apply an analogous technique for finding the degree of GCD(f_1, f_2, f_3). A consecutive process is usually applied, which can be formally written for three polynomials in the form

$$d = \deg(\mathrm{GCD}(f_3, \mathrm{GCD}(f_1, f_2))).$$

Numerically, the determination of GCD is usually based on some minimisation method which is formally written by (4), (5) and the realization means an infinite iterative process where only finite number of iterations is implemented. Moreover, if the calculation is performed in floating point environment the result is inexact and therefore an approximation is obtained as a result of the above mentioned minimization process. This approximation to GCD will be in this paper entitled *approximate greatest common divisor* - AGCD. This concept is studied and discussed in many papers (see for example [6, 13, 16]). The concept AGCD is mentioned in context with STLN algorithm (see [10, 15, 13, 6]). In this paper AGCD is the result of the least square procedures which is realized by the Gauss-Newton method. Exact coefficients are assumed. Let us consider the system (5). By analogy to [16] and [17] we now present the algorithm for several polynomials. The numerical process will be evident from the following algorithm.

Algorithm 2.1 (*AGCD for m polynomials.*)
Input: Real polynomials f_1, f_2,..., f_m of degrees n_1, n_2,..., n_m respectively, vector **b** defined by (3) and a given tolerance θ. It is assumed that

$$n_1 \geq n_2 \geq \cdots \geq n_m.$$

Output: Polynomial g =AGCD$(f_1, f_2, \ldots, f_m)$
begin
$g := f_{n_m}$
for $j = m, m-1, \ldots, 2$ **do**
Calculate g =AGCD$(g, f_{n_{j-1}})$.

end
for $j = 1, m$ **do**
$w_j(x) := f_j(x)/g(x)$
end
Put $d :=\deg(g)$;
form the vector $(\mathbf{x})^T = [(\vec{g})^T, (\vec{w_1})^T, (\vec{w_1})^T, \ldots, (\vec{w_m})^T]^T$ for the initial approximation of Gauss-Newton iteration.
repeat

$$\mathbf{x}^+ = \mathbf{x} - \begin{bmatrix} \vec{r} & 0 & 0 & 0 \\ C_d(w_1) & C_{n_1-d}(g) & \cdot & \\ \cdot & 0 & \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot & 0 \\ C_d(w_m) & 0 & \cdot\ \cdot & C_{n_m-d}(g) \end{bmatrix}^{\dagger} \begin{bmatrix} \vec{r}^T\vec{g} \\ C_d(w_1)\vec{g} - (\vec{f_1}) \\ \cdot \\ \cdot \\ C_d(w_m)\vec{g} - (\vec{f_m}) \end{bmatrix}$$

$\mathbf{x} = \mathbf{x}^+$
until $\|F(\mathbf{x}) - \mathbf{b}\| < \theta$
Once $\|F(\mathbf{x}) - \mathbf{b}\| < \theta$, we extract coefficients of the polynomial $g(x)$ from the vector $\mathbf{x}$.
We now have $g(x) =$GCD $(f_1, f_2, \ldots, f_m)$.
end of algorithm

The matrix is a block matrix, the non-zero blocks are Cauchy matrices. It contains only zero-blocks except for the first column and the diagonal blocks.

3. Calculation of GCD using Bézout matrices

We now present a different approach to computing the GCD of several real univariate polynomials using Bézoutian matrices (see [3], [8]). The size of this kind of matrix depends purely on the degree of one of the polynomials. It will be possible to determine the degree of GCD of a whole set of polynomials at once. Moreover, its coefficients will be computed at the same time. Let p and q be two polynomials,

$$p(x) = a_0x^k + a_1x^{k-1} + \ldots a_{k-1}x + a_k,$$
$$q(x) = b_0x^k + b_1x^{k-1} + \ldots b_{k-1}x + b_k$$

of degrees at most $k > 0$. If $\deg(p) >\deg(q)$ then some of the first coefficients of q equal zero.
The Bézout matrix associated to p and q (see [12]) is

$$B(p, q) = \begin{bmatrix} c_{1,1} & \cdots & c_{1,k} \\ \vdots & & \vdots \\ c_{k,1} & \cdots & c_{k,k} \end{bmatrix},$$

where the coefficients $c_{i,j}$ are defined by the relation

$$\frac{p(x)q(y) - p(y)q(x)}{x - y} = \sum_{i,j=1}^{k} c_{i,j} x^{i-1} y^{j-1}.$$

In the following, the procedure for computing the AGCD of m polynomials is presented. To achieve this, a set of polynomials $f_1, \ldots, f_m$ satisfying

$$k := n_1 = \deg(f_1) > \deg(f_i),\ i = 2, 3, \ldots, m$$

will be assumed. In contrast with Sylvester matrices, all the Bézout matrices $B(f_1, f_i)$, $i = 2, 3, \ldots, m$ are square and of the same dimension. Therefore the matrix

$$B_{f_1}(f_2, \ldots, f_m) = \begin{bmatrix} B(f_1, f_2) \\ B(f_1, f_3) \\ \vdots \\ B(f_1, f_m) \end{bmatrix}$$

can be constructed. Analogously to computation with Sylvester matrices, the degree of the AGCD equals $k - \operatorname{rank}(B_{f_1}(f_2, \ldots, f_m))$. Its coefficients can be computed by determining the linear combinations of column vectors, as described in the algorithm below (for details see [8]). The numerical realization of GCD will be again called AGCD.

Algorithm 3.1 (*AGCD for m polynomials.*)
Input: Real polynomials f_1, f_2,..., f_m of degrees n_1, n_2,..., n_m respectively. It is assumed that $k := n_1 > \max\{n_2, \ldots, n_m\}$.
Output: Polynomial g =AGCD$(f_1, f_2, \ldots, f_m)$.
begin
Determine the $d = k - \operatorname{rank}(B_{f_1}(f_2, \ldots, f_m))$.
Let $\mathbf{t}_1, \ldots, \mathbf{t}_k$ be column vectors of $B_{f_1}(f_2, \ldots, f_m) = [\mathbf{t}_1, \ldots, \mathbf{t}_k]$.
Construct $T_2 = [\mathbf{t}_k, \mathbf{t}_{k-1}, \ldots, \mathbf{t}_{d+1}]$ and $T_1 = [\mathbf{t}_d, \mathbf{t}_{d-1}, \ldots, \mathbf{t}_1]$.
Calculate QR decomposition of T_2, i.e. $T_2 = QR$, where $Q \in \mathbb{R}^{k \times k}$ is orthogonal and $R \in \mathbb{R}^{k \times (k-d)}$ is an upper triangular matrix.
We set $c := (R)^{-1}_{k-d,k-d}$ and compute $w_i^{d+1} = c\left(Q^T T_1\right)_{k-d,i}$, for $i = d, \ldots, 1$.
Setting $h_i := w_{d-i+1}^{d+1},\ i = 1, \ldots, d$ and $h_0 := 1$,
we finally have $g(x) = h_0 x^d + h_1 x^{d-1} + \ldots + h_{d-1} x + h_d = \mathrm{GCD}(f_1, \ldots, f_m)$.
end of algorithm

4. Numerical experiment

To compare the two presented algorithms, let us now have the following polynomials:

$$f_0 = (x-0.9)^5(x-0.8)^5(x-0.7)^5(x+0.3)^5(x+0.5)^5(x+0.7)^5,$$
$$f_1 = (x-2)^5(x-0.9)^5(x-0.8)^5(x+0.5)^5(x+2)^5,$$
$$f_2 = (x-3)^5(x-0.8)^5(x+0.5)^5(x+2)^5 \text{ and}$$
$$f_3 = (x-0.8)^4(x+0.5)^4.$$

It is easily seen, that $\mathrm{GCD}(f_0, \ldots, f_3) = f_3$. Accuracy of these computations is shown in Table 1. The errors made in determining the coefficients are about two orders of magnitude smaller in case of Algorithm 2.1 than in the case of Algorithm 3.1.

Coefficients	Error in coefficients	
GCD	Algorithm 2.1	Algorithm 3.1
1.0000	0.0000e+00	0.0000e+00
-1.2000	-9.1038e-15	1.8050e-12
-1.0600	-6.8834e-15	-7.6916e-13
1.3320	2.6645e-15	-2.6426e-12
0.5361	1.2212e-15	3.3151e-13
-0.5328	-2.6645e-15	1.3358e-12
-0.1696	-2.0539e-15	1.1419e-13
0.0768	-7.2164e-16	-2.3732e-13
0.0256	-3.8164e-16	-5.7697e-14

Table 1: Comparison of computational error in AGCD coefficients produced by Algorithm 2.1 and Algorithm 3.1.

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] Bini, D. and Pan, V. Y.: *Polynomial and matrix computation, vol. 1 fundamental algorithms.* Birkhuser, 1994.

[4] 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. 195–200. ACM Press, 1995.

[5] Eliaš, J.: *Problémy spojené s výpočtem největšího společného dělitele*. Bachelor thesis, Charles University, Faculty of Mathematics and Physics, 2009.

[6] Eliaš, J.: *Approximate polynomial greatest common divisor*. Master thesis, Charles University, Faculty of Mathematics and Physics, 2012.

[7] Diaz-Toca, G. M. and Gonzales-Vega, L.: Barnett's theorems about greatest common divisor of several univariate polynomials through Bézout-like matrices. J. Symbolic Computation **34**, (2002), 59–81
doi: 10.1006/jsco. 2002.0542

[8] Diaz-Toca, G. M. and Gonzales-Vega, L. : Computing greatest common divisors and square free decompositions through matrix method: The parametric and approximate cases. Linear Algebra Appl. **412** (2006), 222–246.

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

[10] Kaltofen, E., Yang, Z., and Zhi, L.: Structured low rank approximation of Sylvester matrix. Preprint, 2005.

[11] Li, T. Y. and Zeng, Z.: A rank-revealing method with updating, downdating and applications. SIAM J. Matrix Anal. Appl. **26** No. 4 (2005), 918–946.

[12] Pták, V.: Explicit expressions for Bézoutians. Linear Algebra Appl. **59** (1984), 43–54.

[13] Sun, D. and Zhi, L.: Structured low rank approximation of a Bézout matrix. In:*MM Research preprints*, pp. 207–218. KLMM, AMSS, Academia Sinica, Beijing 2006.

[14] 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.

[15] Winkler, J. R. and Allan, J. D.: Structured total least norm and approximate GCDs of inexact polynomials. Journal of Comp.and Appl. Math. **215** (2006), 1–13.

[16] Zeng, Z.: The approximate GCD od inexact polynomials, Part I: univariate algorithm. Preprint, 2004.

[17] 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.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

NUMERICAL MODELLING OF A BRIDGE SUBJECTED TO SIMULTANEOUS EFFECT OF A MOVING LOAD AND A VERTICAL SEISMIC GROUND EXCITATION

Cyril Fischer, Ondřej Fischer, Ladislav Frýba

Institute of Theoretical and Applied Mechanics, AS CR, v.v.i.
Prosecká 76, Prague 9, Czech Republic
{fischerc|fischero|fryba}@itam.cas.cz

Abstract

A simple beam subjected to a row of regularly distributed moving forces and simultaneous vertical motions of its supports is described using a simplified theoretical model and a finite differences approach. Several levels of simplification of the structure and input data are supposed. Numerical results confirm legitimacy of the assumptions.

1. Introduction

Although dynamic action of moving loads on structures was studied since middle of the nineteenth century, the combined effect of train and earthquake attracted attention only recently, [5]. In the present work, we concentrate to the problem of vertical vibrations of a beam, which is subjected to a row of regularly spaced fast moving forces and simultaneously to motion of its supports due to an earthquake.

An approximative analytical solution to the problem was formulated at the cost of significant simplification many times, e.g., [2]. However, these formulae bring their own difficulties for numerical enumeration: they involve partial sums of infinite trigonometric series, which can introduce spurious oscillations, or hidden pairs of terms, which cancel themselves under certain conditions and thus they are a potential source of numerical instability. Moreover, simplifying assumptions like lack of damping or a limited number of eigenmodes taken into account lower credibility of the formulae. Such obstacles divert attention to numerical alternatives.

Numerical algorithms for solution to fourth order parabolic PDEs have a long tradition. The available methods comprise explicit and implicit finite difference schemas or several variants of finite element methods. Method of lines gained in popularity for general problems. It reformulates the PDE to the form, which is convenient for application of a standard ODE solver.

In this paper, we present an attempt to employ an implicit difference schema for solution to the PDE describing the transverse vibrations of a beam. The numerical procedure is tested on the benchmark case introduced by Evans in [1] and on a simple model of a real bridge, see [3].

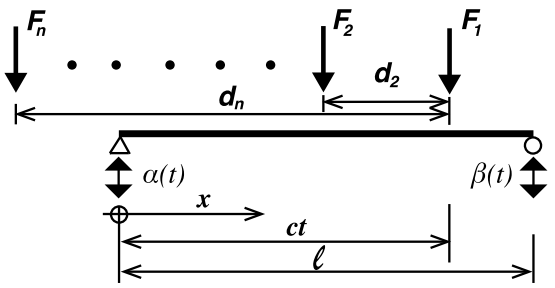


Figure 1: Simplified model of the beam, moving forces, and movement of supports

2. Description of the model and closed form solution

Let us assume a simple damped beam of span ℓ, which is subjected to a row of n moving forces $F_i, i = 1, 2, \ldots, n$ at the distances d_i, see Figure 1. The forces are moving from the left to the right with a constant velocity c. The supports of the beam perform vertical movements $\alpha(t)$ (left support) and $\beta(t)$ (right support), respectively. The problem is governed by the partial differential equation:

$$EI\, v^{\mathrm{IV}}(x,t) + \mu\, \ddot{v}(x,t) + 2\mu\gamma\, \dot{v}(x,t) = \sum\nolimits_{i=1}^{n} F_i \varepsilon_i(t) \delta(x - d_i), \tag{1}$$

$$v(0,t) = \alpha(t),\ v(\ell,t) = \beta(t), \quad v''(0,t) = 0,\ v''(\ell,t) = 0, \tag{2}$$

$$v(x,0) = \dot{v}(x,0) = 0, \tag{3}$$

where $v(x,t)$ is the vertical displacement of the beam at x and time t, respectively, EI is the flexural rigidity of the beam (constant), μ is the mass per unit length of the beam (constant), γ is the circular frequency of the beam damping, $\varepsilon_i(t) = h(t-t_i) - h(t-T_i)$ with $h(t)$ being the Heaviside unit step function, $\delta(x)$ is the Dirac function, $t_i = d_i/c$, $T_i = (\ell + d_i)/c$ is the time when the i-th force enters or leaves the beam, d_i is the distance between the first and i-th force $d_1 = 0$, and primes and dots denote the differentiation with respect to space and time, respectively.

The boundary conditions (2) characterize the "simply supported beam" with prescribed movement of its both ends. The soil displacement functions are usually assumed to be equal $\alpha(t) = \beta(t)$ or shifted $\alpha(t) = \beta(t \pm \Delta t)$ on both ends, however the general choice $\alpha(t) \neq \beta(t)$ is supposed here.

The closed form solution to the problem of beam vibration (1–3) used in this work is described in detail in [3]. Thus, due to space limitation only a few incomplete formulae will be presented here.

The response of the beam $v(x,t)$ is resolved into the so called quasi-static component $v_s(x,t)$ comprising variable boundary conditions and dynamic component $v_d(x,t)$, which includes the moving load on the right hand side:

$$v(x,t) = v_s(x,t) + v_d(x,t). \tag{4}$$

The time-variable boundary conditions $\alpha(t), \beta(t)$ in equation for $v_s(x,t)$ are assumed to be represented by a sum of m selected (dominant) terms of a finite Fourier approximation, possibly modulated by a function of "slow time" τ, $\tau = \sigma t$, $\sigma \ll 1$,

$$\alpha(t) = \sum\nolimits_{k=1}^{m} \gamma(\tau) \sin \omega_k t. \tag{5}$$

Harmonic character of the boundary conditions and assumption of zero damping enables to find analytical solution as a sum of eigenmodes $v_{s,i}(x,t)$:

$$v_s(x,t) = \sum\nolimits_{k=1}^{m} v_{s,k}(x,\tau) \sin \omega_k t, \tag{6}$$

$$v_{s,k}(x,\tau) = C_{k,1} \sin \frac{\lambda_k x}{\ell} + C_{k,2} \cos \frac{\lambda_k x}{\ell} + C_{k,3} \sinh \frac{\lambda_k x}{\ell} + C_{k,4} \cosh \frac{\lambda_k x}{\ell}, \tag{7}$$

where $\lambda_k = \ell \left(\mu \omega_k^2 / EI\right)^{\frac{1}{4}}$ and $C_{k,j}(\tau)$ are given by boundary conditions.

The dynamic component can expressed in the form of eigenmodes expansion:

$$v_d(x,t) = \sum\nolimits_{j=1}^{\infty} q_j(t) \sin \frac{j\pi x}{\ell}, \tag{8}$$

where the functions $q_j(t)$ sum contributions of individual forcing components.

3. Finite difference schema

Let us assume a uniform discretization of the beam with $N-1$ interior points, $0 = x_0 < x_1 < \ldots < x_N = \ell$, $x_i = ih$. The difference schema for the 4^{th} order derivative in (1) with boundary conditions (2) will be deduced from a transformed system ($z(x,t) = v''(x,t)$):

$$\begin{aligned} &z''(x,t) + v(x,t) = f(x,t) \quad \text{and} \quad v(0,t) = \alpha(t),\ v(\ell,t) = \beta(t), \\ &v''(x,t) - z(x,t) = 0, \qquad\qquad z(0,t) = 0, \qquad z(\ell,t) = 0, \end{aligned} \tag{9}$$

which can be discretized using the standard second order difference schema $h^2 v''(x_i) \approx v(x_{i-1}) - 2v(x_i) + v(x_{i+1})$. This procedure avoids explicit formulation of second order boundary conditions. Eliminating the auxiliary variable z the linear algebraic system conforming to (9) with boundary conditions (2) can be written in the matrix form:

$$\frac{1}{h^4} \mathbf{M} \cdot \mathbf{v}_j = \mathbf{f}_j + \frac{1}{h^4} \mathbf{g}_j. \tag{10}$$

Vector $\mathbf{v}_j$ represents unknown displacements of internal nodes x_i, $i = 1, \ldots, N-1$ at time instant $t_j = j \cdot \Delta$. Vector $\mathbf{f}_j = \{f(x_i, t_j)\}_{i=1}^{N-1}$ corresponds to the value of the right hand side in the internal nodes. The symmetric matrix $\mathbf{M} \in \mathbb{R}^{(N-1)\times(N-1)}$ consists of 5 non-zero diagonals with numbers $6, -4, 1$ on the main-, 1^{st}, and 2^{nd} sub- and superdiagonal, respectively, with the exception of the corner values: $M_{1,1} = M_{N-1,N-1} = 5$. Elements of vector $\mathbf{g}_j \in \mathbb{R}^{(N-1)}$ are given as

$$\mathbf{g}_j = \left(2\alpha(t), -\alpha(t), 0, \ldots, 0, -\beta(t), 2\beta(t)\right)^T. \tag{11}$$

The time derivatives at $t = t_j = j \cdot \Delta$ will be approximated by formulae

$$\Delta^2 \ddot{v}(t_j) \approx v(t_{j-2}) - 2v(t_{j-1}) + v(t_j)\,, \qquad 2\Delta \dot{v}(t_j) \approx v(t_{j-2}) - 4v(t_{j-1}) + 3v(t_j). \quad (12)$$

The final implicit recurrence formula can be written in the matrix form for $j = 1, \ldots$

$$\left(b^2\mathbf{M} + \left(1 + \frac{3}{2}\gamma\Delta\right)\mathbf{I}\right) \cdot \mathbf{v}_j = \frac{\Delta^2}{\mu}\mathbf{f}_j + b^2\mathbf{g}_i + 2(1+\gamma\Delta)\mathbf{v}_{j-1} - \left(1 - \frac{1}{2}\gamma\Delta\right)\mathbf{v}_{j-2}, \quad (13)$$

where

$$b = \sqrt{\frac{EI}{\mu}}\frac{\Delta}{h^2}. \quad (14)$$

In compliance with the initial conditions (3) the two starting values can be considered zero: $\mathbf{v}_{-1} = \mathbf{v}_0 = \mathbf{0}$.

The discretization parameters h, Δ should be chosen to allow consistent description of the moving load. The value of h should correspond to axle distances of the supposed train and the time step Δ has to be dependent on the train velocity

$$h = \frac{1}{k}\mathrm{GCD}\{d_1, \ldots, d_n\}\,, \quad \Delta t = \frac{1}{l}\frac{h}{c} \qquad \text{for some } \; k, l \in \mathbb{N}. \quad (15)$$

The consistent distribution of the axle load F_i between two adjacent space nodes is necessary if $l > 1$. This can be assured, e.g., by the choice

$$f(x,t) = \sum\nolimits_{i=1}^{n} F_i \max\left\{0, 1 - \left|\frac{x - (ct - d_i)}{h}\right|\right\}. \quad (16)$$

4. Numerical verification

Problem 1. The simple benchmark case was used first in [1] and then subsequently several times. It considers free vibration case ($f(x,t) = 0$) of an undamped system (1–2) with parameters $\ell = 1, EI = 1, \mu = 1, \gamma = 0, \alpha(t) = \beta(t) = 0$ and

$$v(x,0) = \frac{1}{12}x(2x^2 - x^3 - 1)\,; \qquad \dot{v}(x,0) = 0 \qquad \text{for } 0 \le x \le 1. \quad (17)$$

The exact solution to the continuous problem is obtained by Fourier series analysis:

$$v(x,t) = \sum\nolimits_{s=1}^{\infty} \frac{4}{s^5\pi^5}\left(\cos(s\pi) - 1\right)\sin(s\pi x)\cos(s^2\pi^2 t). \quad (18)$$

Figure 2(a-b) shows numerical approximation (solid curves) of $v(0.5, t)$ computed using the finite difference recurrence (13) together with the corresponding exact solutions (dashed curves) for two different time steps, $\Delta = 0.005, 0.00125$. The relatively high decrease of the computed amplitude in the plot b) is caused by numerical dispersion (damping), see [4]. The rate of numerical dispersion depends on the value of coefficient b (14). The same coefficient occurs in the stability criterion of explicit difference schemas but with different interpretation.

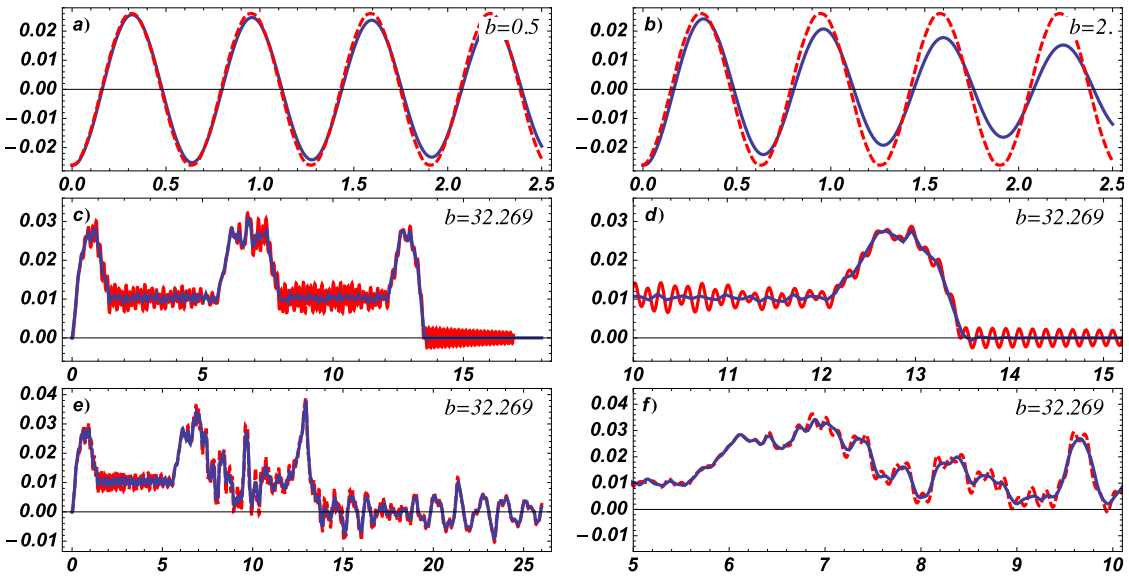


Figure 2: Mid-span deflection ($x = \frac{1}{2}\ell$) of three benchmark cases – numerical approximation (solid, blue) and exact solution (dashed, red).
(a–b) Free vibration benchmark, $h = 0.05$, (a) $\Delta = 0.00125$, (b) $\Delta = 0.005$.
(c–d) Concrete bridge ($\ell = 20$m), train passing at speed $c = 100$ km/h, $h = 0.5, \Delta = 0.0045$, (d) detail for $t \in (10, 15)$.
(e–f) Concrete bridge ($\ell = 20$m), train passing at speed $c = 100$ km/h and an earthquake shock at $t_e = 6.76$s, $\Delta = 0.0045$, (f) detail for $t \in (5, 10)$.
(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Problem 2. The second example is selected from the parametric study presented by authors in [3]. Parameters of the concrete bridge are specified as $\ell = 20$m, $\mu = 8 \cdot 10^3$kg, $EI = 65.5 \cdot 10^6$m^{3}kg.s^{-2}, $\gamma = 1.27$s^{-1}. The train Talgo AV consists of 2 indentical formations with 7 carriages and 20 axles, 16 tons each. Figure 2(c-d) shows the mid-span deflection of the bridge caused by train cruising at speed of c =100km/h. The three significant peaks are caused by the motorized carriages (one on each end of the train and two in the middle). The highly oscillating curve (red) depicts the approximative analytical solution, only first eigenmode is taken into account. The dark smooth curve corresponds to numerical solution (13) with a relatively large quotient $b \approx 32$. It follows approximately the mean value of the analytical solution. Difference of both solutions in greater detail can be seen in part d) of the figure. The high numerical damping wiped out small oscillations as well as the free vibration after the train left the bridge ($t = 13.5$s).

Problem 3. Figure 2(e-f) shows effect of the combined load of train and earthquake. The earthquake is represented by its several Fourier components and a simple modulation function, see [3]. The shock reaches the bridge at the moment when the

first formation of the train leaves the bridge. At this moment is the response due to passing train maximal because the four middle axles forces of the Talgo train represent a pair of engines. It is apparent that after the earthquake shock the amplitude increases. Coincidence between approximate analytical (red, dashed) and numerical (blue, solid) is fairly good: the maximal relative error is $\sim 10\%$ despite the significant simplification of the analytic model and the large time step which leads to $b \approx 32$.

5. Conclusions

We presented a simplified analysis of the vertical vibration of a bridge, which is is caused by a concurrent action of a long sequence of axle forces or their groups distributed in almost regular distances and a support motion due to an earthquake. The implicit finite difference scheme was introduced to verify justifiability of the simplifying assumptions of the approximative closed form solution. The computed responses were compared to those obtained using analytical methods with good results: the agreement between analytical and numerical results for the benchmarks was within desired 1% provided that the time step Δ was sufficiently small. Some problems with high numerical damping are reported.

Acknowledgements

The kind support of the Czech Science Foundation Project No. GC13-34405J and of the RVO 68378297 institutional support are gratefully acknowledged.

References

[1] Evans, D. J.: A stable explicit method for the finite-difference solution of a fourth-order parabolic partial-differential equation. Comput. J. **8** (1965), 280–287.

[2] Frýba, L.: *Vibration of solids and structures under moving loads.* 3rd ed., Academia, Prague, Thomas Telford, London, 1999.

[3] Frýba, L., Urushadze, S. and Fischer C.: Vibration of a beam resting on movable supports and subjected to moving loads. In: A. Cunha, E. Caetano, P. Ribeiro, G. Müller (Eds.), *Proceedings of the 9th International Conference on Structural Dynamics, EURODYN 2014*, pp. 1361–1368 (`190_MS06_ABS_1291`). Porto, 2014.

[4] Hoffman, J. D.: *Numerical methods for engineers and scientists.* 2nd ed., Marcel Dekker, Basel, 2001.

[5] Yau, J. D. and Frýba, L.: Response of suspended beams due to moving loads and vertical seismic ground excitations. Eng. Struct. **29** (2007), 3255–3262.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

AN APPLICATION OF THE BDDC METHOD TO THE NAVIER-STOKES EQUATIONS IN 3-D CAVITY

Martin Hanek[1], Jakub Šístek[2], Pavel Burda[1]

[1] Czech Technical University
Technická 4, Prague, Czech Republic
martin.hanek@fs.cvut.cz, pavel.burda@fs.cvut.cz
[2] Institute of Mathematics AS CR
Žitná 25, Prague, Czech Republic
sistek@math.cas.cz

Abstract

We deal with numerical simulation of incompressible flow governed by the Navier-Stokes equations. The problem is discretised using the finite element method, and the arising system of nonlinear equations is solved by Picard iteration. We explore the applicability of the Balancing Domain Decomposition by Constraints (BDDC) method to nonsymmetric problems arising from such linearisation. One step of BDDC is applied as the preconditioner for the stabilized variant of the biconjugate gradient (BiCGstab) method. We present results for a 3-D cavity problem computed on 32 cores of a parallel supercomputer.

1. Introduction

The Balancing Domain Decomposition by Constraints (BDDC) was developed by Dohrmann in [1] as an efficient method to solve large systems of linear equations arising from partial differential equations discretised by the finite element method. In [1], the method was applied to elliptic problems, namely Poisson problem and linear elasticity. BDDC was extended to the incompressible Stokes problem in [4] considering finite elements with discontinuous approximation of pressure. In [6], the BDDC method was applied to the Stokes problem discretised by Taylor-Hood finite elements with continuous pressure approximation. The interface problem in this monolithic approach contains both velocity and pressure unknowns. An alternative approach was presented in [3]. A generalisation of the BDDC method for systems with nonsymmetric matrix was proposed in [12] and applied to Euler equations of inviscid compressible flows.

In our contribution, we combine the approach to building the interface problem from [6] with the extension to nonsymmetric problems from [12]. The algorithm is applied to nonsymmetric linear systems obtained by Picard's linearisation of the steady Navier-Stokes equations using Taylor-Hood finite elements. Numerical results for flow inside a 3-D lid driven cavity are presented.

2. Navier-Stokes equations and the finite element method

We consider stationary flow of incompressible fluid in three spatial dimensions, governed by the Navier-Stokes equations without body forces (see e.g. [2])

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{0} \quad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{2}$$

where $\boldsymbol{u} = (u_1, u_2, u_3)^T$ is an unknown velocity vector, p is an unknown pressure normalised by (constant) density, ν is a given kinematic viscosity, and Ω is the solution domain. In addition, the following boundary conditions are considered

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \Gamma_D, \tag{3}$$

$$-\nu(\nabla\boldsymbol{u})\mathbf{n} + p\mathbf{n} = 0 \qquad \text{on } \Gamma_N, \tag{4}$$

where Γ_D and Γ_N are parts of the boundary $\partial\Omega$, $\overline{\Gamma_D} \cup \overline{\Gamma_N} = \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$, $\mathbf{n}$ is the outer unit normal vector of the boundary, and $\boldsymbol{g}$ is a given function.

2.1. Weak formulation

In deriving the weak mixed formulation, we multiply equations (1)–(2) by test functions and integrate over the solution domain. Then using the divergence theorem, we get the final weak formulation

We seek $\boldsymbol{u} \in V_g$ *and* $p \in L^2(\Omega)$*, satisfying*

$$\int_\Omega (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{v} \mathrm{d}\Omega + \nu \int_\Omega \nabla\boldsymbol{u} : \nabla\boldsymbol{v} \mathrm{d}\Omega - \int_\Omega p\nabla \cdot \boldsymbol{v} \mathrm{d}\Omega = 0 \quad \forall \boldsymbol{v} \in V, \tag{5}$$

$$\int_\Omega q\nabla \cdot \boldsymbol{u} \mathrm{d}\Omega = 0 \quad \forall q \in L^2(\Omega). \tag{6}$$

Here the spaces are

$$V_g := \left\{ \boldsymbol{u} \in H^1(\Omega)^3, \boldsymbol{u} = \boldsymbol{g} \text{ on } \Gamma_D \right\},$$
$$V := \left\{ \boldsymbol{v} \in H^1(\Omega)^3, \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_D \right\}.$$

2.2. Assembly of the system of algebraic equations

During the assembly of the system of algebraic equations, we substitute into the weak formulation (5)–(6) for $\boldsymbol{u}$, p, $\boldsymbol{v}$, and q their finite element counterparts

$$\boldsymbol{u}_h = \sum_{i=1}^{3n_{\boldsymbol{u}}} u_i \boldsymbol{\phi}_i, \quad p_h = \sum_{i=1}^{n_p} p_i \psi_i, \quad \boldsymbol{v}_h = \sum_{i=1}^{3n_{\boldsymbol{u}}} v_i \boldsymbol{\phi}_i, \quad q_h = \sum_{i=1}^{n_p} q_i \psi_i.$$

Here $\boldsymbol{\phi}_i$ are vector basis functions for velocity, ψ_i are scalar basis functions for pressure, $n_{\boldsymbol{u}}$ is the number of nodes with velocity unknowns, and n_p is the number of nodes with pressure unknowns. For the considered hexahedral Taylor-Hood finite elements (see e.g. [2]), $n_{\boldsymbol{u}}$ is approximately eight times larger than n_p.

We obtain the following system of algebraic equations

$$\begin{bmatrix} \nu\mathbf{A}+\mathbf{N}(\mathbf{u}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{7}$$

where $\mathbf{u}$ is the vector of unknown coefficients of velocity, $\mathbf{p}$ is the vector of unknown coefficients of pressure, $\mathbf{A}$ is the matrix of diffusion, $\mathbf{N}(\mathbf{u})$ is the matrix of advection which depends on the solution, B is the matrix from continuity equation, and $\mathbf{f}$ and $\mathbf{g}$ are discrete right-hand side vectors arising from Dirichlet boundary conditions. Each part of system (7) is assembled as (see [2])

$$\mathbf{A} = [a_{ij}],\ a_{ij} = \int_\Omega \nabla\boldsymbol{\phi}_i : \nabla\boldsymbol{\phi}_j \,\mathrm{d}\Omega, \tag{8}$$

$$\mathbf{N}(\mathbf{u}) = [n_{ij}],\ n_{ij} = \int_\Omega (\mathbf{u}\cdot\nabla)\boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_i \,\mathrm{d}\Omega, \tag{9}$$

$$B = [b_{lj}],\ b_{lj} = -\int_\Omega \psi_l \nabla\cdot\boldsymbol{\phi}_j \,\mathrm{d}\Omega, \tag{10}$$

$$\mathbf{f} = [f_i],\ f_i = -\sum_{j=3n_{\boldsymbol{u}}+1}^{3(n_{\boldsymbol{u}}+\partial n_{\boldsymbol{u}})} u_j \int_\Omega (\mathbf{u}\cdot\nabla)\boldsymbol{\phi}_j\cdot\boldsymbol{\phi}_i \,\mathrm{d}\Omega - \nu \sum_{j=3n_{\boldsymbol{u}}+1}^{3(n_{\boldsymbol{u}}+\partial n_{\boldsymbol{u}})} u_j \int_\Omega \nabla\boldsymbol{\phi}_j : \nabla\boldsymbol{\phi}_i \,\mathrm{d}\Omega, \tag{11}$$

$$\mathbf{g} = [g_l],\ g_l = \sum_{j=3n_{\boldsymbol{u}}+1}^{3(n_{\boldsymbol{u}}+\partial n_{\boldsymbol{u}})} u_j \int_\Omega \psi_l \nabla\cdot\boldsymbol{\phi}_j \,\mathrm{d}\Omega. \tag{12}$$

System (7) is nonlinear due to the matrix $\mathbf{N}(\mathbf{u})$, and for its linearisation, we use the Picard iteration. This leads to solving a sequence of linear systems of equations in the form

$$\begin{bmatrix} \nu\mathbf{A}+\mathbf{N}(\mathbf{u}^k) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{k+1} \\ \mathbf{p}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{13}$$

where $\mathbf{N}(\mathbf{u}^k)$ means that we substitute a solution of velocity from the previous step to the matrix $\mathbf{N}$. This—already linear—nonsymmetric system is solved by means of iterative substructuring.

3. Iterative substructuring

For our calculations, we use decomposition of domain Ω into N nonoverlapping subdomains. In order to explain how the BDDC algorithm fits to problem (13), we assume reordering of unknowns within $\mathbf{u}$ and $\mathbf{p}$ such that the components corresponding to the nodes on the interface are at the end. This leads to the following blocking of the system

$$\begin{bmatrix} \nu\mathbf{A}_{11}+\mathbf{N}_{11} & \nu\mathbf{A}_{12}+\mathbf{N}_{12} & B_{11}^T & B_{21}^T \\ \nu\mathbf{A}_{21}+\mathbf{N}_{21} & \nu\mathbf{A}_{22}+\mathbf{N}_{22} & B_{12}^T & B_{22}^T \\ B_{11} & B_{12} & 0 & 0 \\ B_{21} & B_{22} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{p_1} \\ \mathbf{p_2} \end{bmatrix} = \begin{bmatrix} \mathbf{f_1} \\ \mathbf{f_2} \\ \mathbf{g_1} \\ \mathbf{g_2} \end{bmatrix}, \tag{14}$$

where subscript $_1$ denotes the part with interior unknowns and subscript $_2$ denotes the part with interface unknowns. The whole blocks are now permuted to get an interface problem, similarly as it was done for the Stokes problem in [6],

$$S \begin{bmatrix} \mathbf{u_2} \\ \mathbf{p_2} \end{bmatrix} = g. \tag{15}$$

Here

$$S = \begin{bmatrix} \nu\mathbf{A}_{22} + \mathbf{N}_{22} & B_{22}^T \\ B_{22} & 0 \end{bmatrix} - \begin{bmatrix} \nu\mathbf{A}_{21} + \mathbf{N}_{21} & B_{12}^T \\ B_{21} & 0 \end{bmatrix} \begin{bmatrix} \nu\mathbf{A}_{11} + \mathbf{N}_{11} & B_{11}^T \\ B_{11} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nu\mathbf{A}_{12} + \mathbf{N}_{12} & B_{21}^T \\ B_{12} & 0 \end{bmatrix}$$

is the Schur complement with respect to the interface, and

$$g = \begin{bmatrix} \mathbf{f_2} \\ \mathbf{g_2} \end{bmatrix} - \begin{bmatrix} \nu\mathbf{A}_{21} + \mathbf{N}_{21} & B_{12}^T \\ B_{21} & 0 \end{bmatrix} \begin{bmatrix} \nu\mathbf{A}_{11} + \mathbf{N}_{11} & B_{11}^T \\ B_{11} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f_1} \\ \mathbf{g_1} \end{bmatrix}$$

is the reduced right-hand side.

Problem (15) is solved by the BiCGstab method [10], and one step of BDDC is used as a preconditioner. Thanks to domain decomposition, both the action of the BDDC preconditioner and of the matrix S are parallelised in each iteration. This is realised by the multilevel BDDC implementation in the *BDDCML* library[1] (version 2.4) [8] employed in our computations.

4. BDDC for nonsymmetric systems

The BDDC preconditioner works with a residuum r^k obtained from the k-th iteration of the BiCGstab algorithm

$$r^k = g - S \begin{bmatrix} \mathbf{u_2}^k \\ \mathbf{p_2}^k \end{bmatrix}. \tag{16}$$

The preconditioner provides an approximate solution to problem (15), and it is realised by one iteration of the BDDC method.

A key idea of BDDC is to choose suitable *coarse degrees of freedom*, and then seek solution on the interface in a space of functions that are continuous in these coarse degrees of freedom. Although more advanced choices were introduced for advection-diffusion problem in [9], we restrict ourselves in this study to continuity at coarse nodes, which are selected according to [7], and continuity of arithmetic averages over all faces and edges enforced independently for each component of velocity and for pressure.

[1] `http://users.math.cas.cz/~sistek/software/bddcml.html`

In each action of the BDDC preconditioner, a coarse problem and independent subdomain problems are solved. First we look at one subdomain problem. It takes the total residuum r^k and extracts a local part on the subdomain as

$$r_i = W_i R_i r^k, \tag{17}$$

where R_i is an operator restricting a global interface vector to i-th subdomain, and matrix W_i applies weights to satisfy the partition of unity. Then we solve on each subdomain a saddle-point problem

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} u_i \\ \lambda \end{bmatrix} = \begin{bmatrix} r_i \\ 0 \end{bmatrix}, \tag{18}$$

where λ are Lagrange multipliers, S_i is the Schur complement with respect to the interface of the i-th subdomain, and C_i is the matrix defining coarse degrees of freedom, which has as many rows as is the number of coarse degrees of freedom defined at the subdomain. After solving this problem on each subdomain, we get the *subdomain correction*.

Let us now have a look at the coarse problem. Before solving it in each iteration, one needs to build it in the set-up phase of the preconditioner. This is performed by solving the saddle-point systems from (18) with several right-hand sides

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{19}$$

The solution Ψ_i is the matrix of *coarse basis functions* with every column corresponding to one coarse unknown on the subdomain. These functions are equal to one in one coarse degree of freedom, and they equal to zero in the remaining local coarse unknowns. As introduced in [12], also a set of *adjoint coarse basis functions* Ψ_i^* is needed for nonsymmetric problems. These are obtained by solving

$$\begin{bmatrix} S_i^T & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i^* \\ \Lambda_i^T \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{20}$$

By solving problem (19), we obtain the *local coarse matrix* as a side product,

$$S_{Ci} = \Psi_i^{*T} S_i \Psi_i = -\Lambda_i.$$

Local coarse matrices are then assembled into the global matrix of the coarse problem

$$S_C = \sum_{i=1}^{N} R_{Ci}^T S_{Ci} R_{Ci},$$

where R_{Ci} is the restriction of the global vector of coarse unknowns to those present at i-th subdomain.

In each action of BDDC, we first extract the residuum for the coarse problem as

$$r_C = \sum_{i=1}^{N} R_{Ci}^T \Psi_i^{*T} r_i,$$

solve the coarse problem

$$S_C u_C = r_C, \tag{21}$$

and distribute the coarse solution to individual subdomains

$$u_{Ci} = \Psi_i R_{Ci} u_C.$$

The complete action of the preconditioner $M_{BDDC} : r^k \to u^k$ is obtained by combining the subdomain corrections with the localised coarse corrections,

$$u^k = \sum_{i=1}^{N} R_i^T W_i (u_i + u_{Ci}).$$

5. Numerical results

As the benchmark problem, we consider the 3-D extension of the popular problem in cavity introduced in [11]. The computational domain is a unit cube. The mesh is divided into 32 subdomains using the METIS library (see Figure 1). The computations are performed by a parallel finite element package written in C++ and described in [5], and the *BDDCML* library [8] is used for solving the arising system of equations. Simulations were performed on an SGI Altix UV 100 supercomputer at the Supercomputer center of the CTU in Prague using 32 cores and the same number of subdomains. Our results are compared with [11]. Two directions of the unit tangential velocity vector are considered on the top wall, $\mathbf{u_{top1}} = (1, 0, 0)$ and $\mathbf{u_{top2}} = (1/\sqrt{3}, \sqrt{2}/\sqrt{3}, 0)$. Picard iteration is used for linearisation, with precision $\left\|u^k - u^{k-1}\right\|_2 \leq 10^{-5}$. In [11], FpGMRES method is used for the linearised systems with a block preconditioner. In our computations, the BiCGstab method preconditioned by the BDDC preconditioner is used. The linear iterations are terminated when $\left\|r^k\right\|_2 / \left\|g\right\|_2 \leq 10^{-6}$ or after reaching the maximum number of 100 iterations.

We compare the maximal numbers of linear iterations over all steps of the nonlinear method. These are considered for two equidistant meshes, with $n = 16$ and 32 elements per edge, corresponding respectively to 4096 and 32 768 elements, 35 937 and 274 625 nodes, and 112 724, 859 812 unknowns. Four different values of viscosity ν are tested. Results are presented in Tables 1 and 2. Number of linear iterations from [11] are denoted as 'FpGMRES + block prec.', while our current results are denoted as 'BiCGstab + BDDC'. Finally, numbers of nonlinear iteration required in our calculations are reported in Table 3.

From Tables 1 and 2, we can see that the number of linear iterations is growing with decreasing viscosity, while the dependence is similar for both methods. This

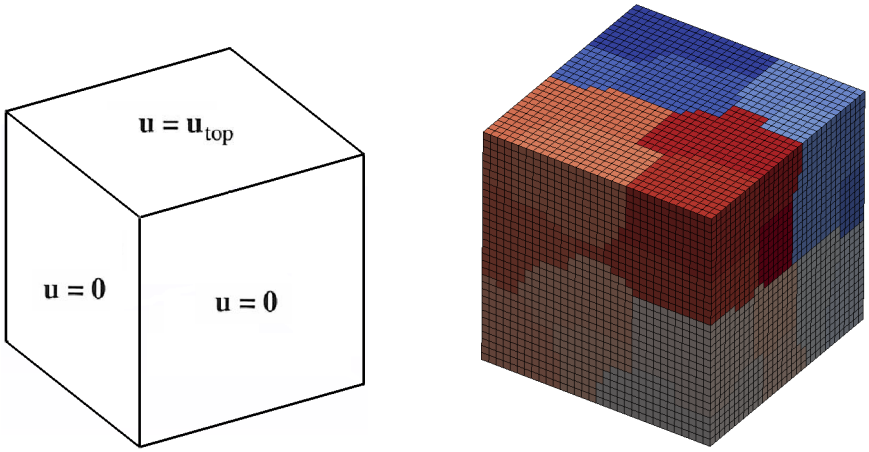


Figure 1: Solution domain with boundary conditions (left) and mesh with 32 subdomains for cavity problem (right)

ν		1/20	1/40	1/80	1/160
$n = 16$	FpGMRES + block prec.	29	32	43	68
	BiCGstab + BDDC	32	31	38	58
$n = 32$	FpGMRES + block prec.	28	32	42	69
	BiCGstab + BDDC	18	19	23	49

Table 1: Number of linear iterations for $\mathbf{u_{top1}} = (1, 0, 0)$

ν		1/20	1/40	1/80	1/160
$n = 16$	FpGMRES + block prec.	29	36	48	64
	BiCGstab + BDDC	29	31	35	53
$n = 32$	FpGMRES + block prec.	28	35	45	61
	BiCGstab + BDDC	18	19	23	49

Table 2: Number of linear iterations for $\mathbf{u_{top2}} = (1/\sqrt{3}, \sqrt{2}/\sqrt{3}, 0)$

confirms that for this problem, the BDDC preconditioner provides a comparable efficiency as the advanced block preconditioner from [11]. As shown in Table 3, a reasonable convergence of the Picard iteration has been obtained for most cases. However, skewing the velocity vector on the lid with respect to coordinate axes had an opposite effect than we expected, with significantly worse convergence for $\mathbf{u_{top1}}$ than for $\mathbf{u_{top2}}$ in the case $\nu = 1/160$. We do not have an explanation for this behaviour. The solution for $\mathbf{u_{top2}}$, $n = 32$, and $\nu = 1/160$ in the slice $x = 0.5$ is shown in Figure 2.

ν		1/20	1/40	1/80	1/160
$\mathbf{u_{top1}}$	$n = 16$	8	11	19	198
	$n = 32$	8	12	21	114
$\mathbf{u_{top2}}$	$n = 16$	8	11	18	39
	$n = 32$	8	12	21	47

Table 3: Number of nonlinear iterations in our calculations

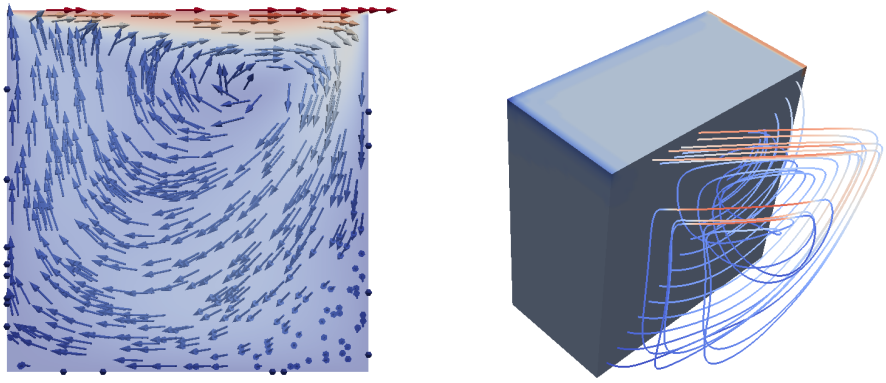


Figure 2: Cavity flow in the plane $x = 0.5$, velocity vectors with magnitude (left) and pressure with several streamtraces (right)

6. Conclusions

In this contribution, we have combined our previous developments on BDDC for the Stokes problem [6], with extensions of the BDDC method to nonsymmetric problems from [12]. An application of the BDDC preconditioner to nonsymmetric linear systems of equations obtained from linearisation of the incompressible Navier-Stokes equations by means of Picard iteration is presented. Taylor-Hood finite elements with continuous approximation of pressure are used for discretisation.

The parallel implementation of the method is employed for solving a 3-D problem of flow in a lid-driven cavity. The required numbers of linear iterations are compared with those by a block preconditioner published in [11], showing a comparable performance of this approach. The BiCGstab method is used for solution of the interface problem, which contains both velocity and pressure unknowns.

Larger tests of parallel scalability and applications to other problems will be the subject of future research.

Acknowledgements

This work was supported by the Czech Ministry of Education, Youth and Sports of the Czech Republic under research project LH11004, by the Czech Science Foundation through grant 14-02067S, by the Academy of Sciences of the Czech Republic through RVO:67985840, and by the Czech Technical University in Prague through the student project SGS13/190/OHK2/3T/12.

References

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

[2] Elman, H. C., Silvester, D. J., and Wathen, A. J.: *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[3] Li, J. and Tu, X.: A nonoverlapping domain decomposition method for incompressible Stokes equations with continuous pressures. SIAM Journal on Numerical Analysis **51** (2013), 1235–1253.

[4] Li, J. and Widlund, O. B.: BDDC algorithms for incompressible Stokes equations. SIAM J. Numer. Anal. **44** (2006), 2432–2455.

[5] Šístek, J. and Cirak, F.: Parallel iterative solution of the incompressible Navier-Stokes equations with application to rotating wings, 2014. To be submitted.

[6] Šístek, J., Sousedík, B., Burda, P., Mandel, J., and Novotný, J.: Application of the parallel BDDC preconditioner to the Stokes flow. Comput. Fluids **46** (2011), 429–435.

[7] Šístek, J., Čertíková, M., Burda, P., and Novotný, J.: Face-based selection of corners in 3D substructuring. Math. Comput. Simulat. **82** (2012), 1799–1811.

[8] Sousedík, B., Šístek, J., and Mandel, J.: Adaptive-Multilevel BDDC and its parallel implementation. Computing **95** (2013), 1087–1119.

[9] Tu, X. and Li, J.: A balancing domain decomposition method by constraints for advection-diffusion problems. Commun. Appl. Math. Comput. Sci **3** (2008), 25–60.

[10] van der Vorst, H. A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **13** (1992), 631–644.

[11] Wathen, A. J., Loghin, D., Kay, D. A., Elman, H. C., and Silvester, D. J.: A new preconditioner for the Oseen equations. In: F. Brezzi, A. Buffa, S. Corsaro, and A. Murli (Eds.), *Numerical mathematics and advanced applications*. Springer-Verlag Italia, Milano, 2003 pp. 979–988. Proceedings of ENUMATH 2001, Ischia, Italy.

[12] Yano, M.: *Massively parallel solver for the high-order Galerkin least-squares method*. Master's thesis, Massachusests Institute of Technology, 2009.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

EXPERIMENTAL COMPARISON OF TRAFFIC FLOW MODELS ON TRAFFIC DATA

Ivan Horňák, Jan Přikryl

Institute of Information Theory and Automation
Pod Vodárenskou věží 2, CZ-182 00 Praha 8, Czech Republic
prikryl@utia.cas.cz

Abstract

Despite their deficiencies, continuous second-order traffic flow models are still commonly used to derive discrete-time models that help traffic engineers to model and predict traffic flow behaviour on highways. We briefly overview the development of traffic flow theory based on continuous flow-density models of Lighthill-Whitham-Richards (LWR) type, that lead to the second-order model of Aw-Rascle. We will then concentrate on widely-adopted discrete approximation to the LWR model by Daganzo's Cell Transmission Model. Behaviour of the discussed models will be demonstrated by comparing the traffic flow prediction based on these models with real traffic data on the southern highway ring of Prague.

1. Introduction

Management systems for highway traffic have existed since 1970s. These complex systems consist of different decision-making tools that address the management of pavement and bridges, public transport, congestion and safety, or traffic data monitoring. In this paper we will concentrate on numerical aspects of three mathematical models that can be used to predict traffic flow behaviour for highway management purposes.

Traffic flow models can be divided into four basic groups according to the level of detail that the model attempts to implement. The most widely employed class of traffic flow models are probably *macroscopic models* that disregard individual vehicles and consider the highway traffic to be an equivalent of compressible fluid flow. As these models are commonly used to predict and control (manage) the highway traffic, the role of such models is crucial for the success of any management action: A good model provides relatively accurate predictions of the future and it is computationally as simple as possible.

In the rest of our paper we will evaluate three possible traffic flow models that may be considered to be good candidates for modelling of highway traffic, and we will examine their performance in predicting highway traffic flow.

2. Macroscopic traffic models

A macroscopic traffic model incorporates traffic flux q [veh/hr],[1] traffic density ϱ [veh/km] and velocity v [km/hr], and describes the so-called *fundamental diagram of traffic flow*. Such a model can be used to predict the behaviour of a road system when applying control or management actions (e.g. ramp metering or speed limits).

Let us first study the number of vehicles N_1 and N_2 entering and leaving a road segment of length Δx metres during Δt seconds. Consider a hypothetical situation where a build-up of vehicles ($N_2 < N_1$) occurs. The change in the flow rate q is $\Delta q = \Delta N/\Delta t$ and the change in vehicle density ϱ is $\Delta \varrho = -\Delta N/\Delta x$.

As the vehicles inside the segment have no possibility to exit the road, vehicles are conserved, with ΔN denoting the number of vehicles inside the segment. Therefore,

$$\Delta q \Delta t = \Delta N = -\Delta \varrho \Delta x \quad \Leftrightarrow \quad \frac{\Delta \varrho}{\Delta t} + \frac{\Delta q}{\Delta x} = 0.$$

This justifies the following relationship for continuous $q(x,t)$ and $\varrho(x,t)$:

$$\frac{\partial \varrho}{\partial t} + \frac{\partial q}{\partial x} = 0 \qquad \text{or} \qquad \partial_t \varrho + \partial_x q = 0. \tag{1}$$

Continuous macroscopic traffic flow models are typically derived from this equation, by introducing a form of speed-influenced density. The most prominent type of the first-order models that result from the direct application of the above equation is the LWR model, described in the next section.

2.1. Lighthill-Whitham-Richards

Lighthill-Whitham-Richards (LWR) model [8, 10] is a first-order model that results from a direct application of the conservation law (1) where the flow rate is function of velocity $v(\varrho)$

$$q(x,t) = v(\varrho(x,t)) \cdot \varrho(x,t).$$

The speed is typically expressed as

$$v(\varrho) = v_\mathrm{f}\,(1 - \frac{\varrho}{\varrho_\mathrm{jam}}),$$

where v_f is the free-flow speed of solitary vehicles, and ϱ_jam denotes so-called *jam density* of the road, that is the maximum possible density of vehicles in the moment when the traffic flow has completely stopped due to traffic jam.

The model is quite simple and numerically stable and even today is often used to study traffic flow phenomena that occur on highways or in road tunnels. According to critical studies [5, 6], the model provides results that correspond well with the theory of kinematic waves, and its output is consistent with empirically observed fundamental diagram data. However, this simple model is unable to capture certain phenomena that occur in everyday traffic, like stop-an-go waves or travel speed adaptivity.

[1] The unit "veh" denotes a *unit vehicle*, an average vehicle that makes it possible to disregard the heterogeneity of traffic flow. Also known as PCE, *passenger car equivalent*.

2.2. Second-order fluid approximations

The inability of LWR-class models to capture more complex traffic flow phenomena led to creation of more elaborate models. These models use an additional set of equations to introduce a relation similar to conservation of momentum in fluids, in the hope that this additional level of detail would lead to a more detailed level of description.

Two seminal works of Payne [9] and Whitham [11] emerged, and sparked a great deal of effort resulting in numerous publications of so-called PW-type flow models, introducing variations and extensions and proposing different numerical schemes. However, 20 year later Daganzo [5] demonstrated that these "higher order" approaches are not appropriately constructed and lead to unrealistic results.

The only continuous traffic flow model of second order that is currently still being studied is due to Aw and Rascle [1]. This AR-type model[2] addresses most of the previous flaws of PW-type models. It takes the composite form of two first-order models,

$$\partial_t \varrho + \partial_x(v\varrho) = 0$$
$$\partial_t(v + p(\varrho)) + v\partial_x\big(v + p(\varrho)\big) = 0$$

where the pressure function of vehicle density $p(\varrho)$ is smooth and increasing.

An AR-type model can be quite conveniently solved using the central upwind scheme [7, 2]. However, as we will see in our experiments, special attention has to be paid to selecting appropriate space- and time-steps.

3. Cell Transmission Model (CTM)

Daganzo in [3] introduced the CTM, where he simplified the first-order models by using a piecewise-linear approximation of the fundamental diagram, depicted in Figure 1. CTM replaces the original LWR state equation (1) by a set of affine functions

$$q = \min\left(v\varrho, q_{\max}, w(\varrho_{\text{jam}} - \varrho)\right).$$

The follow-up paper [4] examines the evolution of traffic on a highway segment divided into I consecutive cells numbered starting at the upstream end of the road, $i = 1, 2, \ldots, I$. The segments are homogeneous and their length is set equal to the distance traveled by typical vehicle in light traffic in one clock tick (time step k of constant length Δt).

The cell transmission model is based on a recursion where the cell occupancy at step $k + 1$ equals its occupancy at step k, plus the inflow and minus the outflow,

$$n_i[k+1] = n_i[k] + y_i[k] - y_{i+1}[k], \tag{2}$$

where the flow from cell $i - 1$ to i during the time interval k is assumed to be

$$y_i[k] = \min\{n_{i-1}[k-1], Q_i[k], N_i[k] - n_i[k]\}, \tag{3}$$

[2]Note that AR in this paper is not related to autoregressive models.

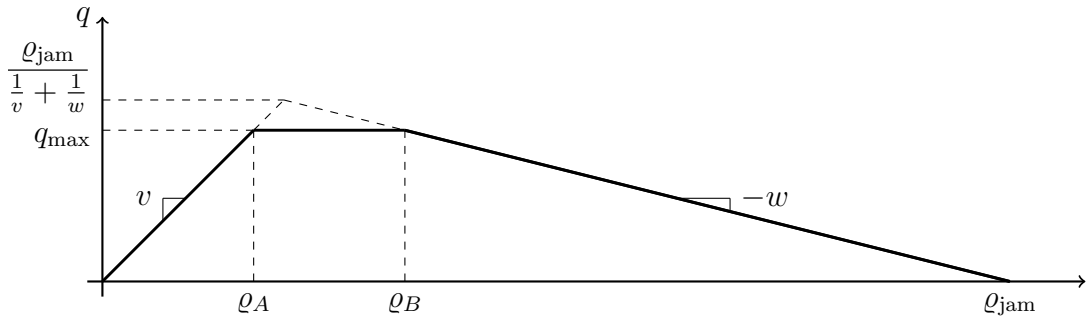


Figure 1: An approximation of the fundamental diagram suggested by Daganzo [3].

where $Q_i[k]$ is the capacity flow into i for time interval t, and $N_i[k] - n_i[k]$ is the amount of empty space in cell i at time step k. Cell occupancies are updated for each step of the clock during the simulation.

4. Experiments

In order to demonstrate the behaviour of all three discussed models, we have tested the prediction capabilities using the data from the southern leg of the Prague Ring (SOKP) section from km 20.1 to km 17.0. We fed the measurements, provided by detectors at km 20.1, as a boundary condition into our models, and used the models to predict the traffic at km 17.0. The predicted data were then compared with the measurements provided by detectors.

The basic parameters for the simulation were the length of a segment $\Delta x = 150\,\text{m}$, and the time step $\Delta t = \Delta x / v_{\text{f}}$. The free flow speed v_{f} has been identified from the measured data as $v_{\text{f}} = 115\,\text{km/h}$, implying $\Delta t \approx 4.7\,\text{s}$. Jam density $\varrho_{\max}$ is given by an average length of a passenger vehicle $d_{\text{avg}} = 6\,\text{m}$ as $\varrho_{\max} = 1000/d_{\text{avg}} = 166\,\text{veh/km}$. Maximum vehicle flow is given by the theoretical speed limit of the highway, which is 130 km/h.

When numerically solving a partial differential equation using a method based on finite differences, a necessary condition of stability of the solution is provided by Courant–Friedrichs–Lewy (CFL) condition [7]. This condition arises if explicit time integration schemes are used for the numerical solution. As a consequence, the time step of such a scheme must be less than a certain time, otherwise the simulation will produce incorrect results. While the rounded time-step $\Delta t = 5\,\text{s}$ is an acceptable value for first-order LWR-type models (even if it violates the CFL condition), integrating an AR-type model with such a large time step does not converge to a plausible solution. The higher-order model unfortunately requires a shorter time step fulfilling the CFL condition. Hence, for an AR-type model, $\Delta t = 0.5\,\text{s}$ has been used.

The results of all three models are compared in Figure 2. We can see that the second-order AR-type model has still issues in following the general trends recogniz-

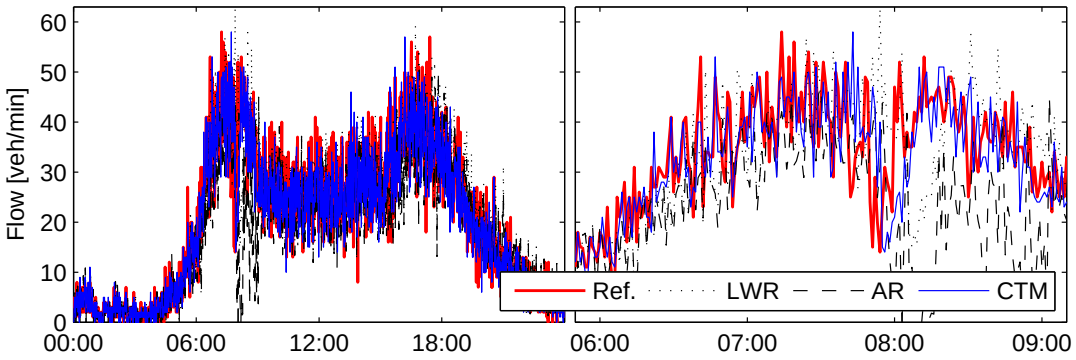


Figure 2: Comparison of predictions of the LWR, AR, and CTM models with the reference flow obtained by measurements. Left: data for one day of traffic, right: the same data between 6:00 and 9:00.

Model	Δt [s]	Steps	Time [s]	MSE	$\max \epsilon_r$ [%]
LWR	5	18017	42	59.58	42%
AR	0.5	180167	1122	80.09	101%
CTM	4.7	18880	25	27.79	14%

Table 1: Comparison of all models on real time data. MSE denotes the mean squared error of the prediction, ϵ_r is the relative prediction error.

able in the traffic data. This is especially visible in the right panel of Figure 2 for times between 8:00 and 9:00. The most probable reason for this anomaly is the higher sensitivity of AR-type models to repetitive changes in the boundary conditions. Most important observation, however, can be found in Table 1 which summarizes the computational times and errors of the models: From the practical point of view the test demonstrates that the AR-type model is almost useless due to the necessary small time-step and resulting long computational time. A simple CTM scheme that resembles cellular automata beats even the simple LWR model in both computational speed and accuracy. Again, our assumption is that the continuous nature of the underlying model is disturbed by the time-variable boundary conditions.

5. Conclusions

We have demonstrated three different traffic flow models and their performance on real-world traffic flow data. Our experiment shows that from the practical point of view, Daganzo's CTM, a simple compartment model based on piecewise linear approximation of the fundamental diagram of traffic flow, provides best results in both accuracy and computational speed. In theory, a continuous higher-order model of AR-type should be able to address traffic phenomena that the CTM is unable to capture, however, the higher order model is significantly less numerically stable. The need for strict fulfillment of the CFL stability condition results in tenfold decrease of the original time-step, rendering the whole model unsuitable for practical application.

The whole Matlab package can be downloaded from the website of the corresponding author at `http://staff.utia.cas.cz/prikryl/panm17.zip`.

Acknowledgments

This work has been supported by the Technology Agency of the Czech Republic under projects no. TA01030603 (NOMŘÍZ) and TA02030522 (SIRID).

References

[1] Aw, A. and Rascle, M.: Resurrection of "second order" models of traffic flow. SIAM J. Appl. Math. **60** (2000), 916–938.

[2] Brandner, M. and Tuma, J.: Modelování dopravního proudu–složitý příběh jednoho typu matematických modelů. PMFA **56** (2011), 106–118.

[3] Daganzo, C. F.: The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological **28** (1994), 269–287.

[4] Daganzo, C. F.: The cell transmission model, part II: Network traffic. Transportation Research Part B: Methodological **29** (1995), 79–93.

[5] Daganzo, C. F.: Requiem for second-order fluid approximations of traffic flow. Transportation Research Part B: Methodological **29** (1995), 277–286.

[6] Gartner, N. H., Messer, C. J., and Rathi, A. K. (Eds.): *Revised Monograph on Traffic Flow Theory*. Federal Highway Administration, Turner-Fairbank Highway Research Center, 2005. URL `http://www.fhwa.dot.gov/publications/research/operations/tft/`.

[7] Larsson, S. and Thomée, V.: *Partial differential equations with numerical methods*, *Texts in Applied Mathematics*, vol. 45. Springer Science & Business, 2008.

[8] Lighthill, M. J. and Whitham, G. B.: On kinematic waves. II. A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **229** (1955), 317–345.

[9] Payne, H. J.: Models of freeway traffic and control. Mathematical models of public systems **21** (1971), 51–61.

[10] Richards, P. I.: Shock waves on the highway. Operations research **4** (1956), 42–51.

[11] Whitham, G. B.: *Linear and non linear waves*. John Wiley, 1974.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

COMPUTATIONAL APPROACHES TO THE DESIGN OF LOW-ENERGY BUILDINGS

Petra Jarošová

Brno University of Technology, Faculty of Civil Engineering
602 00 Brno, Veveří 331/95, Czech Republic
jarosova.p@fce.vutbr.cz

Abstract

European and Czech directives and technical standards, approved in several last years, force substantial changes in thermal behaviour of all buildings, including new and reconstructed one- or more-family houses, block of flats, etc., especially radical decrease of their energy requirements. This stimulates the development of advanced materials, structures and technologies. Since no reliable experience with their design is available, robust and non-expensive computational simulation approaches, compatible with principles of classical thermodynamics, are needed. This paper demonstrates the impact of such requirements on the development of relevant computational algorithms, with the accent on the conception of a building as a thermal system, at various generality levels of analysis of its particular elements and subsystems.

1. Introduction

Phrases like "solar houses", "low-energy houses", "passive houses", with increasing frequency of exploitation in last years, reflect the tendencies to reduce, with help of various non-traditional energy sources, all energy demands of buildings, to their heating and air-conditioning, operation of household equipments, etc. Although i) some ideas of studious utilization of solar radiation can be observed even in the ancient literature and ii) the modern experiments with low-energy houses have their own interesting history, dating back to the first experimental house of the Massachusetts Institute of Technology (1939), most designers of building structures know just iii) the final result of discussion (about 1989) between B. Adamson (Lund University, Sweden) and W. Feist (Institut für Wohnen und Umwelt in Darmstadt, Germany), well-known as the "passive house standard". This result becames 2 decades later, after a lot of practical implementations in various countries, under different climatic conditions, reflected in [6], a part of the European directive [13], as well as of national technical standards, e. g. [15] in the Czech Republic.

Under the Central European conditions, the following requirements follow from [6]: a) the building must be designed to have an annual heating and cooling demand not higher than 15 kWh/m^2 per year, b) the total primary energy consumption (for

Figure 1: Illustrative photos from the Czech Republic, from the left: a) Block of family houses in Židlochovice (2006). b) One-family house "Vějíř" ("Punka") in Brno - Bystrc (2009). c) Cooling towers of the nuclear power plant Dukovany, view from the national nature reserve of xerophilous herbs near the township village Mohelno.

heating, hot water, electricity, etc.) cannot exceed 120 kWh/m^2 per year, c) the building with the total volume V must not leak more air than 0.6 V per hour at pressure 50 Pa, as tested by the blower door, d) as an non-obligatory (unlike a), b) and c)) additional condition: the power requirement for heating under the lowest considerable environmental temperature (typically $-12\,^\circ$C) should not exceed 10 W/m^2. By [6], i) passive solar building design and energy-efficient landscaping, ii) superinsulation and elimination of line and surface "thermal bridges" (locations of massive thermal losses), iii) advanced window technology, iv) airtightness, v) heat recovery ventilation systems, vi) space heating utilizing solar energy and heat pumps and vii) passive and active daylighting techniques and electrical appliances with ecolabel certification marks are needed to reach a), b), c), d). Two examples of such houses of various types are shown on Fig. 1 a), b).

More advanced thermal considerations can be found in the literature in the last decade: e. g. [9] takes even the solar radiation absorbed by bodies of inhabitants into account. However, some critical comments cannot be neglected: installing and maintaining a passive solar energy system is rather expensive, its performance depends strongly on the climate, etc. Even the total economical and ecological benefits may be not clear, namely in comparison with other projects: e. g. the heat pipeline from the nuclear power plant in Dukovany to Brno, designed 1985 (for the urban area with 500 000 inhabitants, 41 km long, working with water and water vapour at the temperature between 56 and 142 $^\circ$C), has never been brought into effect – the waste vapour emits into the surroundings, including the national nature reserve, cf. Fig. 1 c).

2. Modelling of thermal transfer in buildings

From the pragmatical point of view, building designers, respecting [13] and [15] (which do not contain any computational formulae) need not to discuss advantages and drawbacks of the "passive house standard", applied to all new and reconstructed buildings, but must be interested in its practical implementation. Evidently, the

strict requirements of [13] and [15] stimulate the development of advanced materials, structures and technologies. Since no reliable experience with their design is available, reliable and non-expensive computational simulation approaches are needed.

A rather general and transparent approach can be based on the principles of classical thermodynamics, namely on the conservation of scalar quantities $u(x,t)$ on a domain Ω in the Euclidean space R^3 and in the time interval I (t here denotes the non-negative time, $x = (x_1, x_3, x_3)$ the Cartesian coordinate system in R^3, $\partial\Omega$ means the boundary of Ω in R^3, supplied by the unit local formally outward normal $n(x) = (n_1(x), n_2(x), n_3(x))$, dot symbols are reserved for partial time derivatives) with internal volume sources f on $\Omega \times I$ and external surface sources g on $\Gamma \times I$ where $\Gamma \subseteq \partial\Omega$, later also $\Theta = \partial\Omega \setminus \Gamma$, i. e.

$$\begin{array}{rcll} \dot{\varepsilon}(u) - \nabla\eta(u) & = & f & \text{on } \Omega \times I \\ \eta(u) \cdot n & = & g & \text{on } \Gamma \times I \\ \eta(u) \cdot n & = & \psi(u, u_*) & \text{on } \Theta \times I\,; \end{array} \tag{1}$$

here all values u_* must be prescribed on $\Theta \times I$, together with an appropriate transfer function ψ. For simplicity, let us assume the initial condition $u(.,0) = 0$; a simple transform makes it possible to get such initial problem from anyone corresponding to the initial equilibrium. Moreover, (1) contains evolutionary (enthalpic) terms $\varepsilon(u)$ and fluxes $\eta(u)$ (3 components), whose evaluation relies on some reasonable (usually empirical) constitutive relations of the Fourier, Fick, Newton, etc. types, both corresponding to scalar quantities u. In particular, under the assumption of (at least macroscopic) material homogeneity and isotropy, it is possible to write the linearized relations

$$\begin{array}{rcll} \eta(u) & = & -\nabla\beta(u) & \text{on } \Omega \times I\,, \\ \beta(u) & = & \lambda u & \text{on } \Omega \times I\,, \\ \varepsilon(u) & = & \kappa u & \text{on } \Omega \times I\,, \\ \psi(u, u_*) & = & \gamma(u - u_*) & \text{on } \Theta \times I\,, \end{array} \tag{2}$$

with certain constants κ, λ and γ. For instance, in the case of (thermal) energy balance u is usually considered as the (absolute) temperature. Moreover, in (2) κ refers to the thermal capacity (related to the unit volume), λ to the thermal conductivity and γ to some interface heat transfer coefficient; all values u_* should be known from the environment, from the adjacent building component, etc. Let us notice that the first relation of (2), respected in this paper everywhere, forces the potential problem, with zero rotation of $\eta(u)$; this needs to be generalized namely in the analysis of air or moisture flow in rooms and structures, as sketched in [7]. Most computational approaches make use of the weak formulation for some appropriate function space V, typically the Sobolev space $W^{1,2}(\Omega)$ or its subspace: to find such abstract function u, mapping I to V, that

$$(\dot{\varepsilon}(u), v) + (\nabla\beta(u), \nabla v) = (f, v) + \langle g, v\rangle_\Gamma + \langle \psi(u, u_*), v\rangle_\Theta \quad \text{on } I\,; \tag{3}$$

here $(.,.)$ in the simplest case refer to scalar products in $L^2(I, L^2(\Omega))$ or $L^2(I, L^2(\Omega)^3)$,

Figure 2: Photos of crucial details for deterioration of thermal properties of buildings, from the left: a), b) Imperfect connections of particular components. c), d) Moisture condensation on exterior surfaces.

$\langle .,. \rangle_\Theta$ and $\langle .,. \rangle_\Gamma$ to those in $L^2(I, L^2(\Theta))$ and $L^2(I, L^2(\Gamma))$, which can be modified in the sense of dualities in more general spaces.

The usual choices of u in (1), (2) and (3) are: i) the temperature, ii) the (air, material, ...) density, iii) the components of velocity of the motion (related to some appropriate reference configuration); these choices correspond to the conservation of i) energy, ii) mass and iii) (linear and angular) momentum from classical thermodynamics by [1]; f and g from (1) are allowed to be applied to coupling of these approaches, e. g. for the study of simultaneous heat and moisture transfer (i. e. energy and mass balance). An interesting generalization can be found in [5]: in addition to the first thermodynamical principle it takes into account also the second one, involving some entropy considerations, working with so-called "exergy"; however, many users of this term understand "exergy" not in a transparent physical sense, but as certain trinity of i) energy, ii) environment and iii) sustainable development. Moreover, some authors, like [3], advert to the priority of the comfort of individual users (just on the example from Fig. 1 a)), as well as to the quality of the architecture of particular buildings and of the whole urban area, which can be frequently in contradiction with any economical optimization.

3. Practical evaluation of energy consumption

The evaluation of energy consumption of a building from (3) (more precisely: from its finite-dimensional discrete version in practice) seems to be easy now (any thermal flux can be evaluated from the complete information on the temperature development), but hides some difficulties: i) (3) describes one domain (as a constructive or insulation layer, a room, etc., in a building) formally, but the whole building is composed from a high number of such domains, whose mutual interaction relies on the last additive term in (3), ii) advanced building design, from the mathematical point of view, is a complicated optimization problem, seeking for the minimal energy consumption, under a rather large number of conditions coming from technical standards, as those of obligatory temperature levels and temperature stability in rooms – cf. [12]. The list of building energy software [14] contains 417 items now; nevertheless, all of them work with strong (often non-transparent) simplifications – cf. [2].

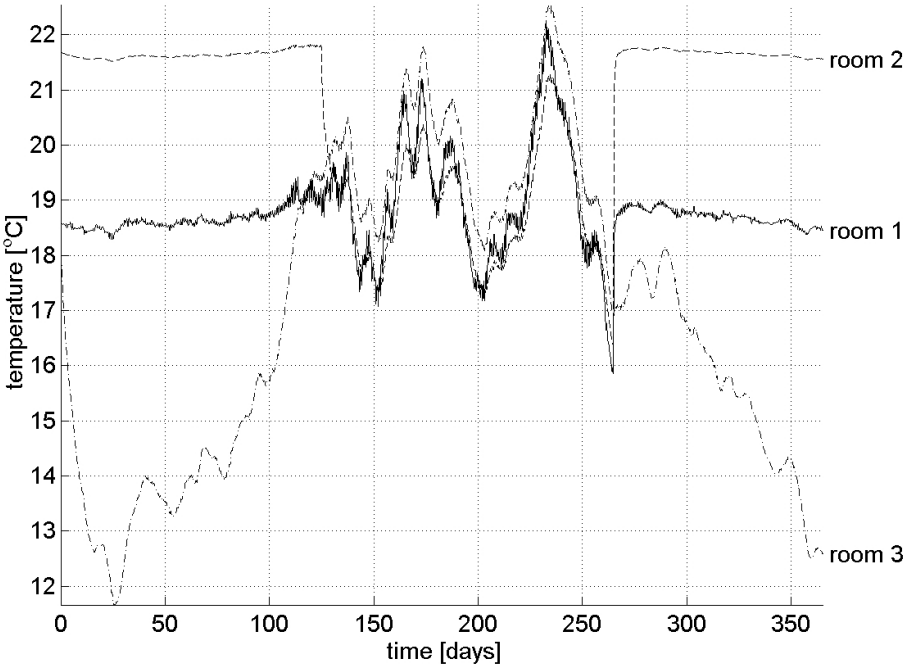


Figure 3: Computational evaluation of the annual energy demand of a low-energy family house during a typical climatic year: temperature development in selected rooms (room 1: full line, room 2: dashed line, room 3: dash-dotted line).

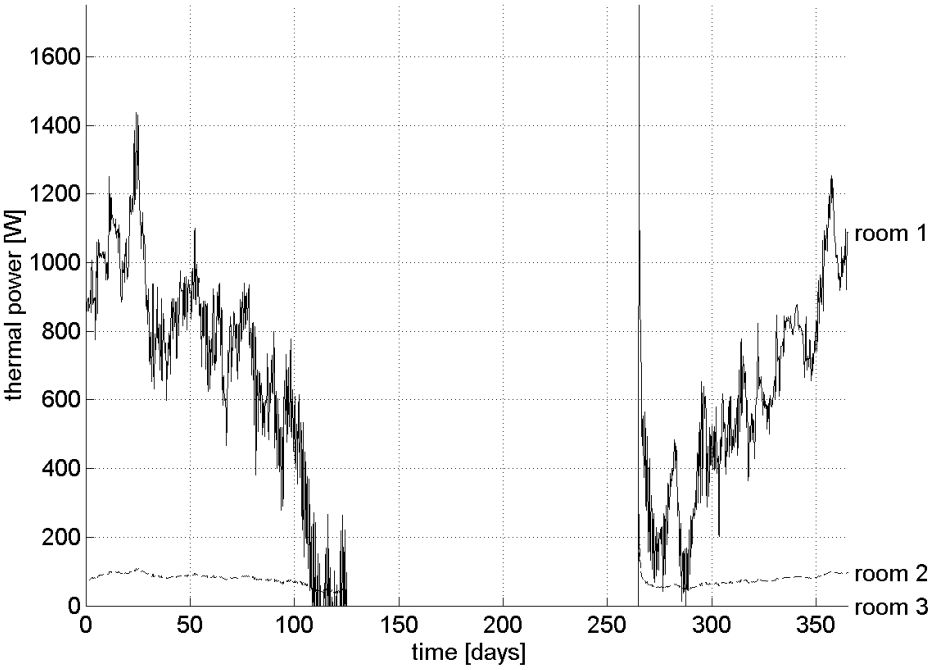


Figure 4: Requirements to artificial heating, corresponding to Fig. 3

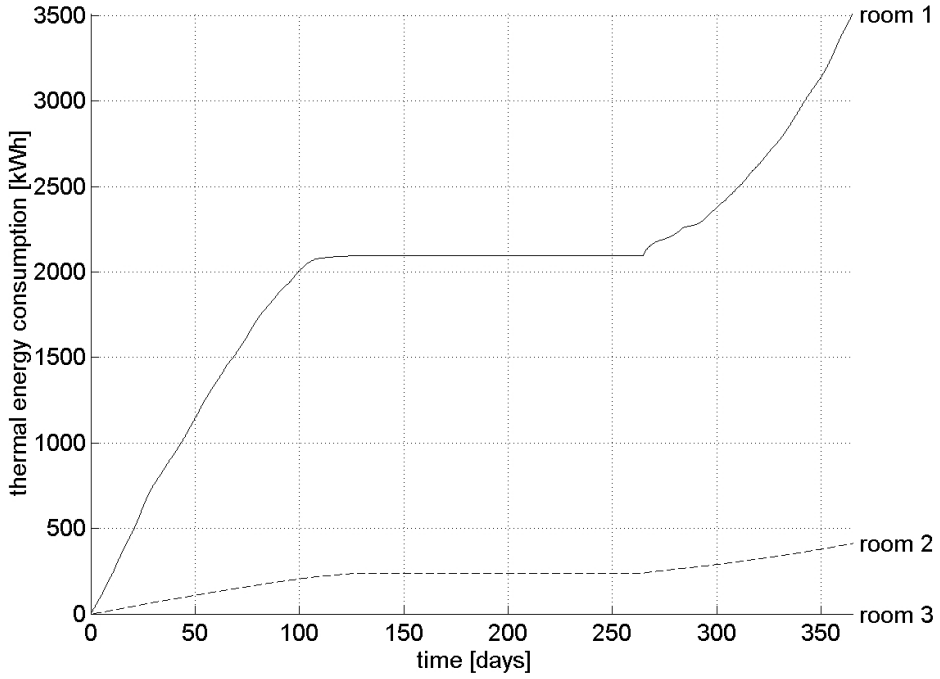


Figure 5: Cumulative energy consumption, related to Fig. 4

The non-commercial software package *Therm Stabil*, developed at the Institute of Technology of Building Materials and Components of Brno University of Technology, Faculty of Civil Engineering (not included in [14]), in the programing language Pascal in the Delphi environment, applies the system approach, taking rooms, walls, roofs, etc., as building elements and subsystems, connected by thermal (and some other physical) fluxes, analyzing (3) using the finite element, volume and difference techniques together with the Rothe sequences (for time discretization). This package is still in progress; its development is a part of research of advanced building materials and their utilization in structures and technologies, as presented at *http://www.fce.vutbr.cz/thd*. Interested reader may request more detailed information from the author of this paper or from the principal author of *Therm Stabil*, Prof. Stanislav Šťastník (e-mail *stastnik.s@fce.vutbr.cz*).

All numerical results presented in this paper have been obtained from this software, except the post-processing for Fig. 3,4,5, prepared in the MATLAB environment. The principal heat fluxes to particular rooms considered in *Therm Stabil* come i) from adjacent building components, ii) through windows, doors, etc., including solar radiation, iii) thanks to the air exchange, iv) from artificial sources of heating and / or air conditioning. Moreover, Fig. 2 demonstrates other effects observed in real building structures, namely so-called potential "thermal bridges" on connections of different building components (photos a), b)), as well as moisture condensation on exterior surfaces with unpleasant consequences of algae population, as discussed in [10] and [11] (photos c), d)).

Fig. 3 shows some results of the computational simulation of thermal fluxes in a new low-energy family house (not passive by the definition of [6], in a village close to Brno), based on the proper description of its location, orientation and composition and on the detailed knowledge of annual climatic data for Brno. Averaged temperatures in 3 typical rooms (from 8 in total: rooms 1 and 2 contain heating equipments of different type, room 3 is heated indirectly) are modulated in the (rather long) winter period by the artificial heating; no air conditioning is applied. Such requirements to artificial heating by Fig. 4 generate the cumulative thermal energy consumption for the whole building, as evident from Fig. 5; no other energy demands (as those from other electrical appliances) are taken into considerations. The measured energy consumption in several first years of existence of this building (and 3 other tested ones) is slightly higher – however, this depends also on the user habits and priorities in heating, ventilation, etc.

The same software package has been applied, inspired by [4], to the simulation of energy consumption in an existing freezing plant in Central Moravia. One could expect much more significant energy reduction, in such an industrial building (containing the freezing space at –24 °C, several offices, etc.), thanks to its sophisticated design, than in a family house. Unfortunately, in this case, described in [8], all available energy consumption data are higher than those predicted by simulation. The a posteriori analysis in situ showed some imperfect connections of building components and the presence of moisture in polyurethane insulation layers (theoretically dry, following the technical standard), as the probable immediate cause of deterioration of their thermal properties.

4. Conclusion

Energy reduction in building structures is a challenge of last two decades, seen by various authors from ecological, engineering, physical, mathematical and computational points of view. As a reasonable compromise between the traditional stationary evaluations of thermal resistances, improper in advanced structures, and complicated models referring to large systems of partial differential equations of evolution, this paper offers an alternative system approach namely to the computational evaluation of energy for heating and air-conditioning; the system complexity can be reduced here thanks to transparent simplifications, compatible with classical thermodynamics. More advanced models (e. g. those containing involving air and moisture flow, driven by Navier - Stokes equations), up to now, suffer from expensive computations and bad correlation between the results of deterministic calculations and available experimental data. However, due to its social significance, the further research is very desirable.

Acknowledgements

This work was supported by the project No. FAST-J-14-2296 of the specific university research at Brno University of Technology.

References

[1] Bermúdez de Castro, A.: *Continuum thermomechanics*. Birkhäuser, Basel, 2005.

[2] Crawley, D. B.: Contrasting the capabilities of building energy performance simulation programs. Building and Environment **43** (2008), 661–673.

[3] Chybík, J.: Interní prostředí pasívního domu v Židlochovicích. In: *Budovy a prostredie – trvalo udržateľná výstavba*, pp. 40–43. Slovak Technical University, Bratislava, 2008. (In Czech.)

[4] Delgado, A. E. and Sun, D.-W.: Heat and mass transfer models for predicting freezing processes – a review. Journal of Food Engineering **47** (2001), 157–174.

[5] Dincer, I.: Exergy – cost – energy – mass analysis of thermal systems and processes. Energy Conversion and Management **44** (2003), 1633–1651.

[6] Feist, W.: *Gestaltungsgrundlagen Passivhäuser*. Das Beispiel, Darmstadt, 1999. (In German.)

[7] Jarošová, P. and Šťastník, S.: Modelling of thermal transfer for energy saving buildings. In: *ICNAAM – International Conference on Numerical Analysis and Applied Mathematics*, pp. 1000–1003. American Institute of Physics, Melville, 2013.

[8] Jarošová, P. and Šťastník, S.: Numerical prediction of energy consumption in buildings with controlled interior temperature. In: *ICNAAM – International Conference on Numerical Analysis and Applied Mathematics*, 4 pp. American Institute of Physics, Melville, 2014, , accepted for publication.

[9] Otto, F.: Einfluss der vom menschlichen Körper absorbierten Solarstrahlung für das Wärmeempfinden in Gebäuden. Bauphysik **31** (2009), 25–37. (In German.)

[10] Siwińska, A. and Grabalińska, H.: Thermal conductivity coefficient of cement-based mortars as air relative humidity function. Heat and Mass Transfer **49** (2011), 1077–1087.

[11] Steuer, R., Šťastník, S., Vala, J., Korjenic, A., and Bednar, T.: Beitrag zur Lösung des Problems der Algenbildung auf Aussenwänden mit Wärmedämmverbundsystemen (WDVS). Bauphysik **31** (2009), 343–353. (In German.)

[12] Šťastník, S. and Vala, J.: On the thermal stability in dwelling structures. Building Research Journal **52** (2004), 31–55.

[13] Directive 2010/31/EU of the European Parliament and of the Council on the energy performance of buildings. Official Journal of the European Union **L 153/13** (2010).

[14] US Department of Energy: *Building Energy Software Tools Directory*. Available at http://apps1.eere.energy.gov/buildings/tools_directory/, 2014.

[15] Vyhláška Ministerstva průmyslu a obchodu č. 78 o energetické náročnosti budov. Sbírka zákonů České republiky **36** (2013). (In Czech.)

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

NUMERICAL MODELLING OF VISCOUS AND VISCOELASTIC FLUIDS FLOW THROUGH THE BRANCHING CHANNEL

Radka Keslerová, Karel Kozel

CTU in Prague, Faculty of Mechanical Engineering,
Department of Technical Mathematics
Karlovo nám. 13, 121 35 Prague, Czech Republic
Radka.Keslerova@fs.cvut.cz, Karel.Kozel@fs.cvut.cz

Abstract

The aim of this paper is to describe the numerical results of numerical modelling of steady flows of laminar incompressible viscous and viscoelastic fluids. The mathematical models are Newtonian and Oldroyd-B models. Both models can be generalized by cross model in shear thinning meaning.

Numerical tests are performed on three dimensional geometry, a branched channel with one entrance and two output parts. Numerical solution of the described models is based on cell-centered finite volume method using explicit Runge–Kutta time integration. Steady state solution is achieved for $t \to \infty$. In this case the artificial compressibility method can be applied.

1. Introduction

The flows in the branching channel are encountered in technical sector as well as in biomedical applications. It is to be in human body in the complex branching system of blood vessels. Therefore the numerical modelling of generalized Newtonian and generalized Oldroyd-B fluids flow is very important for medical science. For the viscoelastic character of blood, the blood flows is numerically simulated by Oldroyd-B mathematical model with generalizing by cross model.

Therefore this work is concerned with the numerical solution of generalized Newtonian and generalized Oldroyd-B fluids flow in the branched channel with T-junction with round cross-section.

2. Mathematical model

The fundamental system of equations is the system of generalized Navier–Stokes equations for incompressible fluids. This system is based on the system of balance laws of mass and momentum for incompressible fluids

$$\operatorname{div} \boldsymbol{u} = 0 \tag{1}$$

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}.\nabla)\boldsymbol{u} = -\nabla P + \text{div } \mathbf{T} \tag{2}$$

where P is the pressure, ρ is the constant density, $\boldsymbol{u}$ is the velocity vector. The symbol $\mathbf{T}$ represents the stress tensor.

For the different choice of mathematical model the different definition of the stress tensor is used. For viscous flows with the representative of Newtonian fluids the Newtonian model is considered (see e.g. [1], [2])

$$\mathbf{T} = 2\mu\mathbf{D} \tag{3}$$

where μ is the dynamic viscosity and tensor $\mathbf{D}$ is the symmetric part of the velocity gradient.

In the case of viscoelastic fluids, the simplest viscoelastic model can be used. This model is denoted as *Maxwell model*

$$\mathbf{T} + \lambda_1\frac{\delta \mathbf{T}}{\delta t} = 2\mu\mathbf{D} \tag{4}$$

where λ_1 is the relaxation time. The symbol $\frac{\delta}{\delta t}$ represents upper convected derivative.

By combination of two mathematical models (Newtonian and Maxwell) the behaviour of mixture of viscous and viscoelastic fluids can be described. This model is called Oldroyd-B model and it has the form

$$\mathbf{T} + \lambda_1\frac{\delta \mathbf{T}}{\delta t} = 2\mu\left(\mathbf{D} + \lambda_2\frac{\delta \mathbf{D}}{\delta t}\right). \tag{5}$$

where symbols λ_1 is relaxation time and λ_2 is the retardation time (with dimension of time).

In the system of equations (1) and (2) is on the right hand side the stress tensor $\mathbf{T}$ which can be decomposed to the Newtonian (viscous) part $\mathbf{T}_s$ and viscoelastic part $\mathbf{T}_e$. The tensor $\mathbf{T}_s$ is defined by Newtonian model (3) and the viscoelastic tensor $\mathbf{T}_e$ is defined by Maxwell model (4)

$$\mathbf{T}_s = 2\mu_s\mathbf{D}, \qquad \mathbf{T}_e + \lambda_1\frac{\delta \mathbf{T}_e}{\delta t} = 2\mu_e\mathbf{D}, \tag{6}$$

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 $\frac{\delta}{\delta t}$ used in the viscoelastic part of the stress tensor is defined by the relation, for more details see [1]

$$\frac{\delta \mathbf{T}_e}{\delta t} = \frac{\partial \mathbf{T}_e}{\partial t} + (\boldsymbol{u}.\nabla)\mathbf{T}_e - (\mathbf{W}\mathbf{T}_e - \mathbf{T}_e\mathbf{W}) - (\mathbf{D}\mathbf{T}_e + \mathbf{T}_e\mathbf{D}) \tag{8}$$

where $\mathbf{D}$ is symmetric part and $\mathbf{W}$ is antisymmetric 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

$$\mathbf{W} = \frac{1}{2}(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^T) = \frac{1}{2}\begin{pmatrix} 0 & u_y - v_x & u_z - w_x \\ v_x - u_y & 0 & v_z - w_y \\ w_x - u_z & w_y - v_z & 0 \end{pmatrix}. \tag{10}$$

These mathematical models for the stress tensor could be generalized. For this case the viscosity is considered as a viscosity function and it's defined by shear-thinning cross model (for more details see [7])

$$\mu(\dot{\gamma}) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (\lambda\dot{\gamma})^b)^a}, \quad \dot{\gamma} = 2\sqrt{\frac{1}{2}\text{tr } \mathbf{D}^2} \tag{11}$$

with special parameters $\mu_0 = 1.6 \cdot 10^{-1}$ Pa.s, $\mu_\infty = 3.6 \cdot 10^{-3}$ Pa.s, $a = 1.23$, $b = 0.64$, $\lambda = 8.2$ s.

3. Numerical solution

The system of equations (1),(2) is solved by the artificial compressibility method, see [3, 4]). In its simplest form, only the continuity equation is modified by the first term in the following equation

$$\frac{1}{\beta^2}\frac{\partial p}{\partial t} + \text{div } \boldsymbol{u} = 0 \tag{12}$$

where β is positive parameter. The inviscid part of modified Navier–Stokes equations is now strongly hyperbolic and can therefore be solved by standard methods for hyperbolic conservation laws. The system including the modified continuity equation and the momentum equations can be written

$$\tilde{R}_\beta W_t + F^c_x + G^c_y + H^c_z = F^v_x + G^v_y + H^v_z + S, \quad \tilde{R}_\beta = \text{diag}(\frac{1}{\beta^2}, 1, \cdots, 1) \tag{13}$$

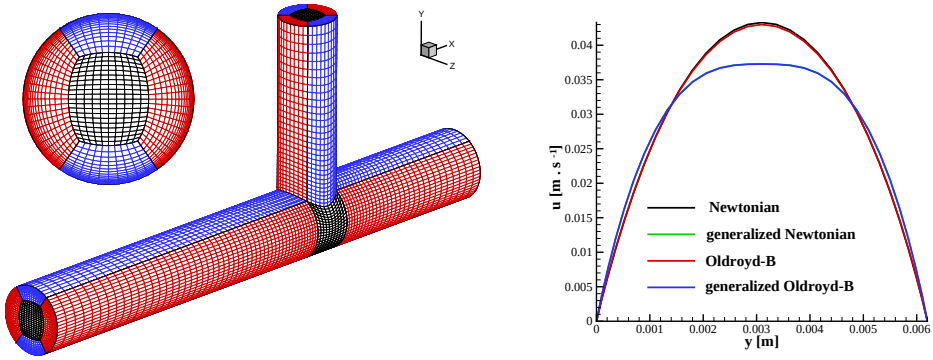
where W is vector of unknowns, $W = (p, u, v, w, t_{e1}, \ldots, t_{e6})$, F^c, G^c, H^c and F^v, G^v, H^v are inviscid and viscous fluxes and S denotes the source term.

Eq. (13) is discretized in space by the finite volume method and the arising system of ODEs is integrated in time by the explicit multistage Runge–Kutta scheme ([5, 6]).

The flow is modeled 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.

4. Numerical results

This section deals with the comparison of the numerical results of generalized Newtonian and generalized Oldroyd-B fluids flow. Numerical tests are performed in an idealized branched channel with the circle cross-section. Fig. 1 (left) shows the shape of the tested domain. The computational domain is discretized using a structured, wall fitted mesh with hexahedral cells. The domain is divided to 19 blocks with 125 000 cells.



(a) structure of the domain (b) axial velocity profile

Figure 1: Structure of the computed domain (left) and axial velocity profile for steady fully developed flow of tested fluids (right)

As initial condition the following model parameters are used: $\mu_e = 0.0004\,\text{Pa.s}$, $\mu_s = 0.0036\,\text{Pa.s}$, $\lambda_1 = 0.06\,\text{s}$, $U_0 = 0.0615\,\text{m.s}^{-1}$, $L_0 = 0.0031\,\text{m}$, $\rho = 1050\,\text{kg.m}^{-3}$. Using these data, fully developed Poiseuille velocity profile (for Newtonian fluid) is prescribed at the inlet (Dirichlet condition). At the outlet homogeneous Neumann conditions for the velocity components and a constant pressure are prescribed (0.0005 Pa (main channel) and 0.00025 Pa (branch)). On the vessel walls no-slip homogeneous Dirichlet conditions are prescribed for the velocity field. In the case of the Oldroyd-B and generalized Oldroyd-B models, homogeneous Neumann conditions are imposed for the components of the extra stress tensor at all boundaries. In Fig. 1 (right) the axial velocity profile for fully developed flow close to the branching is shown. The lines for Newtonian and Oldroyd-B fluids are similar to the parabolic line, as was assumed. From this velocity profile is clear that the shear thinning fluids attain lower maximum velocity in the central part of the channel (close to the axis of symmetry) which is compensated by the increase of local velocity in the boundary layer close to the wall. In Fig. 2 the velocity isolines and the cuts through the main channel and the small branch for Newtonian fluids are shown.

The axial velocity isolines for all tested fluids are shown in the Figure 3. It can be observed from Fig. 3 that the size of separation region for generalized Newtonian and generalized Oldroyd-B fluids is smaller than for Newtonian and Oldroyd-B fluids.

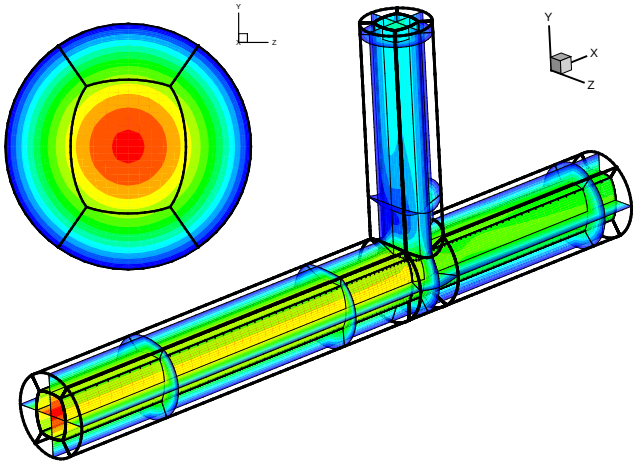
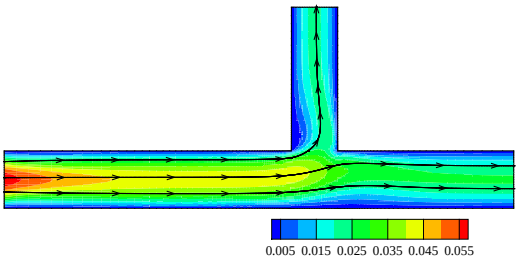
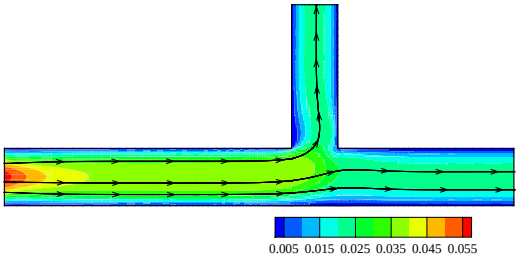


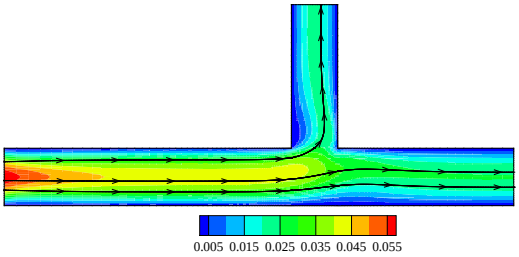
Figure 2: Velocity isolines of steady flows for Newtonian fluids



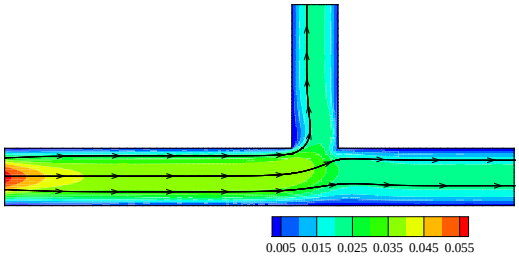
(a) Newtonian



(b) generalized Newtonian



(c) Oldroyd-B



(d) generalized Oldroyd-B

Figure 3: Axial velocity isolines in the center-plane area

5. Conclusion

In this paper a finite volume solver for incompressible laminar viscous and viscoelastic flows in the branching channel with T-junction and circle cross section was described. Newtonian and Oldroyd-B fluids models were generalized by the cross model for numerical solution of generalized Newtonian and Oldroyd-B fluids flow. The explicit Runge-Kutta method was considered for time integrating.

The numerical results obtained by this method were presented and compared. In the case of steady flow in this type of the 3D branching channel the numerical results for Newtonian and Oldroyd-B fluids are similar. Future work will be devoted to an extension of this numerical study to the unsteady simulation.

Acknowledgements

This work was supported by grant SGS13/174/OHK2/3T/12 of the Czech Science Foundation.

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** (2010), 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. Adv. Math. Fluid Mech. (2010), 83–104.

[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*. CTU in Prague, Czech Republic, 1996.

[5] Keslerová, R. and Kozel, K.: Numerical modelling of incompressible flows for Newtonian and non-Newtonian fluids. Math. Comput. Simulation **80** (2010), 1783–1794.

[6] LeVeque, R.: *Finite-volume methods for hyperbolic problems*. Cambridge University Press, 2004.

[7] 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 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

PARALLELIZATION OF ARTIFICIAL IMMUNE SYSTEMS USING A MASSIVE PARALLEL APPROACH VIA MODERN GPUS

Jiří Khun, Ivan Šimeček

Department of Computer Systems,
Faculty of Information Technology,
Czech Technical University in Prague,
Thákurova 9, 160 00 Prague 6, Czech Republic
jiri.khun@fit.cvut.cz, ivan.simecek@fit.cvut.cz

Abstract

Parallelization is one of possible approaches for obtaining better results in terms of algorithm performance and overcome the limits of the sequential computation. In this paper, we present a study of parallelization of the opt-aiNet algorithm which comes from Artificial Immune Systems, one part of large family of population based algorithms inspired by nature. The opt-aiNet algorithm is based on an immune network theory which incorporates knowledge about mammalian immune systems in order to create a state-of-the-art algorithm suitable for the multimodal function optimization. The algorithm is known for a combination of local and global search with an emphasis on maintaining a stable set of distinct local extrema solutions. Moreover, its modifications can be used for many other purposes like data clustering or combinatorial optimization. The parallel version of the algorithm is designed especially for modern graphics processing units. The preliminary performance results show very significant speedup over the computation with traditional central processor units.

1. Introduction

Research behind this paper represents an intersection of several scientific disciplines but two of them are playing a key role: artificial immune systems (AIS) and design of parallel algorithms with an emphasis on general purpose computing via graphic processing units (GPU).

1.1. Artificial immune systems

Artificial immune systems are a part of a large field of computational intelligence approaches inspired by nature. Their basic principles are based on the knowledge obtained by studying real biological immune systems, especially mammalian.

In general, we can imagine any biological immune system as a mechanism which responses on varied incoming threats represented by pathogens and toxic substances in order to protect the host organism. Pathogens can be represented by a wide

range of different types of micro-organisms like parasites, bacteria, viruses, prions and others. Every immune system's main task is to detect such threats and try to eliminate them.

After ages, the mammalian immune system has developed itself into a very complex part of a host body and the development is still in progress based on everyday life experiences. This is the main reason why the immune systems became an inspiration for the computational intelligence area.

Theories behind the biological immune systems have become an inspiration for many algorithms like Clonal Selection, Negative Selection, Danger Theory, Theory of Immune Network and others [1]. Our research is focused on the Immune Network algorithms which enhanced an original theory of the clonal selection.

1.2. General computing using graphics processing unit

Almost every modern GPU is able to provide a general purpose computational performance at least an order of magnitude bigger than a present-day central processing unit (CPU). In the area of high performance computing (HPC) and supercomputers, the use of GPUs is a standard way to achieve a significantly higher performance than delivered by previous generation computers while keeping the same power consumption.

Great performance hidden in GPUs is achieved by large amount of simple processing elements working in parallel. Every individual element deals only with a small subset from the running task. Therefore, it is necessary to carefully design an algorithm for this parallel approach. But there are also many tasks that cannot be solved in parallel at all due to their sequential nature. Parallel approach has to deal with several aspects like data hazards or synchronization and, consequently, is increasing algorithm complexity.

2. Opt-aiNet

The opt-aiNet algorithm [2] is based on the theory of Immune Network which came with an idea that immune cells do not react only to foreign pathogens but can also react to other immune cells, see Algorithm 1. Thanks to this, the whole immune environment becomes a dynamic self-regulated network where individual cells constantly excite and inhibit each other. This behavior also leads to a richly diverse population of immune cells capable to react to a broad spectrum of possible threats.

An original version of the aiNet algorithm was intended for data clustering and was later extended to deal with optimization tasks. In our research, we are focusing on the version for continuous multi-modal optimization, but results of the research will be applicable across all modifications of the algorithm.

Figure 1 shows an example of the algorithm's results: identified local maxima of Schaffer's function of two real-valued variables.

Algorithm 1 Pseudocode for the opt-aiNet aglorithm [1]

```
1: procedure OPT-AINET
Input: PopulationSize, ProblemSize, Nclones, Nrandom, AffinityThreshold
Output: BestCell
2:     Population ← InsertInitialPopulation(PopulationSize, ProblemSize);
3:     while not(TerminationCondition) do
4:         EvaluatePopulation(Population);
5:         BestCell ← GetBestSolution(Population);
6:         Progeny ← (nothing)
7:         AvgPopFitness ← 0;
8:         while CalculateAvgPopFit(Population) > AvgPopFitness do
9:             AvgPopFitness ← CalculateAvgPopFit(Population);
10:            for Cell(i) in Population do
11:                Clones ← CreateClones(Celli, Nclones);
12:                for Clone(i) in Clones do
13:                    Clone(i) ← MutateAccordingFitnessParent(Clone(i), Cell(i));
14:                EvaluatePopulation(Clones);
15:                Progeny ← GetBestSolution(Clones);
16:    SupressLowAffinityCells(Progeny, AffinityThreshold);
17:    Progeny ← CreateRandomCells(Nrandom);
18:    Population ← Progeny;
19:    return BestCell;
```

3. Analysis of parallelization

Design of parallel algorithms for general purpose GPU computations (GPGPU) is not always a straightforward and simple approach. Especially for tasks that aren't completely data parallel. There are several rules that must be met otherwise the computation performance can be even much lower that on a CPU.

The most important rule is full utilization of GPU resources. It is necessary to run thousands or more independent computational threads. Only such amount of threads can hide some problematic architectural areas like relatively big latency between GPU's cores and their memory.

Another limitation during the design of the parallel algorithm for GPU is the fact that individual threads are run in groups containing tens of threads (usually 64). The threads within a group are using the same instruction buffer and must perform the same program's code. Therefore if any of the threads is branching, the computation must be serialized with a significant negative impact on the performance.

3.1. Parallelization of opt-AiNet

The opt-aiNet algorithm is relatively complex and contains a lot of data dependencies. Therefore it is not reasonable to run it in parallel like a one piece of code.

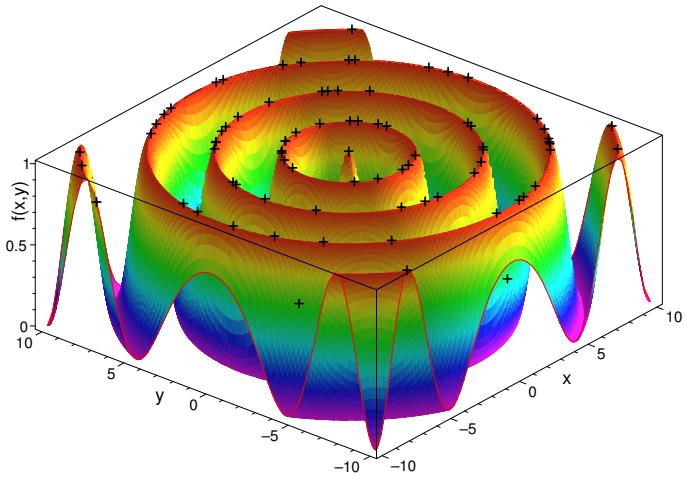


Figure 1: An example of results of multimodal search performed by opt-AiNet on a testing function of two real-valued variables (Schaffer's function). Individual marks represent the found maxima of the function. Redrawn from [2].

During our analysis we discovered 6 individual parallel regions that can be effectively parallelized in a large-scale satisfying the above-mentioned requirements for the efficient GPU computation. The parallel regions will be discussed in the following subsection in detail.

Not all parallel regions are purely data-parallel. Some of them contain internal data dependencies that require intra-thread communication (e.g., via special so-called *shared* memory) and synchronization. Some regions also represent a parallel reduction pattern that require a lot of intra-kernel synchronization.

The thread synchronization and communication represent the biggest challenge during the parallelization because GPU threads are sensitive to synchronization methods and a wrong approach can devastate the overall performance.

3.2. Parallel regions within the opt-AiNet algorithm

As mentioned above, our analysis showed that the algorithm can be divided into several parts with a potential for the large parallelization targeting GPU. These parallel regions are covering almost all necessary steps that must be processed during the algorithm's execution.

3.2.1. Insertion of an initial or an additional population

The insertion of an initial population, consisting of individual immune cells, is the first step of the algorithm. Another cells are also inserted during later stages of the algorithm.

The insertion can be fully done in parallel without any significant obstacle. Every computational thread will insert one or more immune cells and there is not any relation between inserted cells.

It is important to highlight that every computational thread need to have an own independent random number generator because, for proper results of the algorithm, every cell has to be generated with completely random initial configuration values.

3.2.2. Calculation of function values

In this step every cells' internal configuration is used for the calculation of the functional value of the optimized function. The calculation of the function values can be done fully in parallel and in any order because there are not any direct bindings between individual cells during this step.

3.2.3. Calculation of fitness values

The fitness value represents how close the particular immune cell is to the currently best found solution (the maxima of the optimized function). It is an important value that influences another life span of the particular cell (solution).

The calculation of the fitness value is not a data parallel task because at the beginning it is necessary to found the largest function value across all cells in the population. The parallel reduction pattern can be used for this approach. A complication within this approach is the out of order execution of the groups of threads (as mentioned above) on GPUs that requires intergroup synchronization.

3.2.4. Cloning and mutation of the population

In this step, the individual immune cells are cloned and mutated with certain probability. The probability is based on the fitness value due to the fact that the cells with better fitness values have higher chance to clone themselves.

This leads to a varying size of the population that represents another problem for a real implementation of the algorithm on a GPU because current program model needs to specify memory regions in advance before computation.

On the other hand, the mutation as a subsequent step after the cloning does not represent a difficult task and can be done fully in parallel assuming independent random number generator for every computational thread.

3.2.5. Selection of the best cells

During every iteration of the opt-aiNet algorithm, a proportional part of cells is selected for transfer to the next generation. In general, only the best solutions are chosen therefore the algorithm needs to be able to search the whole population in parallel and select them. Logical approach is to sort all cells by the fitness value and then select the part of the cells with required quality.

This can be done in parallel with the help of a parallel sort pattern. There are several types of the parallel sort algorithm suitable for GPU implementation. For example the merge sort.

3.2.6. Suppression of similar cells

This step represents the part of the algorithm where low affinity cells with fitness lower than the required level are suppressed in order to avoid situations where too

many cells within a population are covering the same state space. It is the most challenging part in the whole parallelization processes of the opt-aiNet algorithm because every cell in the population must be compared against the rest.

Our current approach is to sort all cells in parallel by their affinity with the help of the parallel merge pattern. Then we do the suppression on a local level within individual groups of threads (every computational thread represents one cell). The last step is to apply the suppression to the individual groups of threads on a global level.

4. Conclusion

Opt-aiNet represents an important algorithm intended for the multi-modal search. It comes from the family of AiNet algorithms which are influencing wide area of data processing tasks like data clustering, data compression or combinatorial optimization. As many other nature-inspired heuristics, opt-aiNet is capable to perform very well in terms of solutions' quality but it needs a corresponding amount of the compute power. Therefore any possible improvement in this area is welcome.

Within this paper, we discussed possibilities of parallelization of the opt-aiNet algorithm as a potential source of a large improvement from the perspective of computational performance. We have focused especially on massive parallel approach represented by modern GPUs allowing universal non-graphical computations because these devices start to be a common part of almost every present-day supercomputer, work-station or a notebook.

Our analysis is showing a large potential of possible parallelization even for the massive parallel approach represented by GPUs and their thousands of computational threads. On the other hand, there are also obstacles which make the parallelization relatively challenging and non-trivial.

Our preliminary testing implementation is showing promising results and we are expecting a speed-up factor of 5 at least if we compare the original sequential approach running on a present-day average CPU (Intel Core i5 4200M) and the GPU implementation running on a low-end GPU (AMD Radeon HD 8750M).

Acknowledgements

This research has been supported by SGS grant No. SGS14/106/OHK3/1T/18.

References

[1] Brownlee, J.: *Clever Algorithms: Nature-inspired programming recipes*. LuLu, 1st edition, 2011.

[2] de Castro, L. N. and Timmis, J.: An Artificial immune network for multimodal function optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, vol. 1, pp. 699–704, May 2002.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

TANGENTIAL FIELDS IN MATHEMATICAL MODEL OF OPTICAL DIFFRACTION

Jiří Krček, Jaroslav Vlček

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

Abstract

We present the formulation of optical diffraction problem on periodic interface based on vector tangential fields, for which the system of boundary integral equations is established. Obtained mathematical model is numerically solved using boundary element method and applied to sine interface profile.

1. Introduction

Diffraction of optical wave on periodical interface between two media belongs to frequently solved problems, especially, when the grating period Λ is comparable with wavelength λ of incident beam. Among other, this phenomenon is studied and exploited by nanostructured optical elements design. Naturally, theoretical modelling is of great importance in such cases. One of possible approaches has been demonstrated in our previous paper [1], where the boundary integral equations (BIE) for tangential fields have been introduced. Unlike the usually used rigorous coupled waves algorithm (RCWA) advantageous in the far fields analysis [2], the BIE models enable effective modelling of near fields in the spatially modulated region.

2. Formulation of problem

Let $S : x_3 = f(x_1)$ in $\mathbb{R}^3$ be a smooth surface periodically modulated in the coordinate x_1 with period Λ and uniform in the x_2 direction. The interface S with normal vector $\boldsymbol{\nu}$ divides the space into two semi-infinite homogeneous regions $\Omega^{(1)} = \{(x_1, x_2, x_3) \in \mathbb{R}^3,\ x_3 > f(x_1)\}$, $\Omega^{(2)} = \{(x_1, x_2, x_3) \in \mathbb{R}^3,\ x_3 < f(x_1)\}$ 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, the relative permeabilities $\mu^{(1)} = \mu^{(2)} = 1$ (both materials are magnetically neutral), see Fig.1.

We aim to solve optical diffraction problem for monochromatic plane wave with wavelength λ, i.e. with wave number $k_0 = 2\pi/\lambda$, incoming from $\Omega^{(1)}$ under the

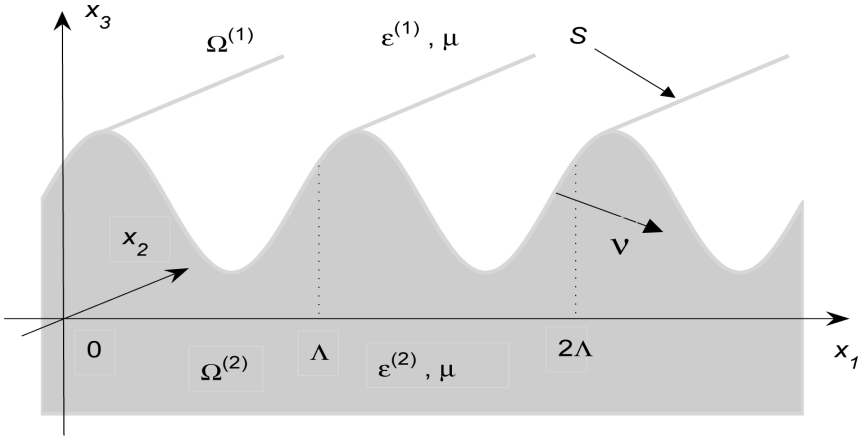


Figure 1: Structure of regions with common periodical boundary

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 $\boldsymbol{E}(x_1, x_2, x_3)\mathrm{e}^{-\mathrm{i}\omega t}$, $\boldsymbol{H}(x_1, x_2, x_3)\mathrm{e}^{-\mathrm{i}\omega t}$, where $\omega = c/\lambda$ and c represents the light velocity in the free space. The unknown intensities can be written as

$$\boldsymbol{E} = \begin{cases} \boldsymbol{E}_0^{(1)} + \boldsymbol{E}^{(1)} & \text{in} \quad \Omega^{(1)}, \\ \boldsymbol{E}^{(2)} & \text{in} \quad \Omega^{(2)}, \end{cases} \qquad \boldsymbol{H} = \begin{cases} \boldsymbol{H}_0^{(1)} + \boldsymbol{H}^{(1)} & \text{in} \quad \Omega^{(1)}, \\ \boldsymbol{H}^{(2)} & \text{in} \quad \Omega^{(2)}, \end{cases} \tag{1}$$

where the subscript 0 denotes incident field. In the media without free charges, the vectors $\boldsymbol{E}^{(j)}$, $\boldsymbol{H}^{(j)}$, $j = 1, 2$ fullfill Maxwell equations (the free-space wave impedance is embedded in the vector $\boldsymbol{H}$) in the form

$$\nabla \times \boldsymbol{E}^{(j)} = \mathrm{i}k_0\mu\boldsymbol{H}^{(j)}\,, \qquad \nabla \times \boldsymbol{H}^{(j)} = -\mathrm{i}k_0\varepsilon^{(j)}\boldsymbol{E}^{(j)} \quad \text{in} \quad \Omega^{(j)}\,, \tag{2}$$

$$\nabla \cdot \boldsymbol{E}^{(j)} = 0, \qquad \nabla \cdot \boldsymbol{H}^{(j)} = 0 \quad \text{in} \quad \Omega^{(j)}\,. \tag{3}$$

The tangential components of the fields are continuous on the boundary, i.e.

$$\boldsymbol{\nu} \times (\boldsymbol{E}^{(1)} - \boldsymbol{E}^{(2)}) = \boldsymbol{o}\,, \qquad \boldsymbol{\nu} \times (\boldsymbol{H}^{(1)} - \boldsymbol{H}^{(2)}) = \boldsymbol{o} \qquad \text{on } S\,. \tag{4}$$

For the far fields, the well-known Sommerfeld's radiation convergence conditions hold that allow to consider the problem on the common interface S only [3].

We solve the problem (2)–(4) for the TM polarization of incident wave, therefore we set $\boldsymbol{E}^{(j)} = (E_1^{(j)}, 0, E_3^{(j)})$, $\boldsymbol{H}^{(j)} = (0, H_2^{(j)}, 0)$. To this purpose, we introduce tangential fields in the next section that enable to reformulate given problem as scalar integral equations at common boundary. Theoretical background of used approach is referred in the article [1]. The boundary element method (BEM) has been chosen to solve obtained system numerically (Sect. 4). Resulting algorithm is tested for sine interface profile in the Sect. 5.

3. Mathematical model

We formulate the problem (2)–(4) as boundary integral equations for tangential fields

$$\boldsymbol{J} = \boldsymbol{\nu} \times \boldsymbol{E}^{(1)} = \boldsymbol{\nu} \times \boldsymbol{E}^{(2)}, \qquad \boldsymbol{I} = -\boldsymbol{\nu} \times \boldsymbol{H}^{(1)} = -\boldsymbol{\nu} \times \boldsymbol{H}^{(2)} \ , \tag{5}$$

where $\boldsymbol{\nu}$ is an unit normal vector of the boundary S oriented as shown in Fig.1. Similarly, $\boldsymbol{\tau}$ represents an unit tangential vector of S. 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, $\boldsymbol{I} = I_\tau \boldsymbol{\tau}$, where $I_\tau = -H_2^{(1)} = -H_2^{(2)}$.

We introduce a parametrization $\boldsymbol{\pi} : \langle 0, 2\pi \rangle \to \mathbb{R}^2$, $\boldsymbol{\pi}(t) = (p(t), q(t))$ of the curve $x_3 = f(x_1)$ having unit normal vector $\boldsymbol{\nu}(t)$ and corresponding tangential vector $\boldsymbol{\tau}(t)$ with the norm $\nu(t) = \sqrt{p'(t)^2 + q'(t)^2}$. Resulting system of boundary integral equations for scalar components I_τ and J_2 derived in [1] is of the following form:

$$\begin{aligned} J_2(s) = -J_{2,0}(s) - \mathrm{i}k_0 \mu \boldsymbol{\tau}(s) \cdot \int_0^{2\pi} I_\tau(t) \boldsymbol{\tau}(t) \left(\Psi^{(1)}(s,t) - \Psi^{(2)}(s,t) \right) \ \nu(t) \ dt \\ - \frac{1}{\mathrm{i}k_0} \boldsymbol{\tau}(s) \cdot \int_0^{2\pi} I'_\tau(t) \nabla_t \left[\frac{1}{\varepsilon^{(1)}} \Psi^{(1)}(s,t) - \frac{1}{\varepsilon^{(2)}} \Psi^{(2)}(s,t) \right] \ dt \\ + \boldsymbol{\nu}(s) \cdot \int_0^{2\pi} J_2(t) \nabla_t \left[\Psi^{(1)}(s,t) - \Psi^{(2)}(s,t) \right] \ \nu(t) \ dt \ , \end{aligned} \tag{6}$$

$$\begin{aligned} I_\tau(s) = -I_{\tau,0}(s) - \mathrm{i}k_0 \int_0^{2\pi} J_2(t) \left(\varepsilon^{(1)} \Psi^{(1)}(s,t) - \varepsilon^{(2)} \Psi^{(2)}(s,t) \right] \nu(t) \ dt \\ + \int_0^{2\pi} I_\tau(t) \boldsymbol{\nu}(t) \cdot \nabla_t \left[\Psi^{(1)}(s,t) - \Psi^{(2)}(s,t) \right] \ \nu(t) \ dt \ . \end{aligned} \tag{7}$$

In the kernels of integral operators, the parametrized periodical Green functions $\Psi^{(j)}(s,t)$, $j = 1, 2$ of Helmholtz equation play important role. We apply these by the relations [4]

$$\Psi^{(j)}(s,t) = \sum_{m=-\infty}^{\infty} \Psi_m^{(j)}(s,t) \, , \qquad \Psi_m^{(j)}(s,t) = \frac{1}{2\mathrm{i}\Lambda\beta_m} \ \mathrm{e}^{\mathrm{i}(\alpha_m(p(s)-p(t)) + \beta_m |q(s)-q(t)|)} \ , \tag{8}$$

where α_m, β_m are the propagation constants defined as

$$\alpha_m = \alpha + (2\pi m)/\Lambda \ , \quad \alpha = k_0 \sqrt{\varepsilon^{(1)}} \sin\theta \ , \quad \alpha_m^2 + \beta_m^2 = k_0^2 \varepsilon \ . \tag{9}$$

Required properties of obtained operators have been established e.g. in references [4, 5]. Note, that the singularity of logarithmic type is of key importance,

because it enables to split the operators into compact ones with continuous kernel and the other with logarithmic singularity:

$$\Psi^{(j)}(s,t) = \Psi_0^{(j)}(s,t) + \frac{1}{2\pi} \ln \left| 2 \sin \frac{s-t}{2} \right| + \Psi_r^{(j)}(s,t) \tag{10}$$

with regular part

$$\Psi_r^{(j)}(s,t) = \sum_{m \in Z, m \neq 0} \left\{ \Psi_m^{(j)}(s,t) - \frac{1}{2\pi} \frac{\mathrm{e}^{-\mathrm{i}m(s-t)}}{2|m|} \right\} . \tag{11}$$

In the way of existence and uniqueness of presented model we refer to the paper [6], where the properties of boundary operators are discussed in detail.

4. Numerical implementation

To solve the system of boundary integral equations (6),(7) we use collocation method with $2N+1$ equidistant collocation points $s_j = \frac{2\pi j}{2N}$, $j = 0, \ldots, 2N$.

We seek for discrete solutions

$$I_\tau(s) = \sum_{k=0}^{2N} c_k \phi_k(s) \quad \text{and} \quad J_2(s) = \sum_{k=0}^{2N} d_k \phi_k(s) \tag{12}$$

with interpolation basis $\{\phi_k\}_{k=0}^{2N}$. Thus, the system of trigonometric polynomials or linear splines (piecewise linear functions) is the usual choice of basis functions. Here, we prefer the last ones with nodes identical with collocation points ($\phi_k(s_j) = \delta_{kj}$). Note that an using of frequently applied cubic splines did not yield better results in the example demonstrated in the Sect. 5.

We find advantageous to take the order N of boundary discretization equal to the order of diffraction modes truncation in the Green function (8), so that

$$\Psi^{(j)}(s,t) \approx \sum_{m=-N}^{N} \Psi_m^{(j)}(s,t) , \quad j = 1, 2 . \tag{13}$$

Since the integral operators in the solved system are splitted by (10), we evaluate numerically the compact operators with continuous kernels – the trapezodial rule with nodes in collocation points (i.e. $t_j = s_j$) gives sufficiently accurate results. The logarithmic-type singular operators can be evaluated analytically.

5. Numerical results

As an example, we consider the smooth sine boundary

$$S: \; x_3 = \frac{h}{2} \left(1 + \cos \frac{2\pi x_1}{\Lambda} \right) , \; x_1 \in \langle 0, \Lambda \rangle , \quad \Lambda = 500 \text{ nm}, \; h = 50 \text{ nm}$$

between two regions with indices of refraction $n_1 = 1$ (air) and $n_2 = 1.5$ (glass), $n_j = \sqrt{\varepsilon^{(j)}}$. Incident beam of wavelength $\lambda = 632.8$ nm propagates under given

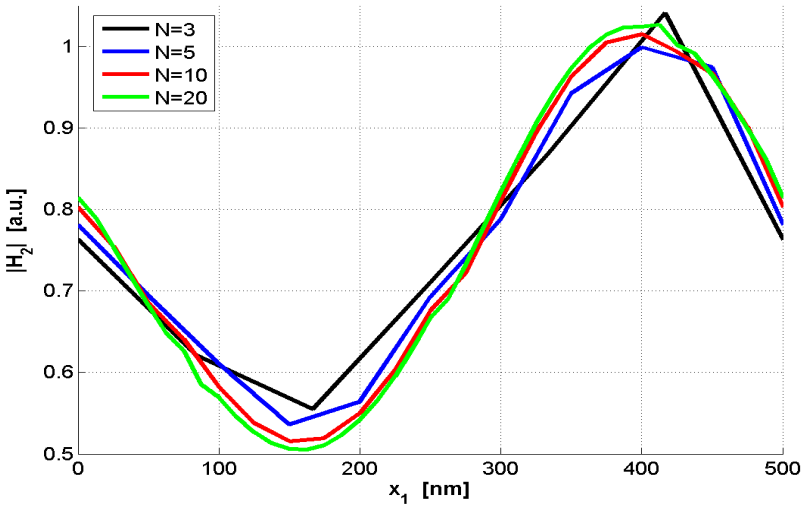


Figure 2: The convergence of used BEM algorithm (incidence angle $\theta = 40°$)

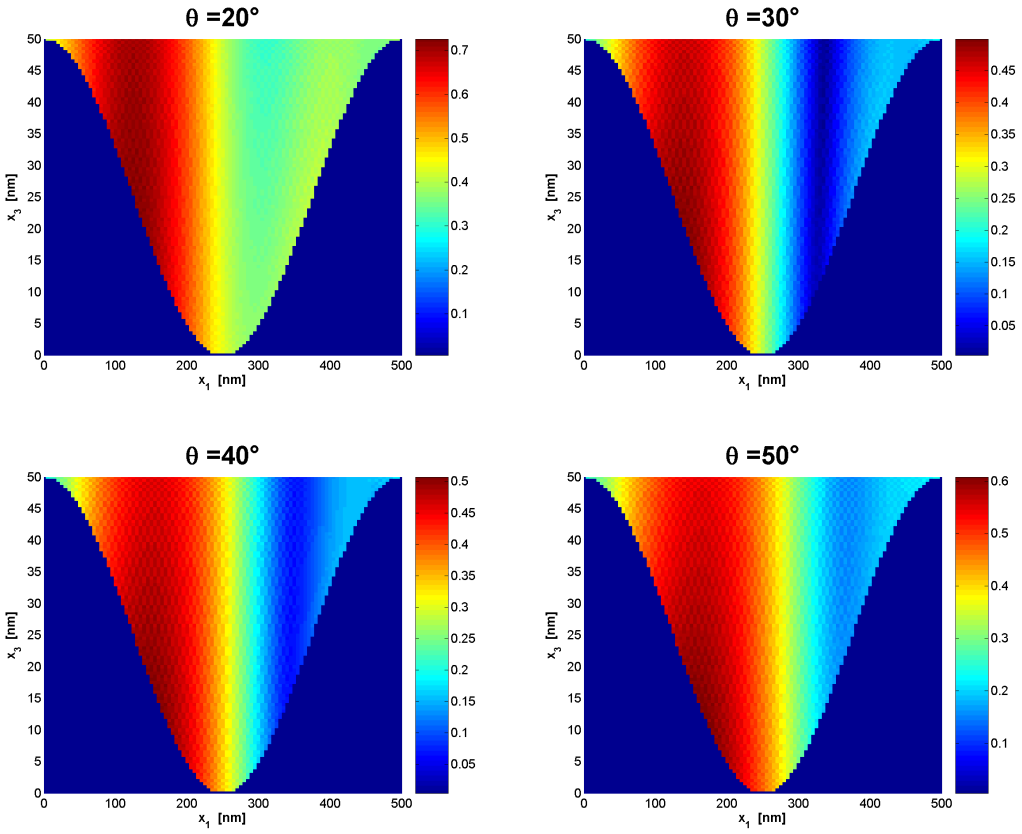


Figure 3: Reflected field $|H_2^{(1)}|$ for chosen incidence angle θ ($N = 50$).

angle of incidence θ. The Fig. 2 illustrates increasing accuracy of approximation with growing discretization order. We present here the absolute value of complex tangential component of the field $\boldsymbol{H}$ at one period of common boundary.

The reflected field $|H_2^{(1)}|$ is demonstrated at the Fig. 3 near to the boundary for several incidence angles. As the both materials are lossless, the field is nearly uniform in vertical direction.

6. Conclusion

The results obtained using presented BEM algorithm show possible applicability of the approach based on tangential fields to many problems, in which the detailed analysis of the diffracted optical field at an interface and/or in the near region is needed. We suppose to exploit this method in future to surface plasmon modelling.

Acknowledgements

This work was partially supported by National Supercomputing Center IT4Innovations at the VŠB – Technical University of Ostrava.

References

[1] Krček, J., Vlček, J., and Žídek, A.: Tangential fields in optical diffraction problems. In: J. Chleboun et al. (Eds.), *Programms and Algorithms of Numerical Mathematics*, vol. 16, pp. 124–129. Institute of Mathematics AS CR, Prague 2013.

[2] Bao, G., Cowsar, L., and Masters, W.: *Mathematical modeling in optical science.* SIAM, Philadelphia, 2001.

[3] Kleemann, B. H., Mitreiter, A., and Wyrowski, F.: Integral equation method with parametrization of grating profile. Theory and experiments. J. Modern Opt. **43** (1996), No. 7, 1323–1349.

[4] Linton, C. M.: The Green's function for the two-dimensional Helmholtz equation in periodic domains. J. Engrg. Math. **33** (1998), 377–402.

[5] 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*, pp. 221–229. Faculty of Mechanical Engineering, Slovak University of Technology, Bratislava 2012.

[6] Chen, X. and Friedmann, A.: Maxwells equations in a periodic structure. Trans. Amer. Math. Soc. **323** (1991), 465–507.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

AN ASYNCHRONOUS THREE-FIELD DOMAIN DECOMPOSITION METHOD FOR FIRST-ORDER EVOLUTION PROBLEMS

Lukáš Krupička, Michal Beneš

Department of Mathematics
Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
lukas.krupicka@fsv.cvut.cz, benes@mat.fsv.cvut.cz

Abstract

We present an asynchronous multi-domain time integration algorithm with a dual domain decomposition method for the initial boundary-value problems for a parabolic equation. For efficient parallel computing, we apply the three-field domain decomposition method with local Lagrange multipliers to ensure the continuity of the primary unknowns at the interface between subdomains. The implicit method for time discretization and the multi-domain spatial decomposition enable us to use different time steps (subcycling) on different parts of a computational domain, and thus efficiently capture the underlying physics with less computational effort. We illustrate the performance of the proposed multi-domain time integrator by means of a simple numerical example.

1. Introduction

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain split into a finite number of non-overlapping subdomains Ω^k $(k = 1, \ldots, N_D)$. Let $\Omega = \overline{\bigcup_{k=1}^{N_D} \Omega^k}$, $\Gamma^k = \partial\Omega^k$, $\Sigma = \bigcup_{k=1}^{N_D} \Gamma^k \backslash \partial\Omega$. We introduce the bilinear form

$$((u, v))_k := \int_{\Omega^k} \sum_{|i|\leq 1} \sum_{|j|\leq 1} (-1)^{|i|} a_{ij}(x) D^j u \, D^i v \, \mathrm{d}x \quad \forall u, v \in H^1(\Omega^k), \tag{1}$$

where $i = (i_1, i_2)$ and $j = (j_1, j_2)$ are two-dimensional vectors, i_1, i_2, j_1, j_2 are nonnegative integers and $|i| = i_1 + i_2$ and $|j| = j_1 + j_2$. The summation in (1) means that summation should be carried out over all i and j, for which $|i| \leq 1$, $|j| \leq 1$ holds. We assume that the coefficient functions a_{ij} belong to $L^\infty(\Omega)$. We assume there exists a positive number ϵ (independent of v) such that $((v, v))_k \geq \epsilon \|v\|^2_{H^1(\Omega^k)}$ for every $v \in H^1_0(\Omega^k)$. Further, for every $u, v \in \prod_k H^1(\Omega^k)$ we set $((u, v)) := \sum_k ((u, v))_k$. From now on we are going to use the following notation: $V := \prod_k H^1(\Omega^k)$ and $M := \prod_k H^{-1/2}(\Gamma^k)$, $(\cdot, \cdot)$ will be the usual inner product in $L^2(\Omega)$, $(\cdot, \cdot)_k$ will be the inner product in $L^2(\Omega^k)$ and $\langle \cdot, \cdot \rangle_k$ will be the duality pairing between $H^{-1/2}(\Gamma^k)$

and $H^{1/2}(\Gamma^k)$. Finally, introduce the space $\Phi := \left\{\varphi \in L^2(\Sigma);\ \exists v \in H_0^1(\Omega),\ \varphi = v\big|_\Sigma\right\}$ equipped with the norm $\|\varphi\|_\Phi = \inf\left\{\|v\|_{H^1(\Omega)};\ v \in H_0^1(\Omega),\ v\big|_\Sigma = \varphi\right\}$.We now consider the following two equivalent model problems. Let $T > 0$ be fixed and assume $u_0 \in H_0^1(\Omega)$, $f \in L^2(0,T;L^2(\Omega))$:
(i) find $u \in L^2(0,T;H_0^1(\Omega))$ with $\partial_t u \in L^2(0,T;L^2(\Omega))$, such that

$$(\partial_t u, v) + ((u,v)) = (f,v) \quad \forall v \in H_0^1(\Omega) \quad \text{and} \quad u(x,0) = u_0(x) \text{ in } \Omega; \tag{2}$$

(ii) find $u^k \in L^2(0,T;H^1(\Omega^k))$ with $\partial_t u^k \in L^2(0,T;L^2(\Omega^k))$, $\lambda^k \in L^2(0,T;H^{-1/2}(\Gamma^k))$ and $w \in L^2(0,T;\Phi)$, such that $u^k(x,0) = u_0(x)\big|_{\Omega^k}$ and (for $k = 1,\dots,N_D$)

$$\left\{\begin{array}{rcll} (\partial_t u^k, v^k)_k + ((u^k,v^k))_k - \langle\lambda^k, v^k\rangle_k &=& (f,v^k)_k & \forall v^k \in H^1(\Omega^k), \\ \langle u^k, \mu^k\rangle_k &=& \langle w, \mu^k\rangle_k & \forall \mu^k \in H^{-1/2}(\Gamma^k), \\ \sum_{k=1}^{N_D}\langle\lambda^k,\varphi\rangle_k &=& 0 & \forall \varphi \in \Phi. \end{array}\right. \tag{3}$$

Let us mention that problem (3) is well suited for domain decomposition methods. By the standard linear parabolic equation theory [4], both problems (2) and (3) admit the unique solution, such that $u = u^k$ in Ω^k, $\lambda^k = \nabla u \cdot \mathbf{n}_A^k$ on $\partial\Omega^k$ and $w = u$ on Σ. To solve problem (3) numerically, we propose a new numerical scheme which is based on the subcycling algorithm using non-standard asynchronous time discretization amenable for parallel computing.

2. Asynchronous multi-domain discretization in time

Let us fix $p \in \mathbb{N}$ and let $\tau := T/p$ be a time step. Next, we introduce a substep time $\tau^k = \tau/s^k$, which is proportional to the system time step $\tau = t_{n+1} - t_n$, where s^k is the number of substeps for domain k, as shown schematically in Figure 1. Further, we introduce the backward difference quotient $\delta_{\tau^k}\phi_{n,j}^k := (\phi_{n,j}^k - \phi_{n,j-1}^k)/\tau^k$ for $n = 0,\dots,p-1$. In view of the assumed relationships between the discretization steps, the present "method of asynchronous discretization in time" consists in the following: find, successively for $n = 0,1,2,\dots,p-1$, functions $u_{n,j}^k \in H^1(\Omega^k)$, $\lambda_{n,j}^k \in H^{-1/2}(\partial\Omega^k)$ and $w_{n+1} \in \Phi$, $k = 1,\dots,N_D$, $j = 1,\dots,s^k$, as solutions of the problems

$$(\delta_{\tau^k} u_{n,j}^k, v_j^k)_k + ((u_{n,j}^k, v^k))_k - \langle\lambda_{n,j}^k, v_j^k\rangle_k = (f_{n,j}^k, v_j^k)_k \qquad \forall v_j^k \in H^1(\Omega^k), \tag{4}$$

$$\langle u_{n,j}^k, \mu_j^k\rangle_k = \langle w_{n,j}^k, \mu_j^k\rangle_k \qquad \forall \mu_j^k \in H^{-1/2}(\Gamma^k), \tag{5}$$

$$\sum_{k=1}^{N_D}\langle\lambda_{n,s^{N_D}}^k, \varphi\rangle_k = 0 \qquad \forall \varphi \in \Phi, \tag{6}$$

starting with the functions $u_{0,0}^k(x) = u_0(x)\big|_{\Omega^k} \in H^1(\Omega^k)$.

In this work, the equation of continuity of fluxes is required only at the final (system) time step, see (6). The unknown $w_{n,j}^k$ on the common interface Σ is linearly interpolated at the intermediate steps by

$$w_{n,j}^k = \left(1 - \frac{j}{s^k}\right) w_n + \left(\frac{j}{s^k}\right) w_{n+1} \quad \forall j = 1,\dots,s^k, \quad k = 1,\dots,N_D.$$

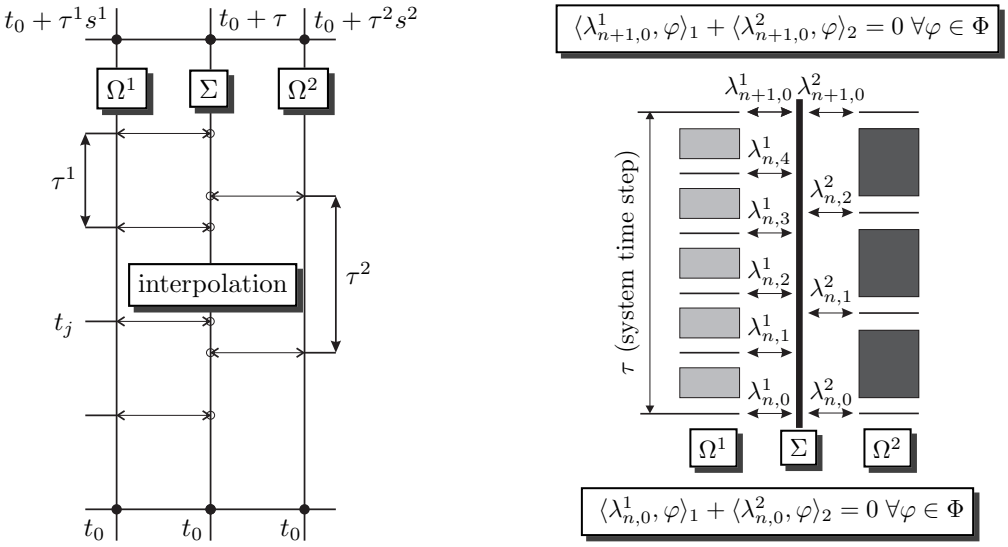


Figure 1: Substeps of the system time step. Example for $N_D = 2$, $s^1 = 5$, $s^2 = 3$.

Theorem 1. *Problem* (4)–(6) *has a unique solution.*

Proof. Without loss of generality we assume $u^k_{n,0} = 0$. First, we associate with any $\varphi \in \Phi$ a vector function

$$\widetilde{\boldsymbol{\varphi}} = (\tilde{u}^1_{n,1}, \tilde{u}^1_{n,2}, \ldots, \tilde{u}^1_{n,s^1}, \tilde{u}^2_{n,1}, \tilde{u}^2_{n,2}, \ldots, \tilde{u}^2_{n,s^2}, \ldots, \tilde{u}^{N_D}_{n,1}, \tilde{u}^{N_D}_{n,2}, \ldots, \tilde{u}^{N_D}_{n,s^{N_D}}) \in \prod_k H^1(\Omega^k)^{s^k},$$

components of which are defined as solutions of the following Dirichlet problems

$$\begin{aligned} (\delta_{\tau^k} \tilde{u}^k_{n,j}, v^k)_k + ((\tilde{u}^k_{n,j}, v^k))_k &= 0 && \forall v^k \in H^1_0(\Omega^k), \\ \langle \tilde{u}^k_{n,j}, \mu^k \rangle_k &= \langle \big(j/s^k\big)\, \varphi, \mu^k \rangle_k && \forall \mu^k \in H^{-1/2}(\Gamma^k) \end{aligned}$$

for $k = 1, \ldots, N_D$, $j = 1, \ldots, s^k$. Note that

$$\| |\widetilde{\boldsymbol{\varphi}}| \|_{\prod_k H^1(\Omega^k)^{s^k}} := \sum_k^{N_D} \sum_j^{s^k} \|\tilde{u}^k_{n,j}\|_{H^1(\Omega^k)} \le c \|\varphi\|_\Phi. \tag{7}$$

Now we assume a given function $\psi \in \Phi$ and set a vector functions

$$\boldsymbol{u} = (u^1_{n,1}, u^1_{n,2}, \ldots, u^1_{n,s^1}, u^2_{n,1}, u^2_{n,2}, \ldots, u^2_{n,s^2}, \ldots, u^{N_D}_{n,1}, u^{N_D}_{n,2}, \ldots, u^{N_D}_{n,s^{N_D}})$$

and

$$\boldsymbol{\lambda} = (\lambda^1_{n,1}, \lambda^1_{n,2}, \ldots, \lambda^1_{n,s^1}, \lambda^2_{n,1}, \lambda^2_{n,2}, \ldots, \lambda^2_{n,s^2}, \ldots, \lambda^{N_D}_{n,1}, \lambda^{N_D}_{n,2}, \ldots, \lambda^{N_D}_{n,s^{N_D}})$$

so that $\boldsymbol{u} = \widetilde{\boldsymbol{\psi}}$ and

$$(\delta_{\tau^k} u^k_{n,j}, v^k)_k + ((u^k_{n,j}, v^k))_k - \langle \lambda^k_{n,j}, v^k \rangle_k = 0 \qquad \forall v^k \in H^1(\Omega^k), \tag{8}$$

$$\langle u^k_{n,j} - \big(j/s^k\big)\, \psi, \mu^k \rangle_k = 0 \qquad \forall \mu^k \in H^{-1/2}(\Gamma^k) \tag{9}$$

for $j = 1, \ldots, s^k$, $k = 1, \ldots, N_D$. We now define the operator $\mathcal{S} : \Phi \to \Phi^*$ by

$$\langle \mathcal{S}(\psi), \cdot \rangle_{\Phi^*,\Phi} = \sum_{k=1}^{N_D} \langle \lambda^k_{n,s^k}, \cdot \rangle_k.$$

From (7) and (8) we easily compute (recall $\boldsymbol{u} = \widetilde{\boldsymbol{\psi}}$)

$$\langle \mathcal{S}(\psi), \varphi \rangle_{\Phi^*,\Phi} = \sum_{k=1}^{N_D} \langle \lambda^k_{n,s^k}, \varphi \rangle_k \leq \alpha \|\psi\|_\Phi \|\varphi\|_\Phi. \tag{10}$$

On the other hand, taking $\varphi = \psi$ we have, combining (8) and (9),

$$\begin{aligned} \langle \mathcal{S}(\psi), \psi \rangle_{\Phi^*,\Phi} = \sum_{k=1}^{N_D} \langle \lambda^k_{n,s^k}, \psi \rangle_k &= \sum_{k=1}^{N_D} (\delta_{\tau^k} u^k_{n,s^k}, u^k_{n,s^k})_k + \sum_{k=1}^{N_D} ((u^k_{n,s^k}, u^k_{n,s^k}))_k \\ &\geq \gamma \|\psi\|^2_\Phi. \end{aligned} \tag{11}$$

In (10) and (11), α and γ are positive constants, independent of ψ and φ. Hence, $\mathcal{S}$ is an isomorphism from Φ onto Φ^*.

We now turn back, for a moment, to (4)–(6) and consider $\breve{u}^k_{n,j} \in H^1_0(\Omega^k)$ and $\breve{\lambda}^k_{n,j} \in H^{-1/2}(\partial\Omega^k)$ as the solution of the problem

$$(\delta_{\tau^k} \breve{u}^k_{n,j}, v^k)_k + ((\breve{u}^k_{n,j}, v^k))_k - \langle \breve{\lambda}^k_{n,j}, v^k \rangle_k = (f^k_{n,j}, v^k)_k \qquad \forall v^k \in H^1(\Omega^k),$$

for $k = 1, \ldots, N_D$, $j = 1, \ldots, s^k$. The existence of such solutions is ensured by [1]. We now define the functional $g \in \Phi^*$ by

$$\langle g, \cdot \rangle_{\Phi^*,\Phi} = \sum_{k=1}^{N_D} \langle -\breve{\lambda}^k_{n,s^k}, \cdot \rangle_k.$$

Problem (4)–(6) can now be reduced to problem

$$\mathcal{S}(\psi) = g.$$

Now with ψ in hand, we determine $u^k_{n,j} \in H^1(\Omega^k)$ and $\lambda^k_{n,j} \in H^{-1/2}(\partial\Omega^k)$ as the solution of decoupled (independent) Dirichlet problems (8) and (9) for $k = 1, \ldots, N_D$, $j = 1, \ldots, s^k$. It is easy to verify, that $u^k_{n,j}$, $\lambda^k_{n,j}$ and $w^k_{n,j} = \left(\frac{j}{s^k}\right) \psi$ solve uniquely problem (4)–(6). Recall that we considered for simplicity $u^k_{n,0} = 0$. The proof is complete. □

Remark 2. *Let us explicitly mention, that $\mathcal{S}$ corresponds to the Poincaré-Steklov operator on Σ, well known in the theory of domain decomposition methods for elliptic problems, see [2, 3].*

3. Numerical example

We approximate the problem (3) in space choosing V_h, M_h and Φ_h finite dimensional subspaces of V, M and Φ and introduce $\boldsymbol{u}_h(x,t) = \mathbf{N}_u(x)\widetilde{\boldsymbol{u}}(t)$, $\boldsymbol{w}_h(x,t) = \mathbf{N}_w(x)\widetilde{\boldsymbol{w}}(t)$ and $\boldsymbol{\lambda}_h(x,t) = \mathbf{N}_{\boldsymbol{\Lambda}}(x)\widetilde{\boldsymbol{\Lambda}}(t)$, such that $\boldsymbol{u}_h(t) \in V_h$, $\boldsymbol{w}_h(t) \in \Phi_h$ and $\boldsymbol{\lambda}_h(t) \in M_h$ for all $t \in (0,T)$, respectively. Application of FEM-discretization in space leads to the following system of equations ($j = 1,\ldots,s^k$, $k = 1,\ldots,N_D$):

$$\left\{\begin{array}{rcl} \mathbf{M}^k\delta_{\tau^k}\boldsymbol{u}^k_{n,j} + \mathbf{K}^k\boldsymbol{u}^k_{n,j} + (\mathbf{C}^k)^T\boldsymbol{\Lambda}^k_{n,j} &=& \boldsymbol{f}^k_{n,j}, \\ \mathbf{C}^k\boldsymbol{u}^k_{n,j} - \left(\frac{j}{s^k}\right)\mathbf{B}^k\boldsymbol{w}_{n+1} - \left(1-\frac{j}{s^k}\right)\mathbf{B}^k\boldsymbol{w}_n &=& \mathbf{0}, \\ \sum_{k=1}^{N_D}(\mathbf{B}^k)^T\boldsymbol{\Lambda}^k_{n,s^k} &=& \mathbf{0}. \end{array}\right.$$

Using the common nomenclature of heat conduction, $\mathbf{M}^k$ is the capacitance matrix, $\mathbf{K}^k$ is the conductance matrix and the vector $\boldsymbol{f}^k$ represents the nodal values of the source corresponding to subdomain Ω^k. Operators $\mathbf{C}^k$ and $\mathbf{B}^k$ are the Boolean matrices extracting the interface degrees of freedom from $\boldsymbol{u}$ and the corresponding degrees of freedom from $\boldsymbol{w}$ for a particular subdomain k. The above system can be written in a matrix form, which has a block-bordered structure amenable to parallel computation. In this work, we solve for all unknowns simultaneously by a monolithic method using a direct solver. In order to briefly present the performance of the proposed algorithm, we consider a simple test problem. A square of size 1.0×1.0 is divided into two equal subdomains, and each subdomain is divided into 5×10 square elements, see Figure 2. We consider the analytical solution given by

$$u^*(x_1,x_2,t) = \sin(\pi x_1)\sin(\pi x_2)\sin(t)$$

and assume the coefficient functions as constants: $a_{ij}(x) = \delta_{ij}10^{-4}$ in Ω. Hence, the right hand side takes the form

$$f^* = \left[\cos(t) + 2\times 10^{-4}\pi^2\sin(t)\right]\sin(\pi x_1)\sin(\pi x_2).$$

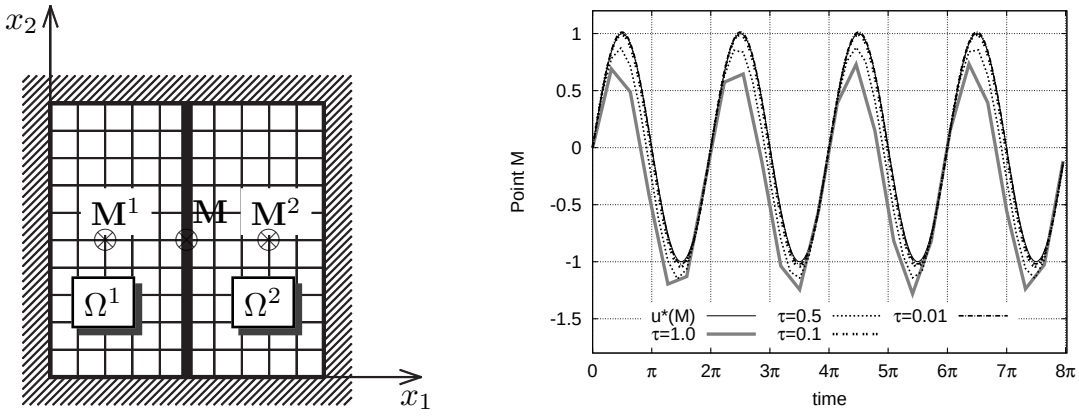


Figure 2: 2D test problem (left). Numerical results at the point M for various system time steps (right).

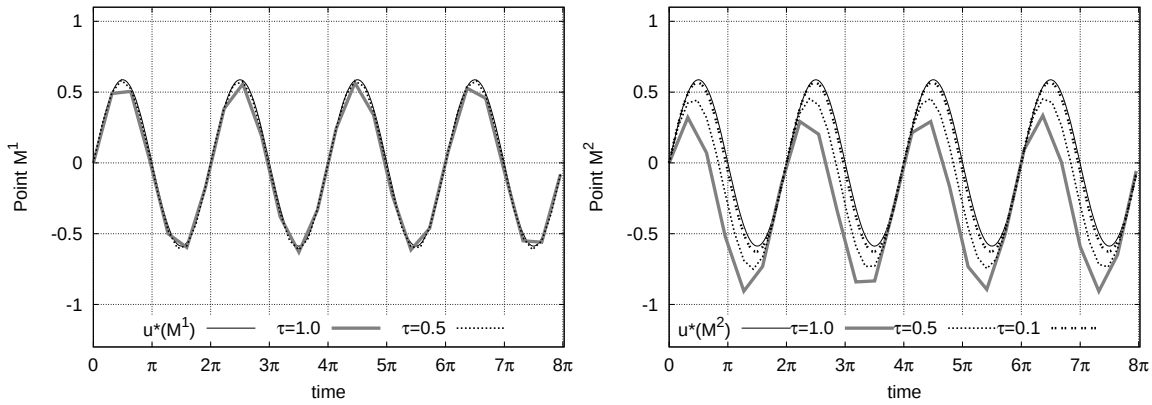


Figure 3: Results at the points M^1 (left) and M^2 (right).

In Figures 2 and 3 we have shown the results at points M, M^1 and M^2 for various system time steps τ, the ratio $s^1 : s^2 = 10 : 1$. As predicted by the theory, the numerical results are stable and match well with the analytical solution for sufficiently small system time step (approx. $\tau \approx 0.1$).

Acknowledgements

This research was supported by the grant SGS14/001/OHK1/1T/11 provided by the Grant Agency of the Czech Technical University in Prague (the first author) and by the project GAČR 14-21450S (the second author).

References

[1] Babuška, I.: The finite element method with Lagrangian multipliers. Numer. Math. **20** (1973), 179–192.

[2] Brezzi, F. and Marini, L.: Macro hybrid elements and domain decomposition methods. In: (J. A. Désideri, L. Fezoui, B. Larrouturou, B. Rousselet (Eds.), *Optimization et Côntrole Cépadue's-Editions*, pp. 89–96. Toulouse, 1993.

[3] Brezzi, F. and Marini, L.: A three-field domain decomposition method. In: (A. Quarteroni, J. Periaux, Y. A. Kuznetof, O. Widlund (Eds.), *Domain Decomposition Methods in Science and Engineering*, vol. 157, pp. 27–34. American Mathematical Society, Series CONM, 1994.

[4] Rektorys, K.: *The method of discretization in time and partial differential equations*. Springer, 1982.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

NUMERICAL SOLUTION OF A NEW HYDRODYNAMIC MODEL OF FLOCKING

Václav Kučera, Andrea Živčáková

Charles University in Prague, Faculty of Mathematics and Physics
Sokolovská 83, 186 75 Praha, Czech Republic
kucera@karlin.mff.cuni.cz, zivcakova@karlin.mff.cuni.cz

Abstract

This work is concerned with the numerical solution of a hydrodynamic model of the macroscopic behavior of flocks of birds due to Fornasier et al., 2011. The model consists of the compressible Euler equations with an added nonlocal, nonlinear right-hand side. As noticed by the authors of the model, explicit time schemes are practically useless even on very coarse grids in 1D due to the nonlocal nature of the equations. To this end, we apply a semi-implicit discontinuous Galerkin method to solve the equations. We present a simple numerical test of the resulting scheme.

1. Continuous problem

In [4], a new hydrodynamic limit of a modification of the famous Cucker-Smale model was derived. The equations describe, using macroscopic quantities, the dynamics of flocks of birds or other self-organizing entities. The equations are highly nonlinear and nonlocal and are therefore extremely expensive to treat numerically. In [4] a first simple simulation was performed using the finite volume method. Here, we discretize the model more efficiently using the discontinuous Galerkin method.

Let $\Omega = (0,1) \subset \mathbb{R}$ and for $0 < M < +\infty$, we set $Q_M := \Omega \times (0, M)$. We treat the following problem written in conservative variables. Find $\boldsymbol{w} : Q_M \to \mathbb{R}^3$ such that

$$\frac{\partial \boldsymbol{w}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial x} = \boldsymbol{g}(\boldsymbol{w}) \ \text{ in } Q_M, \tag{1}$$

where $\boldsymbol{w} = (\rho, \rho u, E)^\top \in \mathbb{R}^3$ is the *state vector* and

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{w}) &= \big(f_1(\boldsymbol{w}), f_2(\boldsymbol{w}), f_3(\boldsymbol{w})\big)^\top = \big(\rho u, \rho u^2 + p, (E+p)u\big)^\top, \\ \boldsymbol{g}(\boldsymbol{w}) &= \lambda\big(0, \mathcal{A}(\boldsymbol{w}), \mathcal{B}(\boldsymbol{w})\big)^\top. \end{aligned} \tag{2}$$

Here ρ denotes the density, u velocity, E energy and p pressure. The right-hand side functions $\mathcal{A}$ and $\mathcal{B}$ are given by

$$\begin{aligned} \mathcal{A}(\boldsymbol{w})(x,t) &= \int_{\mathbb{R}} b(|x-y|)\Big(u(y,t) - u(x,t)\Big)\rho(x,t)\rho(y,t)\,\mathrm{d}y, \\ \mathcal{B}(\boldsymbol{w})(x,t) &= \int_{\mathbb{R}} b(|x-y|)\rho(x,t)\Big(\rho(y,t)u(x,t)u(y,t) - 2E(y,t)\Big)\,\mathrm{d}y, \end{aligned} \tag{3}$$

where

$$b(|x-y|) = \frac{K}{(\lambda + |x-y|^2)^{\beta+1}}, \tag{4}$$

and $K, \lambda > 0$ and $\beta \geq 0$ are given constants. The relations between E, p are the classical laws of a perfect gas,

$$E = \rho\left(\frac{3}{2}T + \frac{u^2}{2}\right), \quad p = \rho T, \tag{5}$$

where T is the thermodynamic temperature.

In (3), we write the right-hand side terms $\mathcal{A}, \mathcal{B}$ as functions of $\boldsymbol{w}$, although the integrals in (3) are written terms of the nonconservative variables ρ, u, T. Expressing $\mathcal{A}, \mathcal{B}$ in $\boldsymbol{w}$ in a suitable way is a key ingredient in our scheme and will be described in detail in Section 2.3. System (1) is equipped with the initial condition $\boldsymbol{w}(x,0) = \boldsymbol{w}^0(x)$ and periodic boundary conditions.

2. Discretization

We shall use the multidimensional notation for $\Omega \subset \mathbb{R}^d$, although in our computations we have $d = 1$. Let $\mathcal{T}_h$ be triangulation of Ω and $\mathcal{F}_h$ the system of all faces (nodes in 1D) of $\mathcal{T}_h$. For each $\Gamma \in \mathcal{F}_h$ we choose a unit normal $n_\Gamma = \pm 1$, which, for $\Gamma \subset \partial\Omega$, has the same orientation as the outer normal to Ω. For each *interior* face $\Gamma \in \mathcal{F}_h$ there exist two neighbours $K_\Gamma^{(L)}$, $K_\Gamma^{(R)} \in \mathcal{T}_h$ such that n_Γ is the outer normal to $K_\Gamma^{(L)}$. For v piecewise defined on $\mathcal{T}_h$ and $\Gamma \in \mathcal{F}_h$ we introduce $v|_\Gamma^{(L)}$ is the trace of $v|_{K_\Gamma^{(L)}}$ on Γ, $v|_\Gamma^{(R)}$ is the trace of $v|_{K_\Gamma^{(R)}}$ on Γ, $\langle v \rangle_\Gamma = \frac{1}{2}\big(v|_\Gamma^{(L)} + v|_\Gamma^{(R)}\big)$ and $[v]_\Gamma = v|_\Gamma^{(L)} - v|_\Gamma^{(R)}$. On $\partial\Omega$, we define $v|_\Gamma^{(L)}, v|_\Gamma^{(R)}$ using periodic boundary conditions. If $[\cdot]_\Gamma, \langle\cdot\rangle_\Gamma, v|_\Gamma^{(L)}, v|_\Gamma^{(R)}$ appear in an integral over $\Gamma \in \mathcal{F}_h$, we omit the subscript Γ.

Let $p \in \mathbb{N}$ and let $P^p(K)$ be the space of polynomials on $K \in \mathcal{T}_h$ of degree $\leq p$. The approximate solution will be sought in the space of discontinuous piecewise polynomial functions

$$\boldsymbol{S}_h := [S_h]^3, \text{ where } S_h = \{v;\, v|_K \in P^p(K), \forall K \in \mathcal{T}_h\}.$$

2.1. Discontinuous Galerkin space semidiscretization

The discrete problem is derived in the following way. We multiply (1) by a test function $\boldsymbol{\varphi}_h \in \boldsymbol{S}_h$, integrate over $K \in \mathcal{T}_h$ and apply Green's theorem in the convective terms. Summing over $K \in \mathcal{T}_h$ and rearranging the boundary terms, we obtain

$$\int_\Omega \frac{\partial \boldsymbol{w}}{\partial t} \cdot \boldsymbol{\varphi}\,\mathrm{d}x + \sum_{\Gamma\in\mathcal{F}_h}\int_\Gamma \boldsymbol{f}(\boldsymbol{w})n\cdot[\boldsymbol{\varphi}]\,\mathrm{d}S - \sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{f}(\boldsymbol{w})\cdot\frac{\partial\boldsymbol{\varphi}}{\partial x}\,\mathrm{d}x = \int_\Omega \boldsymbol{g}(\boldsymbol{w})\cdot\boldsymbol{\varphi}\,\mathrm{d}x. \tag{6}$$

Since $\boldsymbol{w}$ will be approximated by a function from $\boldsymbol{S}_h$, which are discontinuous on edges, we approximate the physical flux $\boldsymbol{f}(\boldsymbol{w})n$ through an edge $\Gamma \in \mathcal{F}_h$ by a so-called *numerical flux* $\mathbf{H}(\boldsymbol{w}^{(L)}, \boldsymbol{w}^{(R)}, n)$ as in the finite volume method. In our computations

we use the *Vijayasundaram* numerical flux, cf. [5, 2]. Now we can define the following forms defined for $\boldsymbol{w}, \boldsymbol{\varphi} \in \boldsymbol{S}_h$:
Convective form:

$$b_h(\boldsymbol{w}, \boldsymbol{\varphi}) = \sum_{\Gamma \in \mathcal{F}_h} \int_\Gamma \mathbf{H}(\boldsymbol{w}^{(L)}, \boldsymbol{w}^{(R)}, n) \cdot [\boldsymbol{\varphi}] \, \mathrm{d}S - \sum_{K \in \mathcal{T}_h} \int_K \boldsymbol{f}(\boldsymbol{w}) \cdot \frac{\partial \boldsymbol{\varphi}}{\partial x} \, \mathrm{d}x,$$

right-hand side source term form:

$$l_h(\boldsymbol{w}, \boldsymbol{\varphi}) = -\int_\Omega \boldsymbol{g}(\boldsymbol{w}) \cdot \boldsymbol{\varphi} \, \mathrm{d}x.$$

Finally, we introduce the space semi-discrete problem: We seek $\boldsymbol{w}_h \in C^1([0, M]; \boldsymbol{S}_h)$:

$$\frac{d}{dt}(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) + b_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) + l_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = 0, \ \forall \boldsymbol{\varphi}_h \in \boldsymbol{S}_h, \ \forall t \in (0, M). \quad (7)$$

2.2. Time discretization

Equation (7) represents a system of nonlinear ordinary differential equations, which must be discretized in time. Due to extreme time step restrictions caused by the nonlocal right-hand side terms, cf. [4], we want to avoid using an explicit scheme. However an implicit time discretization is also very expensive due to its nonlinearity. Therefore we use the semi-implicit scheme of [3] and apply it to our problem.

Let $0 = t_0 < t_1 < t_2 < \dots$ be a partition of time interval $[0, M]$ and define $\tau_k = t_{k+1} - t_k$. We approximate $\boldsymbol{w}_h^k \approx \boldsymbol{w}_h(t_k)$, where $\boldsymbol{w}_h^k \in \boldsymbol{S}_h$. We use a first order backward difference approximation for the time derivative. Following [3], the nonlinear convective terms $b_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h)$ are linearized as

$$\begin{aligned} \tilde{b}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = &- \sum_{K \in \mathcal{T}_h} \int_K \mathbb{A}\,(\boldsymbol{w}_h^k) \boldsymbol{w}_h^{k+1} \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x} \, \mathrm{d}x \\ &+ \int_{\mathcal{F}_h} \Big(\mathbb{P}^+\big(\langle \boldsymbol{w}_h^k \rangle, n\big) \boldsymbol{w}_h^{k+1,(L)} + \mathbb{P}^-\big(\langle \boldsymbol{w}_h^k \rangle, n\big) \boldsymbol{w}_h^{k+1,(R)} \Big) \cdot [\boldsymbol{\varphi}_h] \, \mathrm{d}S, \end{aligned} \quad (8)$$

where $\mathbb{A} = \frac{D\boldsymbol{f}}{D\boldsymbol{w}}$ and $\mathbb{P}^+, \mathbb{P}^-$ are matrices defining the Vijayasundaram numerical flux, cf. [3] for details.

As for the source terms, again we linearize them to obtain the approximation $l_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) \approx \tilde{l}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h)$. The specific construction of this linearization is technical and will be presented separately in Section 2.3.

Collecting all the considerations, we obtain the following semi-implicit DG scheme:

Definition 1. *We say that the sequence $\boldsymbol{w}_h^k \in \boldsymbol{S}_h, k = 0, 1, \dots,$ is a semi-implicit DG solution of problem (1) if for all $\boldsymbol{\varphi}_h \in \boldsymbol{S}_h$*

$$\Big(\frac{\boldsymbol{w}_h^{k+1} - \boldsymbol{w}_h^k}{\tau_k}, \boldsymbol{\varphi}_h \Big) + \tilde{b}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + \tilde{l}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = 0. \quad (9)$$

Equation (9) represents a linear equation for the unknown $\boldsymbol{w}_h^{k+1}$. By choosing basis functions of $\boldsymbol{S}_h$ with supports on only one element, we obtain a sparse, block-tridiagonal matrix with lower left and upper right blocks corresponding to periodic boundary conditions. To solve these systems, we use the direct solver UMFPACK, [1]. It is our goal to construct $\tilde{l}_h$ in such a way so as to preserve the sparsity structure of the systems solved.

2.3. Linearization of the source terms l_h

First, we rewrite the right-hand side integrals $\mathcal{A}, \mathcal{B}$ in terms of the conservative variables. For the integral $\mathcal{A}$, we obtain

$$\mathcal{A} = \int_{\mathbb{R}} b(|x-y|)\boldsymbol{w}(x,t) \cdot \big(w_2(y,t), -w_1(y,t), 0\big)\,\mathrm{d}y. \tag{10}$$

Similarly, we write $\mathcal{B}$ as

$$\mathcal{B} = \int_{\mathbb{R}} b(|x-y|)\boldsymbol{w}(x,t) \cdot \big(-2w_3(y,t), w_2(y,t), 0\big)\,\mathrm{d}y. \tag{11}$$

Therefore, we can rewrite the vector $\boldsymbol{g}(\boldsymbol{w})$ as

$$\boldsymbol{g}(\boldsymbol{w})(x,t) = \lambda \int_{\mathbb{R}} b(|x-y|)\mathbb{U}_2\big(\boldsymbol{w}(y,t)\big)\boldsymbol{w}(x,t)\,\mathrm{d}y, \tag{12}$$

where $\mathbb{U}_2(\boldsymbol{w}) \in \mathbb{R}^{3\times 3}$ is the matrix

$$\mathbb{U}_2(\boldsymbol{w}) = \begin{pmatrix} 0 & 0 & 0 \\ w_2 & -w_1 & 0 \\ -2w_3 & w_2 & 0 \end{pmatrix}.$$

Approximating $\boldsymbol{w}(x,t) \approx \boldsymbol{w}_h^{k+1}(x)$ and $\boldsymbol{w}(y,t) \approx \boldsymbol{w}_h^{k}(y)$, we get the linearized form

$$\tilde{l}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = \int_{\mathbb{R}} \left(\int_{\mathbb{R}} b(|x-y|)\mathbb{U}_2\big(\boldsymbol{w}_h^k(y)\big)\mathrm{d}y \right) \boldsymbol{w}_h^{k+1}(x) \cdot \boldsymbol{\varphi}_h(x)\,\mathrm{d}x. \tag{13}$$

Adding $\tilde{l}_h$ to the scheme (9) does not change the sparsity structure of the system matrix, since it contributes only to the block-diagonal. This is important, since other expressions than (12) are possible, however they lead to a full system matrix, which is undesirable. Nonetheless, the computation of these terms is extremely time consuming due to their nonlocal nature. Even if the basis functions of $\boldsymbol{S}_h$ are local, in order to evaluate $\tilde{l}_h$, we must compute the inner integral $\int_{\mathbb{R}} b(|x-y|)\mathbb{U}_2\big(\boldsymbol{w}_h^k(y)\big)\mathrm{d}y$, which is time consuming due to the slow decay of the function $b(|x-y|)$.

3. Numerical experiment

In this numerical experiment, we start at $t = 0$ with a Gaussian distribution of density $\rho(x) = \exp(-10(x - 0.5)^2)$ along with constant temperature $T = 10$ and the velocity distribution $u(x) = -\sin(2\pi x)$. The triangulation consists of 400 piecewise quadratic elements. We observe the formation of a sharp peak in ρ, as seen in Figure 1. Due to jumps in the solution, artificial diffusion was added, cf. [3].

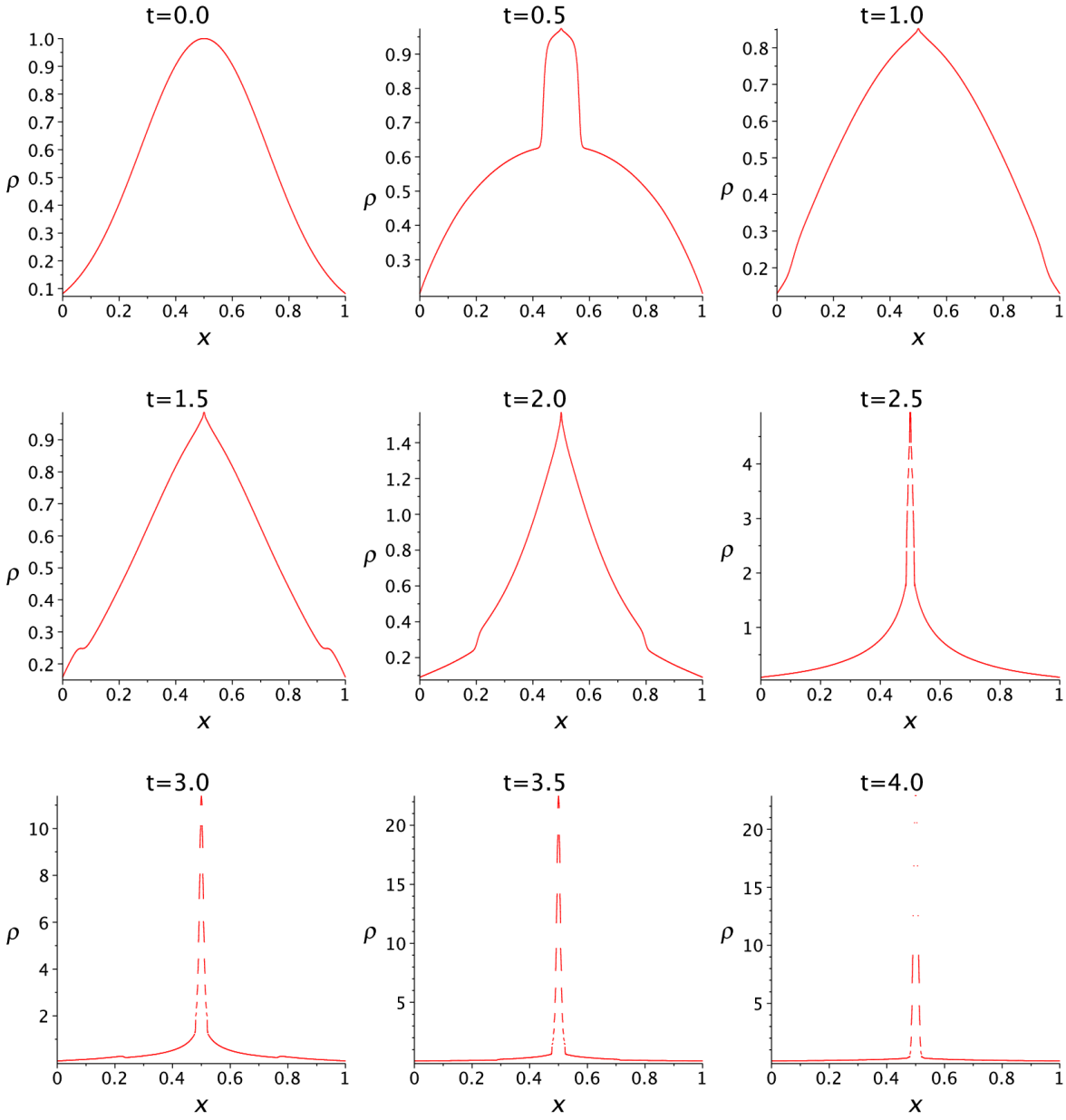


Figure 1: Numerical results for density.

Furthermore, in large regions of Ω, a state close to *vacuum* occurs, i.e. $\rho \approx 0$ and the matrices $\mathbb{A}, \mathbb{P}^+, \mathbb{P}^-$ are no longer defined. To avoid this complication, at each time step, $\boldsymbol{w}_h^k$ was *postprocessed* to avoid the vacuum state: If $\rho < \varepsilon$ or $T < \varepsilon$, then set $\rho := \varepsilon$ or $T := \varepsilon$ and recompute the energy E using relation (5). This defines a new state $\tilde{\boldsymbol{w}}_h^k$ which is used in (9) instead of $\boldsymbol{w}_h^k$ to compute $\boldsymbol{w}_h^{k+1}$. In our case, we use $\varepsilon := 10^{-5}$.

4. Conclusion

We have presented an efficient numerical method for the solution of a nonlinear and nonlocal version of the compressible Euler equations describing the dynamics of flocks of birds from [4]. To avoid severe time step restrictions, a semi-implicit discontinuous Galerkin scheme is applied. A suitable treatment of the nonlocal terms is given, which leads to sparse linear systems. Shock capturing and postprocessing of vacuum are added to obtain a stable scheme. To our knowledge, these are the first numerical results for this model, except for one test case in the original work [4].

Acknowledgements

The research of V. Kučera is supported by the Grant No. P201/13/00522S of the Czech Science Foundation. He is a junior researcher at the University Center for Mathematical Modelling, Applied Analysis and Computational Mathematics (Math MAC). The research of A. Živčáková is supported by the Charles University in Prague, project GA UK No. 758214.

References

[1] Davis, T. A. and Duff, I. S.: A combined unifrontal/multifrontal method for unsymmetric sparse matrices. ACM Transact. on Math. Soft. **25** (1999), 1–19.

[2] Feistauer, M., Felcman, J., and Straškraba, I.: *Mathematical and computational methods for compressible flow*. Clarendon Press, Oxford, 2003.

[3] Feistauer, M. and Kučera, V.: On a robust discontinuous Galerkin technique for the solution of compressible flow. J. Comput. Phys. **224** (2007), 208–221.

[4] Fornasier, M., Haškovec, J., and Toscani, G.: Fluid dynamic description of flocking via Povzner-Boltzmann equation. Physica D **240**, no. 1, (2011), 21–31.

[5] Vijayasundaram, G.: Transonic flow simulation using upstream centered scheme of Godunov type in finite elements. J. Comput. Phys. **63** (1986), 416–433.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

NONLINEAR CONJUGATE GRADIENT METHODS

Ladislav Lukšan, Jan Vlček

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží, 182 07 Praha 8
luksan@cs.cas.cz, vlcek@cs.cas.cz

Abstract

Modifications of nonlinear conjugate gradient method are described and tested.

Conjugate gradient method is frequently used to solve the following problems

$$F(x) = \frac{1}{2}(x - x^*)^T A(x - x^*) \to \min, \quad \text{or} \quad A(x - x^*) = 0,$$

where A is a symmetric positive definite matrix. Linear and nonlinear conjugate gradient methods differ in line search and gradient evaluations

Linear CG :	**Nonlinear CG :**
$g_1 = g(x_1),\ \ s_1 = -g_1,$	$F_1 = F(x_1),\ \ g_1 = g(x_1),\ \ s_1 = -g_1,$
For $i = 1,\ 2\ \ \ldots$ **do**	**For** $i = 1,\ 2\ \ \ldots$ **do**
$\alpha_i = \|g_i\|^2 / s_i^T A s_i,$	$\alpha_i > 0$ and $s_i^T g(x_i + \alpha_i s_i) = 0,$
$x_{i+1} = x_i + \alpha_i s_i,$	$x_{i+1} = x_i + \alpha_i s_i,$
$g_{i+1} = g_i + \alpha_i A s_i,$	$F_{i+1} = F(x_{i+1}),\ \ g_{i+1} = g(x_{i+1}),$
If $\|g_{i+1}\| \le \varepsilon \|g_1\|$ **then stop**.	**If** $\|g_{i+1}\| \le \varepsilon \|g_1\|$ **then stop**,
$\beta_i = \|g_{i+1}\|^2 / \|g_i\|^2,$	$\beta_i = \|g_{i+1}\|^2 / \|g_i\|^2,$
$s_{i+1} = -g_{i+1} + \beta_i s_i.$	$s_{i+1} = -g_{i+1} + \beta_i s_i$

($g(x)$ is the gradient of function F at the point x). Nonlinear CG method serves for seeking minima of a general nonlinear function $F(x)$. Instead of the perfect line search with $s_i^T g(x_i + \alpha_i s_i) = 0$, the (generalized) Wolfe conditions

$$F(x_i + \alpha_i s_i) - F_i \le \varepsilon_1 \alpha_i s_i^T g_i, \qquad \varepsilon_2 s_i^T g_i \le s_i^T g(x_i + \alpha_i s_i) \le \varepsilon_3 |s_i^T g_i| \tag{1}$$

are used, where $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$ and $\varepsilon_3 \ge 0$. Basic versions of the nonlinear conjugate gradient method use direction vectors $s_1 = -g_1$, $s_{i+1} = -g_{i+1} + \beta_i s_i$, $i \in \mathbb{N}$, with the coefficients

$$\beta_i^{HS} = \frac{y_i^T g_{i+1}}{y_i^T s_i}, \qquad \beta_i^{PR} = \frac{y_i^T g_{i+1}}{g_i^T g_i}, \qquad \beta_i^{LS} = \frac{y_i^T g_{i+1}}{|g_i^T s_i|} \tag{2}$$

(HS–Hestenes and Stiefel, PR–Polak and Ribire, LS–Liu and Storey),

$$\beta_i^{DY} = \frac{g_{i+1}^T g_{i+1}}{y_i^T s_i}, \qquad \beta_i^{FR} = \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}, \qquad \beta_i^{CD} = \frac{g_{i+1}^T g_{i+1}}{|g_i^T s_i|} \tag{3}$$

(DY–Dai and Yuan, FR–Fletcher and Reeves, CD–conjugate descent), and

$$\beta_i^{HSP} = \frac{p_i^T g_{i+1}}{y_i^T s_i}, \qquad \beta_i^{PRP} = \frac{p_i^T g_{i+1}}{g_i^T g_i}, \qquad \beta_i^{LSP} = \frac{p_i^T g_{i+1}}{|g_i^T s_i|} \tag{4}$$

(Perry's modifications of HS, PR, LS methods). We use the notation $d_i = x_{i+1} - x_i$, $y_i = g_{i+1} - g_i$, $p_i = y_i - d_i$. If $g_{i+1}^T s_i = 0$ (perfect line search) and function F is quadratic, all these methods are identical.

Methods HS, PR, LS are more efficient than DY, FR, CD (since they keep the conjugacy of direction vectors more successfully), but their global convergence cannot be proved without additional modifications. Methods DY, FR, CD are globally convergent (with some limitations concerning the stepsize selection), but they are less efficient than HS, PR, LS methods. The following simple modifications can be used for improving global convergence of HS, PR, LS methods.

$$\begin{aligned} \beta_i^{HS+} &= \max(0, \beta_i^{HS}), & \beta_i^{HSC} &= \max(0, \min(\beta_i^{HS}, \beta_i^{DY})), \\ \beta_i^{PR+} &= \max(0, \beta_i^{PR}), & \beta_i^{PRC} &= \max(0, \min(\beta_i^{PR}, \beta_i^{FR})), \\ \beta_i^{LS+} &= \max(0, \beta_i^{LS}), & \beta_i^{LSC} &= \max(0, \min(\beta_i^{LS}, \beta_i^{CD})). \end{aligned}$$

In this contribution, we will study further modifications of nonlinear CG methods that improve the efficiency of the basic ones. The following modifications, which use direction vectors

$$s_{i+1} = -\left(1 + \beta_i \frac{g_{i+1}^T s_i}{g_{i+1}^T g_{i+1}}\right) g_{i+1} + \beta_i s_i, \tag{5}$$

$$s_{i+1} = -g_{i+1} + \beta_i \left(s_i - \frac{g_{i+1}^T s_i}{g_{i+1}^T y_i}\, y_i\right) \tag{6}$$

guarantee that the direction vectors are descent.

Theorem 1. *Consider the nonlinear CG method with direction vectors $s_1 = -g_1$ and (6) or (6), $i \in \mathbb{N}$. Then $g_i^T s_i = -g_i^T g_i$, $i \in \mathbb{N}$. Let the parameter β_i be given by (2) and the generalized Wolfe conditions (1) with $\varepsilon_2 \le \varepsilon_3 < \infty$ be used. If the objective function F is strongly convex, Lipschitz continuous and bounded from below on $\mathbb{R}^n$, then the nonlinear CG method is uniformly descent and, therefore, globally convergent.*

The following modifications, which use direction vectors

$$s_{i+1} = -\vartheta_i g_{i+1} + \beta_i s_i, \tag{7}$$

where

$$\vartheta_i^{HS} = \vartheta_i^{DY} = \frac{y_i^T s_i}{y_i^T s_i} = 1, \quad \vartheta_i^{PR} = \vartheta_i^{FR} = \frac{y_i^T s_i}{g_i^T g_i}, \quad \vartheta_i^{LS} = \vartheta_i^{CD} = \frac{y_i^T s_i}{|g_i^T s_i|}, \qquad (8)$$

improve the conjugacy of HS, PR, LS methods and guarantee the global convergence of DY, FR, CD methods.

Theorem 2. *Consider the nonlinear CG method with direction vectors $s_1 = -g_1$ and (7), (8), $i \in \mathbb{N}$. Let the parameter β_i be given by (3) and the generalized Wolfe conditions (1) with $0 < \varepsilon_3 < \infty$ be used. If the objective function F is Lipschitz continuous and bounded from below on $\mathbb{R}^n$, then the nonlinear CG method is globally convergent.*

Further nonlinear CG methods with descent property can be obtained using the following lemma, where symbols s_+, g_+ denote s_{i+1}, g_{i+1} and s, z denote s_i, z_i.

Lemma 1. *Let $s_+ = -\vartheta g_+ + \beta s$, where $0 < \underline{\vartheta} \le \vartheta \le \overline{\vartheta}$ and*

$$\beta = g_+^T z - \frac{\lambda}{\vartheta} z^T z g_+^T s. \qquad (9)$$

If $z \in \mathbb{R}^n$ is an arbitrary nonzero vector and $1/4 < \underline{\lambda} \le \lambda \le \overline{\lambda}$, then

$$-\|g_+\|\|s_+\| \le g_+^T s_+ \le -\underline{s}\,\|g_+\|^2, \qquad \underline{s} = \underline{\vartheta}\left(1 - \frac{1}{4\underline{\lambda}}\right) > 0,$$

so that $\|s_+\| \ge \underline{s}\|g_+\|$.

Vector z is chosen in such a way that the first member in (9) corresponds to some basic nonlinear CG method. Let $\vartheta = 1$. Using vectors $z = y/y^T s$, $z = y/g^T g$, $z = y/|g^T s|$ in Lemma 1, we obtain the descent modification of HS, PR, LS methods with

$$\beta^{HSD} = \beta^{HS} - \lambda\frac{y^T y g_+^T s}{(y^T s)^2}, \quad \beta^{PRD} = \beta^{PR} - \lambda\frac{y^T y g_+^T s}{(g^T g)^2}, \quad \beta^{LSD} = \beta^{LS} - \lambda\frac{y^T y g_+^T s}{(g^T s)^2}.$$

Using vectors $z = g_+/y^T s$, $z = g_+/g^T g$, $z = g_+/|g^T s|$ in Lemma 1, we obtain the descent modification of DY, FR, CD methods with

$$\beta^{DYD} = \beta^{DY} - \lambda\frac{g_+^T g_+ g_+^T s}{(y^T s)^2}, \quad \beta^{FRD} = \beta^{FR} - \lambda\frac{g_+^T g_+ g_+^T s}{(g^T g)^2}, \quad \beta^{CDD} = \beta^{CD} - \lambda\frac{g_+^T g_+ g_+^T s}{(g^T s)^2}.$$

Theorem 3. *Consider the nonlinear CG method with direction vectors $s_1 = -g_1$ and $s_{i+1} = -g_{i+1} + \beta_i s_i$, $i \in \mathbb{N}$. Let $\beta_i = \beta_i^{HSD}$ or $\beta_i = \beta_i^{LSD}$ with $1/4 < \underline{\lambda} \le \lambda_i \le \overline{\lambda}$ and the generalized Wolfe conditions (1) with $0 < \varepsilon_3 < \infty$ be used. If the objective function F is uniformly convex, Lipschitz continuous and bounded from below on $\mathbb{R}^n$, then the nonlinear CG method is uniformly descent and, therefore,globally convergent.*

The idea of the proof of the above theorem cannot be used for the PRD method, since the upper bound for $|g_{i+1}^T s_{i+1}|/\|g_{i+1}\|^2$ is unavailable. Setting $\lambda_i = 2$ in the HSD method, we obtain the Hager–Zhang method with

$$\beta_i^{HZ} = \beta_i^{HS} - 2\frac{y_i^T y_i g_{i+1}^T s_i}{(y_i^T s_i)^2}.$$

Setting $\lambda_i = \rho_i y_i^T d_i / y_i^T y_i$ in the HSD method, we obtain the Dai–Liao method with

$$\beta_i^{DL} = \beta_i^{HS} - \rho_i \frac{g_{i+1}^T d_i}{y_i^T s_i}.$$

Further useful nonlinear CG methods can be obtained by the modification of the numerator in (2). In such a way we obtain coefficients

$$\beta_i^{HSM} = \frac{\hat{y}_i^T g_{i+1}}{y_i^T s_i}, \quad \beta_i^{PRM} = \frac{\hat{y}_i^T g_{i+1}}{g_i^T g_i}, \quad \beta_i^{LSM} = \frac{\hat{y}_i^T g_{i+1}}{|g_i^T s_i|}, \tag{10}$$

where

$$\hat{y}_i = g_{i+1} - \frac{\|g_{i+1}\|}{\|g_i\|} g_i \quad \Rightarrow \quad \hat{y}_i^T g_{i+1} = \|g_{i+1}\|^2 - \frac{\|g_{i+1}\|}{\|g_i\|} g_{i+1}^T g_i.$$

Since $|g_{i+1}^T g_i| \le \|g_{i+1}\| \, \|g_i\|$, one has

$$0 \le \|g_{i+1}\|^2 - \frac{\|g_{i+1}\|}{\|g_i\|} g_{i+1}^T g_i \le 2\|g_{i+1}\|^2,$$

or $0 \le \hat{y}_i^T g_{i+1} \le 2\|g_{i+1}\|^2$.

Theorem 4. *Values β_i^{HSM}, β_i^{PRM}, β_i^{LSM} satisfy the inequalities $0 \le \beta_i^{HSM} \le 2\beta_i^{DY}$, $0 \le \beta_i^{PRM} \le 2\beta_i^{FR}$, $0 \le \beta_i^{LSM} \le 2\beta_i^{CD}$. Assume that the strong Wolfe conditions (1) with $0 < \varepsilon_3 = \varepsilon_2$ hold. If $\varepsilon_2 < 1/2$, the LSM method is descent. If $\varepsilon_2 < 1/3$, the HSM method is descent. If $\varepsilon_2 < 1/4$, the PRM method is descent.*

Nonlinear CG methods can be also derived by using variable metric updates. The one step limited memory BFGS method has the form

$$\begin{aligned} s_{i+1} &= -H_{i+1} g_{i+1}, \qquad H_{i+1} y_i = \rho_i d_i, \\ H_{i+1} &= I + \left(\frac{y_i^T y_i}{y_i^T d_i} + \rho_i \right) \frac{d_i d_i^T}{y_i^T d_i} - \frac{y_i d_i^T + d_i y_i^T}{y_i^T d_i}, \end{aligned}$$

where $d_i = x_{i+1} - x_i = \alpha_i s_i$, $y_i = g_{i+1} - g_i$ a $\rho_i > 0$. If $d_i^T g_{i+1} = 0$, we can write

$$\begin{aligned} s_{i+1} &= -g_{i+1} - \left(\frac{y_i^T y_i}{y_i^T d_i} + \rho_i \right) \frac{d_i^T g_{i+1}}{y_i^T d_i} d_i + \frac{d_i^T g_{i+1}}{y_i^T d_i} y_i + \frac{y_i^T g_{i+1}}{y_i^T d_i} d_i \\ &= -g_{i+1} + \frac{y_i^T g_{i+1}}{y_i^T s_i} s_i = -g_{i+1} + \beta_i^{HS} s_i. \end{aligned}$$

Of course, we can omit only selected members containing $d_i^T g_{i+1}$. Setting

$$s_{i+1} = -g_{i+1} + \rho_i \frac{d_i^T g_{i+1}}{y_i^T d_i} d_i - \frac{y_i^T g_{i+1}}{y_i^T d_i} d_i$$

and $\rho_i = 1$, we obtain the HSP method (Perry's modification of the HS method) introduced in (4).

Further interesting nonlinear CG methods can be derived by using the generalized quasi-Newton condition. The generalized QN condition can be written in the form

$$s_{i+1} = -H_{i+1} g_{i+1}, \qquad H_{i+1} y_i = \rho_i d_i \quad \Rightarrow \quad y_i^T s_{i+1} = -\rho_i d_i^T g_{i+1}.$$

Since $s_{i+1} = -g_{i+1} + \beta_i s_i$, this condition is satisfied for $\beta_i = \beta_i^{DL}$, where

$$\beta_i^{DL} = \frac{y_i^T g_{i+1} - \rho_i d_i^T g_{i+1}}{y_i^T s_i} = \beta_i^{HS} - \rho_i \frac{d_i^T g_{i+1}}{y_i^T s_i}.$$

This way leads to the Dai–Liao modifications

$$\beta^{HSDL} = \beta^{HS} - \rho \frac{g_+^T d}{y^T s}, \quad \beta^{PRDL} = \beta^{PR} - \rho \frac{g_+^T d}{g^T g}, \quad \beta^{LSDL} = \beta^{LS} - \rho \frac{g_+^T d}{|g^T s|} \qquad (11)$$

(here $\beta^{HSDL} = \beta^{DL}$).

Nonlinear CG methods can be improved by using restarts. In this case, we set $\beta_i = 0$ if the prescribed condition is not satisfied. The uniform descent condition

$$-g_{i+1}^T s_{i+1} \geq \underline{\eta} \|g_{i+1}\| \|s_{i+1}\|$$

(where, e.g., $\underline{\eta} = 10^{-8}$) guarantees the global convergence of the CG method. The uniform conjugacy condition

$$|y_i^T s_{i+1}| \leq \overline{\eta} \|s_{i+1}\| \|y_i\|$$

(where, e.g., $\overline{\eta} = 5.10^{-2}$) improves efficiency of methods DY, FR, CD and their modifications.

In the tables introduced below, we demonstrate efficiency of selected nonlinear CG methods. The first table contains results obtained using the collection of 73 problems with 10000 variables (TEST 12, `http://camo.ici.ro/neculai/ansoft.htm`). The second table contains results obtained using the collection of 73 problems with 1000 variables (TEST 25, `http://www.cs.cas.cz/luksan/test.html`). In these tables, NIT is the total number of iterations, NFV is the total number of function evaluations and TIME is the total CPU time. At the same time, M denotes basic methods (2), (3), (4), MS denotes modifications (6), MT denotes modifications (6), MI denotes modifications (7), MD denotes modifications based on Lemma 1, MM denotes modifications (10) a MDL denotes modifications (11). Moreover, character + means that value β_i is replaced by $\max(0, \beta_i)$ and character $*$ means that values β^{HS+}, β^{PR+}, β^{LS+} are used in the formulas for β^{HSDL}, β^{PRDL}, β^{LSDL}.

Further details and references can be found in Chapter 3 of the report V1152-14, which is available at `http://www.cs.cas.cz/luksan/lekce4.pdf`.

	Methods of HS type		Methods of PR type		Methods of LS type	
Method	NIT - NFV	TIME	NIT - NFV	TIME	NIT - NFV	TIME
M	73500 - 146562	45.5	97522 - 153458	52.5	90844 - 182707	59.2
M+	64776 - 130153	42.2	99012 - 199048	52.2	109072 - 217871	59.1
MS+	64267 - 127877	39.4	81135 - 162484	46.9	98472 - 197386	54.6
MI+	64776 - 130153	42.2	59242 - 118194	37.5	92908 - 185231	49.8
MT+	56465 - 113023	37.8	66533 - 132821	41.1	69851 - 139348	41.5
MD+	63923 - 128143	42.0	93105 - 187343	51.3	70265 - 140260	41.7
MDL∗	70630 - 138223	46.7	89794 - 180989	50.8	106829 - 214506	65.0
MM	63761 - 127077	38.2	69206 - 139422	40.1	98169 - 196718	48.2
	Methods of DY type		Methods of FR type		Methods of CD type	
Method	NIT - NFV	TIME	NIT - NFV	TIME	NIT - NFV	TIME
M	72624 - 145100	47.1	81152 - 162513	48.0	87805 - 176088	63.7
MS	85372 - 161985	57.9	84886 - 170639	68.1	69839 - 140434	42.2
MI	72624 - 145100	47.1	70155 - 141153	49.1	83105 - 166870	49.6
MT	85249 - 169741	51.1	84001 - 175873	63.8	88634 - 184105	76.1
MD+	84267 - 170722	52.5	82341 - 164020	61.3	75449 - 151144	46.6
	Methods of HSP type		Methods of PRP type		Methods of LSP type	
Method	NIT - NFV	TIME	NIT - NFV	TIME	NIT - NFV	TIME
M	94217 - 189553	99.9	98579 - 195634	52.3	89764 - 168900	55.3
M+	75175 - 150631	46.9	65729 - 132372	40.6	85626 - 164338	48.9
MS+	63356 - 126299	39.6	65561 - 131168	41.6	84874 - 170016	50.0
MI+	75175 - 150631	47.0	66181 - 133055	43.6	68377 - 136899	43.8
MT+	67290 - 136028	48.5	69115 - 138680	44.0	66094 - 132739	44.0
MD+	80467 - 154308	51.4	71019 - 143753	47.6	87721 - 165860	53.5

	Methods of HS type		Methods of PR type		Methods of LS type	
Method	NIT - NFV	TIME	NIT - NFV	TIME	NIT - NFV	TIME
M	182719 - 362799	44.5	193715 - 382239	48.7	186195 - 365074	47.9
M+	181090 - 357804	45.0	194625 - 385349	47.3	171949 - 339448	38.5
MS+	176027 - 348089	44.7	180893 - 356713	45.4	181363 - 357095	46.6
MI+	181090 - 357804	45.0	192212 - 377671	49.0	182165 - 358848	46.9
MT+	179137 - 354722	39.9	166227 - 327249	36.1	172590 - 339757	37.4
MD+	189405 - 372372	48.8	200779 - 394240	49.9	182981 - 361565	45.5
MDL∗	185031 - 366172	45.1	196460 - 583719	51.0	188953 - 373247	45.5
MM	175646 - 346092	45.7	188722 - 373911	47.4	190902 - 376303	46.4
	Methods of HSP type		Methods of PRP type		Methods of LSP type	
Method	NIT - NFV	TIME	NIT - NFV	TIME	NIT - NFV	TIME
M	180076 - 356292	45.4	183742 - 362432	47.0	194143 - 381417	48.1
M+	174388 - 344643	44.3	173541 - 342704	38.3	204033 - 397429	51.3
MS+	185629 - 366182	46.0	185214 - 365439	47.7	181322 - 358282	45.8
MI+	174388 - 344643	44.3	175264 - 346739	44.9	183064 - 361115	45.4
MT+	174902 - 345601	38.8	163751 - 322111	35.8	178082 - 349536	38.7
MD+	190318 - 374073	42.5	191386 - 377971	48.0	183564 - 361783	45.7
MDI+	190318 - 374073	42.5	185624 - 367332	46.1	189522 - 373814	47.1

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

ACCELERATION OF LE BAIL FITTING METHOD ON PARALLEL PLATFORMS

Ondřej Mařík, Ivan Šimeček

Department of Computer Systems,
Faculty of Information Technology,
Czech Technical University in Prague,
Thákurova 9, 160 00 Praha 6, Czech Republic
xsimecek@fit.cvut.cz, marikond@fit.cvut.cz

Abstract

Le Bail fitting method is procedure used in the applied crystallography mainly during the crystal structure determination. As in many other applications, there is a need for a great performance and short execution time. In this paper, we describe utilization of parallel computing for mathematical operations used in Le Bail fitting. We present an algorithm implementing this method with highlighted possible approaches to its aforementioned parallelization. Then, we propose a sample parallel version using the OpenMP API and its performance results on the real multithreaded system. Further potential for the massive parallelization is also discussed.

1. Introduction

The crystal structure determination from powder diffraction is an important part of applied crystallography science and its detailed description is out of scope of this paper. However, certain basic principles are needed for better understanding of Le Bail fitting method and its application (for details see [2, 3]).

Starting from the most elementary knowledge, one type of matter in solid state is crystalline matter which has ordered, even periodical internal structure and therefore can be described by a single cell of this structure. This fact also simplifies the process of obtaining structure of an unknown sample because x-ray diffraction can be used to obtain complete (from single crystal) or partial (from crystalline powder) image of inner structure. Using powder for diffraction is becoming still more common because powder sample is usually much easier to obtain than a single crystal of studied substance. The main drawback of using powder is the need to extrapolate part of the structure information which is lost due to the random orientation of crystallites in sample. This fact (illustrated by Figure 1) requires great computing power available only relatively recently to solve structure in acceptable time.

An example of data obtained by powder diffraction is depicted in Figure 2. Most important information from diffraction profile (pattern) for further analysis are the

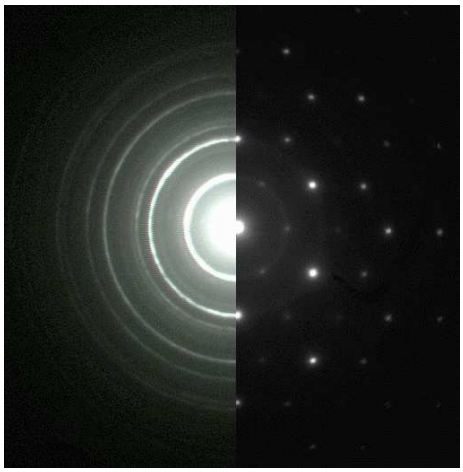


Figure 1: Diffraction pattern image of polycrystalline (left) and single crystal (right) of Cr_2O_3 (reprinted from [3])

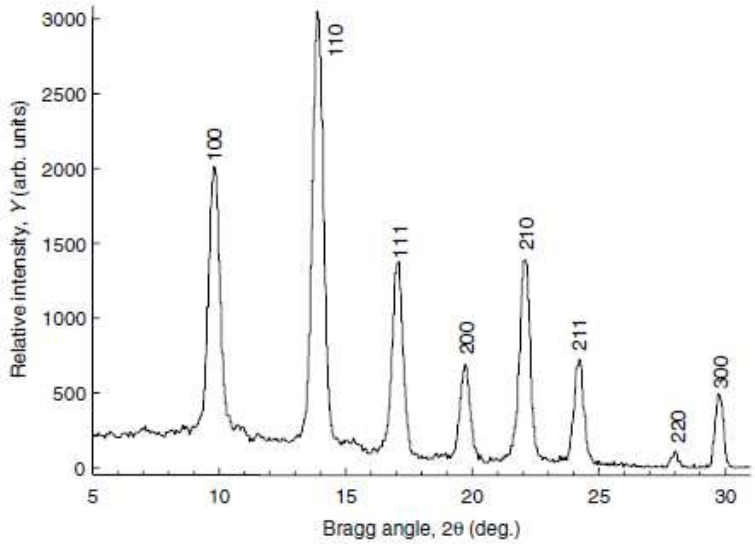


Figure 2: Powder diffraction pattern of LaB_6 with Bragg's peaks labeled by Miller indices of corresponding lattice plane sets (reprinted from [2])

peak positions and their integrated intensities. Part of the result of structure determination are the properties of unit cells, namely its size (a, b, c) and angles between them (α, β, γ). The relationship between these structure parameters and diffraction profile can be summarized based on Bragg's law into following equation:

$$2\Theta_{hkl} = 2 \arcsin \frac{\lambda}{2d_{hkl}}, \tag{1}$$

where peak position is denoted by 2Θ (diffraction angle), λ is wavelength of used radiation and d_{hkl} is interplanar distance, which can be calculated from structure parameters. For the simplification of equations, intermediary variable Q is usually used:

$$d_{hkl} = \sqrt{\frac{1}{Q_{hkl}}} \tag{2}$$

The calculation of Q itself then has to take into account level of symmetry present in crystal structure, for example in monoclinic crystal system it can be obtained as:

$$Q_{hkl} = \frac{h^2}{a^2 \sin^2 \beta} + \frac{k^2}{b^2} + \frac{l^2}{c^2 \sin^2 \beta} - \frac{2hl \cos \beta}{ac \sin^2 \beta} \tag{3}$$

The reason for relatively complicated and computationally complex determination of unit cell parameters from the powder diffraction profile is apparent from Equation (3). It is easy to calculate Q if the unit cell parameters are known but not vice versa (for details see [4, 5]). Here comes the advantage of parallel computing, the details of which will be explained later.

2. Le Bail fitting

During the crystal structure determination is it often needed to refine the estimated structure parameters to better fit the observed (measured) diffraction profile. Le Bail fitting can be used exactly for this purpose. Usually, a single sample can be evaluated by multiple methods, each providing slightly different results in the form of different unit cell parameters. Le Bail fitting is then used to further refine these results based on observed diffraction profile, to either increase their accuracy or to select the best estimates for the next step of structure determination.

Le Bail fitting itself is an iterative process which can be difficult to parallelize. The process can be crudely described as adjustment of structure parameters based on difference between observed diffraction profile and diffraction profile computed from current structure parameters estimates. This is achieved by the decomposition of diffraction profile into separate peaks with approximately Gaussian shape and then applying following operations on all peaks. The main idea behind profile decomposition is illustrated by Figure 3. This principle is of course applicable on an arbitrary number of peaks and variable profile size, which allows to regard the peaks as independent data and thus allow simple parallelization.

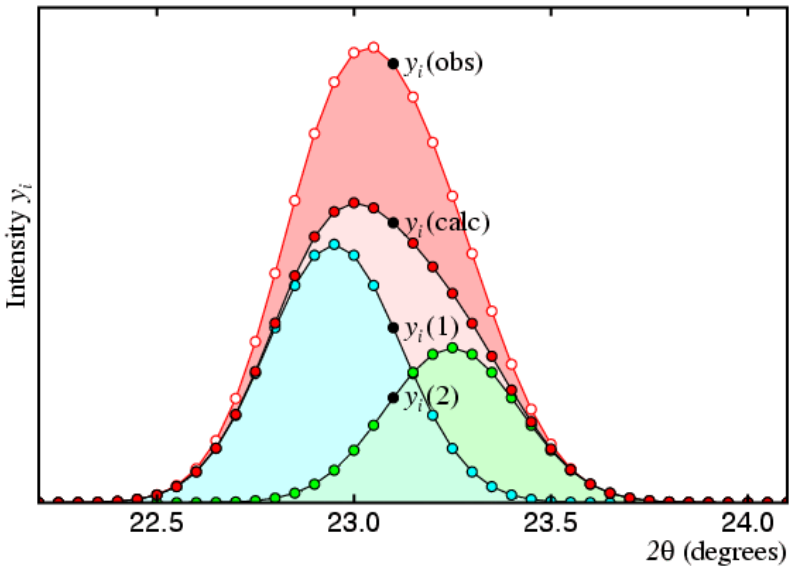


Figure 3: Pattern decomposition principle

Each iteration of Le Bail fitting consists of two main steps. The goal of the first one is to determine the integrated intensities of individual peaks in the calculated profile, basically performing the aforementioned decomposition. That itself is an iterative process of applying the following equation:

$$I_K(obs) = \sum_i \left(w_{i,K} \cdot S_K^2(obs) \cdot \frac{y_i(obs)}{y_i(calc)} \right), \quad \text{where} \tag{4}$$

- $I_K(obs)$ is the new integrated intensity of a peak, which is calculated for all peaks in profile,
- $w_{i,K}$ is the weight of considered point meaning its distance from currently calculated peak,
- $S_K^2(obs)$ is the integrated intensity obtained in previous iteration (or selected constant in the first iteration),
- $y_i(obs)$ is the observed value of y_i (single point in the profile),
- $y_i(calc)$ is the calculated value of y_i (single point in the profile).

While it may seem that all points in profile are considered for increments of each peak, in practice only points in certain neighbourhood of a peak position are actually involved in calculation and this fact is reflected by $w_{i,K}$.

The second step in each iteration of Le Bail fitting includes application of non-linear least squares method on the structure parameters. The detailed explanation of least squares is again out of scope of this document but it uses the fact that each

point of diffraction profile can be viewed as a result of certain function of structure parameters. As the set of equations for all points in profile is non-linear, the non-linear least squares method has to be used to adjust the structure parameters to better fit the calculated profile to the observed one. The usual solution can be expressed as:

$$\Delta\vec{x} = (\mathbf{A^T A})^{-1}\mathbf{A^T}\vec{y}, \quad \text{where} \tag{5}$$

- $\vec{x}$ is the result of this step (adjustments to the structure parameters),
- $\mathbf{A}$ is the Jacobi matrix (matrix of partial derivations with one row per point in profile),
- $\vec{y}$ is the vector of differences between observed and calculated diffraction profile.

The structure parameters adjustments are then applied on current parameters, diffraction profile replotted and another iteration starts if needed. The sequential algorithm basically follows the described process.

3. Parallelization using OpenMP API

There are several possible approaches to parallelizing Le Bail fitting algorithm. First and most obvious option offers the number of processed samples, that is sets of parameter estimates. Since these are independent, except in using common observed diffraction profile for reading only, each sample can be assigned to one process or thread.

Second approach is based on number of points into which is the diffraction profile discretized. This is applicable mostly in the first step in main iterations since the second step consists predominantly of matrix operations.

Another way to parallelize can be derived from a number of peaks into which the diffraction profile is decomposed. Even though it is usually less independent data than in previous options, it is nonetheless still a viable option.

Now that the possible parallelism in the algorithm was discussed, let's focus on the possible implementation tools. First step is usually to utilize full CPU capabilities which means to use threads or processes to run code in parallel. The OpenMP API can be used for this purpose. OpenMP API is defined by a collection of compiler directives, library routines and environment variables extending the C, C++ and Fortran languages [1]. These can be used to create portable parallel programs utilizing shared memory. It has the *fork-join* execution model meaning the program starts as single thread and certain blocks of code are run in separate threads. Shared memory implies requirements on memory management during implementation to avoid inconsistencies and undefined behaviour. Even though the changes in the parallel version using OpenMP are relatively small, the speed-up it provides is significant on systems with multiple CPUs or multi-core CPU as is apparent from Table 1 with data measured on single CPU Intel Core i5-2500@4 Ghz with 4 cores.

#data sets	40	80	120	160	200
Sequential (sec)	3.39	7.11	10.04	13.71	16.89
OpenMP@4 cores (sec)	1.23	2.37	3.30	4.16	4.99
Speedup	2.75	3.00	3.04	3.30	3.38

Table 1: Achieved performance of OpenMP accelerated version.

Moreover only parallelization based on number of processed samples is used, showing only the fraction of potential for more massively parallel platforms.

4. Conclusions

Le Bail fitting is a great example of acceleration of practical applications by the parallel computing. The multithreaded version using OpenMP API achieves a great performance and almost linear speedup. Massively parallel platforms like GPU (CUDA, OpenCL) will be able to enhance the application's performance even more, but at a cost of more extensive code changes.

Acknowledgements

This work was supported by grant No. SGS14/106/OHK3/1T/18 of the Czech Technical University in Prague.

References

[1] OpenMP Architecture Review Board: Openmp application program interface, online, 2013. URL `http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf`

[2] Pecharsky, V. K. and Zavalij, P. Y.: *Fundamentals of powder diffraction and structural characterization of materials*. Springer Science, New York, second edition, 2009.

[3] The Australian National University: Electron diffraction, online, 2009, URL `http://people.physics.anu.edu.au/ web107/research/highpage4.php`

[4] Šimeček, I.: A new approach for indexing powder diffraction data based on dichotomy method. In: *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*, CSE'2012, pp. 124–129, 2012, doi: 10.1109/ICCSE.2012.27.

[5] Šimeček, I.: A new approach for indexing powder diffraction data suitable for gpgpu execution. *Advances in Intelligent Systems and Computing*, 188 AISC, pp. 409–416, 2013. 7th International Conference on Soft Computing Models in Industrial and Environmental Applications.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

COMPARISON OF CRACK PROPAGATION CRITERIA IN LINEAR ELASTIC FRACTURE MECHANICS

Karel Mikeš

Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
karel.mikes.1@fsv.cvut.cz

Abstract

In linear fracture mechanics, it is common to use the local Irwin criterion or the equivalent global Griffith criterion for decision whether the crack is propagating or not. In both cases, a quantity called the stress intensity factor can be used. In this paper, four methods are compared to calculate the stress intensity factor numerically; namely by using the stress values, the shape of a crack, nodal reactions and the global energetic method. The most accurate global energetic method is used to simulate the crack propagation in opening mode. In mixed mode, this method is compared with the frequently used maximum circumferential stress criterion.

1. Introduction

The description of crack propagation is one of the most important ingredients of linear elastic fracture mechanics (LEFM). The main questions are: At which loading level will the crack propagation begin and in which direction will the crack propagate?

The aim of this paper is to compare numerical implementations of most frequently used crack propagation criteria for opening mode and mixed mode in 2D.

2. Stress intensity factor concept

The stress intensity factor is a quantity used in LEFM to describe the asymptotic singular stress field near the crack tip. The stress in the vicinity of the crack tip is unbounded and grows in inverse proportion to the square root of distance from the tip. Under plane stress, the asymptotic stress field is described by

$$\sigma_x(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1-\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) - \frac{K_{II}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\left(2-\cos\frac{\theta}{2}\cos\frac{3\theta}{2}\right), \quad (1)$$

$$\sigma_y(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1+\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) + \frac{K_{II}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2}, \quad (2)$$

$$\tau_{xy}(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} - \frac{K_{II}}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1-\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right), \quad (3)$$

where K_I and K_{II} are the stress intensity factors for modes I and II which represent the loading and geometry conditions, r is the distance from the crack tip and θ is the polar angle; see e.g. [7].

3. Crack propagation in mode I (opening mode)

In mode I, the crack is opening without sliding. Therefore, we can assume that the crack will propagate in the original direction and we have to decide when the propagation starts.

3.1. Local Irwin criterion

This concept was introduced by Irwin [4] in 1957. The stress intensity factor is used to decide about the crack propagation. The propagation will start when the value of the stress intensity factor K_I reaches its critical value so-called fracture toughness denoted by K_c .

3.2. Global Griffith criterion

This criterion was introduced by Griffith [3] in 1920. The crack will grow if a sufficient amount of energy is released by its propagation. The criterion is based on the strain energy release rate, defined as

$$\mathcal{G}(u,a) = -\frac{1}{t}\frac{\partial W_e(u,a)}{\partial a}, \tag{4}$$

where $W_e(u,a)$ is the elastic strain energy considered as a function of the imposed displacement u and the crack length a. The beam thickness is denoted by t; see Figure 1.

Under plane stress and in mode I, both criteria are equivalent [2]. We can write

$$\mathcal{G}(u,a) = \frac{{K_I}^2}{E}. \tag{5}$$

The rules for crack propagation according to the local Irwin and global Griffith criteria are summarized in Table 1, where K_c resp. G_f are material properties called fracture toughness [$\mathrm{Nm}^{-3/2}$] and fracture energy [Nm^{-1}], resp.

Local criterion	**Global criterion**		**Crack behaviour**
$K_I < K_c$	$\mathcal{G} < G_f$	$\Rightarrow$	no crack propagation
$K_I = K_c$	$\mathcal{G} = G_f$	$\Rightarrow$	crack propagation
$K_I > K_c$	$\mathcal{G} > G_f$	$\Rightarrow$	inadmissible (in statics)

Table 1: Crack propagation rules according to the local and global criteria

3.3. Simulation in opening mode

Four methods have been used to calculate the stress intensity factor or the strain energy release rate in opening mode; namely by using (i) the stress values, (ii) the crack opening, (iii) nodal forces and (iv) the release of strain energy. The first three methods have a local character and deal with the values near the crack tip to calculate the stress intensity factor. The fourth method evaluates the change of the energy of the whole beam when the crack is extended. Three different types of triangular finite elements have been used for each method; namely (i) three-node element with linear approximation of displacement, (ii) six-node element with quadratic approximation and (iii) six-node quadratic element with singular shape functions on the edges starting from the crack tip; see Figure 2. Numerical simulations have been performed using the open-source finite element code OOFEM [6]. The relative error of all methods in a three-point bending test with geometry according to Figure 1 have been evaluated by comparing the computed value of the stress intensity factor with the ,,exact" values obtained by using approximate analytic formulas available in [2], [5] and [7]. The relative errors of all methods for all types of elements are shown in Table 2.

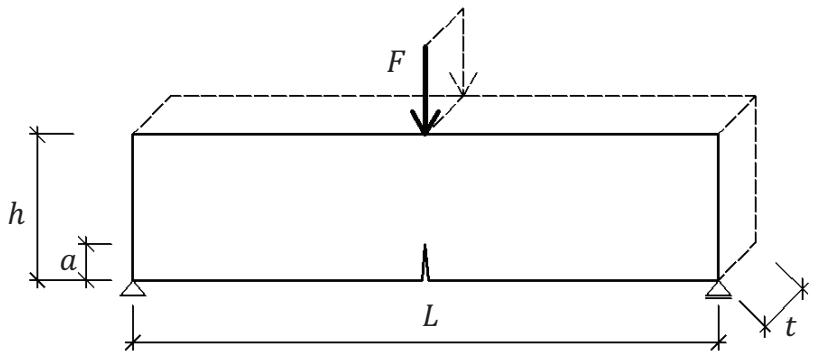


Figure 1: Geometry of the three-point bending test

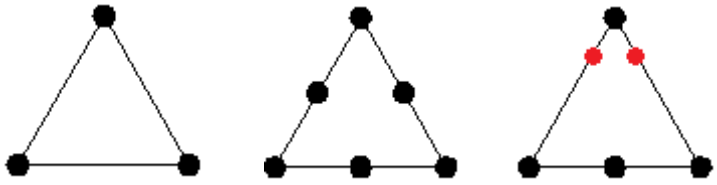


Figure 2: Used finite elements: linear (left), quadratic (middle), modified singular quadratic (right)

	Linear	Quadratic	Singular quadratic
Values of stress	> 30 %	10 - 30 %	10 - 30 %
Shape of a crack	< 10 %	2 %	1 %
Node reactions	5 - 10 %	< 5 %	< 5 %
Energetic method	5 %	2 %	0.5 %

Table 2: The relative errors of the computed stress intensity factor K_I for different methods and different finite elements

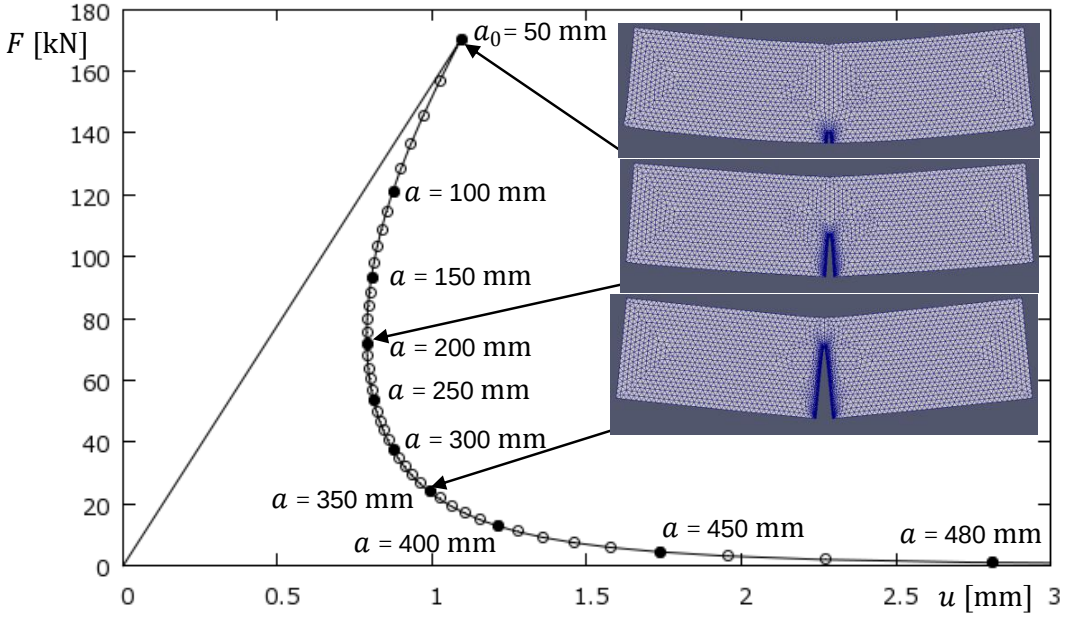


Figure 3: Force-displacement diagram of the three-point bending test

The global energy method using the modified quadratic elements is the most accurate but also the most time-consuming one. Figure 3 shows the load-displacement diagram obtained with this method for a beam of the following geometry: height $h = 0.5$ m, length $L = 2$ m, thickness $t = 0.2$ m, initial crack length $a_0 = 0.05$ m, elastic modulus $E = 20$ GPa, Poisson ratio $\nu = 0.2$ and fracture toughness $K_c = 4$ MNm$^{-3/2}$.

4. Crack propagation in mixed mode

The mixed mode represents a combination of tensile opening and in-plane shear. The direction of crack propagation is not known in advance and has to be determined by a suitable criterion.

4.1. Maximum circumferential stress criterion (MCSC)

This criterion determines the crack propagation direction based on the maximal circumferential stress σ_θ, which is defined as

$$\sigma_\theta(r,\theta) = \sigma_y \cos^2\theta + \sigma_x \sin^2\theta - 2\tau_{xy}\sin\theta\cos\theta. \tag{6}$$

Substituting from (1)–(3), we obtain

$$\sigma_\theta(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}\cos^3\frac{\theta}{2} - 3\frac{K_{II}}{\sqrt{2\pi r}}\cos^2\frac{\theta}{2}\sin\frac{\theta}{2}. \tag{7}$$

The angle θ with maximal circumferential stress (MCSC1) is obtained by solving the equation

$$\sin\theta + \frac{K_{II}}{K_I}(3\cos\theta - 1) = 0 \tag{8}$$

with the conditions

$$\theta \in (-\pi,\pi); \qquad K_I > 0; \qquad K_{II}\sin\frac{\theta}{2} < 0, \tag{9}$$

where the ratio K_{II}/K_I is obtained by fitting (1)–(3) to the values of the stress field in a number of Gauss points near the crack tip.

Another approach (referred to as MCSC2) is based on substituting the values of the stress field at Gauss points into the original definition of circumferential stress (5). After smooth of these data by a polynomial function, the angle θ that maximizes this function can be found.

Both approaches give almost the same results; see Figure 4. The first method (MCSC1) turned out to be numerically preferable and therefore is used in the following examples.

4.2. Maximum strain energy release rate criterion (MSERRC)

This criterion determines the crack propagation in the direction that leads to the maximum strain energy release rate defined in (4). Numerical realization consists in simulation of a number of sufficiently small crack extensions in several different directions. For each direction, the strain energy release rate is evaluated by subtracting the final strain energy from the original one and dividing by the increment of the crack area. The obtained values are smoothed using a polynomial function, for which the maximum is then found.

This criterion predicts, in most cases, similar crack trajectories to the MCSC. However, application to the three-point bending test with an eccentric initial crack leads to a different crack path; see Figure 4.

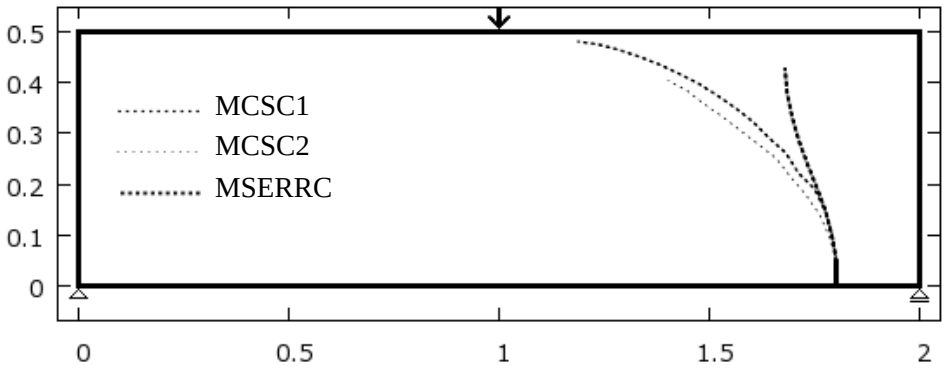


Figure 4: Comparison of crack paths according to different criteria for the three-point bending test with an eccentric initial crack

4.3. Comparative example

The last example is taken from [1]. It is a rectangular panel with two holes and two initial cracks subjected to tension in the vertical direction. In this example, both criteria lead to almost the same crack paths and the results are similar to those from [1]; see Figure 5. The load-displacement diagrams are depicted in Figure 6. Both criteria predict the same behaviour but the curve obtained with MCSC is not smooth. This means that MCSC is less accurate when used to decide whether the crack is propagating or not.

5. Conclusion

In the opening mode, the best results are obtained by the global energy method. In the mixed mode that arises in the three-point bending test, the MSERRC criterion

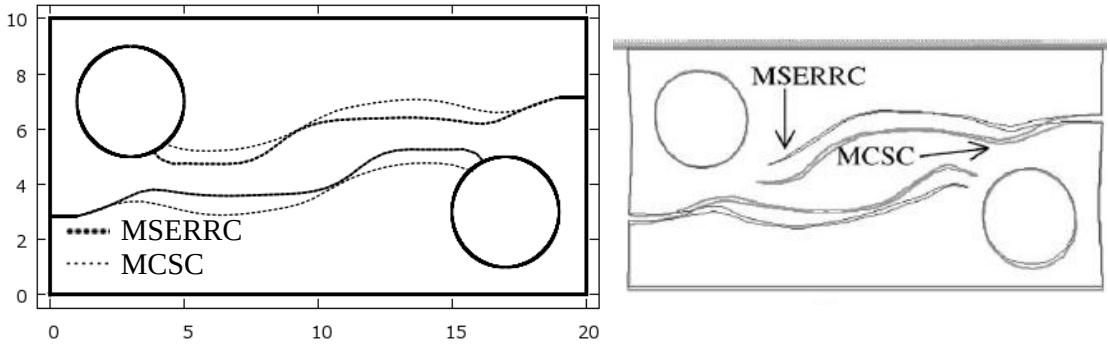


Figure 5: Comparison of crack paths according to different criteria in a vertical tensile test. Simulation in OOFEM (left) against results taken from [1] (right).

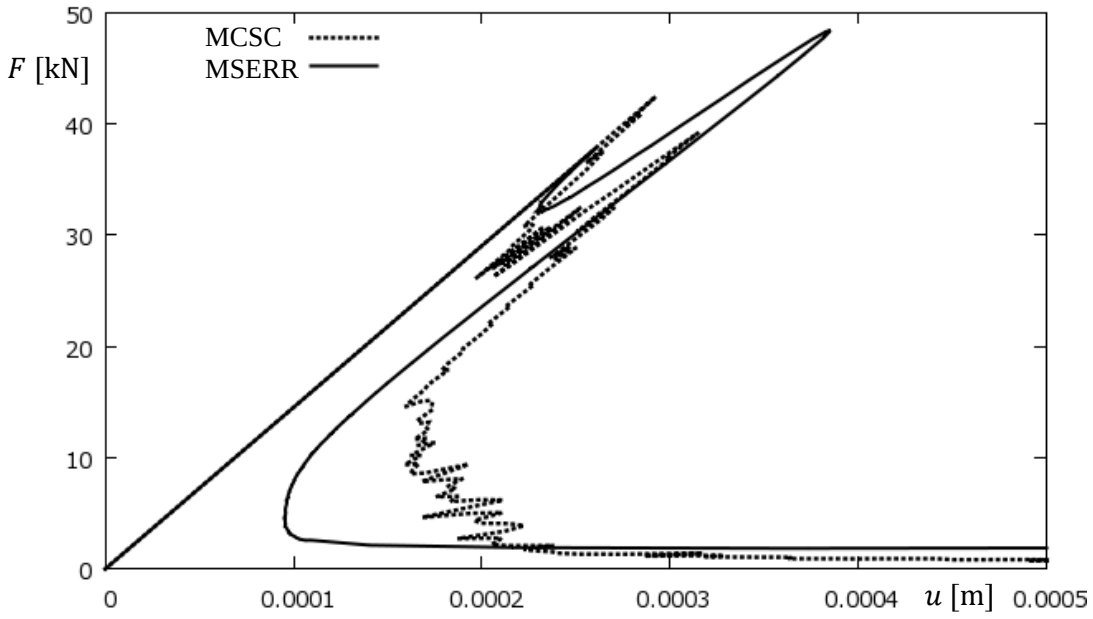


Figure 6: Comparison of load-displacement diagrams for different criteria

based on this method leads to a different crack path than the MCSC criterion using the circumferential stress. But both MSERRC and MCSC give similar results in other examples in mixed mode. Both criteria seem to be accurate in prediction of the propagation angle, but to determine whether the crack propagates it is more appropriate to use MSERRC.

Acknowledgements

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/029/OHK1/1T/11.

References

[1] Bouchard, P. O., Bay, F., and Chastel. Y.: Numerical modelling of crack propagation: automatic remeshing and comparison of different criteria. Comput. Methods Appl. Mech. Engrg. **192** (2003), 3887–3908.

[2] Broek, D.: *Elementary engineering fracture mechanics*. Martinus Nijhoff Publishers, Hague, 1984.

[3] Griffith, A. A.: The phenomena of rupture and flow in solids. Philos. Trans. R. Soc. Lond. **221** (1921), 163–198.

[4] Irwin, G. R.: Analysis of stresses and strains near the end of a crack traversing a plate. J. Appl. Mech. **24** (1957), 361–364.

[5] Jirásek, M. and Zeman, J.: *Přetváření a porušování materiálů*. ČVUT, Praha, 2008.

[6] Patzák, B.: OOFEM – an object-oriented simulation tool for advanced modeling of materials and structures. Acta Polytechnica **52** (2012), 59–66.

[7] Wang, C. H.: *Introduction to fracture mechanics*. DSTO Aeronautical and Maritime Research Laboratory, Melbourne, 1996.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

THE USE OF GRAPHICS CARD AND NVIDIA CUDA ARCHITECTURE IN THE OPTIMIZATION OF THE HEAT RADIATION INTENSITY

Jaroslav Mlýnek[1], Radek Srb[2], Roman Knobloch[1]

[1] Department of Mathematics and Didactics of Mathematics, FP
jaroslav.mlynek@tul.cz, roman.knobloch@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

The paper focuses on the acceleration of the computer optimization of heat radiation intensity on the mould surface. The mould is warmed up by infrared heaters positioned above the mould surface, and in this way artificial leathers in the automotive industry are produced (e.g. for car dashboards). The presented heating model allows us to specify the position of infrared heaters over the mould to obtain approximately even heat radiation intensity on the whole mould surface. In this way we can obtain the uniform material structure of artificial leather. The gradient methods are not suitable to optimize the position of heaters because the minimized function contains many local extremes. Therefore, we used an evolutionary algorithm, specifically the differential evolution algorithm. In this case the optimization procedure needs a lot of operations (especially when the mould volume is large and we use a large number of heaters). A substantial acceleration of the calculation can be achieved by parallel programming using a graphic card and nVidia CUDA architecture. The numerical calculations were performed by the Matlab code written by the authors and were run on a standard PC.

1. Introduction

The article describes the application of parallel programming for the utilization of a graphic card and nVidia CUDA architecture on a standard PC when calculating heat radiation intensity on a shell nickel mould surface and optimization of heat radiation intensity. In practice, a nickel mould is at first preheated by infrared heaters located above the outer mould surface. Then the inner mould surface is sprinkled with a special PVC powder and the outer mould surface is continually heated by infrared heaters.

The goal of the optimization is to determine the position of heaters over the mould so that their position ensures approximately the same heat radiation intensity on the whole mould surface. In this way we obtain uniform material structure

and colour tone of the artificial leather. During the optimization process we have to avoid possible collisions of two heaters as well as a heater and the mould surface. Therefore, the optimization process is more complicated. The minimized function has many local extremes and it is not suitable to use gradient methods in the optimization process. We used an evolutionary algorithm, specifically the differential evolution algorithm. Evolutionary algorithms generally require a lot of operations and long computation time (especially if the mould volume is larger and we use a higher number of heaters). This was the main reason for the use of parallel programming techniques.

In the following part of the article we focus on the implementation of parallel algorithms using the Matlab Parallel Computing Toolbox. The solved technical problem of the heat radiation intensity optimization, the used mathematical model and the calculation of the heat radiation intensity on a mould surface are described in more detail in [1] and [2].

2. Mathematical model and optimization of heat radiation intensity

The heater and the warmed mould are represented in 3-dimensional Euclidean space E_3 using the Cartesian coordinate system (O, x_1, x_2, x_3) with basis vectors $e_1 = (1, 0, 0), e_2 = (0, 1, 0), e_3 = (0, 0, 1)$.

The heater is represented by a straight line segment with a given length. The position of every heater Z can be defined by the following 6 parameters $Z : (s_1, s_2, s_3, u_1, u_2, \varphi)$, where the first three parameters are coordinates of the heater centre, the following two parameters are the first two coordinates of the unit vector u of the heat radiation direction (the third coordinate is negative, i.e. the heater radiates "downward") and the last parameter is the angle φ between the vertical projection of unit vector r of the heater axis onto the x_1x_2-plane and the positive part of axis x_1 (the vectors u and r are orthogonal).

The outer mould surface P is described by elementary surfaces p_j, where $1 \leq j \leq N$. It holds that $P = \cup p_j$, where $1 \leq j \leq N$ and $int\ p_i \cap int\ p_j = \emptyset$ for $i \neq j$, $1 \leq i, j \leq N$. Each elementary surface is described by the centre of gravity $T_j = [t_1^j, t_2^j, t_3^j]$, by the unit outer normal vector $v_j = (v_1^j, v_2^j, v_3^j)$ at the point T_j (we suppose v_j faces "upwards "and therefore is defined through the first two components v_1^j and v_2^j) and by the area of elementary surface w_j. Each elementary surface can thus be defined by the following 6 parameters $p_j : (t_1^j, t_2^j, t_3^j, v_1^j, v_2^j, w_j)$.

Now, we briefly describe the numerical computation procedure 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 position of heaters, and I_{jl} the heat radiation intensity of the lth heater on the p_j elementary surface (I_{jl} is a constant value on the whole p_j in our model). Then the total radiation intensity I_j on the elementary surface p_j is given by the following relation

$$I_j = \sum_{l \in L_j} I_{jl} \ . \tag{1}$$

The producer of artificial leathers recommends the constant value of heat radiation intensity I_{rec} on the whole outer mould surface. We can define F (respectively $\tilde{F}$), the deviation of the heat radiation intensity, by the relation

$$F = \frac{\sum_{j=1}^{N} |I_j - I_{\mathrm{rec}}| w_j}{\sum_{j=1}^{N} w_j}, \quad \tilde{F} = \sqrt{\sum_{j=1}^{N} (I_j - I_{\mathrm{rec}})^2 w_j} \,. \tag{2}$$

Function F defined by relation (2) (and analogously function $\tilde{F}$) has many local extremes. As we stated in this chapter, the position of every heater is defined by 6 parameters. Therefore, $6M$ parameters are necessary to define the position of all M heaters. We will successively construct a population of individuals y in the differential evolution optimization algorithm. Every population includes NP individuals, where every individual y represents one possible position of heaters above the mould. The generated individuals are saved in the matrix $\mathbf{B}_{NP\times(6M+1)}$. Every row of this matrix represents one individual, y, and its evaluation, $F(y)$. We seek the individual $y_{\min} \in C$ satisfying the condition

$$F(y_{\min}) = \min\{F(y); y \in C\}, \tag{3}$$

where $C \subset E_{6M}$ is the examined set. Every element of C is formed by a set of $6M$ allowable parameters and this set defines just one position of the heaters above the mould. The identification of the individual $y_{\min}$ defined by relation (3) is not realistic in practice. But we are able to determine an optimized solution y_{opt}.

3. Differential evolution algorithm and use of parallel programming

Now we describe schematically the particular steps of the differential evolution algorithm named *DE/rand/1/bin* (for more details see [3]) which is applied to our problem and was programmed in Matlab code by the authors.

Differential evolution algorithm

Input: the initial individual y_1, population size NP, the number of used heaters M (dimension of the problem is $6M$), crossover probability CR, mutation factor f, the number of calculated generations NG.

Internal computation:

1. create an initial generation ($G = 0$) of NP individuals $y_i^G, 1 \le i \le NP$,
2.a) evaluate all the individuals y_i^G of the generation G (calculate $F(y_i^G)$ for every individual y_i^G), b) store the individuals y_i^G and their evaluations $F(y_i^G)$ into the matrix $\mathbf{B}$,
3. *repeat until* $G \le NG$
a) *for* $i := 1$ *step* 1 *to* NP *do*
   (i) randomly select index $k_i \in \{1, 2, \ldots, 6M\}$,
   (ii) randomly select indexes $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$,

where $r_t \neq i$ for $1 \leq t \leq 3$ and
$r_1 \neq r_2, r_1 \neq r_3, r_2 \neq r_3$;

(iii) *for* $j := 1$ *step* 1 *to* $6M$ *do*
if $(rand(0,1) \leq CR$ *or* $j = k_i)$ *then*
$y_{i,j}^{\text{trial}} := y_{r_3,j}^G + f\left(y_{r_1,j}^G - y_{r_2,j}^G\right)$
else
$y_{i,j}^{\text{trial}} := y_{i,j}^G$
end if
end for (j)

(iv) *if* $F\left(y_i^{\text{trial}}\right) \leq F\left(y_i^G\right)$ *then* $y_i^{G+1} := y_i^{\text{trial}}$
else
$y_i^{G+1} := y_i^G$

end for (i),

b) store individuals y_i^{G+1} and their evolutions $F\left(y_i^{G+1}\right)$ $(1 \leq i \leq NP)$ of the new generation $G+1$ into the matrix $\mathbf{B}$, $G := G+1$
end repeat.

Output:
the row of matrix $\mathbf{B}$ that contains the corresponding value $\min\{F(y_i^G); y_i^G \in \mathbf{B}\}$ represents the best found individual y_{opt}.
Note that function $rand(0,1)$ randomly chooses a number from the interval $\langle 0, 1\rangle$. The denomination $y_{i,j}^G$ means the jth component of an individual y_i^G in the Gth generation. The individual y_{opt} is the final optimized solution and includes information about the position of each heater.

The parallel programming tools (using the graphics card and nVidia CUDA architecture, see [4]) can be successfully applied in the optimization process. Randomly generated individuals y and their evaluation $F(y)$ (given by relation (2)) are completely independent. Therefore, it is appropriate to use *the central processing unit* (CPU) *for parallel computing* during the creation of a new generation G of individuals y. Calculation of the function value $F(y)$ given by relation (2) of a new individual y is numerically rather demanding. We gradually calculate the heat radiation intensity I_j at each elementary surface p_j given by relation (1). In doing so, we use the experimentally measured values of the heat radiation intensity in the neighbourhood of the heater when calculating the value I_{jl} in the relation (1). Calculations of heat radiation intensities I_j and I_k on different elementary surfaces p_j and p_k are completely independent. Thus we conveniently use the *graphics processing unit* (GPU) *for parallel computing* when evaluating the relation (1). It is tested whether two different heaters of individual y are in a collision or the heater and mould surface are in collision. If a collision occurs, the individual y is penalized and value $F(y)$ is significantly increased. The determination of heater collisions with two different elementary surfaces p_j and p_k is entirely independent and we also use GPU for parallel computing during testing possible collisions of all heaters (the position of heaters is given by individual y) with the mould surface.

4. Practical examples of the use of parallelization

We made calculations on a PC with CPU: IntelCore i7-3770 CPU @3.4 GHz, RAM: 32 GB and GPU: GeForce GTX 460. We choose the common input parameters of the algorithm in the following examples (Example 1, Example 2): CR (crossover factor) = 0.98, f (mutation factor) = 0.60, NP (population size) = 200 individuals. The initial individual y_1 represents even distribution of the heaters over the mould and in the plane parallel with $xy-$plane and in distance 10[cm] over the mould surface. Type of heater: capacity 1,600 [W], length 15 [cm], width 4 [cm].

Example 1

The heated surface is a part of a spherical surface, sphere centred at the origin of the coordinate system, radius of the sphere $r = 0.4$[m], the ground plan of the surface is 0.5×0.5[m^2], I_{rec} (recommended heat radiation intensity) = 68[kW/m^2], M(number of heaters) = 16, N (number of elementary surfaces) = 1,000, NG (number of calculated generations) = 10,000 . The value of the function F for the initial individual y_1 is $F(y_1) = 20.87$. We received y_{opt} with deviation $F(y_{\text{opt}}) = 1.72$ after creation of 10,000 generations. The position of heaters over the testing surface corresponding to the individual y_{opt} is shown on the left-hand side of Figure 1.

Real times of the calculations y_{opt} for different default parameters are shown in Table 1. The first column includes different numbers of elementary surfaces (N). The following columns contain the corresponding times of calculations when using ordinary calculation (column labelled CPU), GPU (labelled CPU+GPU), CPU with quad-core (labelled CPUPAR) and simultaneous use of a CPU with quad-core and GPU (labelled CPUPAR+GPU). The values given in parentheses from the third to the fifth column indicate the reduction of time calculation relative to the corresponding ordinary calculation. Time-consuming calculations in the table were estimated based on the average duration of one generation calculating.

Example 2

We will heat a shell nickel mould (see right-hand side part of Figure 1, this mould is used in production of artificial leathers for dashboards of passenger cars). The size of the mould is $1.5 \times 0.4 \times 0.4$[m^3], I_{rec} (recommended heat radiation intensity) = 68[kW/m^2], M(number of heaters) = 96, N (number of elementary surfaces) = 40,663, NG (number of calculated generations) = 20,000.

Number of elementary surfaces	CPU	CPU+GPU	CPUPAR	CPUPAR+ +GPU
10^3	9.98	7.03 (1.42x)	3.68 (2.71x)	2.87 (3.48x)
10^4	31.13	7.39 (4.21x)	10.39 (3.00x)	3.15 (9.90x)
10^5	261.37	9.80 (26.67x)	98.88(2.64x)	5.00 (52.26x)
10^6	2552.73	34.59 (73.81x)	1153.96 (2.21x)	27.24 (93.72x)

Table 1: Time ([h]) required for the optimization procedure for 10,000 generations

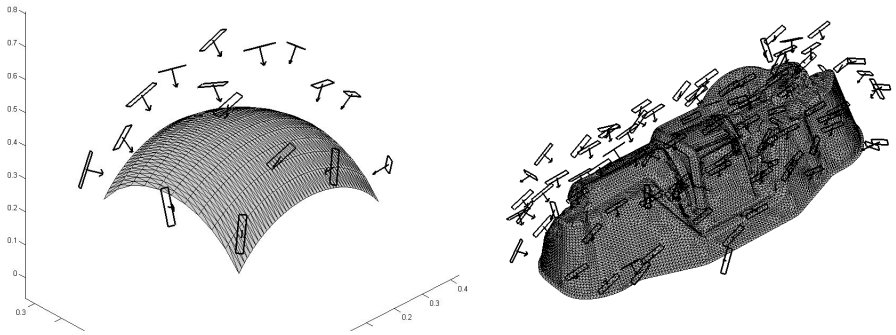


Figure 1: The position of the heaters corresponding to the individual y_{opt}.

The value of the function F for the initial individual y_1 is $F(y_1) = 40.14$. We received y_{opt} with deviation $F(y_{\text{opt}}) = 6.21$ after $20,000$ generations. The calculation lasted 41.06 hours using parallel computing (simultaneously GPU and CPU with quad-core). The calculation time with only "CPU" used would take 1,072 hours (44.66 days). The position of heaters over the mould surface corresponding to individual y_{opt} is shown on the right-hand side of Figure 1. The presented examples demonstrate that the computing time can be significantly reduced even on an ordinary PC with the use of parallel programming supported by a graphic card and nVidia CUDA architecture.

Acknowledgements

This work was supported by the grant SGS-FP-TUL 21049/2014 and by the grant SGS-FM-TUL, Technical University of Liberec.

References

[1] Mlýnek, J. and Srb, R.: The process of an optimized heat radiation intensity calculation on a mould surface. In: K. G. Troitzsch (Ed.), *Proc. of the 29th European Conference on Modelling and Simulation*, Digitaldruck Pirrot GmbH, Koblenz, Germany, pp. 461-467, May 2012.

[2] Mlýnek, J. and Srb, R.: The optimization of heat radiation intensity. In: J. Chleboun, K. Segeth (Eds.), *Proc. of the 16th Conference Programs and Algorithms of Numerical Mathematics*, Horní Maxov, pp. 142-148, June 2012.

[3] Price, K. V., Storn, R. M., and Lampien, J. A.: *Differential evolution.* Springer-Verlag Berlin, Heidelberg, 2005.

[4] Cook, S.: *CUDA programming: a developer's guide to parallel computing with GPUs.* Elsevier, Walthan, USA, 2013.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

WAVELETS AND PREDICTION IN TIME SERIES

Vratislava Mošová

Moravian College Olomouc
tř. Kosmonautů 1, 779 00 Olomouc
Czech Republic
vratislava.mosova@mvso.cz

Abstract

Wavelets (see [2, 3, 4]) are a recent mathematical tool that is applied in signal processing, numerical mathematics and statistics. The wavelet transform allows to follow data in the frequency as well as time domain, to compute efficiently the wavelet coefficients using fast algorithm, to separate approximations from details. Due to these properties, the wavelet transform is suitable for analyzing and forecasting in time series. In this paper, Box-Jenkins models (see [1, 5]) combined with wavelets are used to the prediction of a time series behavior. The described method is demonstrated on an example from practice in the conclusion.

1. Introduction

It is possible to get the first impression of a time series behavior from the line graph. However, the conclusions received are highly subjective. More accurate information can be provided for instance by the Box-Jenkins methodology. Box-Jenkins models use the fact that every time series $\{y_t \mid t = 1, \ldots, T\}$ is a realization of some stochastic process. Because such models are based on the stochastic nature of time series, correlations have important place in drawing them up. A prediction for the time series is then created on the basis of the mathematical model received. This paper deals with linking wavelets and Box-Jenkins models. The ability of wavelets to decorrelate data is then used to specify forecast in time series.

The contribution is divided into following parts: The description of standard Box-Jenkins models built is given in Section 2. Wavelets and their usage in forecasting time series are discussed in Section 3. The procedures described are presented in the example in Section 4.

2. Box-Jenkins models

The Box-Jenkins models are constructed for the stationary time series. It means that the mean value and the variance function are constant, the correlation and the covariation functions depend only on the time distance of random variables.

A special case of the stationary process is the series $\{a_t\}$ of uncorrelated random variables with constant mean value and constant variance function that is called the white noise. In what follows, suppose that every time series consists of an unsystematic component $\{a_t\}$ and of systematic components such as a trend, a seasonal component or a cyclical component.

The stationary Box-Jenkins process is denoted by ARMA(p, q). It is a process composed of an autoregressive process of order p and a process of moving averages of order q. The mathematical model of it is

$$y_t = \Phi_1 y_{t-1} + \cdots + \Phi_p y_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}, \tag{1}$$

where $\Phi_1, \ldots, \Phi_p$ are parameters of the autoregressive part and $\theta_1, \ldots, \theta_q$ are parameters of the moving averages part of the model. This model can be rewritten using a backshift operator $B^i y_t = y_{t-i}$ in the form

$$\Phi_p(B) y_t = \theta_q(B) a_t, \tag{2}$$

where $\Phi_p(B) = (1 - \Phi_1 B - \cdots - \Phi_p B^p)$ and $\theta_q(B) = (1 - \theta_1 B - \cdots - \theta_q B^q)$.

The process AR(p) is stationary in case when roots of the polynomial $\Phi_p(B)$ lie outside the unit circle. The process MA(q) is invertible, when roots of the polynomial $\theta_q(B)$ lie outside the unit circle. But stationary models nearly absent in the economic practice. Fortunately, it is possible to convert a nonstationary model to a stationary one.

If d roots of the polynomial $\Phi_p(B)$ lie on the unit circle, the process is not stationary but it has a stochastic trend. Such process is denoted $I(d)$ and it is called the integrated process of order d. Its model has the form

$$(1 - B)^d y_t = a_t. \tag{3}$$

This model can be converted to a stationary one if d-times differentiation is applied to it. The combination of the stationary and the integrated process leads to the nonstationary process ARIMA(p, d, q),

$$\Phi_p(B)(1 - B)^d y_t = \theta_q(B) a_t. \tag{4}$$

When a seasonal oscillation with period s occurs in a time series, it is necessary to capture the dependence among the components of the original series and also the dependence among the components, which correspond to the different seasons. The seasonal model SARIMA$(p, d, q)(P, D, Q)$,

$$\Phi_P(B^s)\Phi_p(B)(1 - B)^d(1 - B^s)^D y_t = \theta_q(B)\theta_Q(B^s) a_t, \tag{5}$$

where P, D and Q are seasonal parameters of process, is used in this case. The left-hand side of (5) supplies the dependence inside the period and the right-hand side represents only the seasonal dependences.

Constructions of Box-Jenkins models are especially based on the information that is obtained from the correlograms, i.e. the graphs of values of the autocorrelation function ACF and the partial autocorrelation function PACF.

For stationary time series, the residual ACF is defined through autocorrelations with the delay k,

$$\rho_k = \frac{\gamma_k}{\gamma_0}, \tag{6}$$

where $\gamma_k = E[(y_t - \mu)(y_{t-k} - \mu)]$. The residual ACF indicates the range of the linear dependence between y_t and y_{t-k}.

The partial autocorrelation with delay k is defined through partial regressive coefficients Φ_{kk} in the autoregression of order k

$$y_t = \Phi_{k1} y_{t-1} + \Phi_{k2} y_{t-2} + \cdots + \Phi_{kk} y_{t-k} + a_t, \tag{7}$$

where a_t is a value that is uncorrelated with $y_{t-1}, y_{t-2}, \ldots, y_{t-k}$. The function PACF gives the information cleaned from the influence of the variables $y_{t-1}, y_{t-2}, \ldots, y_{t-k}$.

First estimation properties of a given time series are based on the line graph, periodogram, ACF and PACF. Peaks in the periodogram indicate the presence of seasonal oscillations. It means that it is necessary to work with a seasonal model. Values greater than 1 in ACF and PACF mean that the series is not stationary. In this case, it is necessary to consider an integrated model. Removal of non-stationarity in the variance can be achieved by the Box-Cox transformation.

The model chosen has to be verified, i.e. monitored whether autocorrelation unsystematic components are zero by the Box-Pearson test and how good the received estimates of the parameters μ, ϕ, θ are by t-tests.

The model selected is the basis for the estimate of further development of the series. The calculation of the forecasted value y_{T+h} is done by means of the conditional mean value $E(y_{T+h} \mid y_{T-1}, y_{T-2}, \ldots)$.

3. Wavelet transform

The wavelet transform is a useful tool for detecting local properties and investigating nonstationary data. It is defined using wavelets, which form a basis in the space $L^2(R)$. Multiresolution analysis (MRA) is the most commonly used method to the construction of such basis.

During MRA, subspaces $V_j \subset L^2(R)$ are constructed with properties

1) $V_j \subset V_{j+1}$,
2) there exists $\varphi \in V_0$ such that $\{\varphi_{0,k}\}$, where $\varphi_{0,k}(x) = \varphi(x-k)$, is orthogonal and complete in $L^2(R)$,
3) $f(x) \in V_0$ if and only if $f(2^j x) \in V_j$,
4) $\cap_j V_j = \{0\}$,
5) $\overline{\cup_j V_j} = L^2(R)$.

When $\{V_j\}$ is MRA with scaling function φ, then there exists a scaling vector $\mathbf{u} = (\dots, u_{-1}, u_0, u_1, \dots)$ such that

$$\varphi(x) = \sqrt{2} \sum_{k \in Z} u_k \varphi(2x - k). \tag{8}$$

In this case, the associated wavelet ψ is defined by the formula

$$\psi(x) = \sqrt{2} \sum_{k \in Z} v_k \varphi(2x - k), \;\; v_k = (-1)^k u_{1-k}. \tag{9}$$

It follows from the MRA that there exists a subspace $W_j \subset L^2(R)$ such that

$$V_{j+1} = V_j \oplus W_j. \tag{10}$$

The subspaces V_j and W_j can be generated by means of dilations and traslations of the functions φ and ψ. It holds

$$W_j = \overline{\text{span}\{\psi_{j,k}(x)\}}, \quad \text{where} \;\; \psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k), \tag{11}$$

$$V_j = \overline{\text{span}\{\varphi_{j,k}(x)\}}, \quad \text{where} \;\; \varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k). \tag{12}$$

Moreover, it can be seen that

$$V_{j+1} = V_J \oplus W_J \oplus W_{J+1} \oplus \cdots \oplus W_j. \tag{13}$$

It means that it is possible to expand every function $f \in L^2(R)$ into the series

$$f(x) = \sum_{k \in Z} y_{J,k} \varphi_{J,k} + \sum_{j=J}^{\infty} \sum_{k \in Z} x_{j,k} \psi_{j,k}. \tag{14}$$

The scaling coefficients $y_{j,k}$ and the wavelet coefficients $x_{j,k}$ are calculated as inner products. It holds

$$y_{j,k} = \langle f, \varphi_{j,k} \rangle, \qquad x_{j,k} = \langle f, \psi_{j,k} \rangle. \tag{15}$$

It follows from (15), (11), (12), (8), (9) and (14) that

$$y_{j,k} = \frac{1}{\sqrt{2}} \sum_l u_l y_{j+1,2k+l}, \quad x_{j,k} = \frac{1}{\sqrt{2}} \sum_l (-1)^l u_{1-l} y_{j+1,2k+1}, \tag{16}$$

$$y_{l+1,k} = \frac{1}{\sqrt{2}} \sum_m u_{m-2k} y_{l,m} + \frac{1}{\sqrt{2}} \sum_m (-1)^m u_{1-m-2k} x_{l,m}. \tag{17}$$

Computation of the wavelet coefficients is realized by means of the Mallat algorithm. The relations (16) and (17) are the basis of this algorithm. First, approximations and details are computed from the data given (the decomposition phase). The approximations correspond to the trend and the details correspond to random

components of the time series. The process can be repeated more times. The wavelet coefficients can be adapted or not. In the end, modified or original data are obtained from this set of values (the reconstruction phase).

In the following example, the wavelet transform is used to construct the prediction for the time series given. First, a decomposition into approximations and details is done. Then the proper ARIMA model and prediction are found for each of these parts. The resulting prediction is a sum of values from these two partial predictions.

4. Example

The monthly values of CPI inflation in the Czech Republic in the years 2004–2014 are given in Table 1. Find a suitable ARIMA model for this series from January 2004 to December 2012. Make a forecast for the rest of the series using the ARIMA model and then using ARIMA model modified by wavelets. Compare the results received to each other.

	1	2	3	4	5	6	7	8	9	10	11	12
2004	0.3	0.5	0.8	1.	1.2	1.4	1.7	2.	2.2	2.5	2.7	2.8
2005	2.8	2.7	2.6	2.6	2.5	2.4	2.2	2.1	2.	2.	1.9	1.9
2006	2.	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.7	2.6	2.5
2007	2.4	2.3	2.2	2.2	2.1	2.1	2.1	2.	2.	2.2	2.5	2.8
2008	3.4	3.9	4.3	4.7	5.	5.4	5.8	6.1	6.4	6.6	6.5	6.3
2009	5.9	5.4	5.	4.6	4.1	3.7	3.1	2.6	2.1	1.6	1.3	1.
2010	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.9	1.1	1.2	1.4	1.5
2011	1.6	1.7	1.7	1.8	1.8	1.9	1.9	1.9	1.8	1.9	1.9	1.9
2012	2.1	2.2	2.4	2.6	2.7	2.8	2.9	3.1	3.2	3.3	3.3	3.3
2013	3.2	3.	2.8	2.7	2.5	2.3	2.2	2.	1.8	1.6	1.5	1.4
2014	1.3	1.1	1.									

Table 1: Inflation 2000–2014

Solution. On the basis of ACF and PACF, the original time series were modeled through ARIMA(3,2,1) model.

Further, the decomposition of the time series to approximations and details by using the Daubechies wavelet Db3 was done. The first order extrapolation was used to expand the data beyond boundary. This allowed receiving such approximation coefficients that are close to the values of the original time series.

In the next step, appropriate ARIMA models were selected for the approximations and for the details separately. The approximation coefficients are not identical with the time series values, because the information is lost when wavelet decomposition is made. Moreover, a small change of range (e.g. a truncation of the time series or an extension of the data beyond boundary) may affect the shape of the ARIMA model. Therefore, Box-Jenkins models are different for the original data and for approximations.

The approximations were modeled with the help of ARIMA(2,2,1) process and the details were modeled as ARIMA(2,0,1) process in this example. The prognosis for the time series was obtained by adding up the forecasts for approximation and

details. Note that it is possible to realize prediction using approximations only and ignore details, when the details are detected like random noise.

Choice of ARIMA models affects the shape of the predictions. Adequacy of the models is assessed by means of corresponding graphs ACF and PACF. Comparison of the predictions for the next 15 months is shown in Figure 1. Comparison of the values received is presented in Table 2.

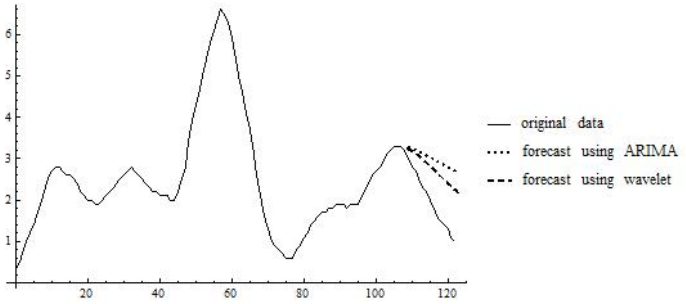


Figure 1: Comparison of predictions

	1	2	3	4	5	6	7	8
original	3.2	3.	2.8	2.7	2.5	2.3	2.2	2.
ARIMA	3.2789	3.24318	3.20814	3.16497	3.12164	3.07674	3.03056	2.98447
wavelets	3.24865	3.18837	3.12578	3.05159	2.97491	2.89764	2.81764	2.73695

	9	10	11	12	13	14	15
original	1.8	1.6	1.5	1.4	1.3	1.1	1.
ARIMA	2.93772	2.89095	2.84405	2.79705	2.75005	2.70301	2.65596
wavelets	2.65611	2.57463	2.49296	2.41125	2.32939	2.24748	2.16556

Table 2: Comparison of predictions

The root mean square error $RMSE = 1.07178$ in case of the ARIMA model and $RMSE = 0.78320$ in case of the model that uses the wavelet transform. It can be seen that the prognosis was improved by 36.8% when the wavelet modification was used.

5. Conclusion

The example has shown that the ARIMA model modification can lead to improved estimation of time series evolution. Note that wavelets can be used not only in forecasting non-stationary time series, but also to detect sudden changes, or to select cycles or fractal nature of time series.

6. Acknowledgement

This work was supported by project GAČR P403/12/1811: Unconventional managerials decision making methods development in enterprise economics and public economy.

References

[1] Artl, J., Artlová, M., and Rublíková, E.: *Analýza ekonomických časových řad s příklady*. VŠE, Praha, 2004.

[2] Najzar, K.: *Základy teorie waveletů*. Karolinum, Praha, 2004.

[3] Jansen, M. and Oonix P.: *Second generation wavelets and applications*. Springer-Verlag, London, 2005.

[4] Sebera, M. and Seberová, H.: Časové řady a wavelety. In: *XX. mezinárodní kolokvium o řízení osvojovacího procesu*, pp. 354–357. Vyškov, 2001.

[5] Siegel, A. F.: *Practical business statistics*. Elselvier, Oxford, 2012.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

ON TWO METHODS FOR THE PARAMETER ESTIMATION PROBLEM WITH SPATIO-TEMPORAL FRAP DATA

Štěpán Papáček[1], Jiří Jablonský[1], Ctirad Matonoha[2]

[1] University of South Bohemia in České Budějovice,
Faculty of Fisheries and Protection of Waters, CENAKVA, Institute of Complex Systems
Zámek 136, 373 33 Nové Hrady, Czech Republic
spapacek@frov.jcu.cz, jiri.jablonsky@gmail.com
[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

Abstract

FRAP (Fluorescence Recovery After Photobleaching) is a measurement technique for determination of the mobility of fluorescent molecules (presumably due to the diffusion process) within the living cells. While the experimental setup and protocol are usually fixed, the method used for the model parameter estimation, i.e. the data processing step, is not well established. In order to enhance the quantitative analysis of experimental (noisy) FRAP data, we firstly formulate the inverse problem of model parameter estimation and then we focus on how the different methods of data pre-processing influence the confidence interval of the estimated parameters, namely the diffusion constant p. Finally, we present a preliminary study of two methods for the computation of a least-squares estimate $\hat{p}$ and its confidence interval.

1. Introduction

The FRAP technique is based on measuring the change in fluorescence intensity in a region of interest (ROI - generally a Euclidean 2D or 3D domain). These changes are induced by an external stimulus, a high-intensity laser pulse provided by the CLSM (Confocal Laser Scanning Microscopy). The stimulus, also called *bleaching*, causes an (ir)reversible loss in fluorescence in the bleached area without any damage to intracellular structures. After the bleach, the observed recovery in fluorescence reflects the mobility (related to diffusion) of fluorescent compounds from the area outside the bleach.

Based on spatio-temporal 2D FRAP images, the process of diffusive transport can be reconstructed using either a closed form model or a numerical simulation based model. In this paper, we study both approaches. We show the results for the oversimplified one-spatial-point Moullineaux method [4] and the results based on the numerical integration of the Fick diffusion PDE (Partial Differential Equation) with the realistic initial and boundary conditions [5].

2. Parameter estimation based on spatio-temporal data

We aim to present a parameter estimation problem with spatio-temporal experimental observation in a comprehensive mathematical framework allowing simultaneously to determine both the parameter value p (generally $p \in \mathbb{R}^q,\ q \in \mathbb{N}$)[1] and the corresponding confidence interval proportional to the output noise and a quantity related to the sensitivity, see (7). The data are represented by a (measured) signal on a Cartesian product of the space-points $(x_i)_{i=1}^n$ and time-points $(t_j)_{j=1}^m$; let $N_{\text{Data}} := m \times n$ be the total number of spatio-temporal data points. We define the operator $S : \mathbb{R}^q \to \mathbb{R}^{N_{\text{Data}}}$ that maps parameter values $p_1, \ldots, p_q$ to the solution of the underlying initial-boundary value problem, e.g. (9), evaluated at points (x_i, t_j):

$$S(p) = \{y(x_i, t_j, p) \in \mathbb{R}, \quad 1 \le i \le n, \quad 1 \le j \le m\}. \tag{1}$$

Some commonly used FRAP methods do not employ all the N_{Data} measured values at points $\{(x_i, t_j),\ i = 1, \ldots, n,\ j = 1, \ldots, m\}$. They either employ some of the values or perform some preprocessing (e.g. space averaging, see [6]). Hence we further define the observation operator $G : \mathbb{R}^{N_{\text{Data}}} \to \mathbb{R}^{N_{\text{data}}}$ that evaluates the set of values $S(p)$ on a certain subset of the full data space ($N_{\text{data}} \le N_{\text{Data}}$):

$$G(S(p)) = (z(x_l, t_l, p))_{l=1}^{N_{\text{data}}} \tag{2}$$

We now define the forward map $F : p \to z(x_l, t_l, p)_{l=1}^{N_{\text{data}}}$. Here, $F = G \circ S$ represents the parameter-to-output map, defined as the composition of the PDE solution operator S and the observation operator G.[2] Our regression model is now

$$F(p) = \text{data}, \tag{3}$$

where the data are modeled as contaminated with additive Guassian noise

$$\text{data} = F(p_T) + e = (z(x_l, t_l, p_T))_{l=1}^{N_{\text{data}}} + (e_l)_{l=1}^{N_{\text{data}}}.$$

Here $p_T \in \mathbb{R}^q$ denotes the true values and $e \in \mathbb{R}^{N_{\text{data}}}$ is a data error vector which we assume to be normally distributed with variance σ^2, i.e. $e_i = \mathcal{N}(0, \sigma^2)$ $i = 1, \ldots, N_{\text{data}}$.

Given some data, the aim of the parameter estimation problem is to find p_T, such that (3) is satisfied in some appropriate sense. Since (3) usually consists of an overdetermined system (there are more data points than unknowns), it cannot be expected that (3) holds with equality, but instead an appropriate notion of a solution is that of a least-squares solution $\hat{p}$ (with $\| \ . \ \|$ denoting the Euclidean norm on $\mathbb{R}^{N_{\text{data}}}$):

$$\| \text{ data} - F(\hat{p}) \|^2 = \min_p \| \text{ data} - F(p) \|^2 . \tag{4}$$

[1] We prefer this more general definition of the model parameter vector instead of the single scalar quantity because we aim to work with more complex model than (9) in the near future.

[2] For the one-point Moullineaux method [4], only the point with the spatial coordinate $x = 0$ is measured, i.e. $G_M : z(t_j, p) := z(0, t_j, p) = y(0, t_j, p),\ j = 1, \ldots, N_{\text{data}} = m$. For the second method, we reduce the data space taking the so-called relevant data only [6], i.e. $G_{\text{PDE}} : z(x_l, t_l, p) = y(x_i, t_j, p),\ i = 1, \ldots, n^* \le n,\ j = 1, \ldots, m^* \le m,\ l = 1, \ldots, N_{\text{data}} = m^* \times n^*$.

Sensitivity analysis and confidence intervals

For the sensitivity analysis we require the Fréchet-derivative $F'[p_1, \dots, p_q] \in \mathbb{R}^{N_{\text{data}} \times q}$ of the forward map F, that is

$$\begin{aligned} F'[p_1, \dots, p_q] &= \begin{pmatrix} \frac{\partial}{\partial p_1} F(p_1, \dots, p_q) & \dots & \frac{\partial}{\partial p_q} F(p_1, \dots, p) \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial p_1} z(x_1, t_1, p) & \dots & \frac{\partial}{\partial p_q} z(x_1, t_1, p) \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ \frac{\partial}{\partial p_1} z(x_{N_{\text{data}}}, t_{N_{\text{data}}}, p) & \dots & \frac{\partial}{\partial p_q} z(x_{N_{\text{data}}}, t_{N_{\text{data}}}, p) \end{pmatrix}. \end{aligned}$$

A corresponding quantity is the Fisher information matrix (FIM)

$$M[p_1, \dots, p_q] = F'[p_1, \dots, p_q]^T F'[p_1, \dots, p_q] \in \mathbb{R}^{q \times q}. \tag{5}$$

Based on the book of Bates and Watts [1], we can estimate confidence intervals. Suppose we have computed $\hat{p}$ as a least-squares solution in the sense of (4). Let us define the residual as

$$res^2(\hat{p}) = \|F(\hat{p}) - \text{data}\|^2 = \sum_{i=1}^{N_{\text{data}}} \left[\text{data}_i - z(x_i, t_i, \hat{p})\right]^2. \tag{6}$$

Then according to [1], it is possible to quantify the error between the computed parameters $\hat{p}$ and the true parameters p_T.

Having only one single scalar parameter p as unknown, the Fisher information matrix M collapses into the scalar quantity $\sum_{i=1}^{N_{\text{data}}} \left[\frac{\partial}{\partial p} z(x_i, t_i, p) \mid_{p=\hat{p}}\right]^2$, and the $1-\alpha$ confidence interval for full observations is described as follows

$$(\hat{p} - p_T)^2 \sum_{i=1}^{N_{\text{data}}} \left[\frac{\partial}{\partial p} z(x_i, t_i, p) \mid_{p=\hat{p}}\right]^2 \leq \frac{res^2(\hat{p})}{N_{\text{data}} - 1} f_{1, N_{\text{data}}-1}(\alpha), \tag{7}$$

where $f_{1, N_{\text{data}}-1}(\alpha)$ corresponds to the upper α quantile of the Fisher distribution with 1 and $N_{\text{data}} - 1$ degrees of freedom.

In (7), several simplifications are possible. Note that according to our noise model, the residual term $\frac{res^2(\hat{p})}{N_{\text{data}}-1}$ is an estimator of the error variance [1] such that the approximation

$$\frac{res^2(\hat{p})}{N_{\text{data}} - 1} \sim \sigma^2 \tag{8}$$

holds if N_{data} is large. Moreover, we remind the reader that the Fisher distribution with 1 and $N_{\text{data}} - 1$ degrees of freedom converges to the χ^2-distribution as $N_{\text{data}} \to \infty$. Hence, the term $f_{1, N_{\text{data}}-1}(\alpha)$ can approximately be viewed as independent of N_{data} as well and of a moderate size.

3. Two FRAP methods: Assessment of uncertainty

Let us proceed to the FRAP measurement technique [4, 5]. We assume the special geometry residing in one-dimensional simplification getting the measured fluorescent intensity level y as a function of the spatial coordinate x, time t and diffusion coefficient p (generally time dependent, e.g. $p = (p_j)_{j=1}^m$):

$$\frac{\partial y}{\partial t} - p\frac{\partial^2 y}{\partial x^2} = 0 \ , \tag{9}$$

in $(t_0, T)\times\Omega$, with suitable boundary conditions on $(t_0, T)\times\partial\Omega$ and initial conditions in Ω, where $\Omega \subset \mathbb{R}$. Problem (9) represents a reliable model of the FRAP process. The corresponding inverse formulation is used in our software CA-FRAP[3] for the processing of the real FRAP data resulting in the solution vector $(\hat{p}_j)_{j=1}^m$. Here, in this paper, the software CA-FRAP is further used (in Subsection 3.2) for the simulation of virtual FRAP data and the subsequent evaluation of the FIM.

According to [2], the standard error of a parameter p_k estimate, i.e. $SE(\hat{p}_k)$, is

$$SE(\hat{p}_k) = \hat{\sigma}\sqrt{{M_{kk}}^{-1}}, \tag{10}$$

where $\hat{\sigma}$ is the data error variance estimate. Relation (10) highlights the importance of the FIM and is further used for the comparison of two FRAP data processing methods.

3.1. The one-point Moullineaux method

C. W. Moulineaux *et al.* [4] measured one-dimensional bleaching profiles (with common variance σ^2) along the specimen long axis. They took the ROI as coincident with the real axis ($x \in \mathbb{R}$) and the initial bleaching profile (of bleached particles) as the Gaussian with half-width r_0 at height $y_{0,0}e^{-2}$, i.e. $y(x, t_0, p) = y_{0,0}\exp\frac{-2x^2}{{r_0}^2}$. Here t_0 corresponds to the initial time and can be set to zero. Then, the solution $y(x, t, p)$ of the diffusion equation (9) for the bleached particles is

$$y(x, t, p) = \frac{y_{0,0}\ r_0}{\sqrt{{r_0}^2 + 8pt}}\exp\frac{-2x^2}{{r_0}^2 + 8pt}, \quad x \in \mathbb{R}, \quad t \in [0, T]. \tag{11}$$

The time evolution of the maximum depth $y(0, t, p)$ is taken by Moullineaux *et al.* as the single observed spatial data point $z_M(t, p)$.[4] It holds $z_M(t, p) = \frac{r_0\ y_{0,0}}{\sqrt{{r_0}^2+8pt}}$.

The FIM, based on the semi-relative sensitivities, collapses to a scalar quantity $M_M = \sum_{j=1}^m \left[\frac{\partial z_M(t_j,p)}{\partial p}p\right]^2 = \sum_{j=1}^m \frac{(4r_0pt_j)^2}{({r_0}^2+8pt_j)^3} = \frac{1}{4}\sum_{j=1}^m \frac{(8s_j)^2}{(1+8s_j)^3}$, where $s_j := \frac{pt_j}{{r_0}^2}$ and an estimate $\hat{p}$ is taken instead of p.

[3] See [3, 5] for more details or mail to: matonoha@cs.cas.cz.

[4] The authors of [4] used the weighted linear regression based on equation $z_M(t, p) = \frac{r_0\ y_{0,0}}{\sqrt{{r_0}^2+8pt}}$ in order to estimate the diffusion coefficient p. They calculated neither the FIM nor the standard error of the parameter p estimate using (10).

Let us assume that we have an equidistant spacing $\Delta s := \frac{s_m - s_1}{m-1}$ such that the sum can be approximated by an integral.[5] Then we get the following expression for the FIM (after some algebraic manipulation assisted by the Mathematica software)

$$M_M \approx \frac{m-1}{32(s_m - s_1)} \left[\ln\left(\frac{1+8s_m}{1+8s_1}\right) - \frac{8(s_m - s_1)(1 + 12(s_1 + s_m) + 128 s_1 s_m)}{(1+8s_m)^2(1+8s_1)^2} \right] + \left[\frac{8{s_1}^2}{(1+8s_1)^3} + \frac{8{s_m}^2}{(1+8s_m)^3} \right]. \quad (12)$$

The expression for M_M is positive, increasing with the number of measurement points, i.e. with $T = t_1 + (m-1)\Delta t$ (for fixed Δt and t_1), and represents the lower bound for the FIM as a scalar quantity (when a scalar p is estimated).[6]

3.2. Initial boundary value problem for PDE (9) and the FIM

As the above approach has several limitations, e.g. cell geometry restriction (infinite domain is required), bleach profile must be gaussian-like, etc., we propose to model the diffusion process by the Fick diffusion equation with realistic initial and boundary conditions instead. Then the parameter estimation problem is formulated as an ordinary least squares problem (4) resulting in the estimate $\hat{p}_{PDE}$. This problem is treated elsewhere [3, 5, 6]. Here, we present the uncertainty assessment based on the numerical evaluation of the FIM (implemented in the CA-FRAP). For each time instant t_j we denote $p_j = \hat{p}_{PDE}$. The CA-FRAP solves the inverse problem (9) and takes the simulated output $y(x_i, t_j, p_j)$, $i = 1, \ldots, n$. Then, according to (5), we obtain the FIM (diagonal in this case) using central differences as

$$M^j_{PDE} = \sum_{i=1}^{n} \left[\frac{\partial y(x_i, t_j, p)}{\partial p} \, |_{p=p_j} \right]^2 \approx \sum_{i=1}^{n} \left[\frac{y(x_i, t_j, p_j + \varepsilon) - y(x_i, t_j, p_j - \varepsilon)}{2\varepsilon} \right]^2 \quad (13)$$

where ε is a small positive number. The corresponding quantity M_{PDE} for the estimation of an overall $\hat{p}_{PDE}$ is the sum $\sum_{j=1}^{m} M^j_{PDE}$, cf. (5).

3.3. Numerical evaluation and comparison of the FIM for both method

We have performed several computations of the FIM for both above mentioned approaches. For a particular case $y_{0,0} = r_0 = p = 1$ and the time step between $m = 10$ measurements equal to 0.1, the evaluation of (12) is straightforward and gives $M_M \approx 0.296$ for $s_1 = 0.1$ and $s_m = 1$. The evaluation of M_{PDE} is more complicated. In order to compare both method, the output $y(x_i, t_j, p_j)$ were computed for the same parameter settings as before by solving the forward problem (9), showing the correspondence with (11), indeed. The numerical evaluation of (13) gives then $M_{PDE} \approx 0.363$. We see that the PDE based method, which uses more spatial points at each time level, gives a lower standard error of the estimated parameter p.

[5] The quantities corresponding to the first t_1 and final T measurement time are $s_1 = pt_1/r_0^2$ and $s_m = pT/r_0^2$, $t_1 < T$, respectively.

[6] The one-point Moullineaux method is the simplest method. Other methods, see e.g. [6] for review, use more data points, thus add more (positive) terms to the FIM.

4. Conclusion

We present two methods for the estimation of the fluorescent compounds mobility from the spatio-temporal FRAP measurement. The first and simplest method is based on the curve fitting to a closed formula and needs some unrealistic or hard-to-accomplish conditions. The second method is based on a numerical approximation of the Fick diffusion PDE with either a scalar or time dependent diffusion coefficient p. Both methods are implemented in our software CA-FRAP, which simultaneously provides the parameter estimate (this is not discussed here, in this paper) and the corresponding standard error (using (10)). This aims to promote the following idea across the FRAP community. The bioprocesses are inherently stochastic, thus the mathematical framework related to the model parameter identification should determine both a parameter mean value and a certain confidence interval, which depends on the output noise and the corresponding sensitivity, cf. (10).

Acknowledgements

This work was supported by the MEYS of the Czech Republic – projects "CENAKVA" (No. CZ.1.05/2.1.00/01.0024), "CENAKVA II" (No. LO1205 under the NPU I program), by Postdok JU CZ.1.07/2.3.00/30.0006, and by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Bates, D. M. and Watts, D. G.: *Nonlinear regression analysis: Its applications.* John Wiley & Sons, New York, 1988.

[2] Cintrón-Arias, A., Banks, H. T., Capaldi, A., and Lloyd, A. L.: A sensitivity matrix based methodology for inverse problem formulation. J. Inv. Ill-Posed Problems **17** (2009), 545–564.

[3] Matonoha, C. and Papáček, Š.: On the connection and equivalence of two methods for solving an ill-posed inverse problem based on FRAP data. J. Comput. Appl. Math., submitted.

[4] Moullineaux, C. W., Tobin, M. J., and Jones, G. R.: Mobility of photosynthetic complexes in thylakoid membranes. Nature **390** (1997), 421–424.

[5] Papáček, Š., Kaňa, R., and Matonoha, C.: Estimation of diffusivity of phycobilisomes on thylakoid membrane based on spatio-temporal FRAP images. Math. Comput. Modelling **57** (2013), 1907–1912.

[6] Papáček, Š., Jablonský, J., Kaňa, R., Matonoha, C., and Kindermann, S.: From data processing to experimental design and back again: A parameter identification problem based on FRAP images. Accepted for publication in ICDIP 2015: XIII International Conference on Digital Image Processing, Dubai 2015.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

2D SIMULATION OF FLOW BEHIND A HEATED CYLINDER USING SPECTRAL ELEMENT APPROACH WITH VARIABLE COEFFICIENTS

Jan Pech[1,2]

[1] New Technologies-Research Center, University of West Bohemia in Pilsen
Universitní 8, 30614 Pilsen, Czech Republic
[2] Institute of Thermomechanics, Academy of Sciences of the Czech Republic
Dolejškova 1402/5, 182 00 Praha 8, Czech Republic
jpech@it.cas.cz

Abstract

The scheme for the numerical solution of the incompressible Navier-Stokes equations coupled with the equation for temperature through the temperature dependent viscosity and thermal conductivity coefficients is presented. It is applied, together with the spectral element method, to the 2D calculations of flow around heated cylinder. High order polynomial approximation is combined with the decomposition of whole computational domain to only a few elements. Resulting data are compared with the experimental data.

1. Introduction

The viscosity and the thermal conductivity of water and air depend on the temperature. As a consequence, a wake behind an obstacle in an isothermal setting differs from the situation, when the body and the fluid temperatures do not coincide. The experimental data, see [5], for the flow around the heated cylinder are available for both water and air in the flow regimes exhibiting regular vortex shedding. The cited experimental data are available for Reynolds numbers ($Re = DV_\infty/\nu_\infty$) in the range $50 < Re < 170$, when Re is related to the cylinder diameter D (V_∞ is the upstream velocity magnitude and ν_∞ is the upstream value of the kinematic viscosity). The conditions and flow parameters in the mentioned experiment were such, that the compressibility of both water and air can be neglected. Therefore the fluid density (ρ) may be assumed to be a constant and the incompressible model will be used. The system of equations describing the heated flow consists of the Navier-Stokes equations (1) with the incompressible constraint (2)

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = -\nabla p + \nabla \cdot \left[\nu \left(\nabla \vec{v} + (\nabla \vec{v})^{\mathbf{T}} \right) \right] , \tag{1}$$

$$\nabla \cdot \vec{v} = 0 \tag{2}$$

and the convection-diffusion equation for temperature (T):

$$\rho c_p \left(\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T \right) = \nabla \cdot (\lambda \nabla T) \ , \tag{3}$$

where in the system (1)-(3) $\vec{v}$ denotes the fluid velocity vector, p is the kinematic pressure, λ the thermal conductivity and the constant c_p is the specific heat at the constant pressure. Due to the nature of the pressure in the incompressible models the above system is complete without the equation of state, on the other hand the variability of the material coefficients causes strong coupling of equations (1) and (3). The thermal dependencies of ν and λ can be approximated by power function obtained from a tabulated data as in [3]:

$$\nu(T) = \nu_\infty (T/T_\infty)^{\omega_\nu} \ , \text{(air: } \omega_\nu = 0.7774, \text{ water: } \omega_\nu = -7 \text{)}, \tag{4}$$

$$\lambda(T) = \lambda_\infty (T/T_\infty)^{\omega_\lambda} \ , \text{(air: } \omega_\lambda = 0.85, \text{ water: } \omega_\lambda = 0.71 \text{)}, \tag{5}$$

where $1 \leq (T/T_\infty) \leq \tilde{T} = (T_W/T_\infty)$ (T_W is the constant temperature of the cylinder wall).

The system of equations (1)-(3) generally admits non-smooth or even discontinuous solutions, but observations do not confirm any shocks in the mentioned range of Re for the fluids in the state which coincides with description in [5]. This suggests possible existence of a smoother solution. Therefore we will use the computational method based on the assumption of smooth data and solution, as is the spectral method (see e.g. [2]). This method converges with increasing (e.g. polynomial) order of the expansion basis. If the method is applicable, its minimization of the number of degrees of freedom and the convergence rate are superior to methods of lower, fixed order, which converge by dividing the computational domain to smaller parts. On the other hand, spectral methods are not always applicable. Already the fact, that the cylinder is in our case represented as a circular hole inside the domain, forces us to leave pure spectral method and use the spectral element method, which combines the geometrical flexibility of the finite element method with the approach of the spectral method. This leads us to use of minimal number of elements and application of very high order expansion basis. However, the class of equations where the high orders are advantageous for numerical computation is limited and this fact must be taken into account in design of the numerical scheme.

2. Numerical scheme

The numerical scheme for the system (1)–(3) was developed on the base of the splitting scheme for the Navier-Stokes equations with a variable viscosity ([1]), where the temperature dependent viscosity was decomposed to the sum of the constant ν_∞ and the variable part ν_s: $\nu(T(\vec{x},t)) = \nu_\infty + \nu_s(\vec{x},t)$. Denoting by " ˆ " and " ˜ "

intermediate fields, by superscript the values of the variables in the n-th time level and by Δt the time step, we arrive to the first order scheme in time (see [4]):

$$\frac{\hat{\vec{v}} - \vec{v}^n}{\Delta t} = -(\vec{v}^n \cdot \nabla)\vec{v}^n + \nabla \cdot [\nu_s^n (\nabla \vec{v}^n + \nabla^T \vec{v}^n)] \ , \tag{6}$$

$$\frac{\tilde{\vec{v}} - \hat{\vec{v}}}{\Delta t} = -\nabla p^{n+1} \underbrace{\Rightarrow}_{\nabla \cdot \tilde{\vec{v}} = 0} \nabla^2 p^{n+1} = \nabla \cdot \left(\frac{\hat{\vec{v}}}{\Delta t} \right) \ , \tag{7}$$

$$\frac{\vec{v}^{n+1} - \tilde{\vec{v}}}{\Delta t} = \nu_\infty \nabla \cdot \nabla \vec{v}^{n+1} \ . \tag{8}$$

The key role plays the high order pressure boundary condition (HOPBC), which is asserted on the boundaries, where the Dirichlet condition for velocity is prescribed:

$$\frac{\partial p^{n+1}}{\partial \vec{n}} = \vec{n} \cdot \left[-(\vec{v}^n \cdot \nabla)(\vec{v}^n) + \nabla \cdot \left(\nu^n \nabla \vec{v}^n + \nu^n (\nabla \vec{v}^n)^{\mathbf{T}} \right) \right] \ . \tag{9}$$

Temperature dependence of the thermal conductivity is needed to keep the correct Prandtl number ($Pr = \nu \rho c_p / \lambda$). The scheme for the temperature equation was derived again by the operator splitting combined with the splitting of λ to the constant λ_∞ and the variable part λ_s, i.e. $\lambda(T(\vec{x}, t)) = \lambda_\infty + \lambda_s(\vec{x}, t)$.

The operator splitting then allows the implicit treatment of the diffusion operator with the constant coefficient (λ_∞) and the explicit treatment of the part with the variable conductivity (λ_s). The first order scheme in time for temperature reads:

$$\frac{\hat{T} - T^n}{\Delta t} = -(\vec{v}^n \cdot \nabla) T^n - \frac{1}{\rho c_p} \nabla \cdot (\lambda_s^n \nabla T^n) \ . \tag{10}$$

As in (7) and (8) the spectral element method is applicable to the implicit step of the scheme for temperature:

$$\frac{\lambda_\infty}{\rho c_p} \nabla^2 T^{n+1} - \frac{T^{n+1}}{\Delta t} = -\frac{\hat{T}}{\Delta t} \ . \tag{11}$$

The whole scheme (6)–(11) was implemented on the base of the modified Nektar++ library of version 3.3.0 and the deeply modified incompressible Navier-Stokes solver provided with the same library.

3. Mesh and parameters of the computation

The model assumed the flow in an open channel, which is not significantly influenced by a tank walls (in experiment) or Dirichlet boundary conditions on outer boundaries (in computation). Therefore the dimensions of the computational domain must be large enough. The cylinder diameter $D = 1$ was chosen for simplicity and then the spatial dimensions of the computational domain were: $20D$ upstream,

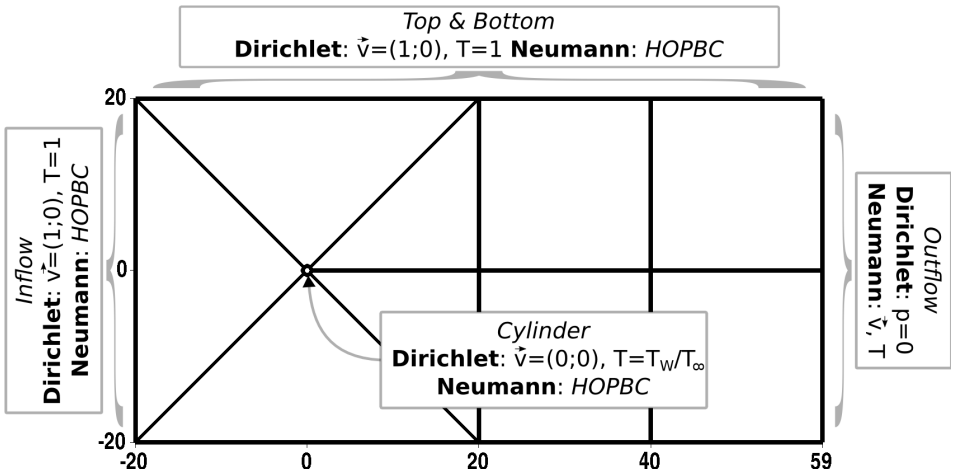


Figure 1: The computational mesh consisting of 9 elements with description of the boundary conditions (*HOPBC* is given by eq. (9)). The curve of the cylinder wall was given by 10^{th} order polynomial for each of the adjacent elements.

$60D$ downstream and $20D$ above and under the cylinder. We divided the computational domain to small number of elements ($NEL = 9$) and used the rich expansion basis, having polynomial orders up to $p = 49$ in each coordinate variable (2500 DOFs per element). The *no slip* condition and value of relative temperature $\tilde{T}$ was prescribed at the cylinder wall. Figure 1 shows the boundary conditions and the computational mesh with all its elements. The chosen values of the inlet boundary conditions imply $\nu_\infty = 1/Re$ and $\lambda_\infty/(\rho c_p) = 1/(RePr)$, so we can set Re and Pr as independent, dimensionless parameters and avoid the explicit specification of the constants ρ and c_p. The initial conditions for both velocity and temperature were constants equal to the values on the inflow boundary. The final quantity for the comparison with the experimental results was the *Strouhal number* $St = fD/V_\infty$ (f denotes here the frequency of the vortex shedding). The effects of the heating as a relation of St, Re and Pr numbers was studied also theoretically, see the empirical formula derived in [3], which shall also be used for the comparison with the results of the computation.

4. Results

The value of the Strouhal number can be determined from the temporal oscilations of the approximative values of the forces acting on the cylinder. These forces are often denoted as *lift* and *drag* force. As the flow develops to the von Kármán vortex street, the oscillations tend to stable frequency. The stabilized periodicity was recognized in the data and the averaged frequency through multiple periods was computed (see Figure 3). Since the flow develops slowly from the constant initial conditions, a long time computation was needed (about 300000 time steps $\Delta t = 0.001$, depending on the Reynolds number).

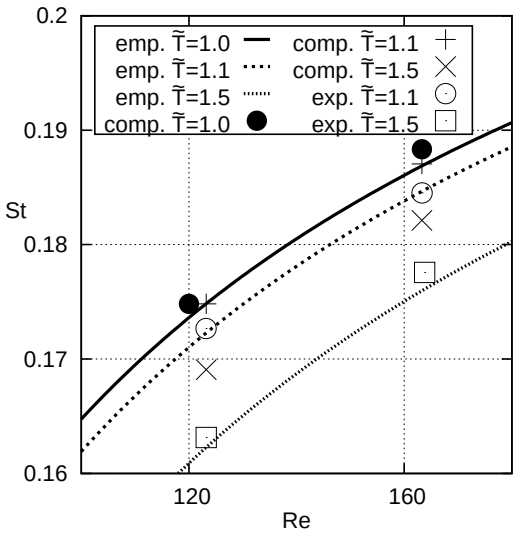


Figure 2: The resulting dependence of Strouhal number on Reynolds number for various temperatures ($\tilde{T} = T_W/T_\infty$) and flow of air.

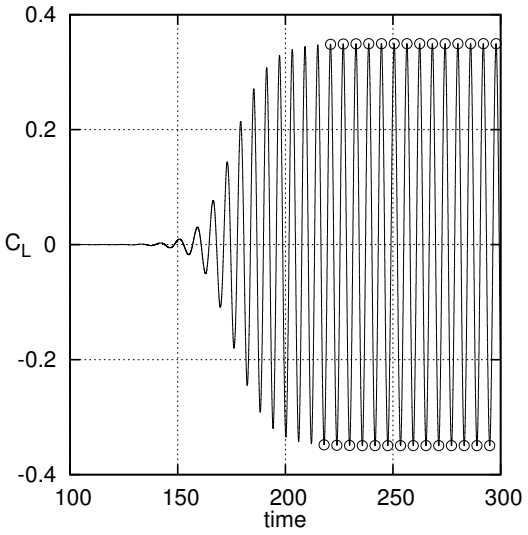


Figure 3: Plot of the lift coefficient C_L for $Re = 123.2$, $T_W/T_\infty = 1.5$ in the flow of air. Rings indicate the extremes taken for computation of the Strouhal number.

The resulting graphs of $St - Re$ dependencies for both air and water, for various cylinder temperatures, is shown in Fig. 2 and Fig. 4. The continuous curve is given by the empirically obtained formula, see [3].

5. Conclusion

The presented results demonstrate applicability of the computational scheme (6)–(11) introduced in combination with high order spatial approximation. Obtained $St - Re$ dependencies show qualitatively good agreement with the experimental results ([5, 3]) across various cylinder temperatures. Observed shift of the data is mostly caused by insufficient expansion basis, since the expansion coefficients of the highest orders were converged only to the values around 0.01. This setting of the expansion basis was chosen due to high memory demands of the matrix system of reference computations, since the work was performed on single CPU.

The increase/decrease of the Strouhal number caused by heating is smaller than predictions of both the experiment and the empirical formula. On the other hand, in case of water flow, the differences of the experiment from the empirical formula are on the same level as the error of the computation.

The results are well comparable with standard approaches using hundreds of thousands low order elements. The main advantage of the spectral approach stays in the significant reduction of number of DOFs and possible reaching of exponential error decay. Achievement of more accurate solutions will be the goal of future computations.

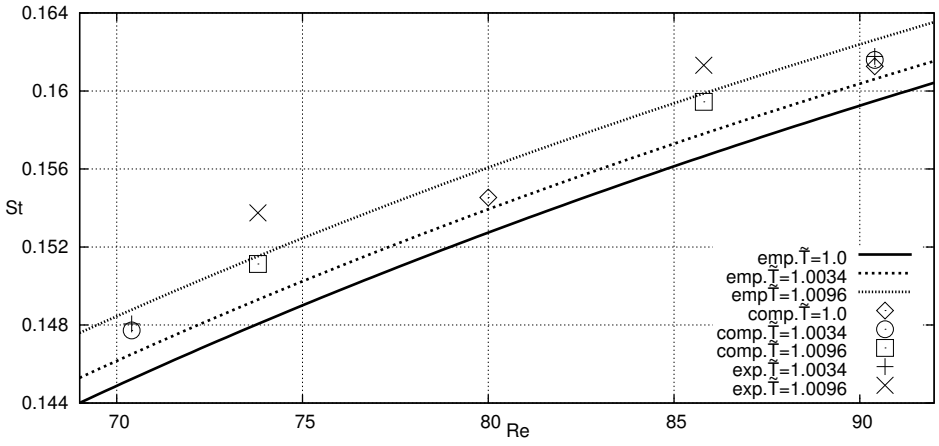


Figure 4: Resulting $St - Re$ dependence for various temperatures in flow of water.

Acknowledgements

The result was developed within the CENTEM project, no. CZ.1.05/2.1.00/ 03.0088, co-funded by the ERDF as part of the Ministry of Education, Youth and Sports OP RDI programme.

References

[1] Karamanos, G. and Sherwin, S.: A high order splitting scheme for the Navier-Stokes equations with variable viscosity. Applied Numerical Mathematics **33** (2000), 455–462.

[2] Karniadakis, G. and Sherwin, S.: *Spectral/hp element methods for computational fluid dynamics.* Oxford University Press, 2005.

[3] Maršík, F., Trávníček, Z., Yen, R., Tu, W., and Wang, A.: Sr-Re-Pr relationship for a heated/cooled cylinder in laminar cross flow. In: *Proceedings of CHT-08 ICHMT International Symposium on Advances in Computational Heat Transfer*, 2008.

[4] Pech, J. and Maršík, F.: 2D simulation of flow around heated circular cylinder with variable viscosity using high order spectral elements. In: *Proceedings of Conference Topical Problems of Fluid Mechanics 2014*, pp. 89–92, 2014.

[5] Vít, T., Ren, M., Trávníček, Z., Maršík, F., and Rindt, C.: The influence of temperature gradient on the Strouhal–Reynolds number relationship for water and air. Experimental Thermal and Fluid Science **31** (2007), 751–760.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

MINIMIZATION OF A CONVEX QUADRATIC FUNCTION SUBJECT TO SEPARABLE CONICAL CONSTRAINTS IN GRANULAR DYNAMICS

Lukáš Pospíšil, Zdeněk Dostál

FEECS VŠB-Technical University of Ostrava
17. listopadu 15, CZ-70833 Ostrava, Czech Republic
lukas.pospisil@vsb.cz, zdenek.dostal@vsb.cz

Abstract

The numerical solution of granular dynamics problems with Coulomb friction leads to the problem of minimizing a convex quadratic function with semidefinite Hessian subject to a separable conical constraints. In this paper, we are interested in the numerical solution of this problem. We suggest a modification of an active-set optimal quadratic programming algorithm. The number of projection steps is decreased by using a projected Barzilai-Borwein method. In the numerical experiment, we compare our algorithm with Accelerated Projected Gradient method and Spectral Projected Gradient method on the solution of a particle dynamics problem with hundreds of spherical bodies and static obstacles.

1. Time-stepping scheme and formulation of optimization problem

In our simulation, we consider a system of $nb \in \mathbb{N}$ *particles* in vector space $\{(x, y, z) \in \mathbb{R}^3\}$. The position of each particle in time t is defined by the *vector of generalized position* $q_i^{(t)} \in \mathbb{R}^7$, which consists of the position of the centre of gravity $[r_x, r_y, r_z]^T$ and the unit quaternion of rotation $[e_0, e_1, e_2, e_3]^T$. The velocity of the body is defined by the *vector of generalized velocities* $v_i^{(t)} \in \mathbb{R}^6$, it includes the velocity corresponding to the position of the centre of the body and angular velocities represented in Euler angles.

We use the well-known time-stepping scheme, see Heyn [9] or Heyn et al. [10]

$$\begin{aligned} \boldsymbol{q}^{(t+h)} &= \boldsymbol{q}^{(t)} + hQ\boldsymbol{v}^{(t)} \ , \\ \boldsymbol{v}^{(t+h)} &= \boldsymbol{v}^{(t)} + hM^{-1}(\boldsymbol{F}_{ext} + \boldsymbol{F}_C) \ , \end{aligned} \tag{1}$$

where h is a time step, Q denotes the matrix of linear mapping between the derivative of the position vector and the vector of velocities, M is a generalized mass matrix, $\boldsymbol{F}_C$ is a vector of forces induced by contact constraints, and $\boldsymbol{F}_{ext}$ is a vector of external forces. In our simulation, the vector of external forces represents the gravity force applied to each body.

Let us denote the number of contacts by $m \in \mathbb{N} \cup \{0\}$. The contact force applied to each body can be separated into the sum of the normal force and the tangential force, i.e.,

$$\boldsymbol{F}_C = \boldsymbol{F}_n + \boldsymbol{F}_T = \gamma_n \boldsymbol{n} + \gamma_u \boldsymbol{u} + \gamma_w \boldsymbol{w} \ ,$$

where $\gamma_n > 0$ is the size of the normal component of the friction force, and $\gamma_u, \gamma_w \in \mathbb{R}$ are the sizes of the tangential components of the friction force. Here, $\{\boldsymbol{n}, \boldsymbol{u}, \boldsymbol{w}\}$ is an orthonormal basis of the tangential space at the contact point. The relation between the components of $\boldsymbol{\gamma}_j := [\gamma_n, \gamma_u, \gamma_w]$ for j-th contact ($j = 1, \ldots, m$) can be described by the *Coulomb friction model*. The unknown vector of all components in all contacts can be denoted by $\boldsymbol{\gamma} := [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_m] \in \mathbb{R}^{3m}$ and can be found by solving the problem of minimizing a convex quadratic function subject to separable conical constraints (see Heyn [9]). The proof of equivalency is based on the maximum dissipation principle and duality.

The optimization problem is given by

$$\text{find} \quad \boldsymbol{\gamma} := \arg\min_{\boldsymbol{x} \in \Omega} f(\boldsymbol{x}), \quad f(\boldsymbol{x}) := \frac{1}{2} \boldsymbol{x}^T A \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x} \ , \tag{2}$$

where $A \in \mathbb{R}^{3m,3m}$ is a symmetric positive semidefinite matrix, $\boldsymbol{b} \in \mathbb{R}^{3m}$, and $\Omega \subset \mathbb{R}^{3m}$ is a non-empty convex feasible set defined by separable conical constraints

$$\Omega := \{\boldsymbol{x} \in \mathbb{R}^{3m} : h_j(x_{2j-2}, x_{2j-1}, x_{2j}) \leq 0, j = 1, \ldots, m\} \ ,$$

where $h_j : \mathbb{R}^3 \to \mathbb{R}$ are conical constraints functions

$$h_j(x, y, z) := \sqrt{y^2 + z^2} - \mu_j x, \quad j = 1, \ldots, m \ ,$$

and $\mu_j \geq 0$ are given friction coefficients that define the interior angles of cones. Let us notice, that if we consider the problem without friction, then $\mu_j = 0$, and the optimization problem (2) becomes a quadratic programming problem with bound constraints.

For the sake of simplicity we denote the triplet of compoments of $\boldsymbol{x} \in \mathbb{R}^n$ constrained by j-th constraint function using the notation of index sets

$$\mathcal{I}_j := \{3j - 2, 3j - 1, 3j\}, \quad \bigcup_{j=1}^{m} \mathcal{I}_j = \{1, \ldots, n\}, \quad j = 1, \ldots, m \ .$$

2. Active-set method

For numerical solution of the problem (2), we are using the variant of Modified Proportioning with Gradient Projection (MPGP), see Dostál [5] and Dostál et al. [7, 4], or Pospíšil [12]. This active-set algorithm is based on the decomposition of the set of all constraint indices $\mathcal{M} := \{1, \ldots, m\}$ into two disjoint subsets based on the values of constraint functions

$$\begin{aligned} \mathcal{F}(\boldsymbol{x}) &:= \{j \in \mathcal{M} : h_j(\boldsymbol{x}_{\mathcal{I}_j}) < 0\} \ , \\ \mathcal{A}(\boldsymbol{x}) &:= \{j \in \mathcal{M} : h_j(\boldsymbol{x}_{\mathcal{I}_j}) = 0\} \ . \end{aligned}$$

The gradient of the objective function $\boldsymbol{g} := \nabla f(\boldsymbol{x}) = A\boldsymbol{x} - b \in \mathbb{R}^n$ can be used to define the *free* and the *chopped* gradient with components

$$\begin{array}{ll} \boldsymbol{\varphi}_{\mathcal{I}_j}(\boldsymbol{x}) = \boldsymbol{g}_{\mathcal{I}_j} \text{ for } j \in \mathcal{F}(\boldsymbol{x}), & \boldsymbol{\varphi}_{\mathcal{I}_j}(\boldsymbol{x}) = 0 \text{ for } j \in \mathcal{A}(\boldsymbol{x}), \\ \boldsymbol{\beta}_{\mathcal{I}_j}(\boldsymbol{x}) = 0 \text{ for } j \in \mathcal{F}(\boldsymbol{x}), & \boldsymbol{\beta}_{\mathcal{I}_j}(\boldsymbol{x}) = \boldsymbol{g}_{\mathcal{I}_j} - \min\{n_j^T(\boldsymbol{x}_{\mathcal{I}_j})\boldsymbol{g}_{\mathcal{I}_j}, 0\} n_j(\boldsymbol{x}_{\mathcal{I}_j}) \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j \in \mathcal{A}(\boldsymbol{x}), \end{array}$$

where $n_j(x, y, z)$ is the unit outer normal of j-th constraint $h_j(x, y, z)$. We consider a problem with conical constraints, so outer normal is given by

$$n_j(x, y, z) := \begin{cases} [-1, 0, 0]^T & \text{if } x = y = z = 0\ , \\ \left[-\mu_j, y/\sqrt{y^2 + z^2}, z/\sqrt{y^2 + z^2}\right]^T & \text{elsewhere.} \end{cases}$$

Algorithm 1: **Modified Proportioning with Barzilai-Borwein Gradient Projection (MPGPS-BB).**

Choose $\boldsymbol{x}^0 \in \Omega$
for $k = 0, 1, 2, \ldots$ *(while a stopping criterion is not achieved)*
  if $\|\varphi(x_k)\| \geq \|\beta(x_k)\|$ (proportioning condition)
    Control the solvability
     if $\min\{\alpha_f, \alpha_{cg}\} = \infty$, *then the problem has no solution.*
    CG step *or* CG halfstep
     make one CG step to solve problem on free set
     if this step means leaving Ω*, do only a half-step and restart CG*
  else
    Gradient projection step.
     make projected Barzilai-Borwein step
     restart CG on free set
  endif
  $k := k + 1$
endfor

Our algorithm is based on using the free and chopped gradient to minimize the objective function on the free set and afterwards on the active set. The switching between these processes is realized by the proportioning condition. The implementation details of each step are the same as in the original Modified Proportioning with Gradient Projections algorithm (MPGP) in Dostál [5], Dostál et al. [7, 4]. Nevertheless, MPGP was developed to solve the problems with a symmetric positive definite Hessian matrix. The recent generalization to the problems with symmetric positive semidefinite Hessian suggests only one difference from the original algorithm,

specifically a test of the problem solvability, see Algorithm 1. The coefficient α_f is the maximal feasible step-size and α_{cg} is a coefficient of the conjugate gradient computed from the free gradient. If both of these coefficient are equal to infinity, then the problem has no solution. The theory will be published in [6].

To solve a problem with separable conical constraints, we suggest to use the projected version of Barzilai-Borwein method [2] instead of the projected gradient method with constant step-length as in original MPGP algorithm. Constant step-length always induces the descend of cost function, as it was shown by Dostál and Schöberl [8]. However, the numerical experiments show that using non-monotone algorithms, such as projected Barzilai-Borwein (PBB) given by

$$\boldsymbol{x}^{k+1} = P_\Omega(\boldsymbol{x}^k - \alpha_k^{BB}\nabla f(x^k)), \quad \alpha_k^{BB} = \frac{\boldsymbol{s}_k^T\boldsymbol{s}_k}{\boldsymbol{s}_k^T A\boldsymbol{s}_k}, \quad \boldsymbol{s}_k = \boldsymbol{x}^k - \boldsymbol{x}^{k-1},$$

usually evokes the decrease of the projection steps number. This modification was inspirated by the Spectral Projected Gradient method (SPG), which uses the similar type of steps, see Birgin et al. [3]. The idea of the combination of MPGP and PBB was firstly presented by Pospíšil [12] and tested on the problem with separable quadratic constraints.

The main shortage of the presented MPGPS-BB algorithm is the absence of the proof of convergence. The PBB method is non-monotone and hardly analyzable. Therefore, the SPG method is using an additional line-search method to control the descend of the objective function, i.e. the global convergence. In our algorithm, we tried to omit this line-search. Our idea is well-founded by the numerical experiment presented in the final section of this paper.

As a stopping criterion in our algorithm, we are using the norm of the *scaled projected gradient* defined by

$$\tilde{\boldsymbol{g}}_\alpha^P(\boldsymbol{x}) := \frac{1}{\alpha}(\boldsymbol{x} - P_\Omega(\boldsymbol{x} - \alpha\nabla f(\boldsymbol{x}))) \ .$$

The equivalency of this gradient and the fulfilment of Karush-Kuhn-Tucker optimality conditions for problems with feasible sets with strong curvature was discussed and proved by Bouchala et al. [4].

3. Numerical experiments

In this section, we present the numerical results showing the efficiency of our algorithm on the simulation of 339 spherical particles with friction. In our benchmark, the particles are scattered into simple box represented by six walls. The initial position of the partices and final position can be found in Fig. 1, where we depicted only the partices and the bottom side of the static box. The material of the bodies is represented by density $\rho = 730$ kg.m^{-3} and friction parameter $\mu = 0.3$. The stepsize of the time-stepping scheme is $h = 6.25 \cdot 10^{-4}$ s.

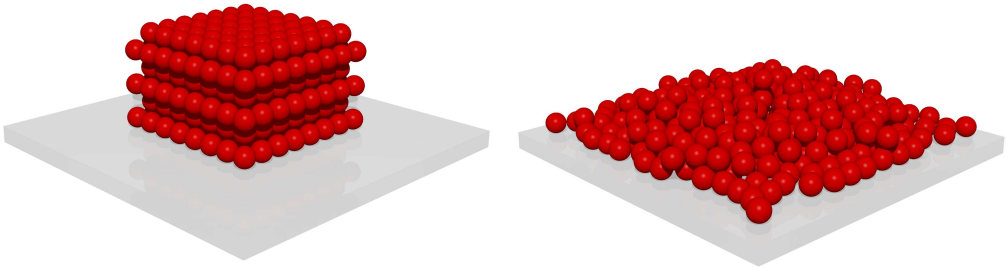


Figure 1: State of testing benchmark at $t = 0$s (left) and $t = 5$s (right).

t	contacts	n	active	MPGPS-BB	SPG	APGD
1s	738	2214	562 (76%)	274 (7.6s)	2360 (37.8s)	754 (9.4s)
2s	702	2106	574 (82%)	137 (3.4s)	449 (6.0s)	346 (2.9s)
3s	730	2190	558 (76%)	137 (3.7s)	449 (6.0s)	346 (4.1s)
4s	814	2442	640 (79%)	338 (9.9s)	2931 (56.2s)	1345 (18.9s)
5s	818	2454	652 (80%)	425 (12.3s)	4176 (88.0s)	1742 (25.6s)

Table 1: The optimization problems at selected times of the simulation; number of contacts, dimension of the problem, the number of iterations and computing time of the algorithms.

We compare our algorithm with SPG and the Accelerated Projected Gradient Descend method (APGD [11]). In SPG, because the minimum of the quadratic function in a given direction is known, we use the Cauchy step-size instead of using an additional Grippo-Lampariello-Lucidi line-search. All algorithms were implemented in the Matlab environment. For contact detection, we are using our own implementation of the *Moving Bounding-Box algorithm* [13]. The number of iterations at selected times of the simulation can be found in Table 1. We demand the relative stopping tolerance $\|\tilde{\boldsymbol{g}}_\alpha^P(\boldsymbol{x})\| < \epsilon\|b\|, \epsilon = 10^{-6}$.

4. Conclusions

In our paper, we proposed the modification of our active-set algorithm for the solution of optimization problem in particle dynamics with friction. Our numerical experiment shows the efficiency of the modifications. Unfortunately, the basic disadvantage of using the projected Barzilai-Borwein method is the absence of a convergence proof as well as of an estimate of the speed of convergence.

5. Acknowledgements

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and project SGS SP2014/204.

References

[1] Anitescu, M.: Optimization-based simulation of nonsmooth rigid multibody dynamics. Math. Program. **105** (2006), 113–143.

[2] Barzilai, J. and Borwein, J. M.: Two point step size gradient methods. IMA J. Numer. Anal. **8** (1988), 141–148.

[3] Birgin, E. G., Martinez, J. M., and Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10** (2000), 1196–1211.

[4] Bouchala, J., Dostál, Z., Kozubek, T., Pospíšil, L., and Vodstrčil, P.: On the solution of convex QPQC problems with elliptic and other separable constraints. Appl. Math. Comput. **247** (2014), 848–864.

[5] Dostál, Z.: *Optimal quadratic programming algorithms, with applications to variational inequalities*, 1st edition. SOIA 23. Springer US, New York, 2009.

[6] Dostál, Z. and Pospíšil, L.: Minimization of the quadratic function with semidefinite Hessian subject to the bound constraints. in preparation.

[7] Dostál, Z. and Pospíšil, L.: Optimal iterative QP and QPQC algorithms. Ann. Oper. Res. (2013).

[8] Dostál, Z., and Schöberl, J.: Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. Comput. Optim. Appl. **30** (2005), 23–44.

[9] Heyn, T.: *On the modeling, simulation, and visualization of many-body dynamics problems with friction and contact*. Ph.D. Thesis, University of Wisconsin-Madison, 2013.

[10] Heyn, T., Anitescu, M., Tasora, A., and Negrut, D.: Using Krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation. Internat. J. Numer. Methods Engrg. **95** (2012), 541–561.

[11] Nesterov, Y.: *Introductory lectures on convex optimization: a basic course*. Volume 87, Springer, 2003.

[12] Pospíšil, L.: An optimal algorithm with Barzilai-Borwein steplength and superrelaxation for QPQC problem. In: J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 16*, pp. 155–161. IM ASCR, Prague, 2012.

[13] Schinner, A.: Fast algorithms for the simulations of polygonal particles. Springer-Verlag Granular Matter **2** (1999), 35–43.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

PROCESSES IN CONCRETE DURING FIRE

Petra Rozehnalová, Anna Kučerová, Petr Štěpánek

Brno University of Technology, Faculty of Civil Engineering
Veveří 331/95, 602 00 Brno, Czech Republic
rozehnalova.p@fce.vutbr.cz, kucerova.a@fce.vutbr.cz, stepanek.p@fce.vutbr.cz

Abstract

Paper deals with hydro-thermal performance of concrete exposed to a fire. It is introduced mathematical model, numerical approach and some results provided by the model.

1. Introduction

Behavior of concrete exposed to the high temperature plays crucial role in the assessment of the reliability of concrete structure. There exist several mathematical models that aim to predict and simulate such a behavior. One of the first models was developed by Bažant and Thonguthai. Its improved version is described in [3] or in [4]. Another model was formulated by Gawin et al. [6] or by Dwaikat and Kodur in [5]. These models differ in its complexity, dimension, number of variables and equations. Their common characteristic is that models contain nonlinear differential equations and lot of empirical data.

In the paper we introduced mathematical model which is slightly revised and modified approach of [4]. The model belongs to the simpler ones because the only one phase (free water) is assumed. Surprisingly some phenomena observed in experiments can by explained by the analysis of the model.

2. Physical phenomena

Let us describe physical processes, which occur in concrete during fire. Concrete is non-combustible material with low thermal conductivity. Although concrete does not contribute to fire load of the structures significant changes occur in its structure during a fire exposure. Besides reduction of mechanical, deformation and material properties also chemical composition of concrete is varied during heating [3].

Concrete, as a porous material, contains a large amount of pores, which can be filled fully (saturated concrete) or just partially with water. The water occurred in the pores is evaporable water and starts to evaporate at early beginning of the fire. The first changes of concrete structure arise at 105 °C as stated in [8], when

chemically bounded water is released from cement gel to the pores. Some small micro-cracks start to appear as the capillary porosity arises. The peak of the dehydration process is reached around 270 °C. The color of concrete is changed and a slight decrease of strength, modulus of elasticity and changes in material properties like thermal conductivity can be noted. Temperature of 300 °C is the extreme temperature beside which the concrete structure is irreversibly damaged [7]. In range of 400–600 °C calcium hydroxide decomposes into calcium oxide plus water (rise of amount of free water) and transition of α and β quartz, accompanied by increase in its volume, induces another creation of severe cracks in concrete.

Simultaneously with a change of temperature can be investigated also the change of mass of free water (mostly vapor) and distributions of pore pressure. The pore pressure is one of the main reasons of concrete spalling, which happened at the beginning of heating (10–30 minutes) and is accidental. Small or grater areas of concrete cover can be broken and cross section of member is reduced then. Furthermore in most cases the reinforcement is exposed directly to the fire and the member is heated faster, which can lead to loss of loadbearing capacity.

3. Mathematical model

The aim is to model behavior described above. We consider two-dimensional model. Let Ω be a domain representing a concrete skeleton with the points $\mathbf{x}=(x_1, x_2)$. Let us denote by Γ the boundary of domain Ω. The boundary consists of two parts: Γ_R, which represents part exposed to the fire and Γ_N, which is exposed to the atmosphere. It is supposed that Γ_R and Γ_N are non-intersecting sets and $\Gamma_R \cup \Gamma_N = \Gamma$. By $\mathbf{n} = (n_1, n_2)$ is denoted outer unite normal of Ω.

In the model, there are three unknowns: $w(\mathbf{x}, t)$ denotes amount of free water, $P(\mathbf{x}, t)$ is pore pressure and $T(\mathbf{x}, t)$ is temperature in the point $\mathbf{x}$ and time t.

Mass balance equation of free water takes into account diffusive flow (L 1.2) and variation (L 1.1) of free water. Source of the free water is water dehydrated from the skeleton (R 1.1). The equation is:

$$\underbrace{\frac{\partial w}{\partial t}}_{L\,1.1} + \underbrace{\nabla \cdot \mathbf{J}}_{L\,2.1} = \underbrace{\frac{\partial w_{\mathrm{deh}}}{\partial t}}_{R\,1.1} \quad \text{in } \Omega \times (0, \infty) \,, \tag{1}$$

where $\mathbf{J}$ is flow of free water. Function $w_{\mathrm{deh}} = w_{\mathrm{deh}}(T)$ gives mass of dehydrated water. It is empirical function, we adopted the one specified in [5].

Enthalpy balance equation considers conductive (L 2.2) and convective (L 2.3) heat flows. Source terms in the equation describes effects caused by dehydration of skeleton (R 2.1) and evaporation of free water (R 2.2). Then, the equation is:

$$\underbrace{\rho_s C_s \frac{\partial T}{\partial t}}_{L\,2.1} + \underbrace{\nabla \cdot \mathbf{q}}_{L\,2.2} - \underbrace{C_w \nabla T \cdot \mathbf{J}}_{L\,2.3} = - \underbrace{\Delta H_{\mathrm{deh}} \frac{\partial w_{\mathrm{deh}}}{\partial t}}_{R\,2.1} + \underbrace{\Delta H_{\mathrm{evap}} \frac{\partial w}{\partial t}}_{R\,2.2} \quad \text{in } \Omega \times (0, \infty) \,, \tag{2}$$

where $\mathbf{q}$ means heat flux, $\rho_s = \rho_s(T)$ density of concrete , $C_s = C_s(T)$ specific heat of concrete, C_w specific heat of water, ΔH_{deh} enthalpy of dehydration, $\Delta H_{\text{evap}} = \Delta H_{\text{evap}}(T)$ enthalpy of evaporation.

State equation Now, we have three unknowns and only two equations, (1) and (2). For that reason we add state equation

$$w = \Phi(P, T), \tag{3}$$

where Φ is empirical function described in [10], p. 530.

Constitutive relationship: According to [3], the heat and moisture flux can be considered in the form of Fourier's respectively Darcy's law, i. e.:

$$\mathbf{J} = -\frac{K}{g}\nabla P \quad \text{and} \quad \mathbf{q} = -\lambda \nabla T,$$

where $K = K(T, P)$ denotes permeability of concrete and $\lambda = \lambda(T)$ thermal conductivity and g gravitational acceleration (included for the reasons of dimensionality).

Boundary conditions: The model is completed with boundary conditions. They are of the Robin type:

$$-\mathbf{J}\cdot\mathbf{n} = \beta_N(P - P_\infty) \qquad \text{on } \Gamma_N \times (0,\infty), \tag{4}$$
$$-\mathbf{J}\cdot\mathbf{n} = \beta_R(P - P_\infty) \qquad \text{on } \Gamma_R \times (0,\infty), \tag{5}$$
$$-\mathbf{q}\cdot\mathbf{n} = \alpha_N(T - T_\infty) \qquad \text{on } \Gamma_N \times (0,\infty), \tag{6}$$
$$-\mathbf{q}\cdot\mathbf{n} = \alpha_R(T - T_{\text{en}}) + e\sigma(T^4 - T_{\text{en}}^4) \qquad \text{on } \Gamma_R \times (0,\infty), \tag{7}$$

where α_R, α_N are heat transfer coefficients for boundary exposed to the high temperature and to the atmosphere , β_R, β_N denote coefficients of moisture transfer through the boundary Γ_R resp. Γ_N, e emissivity of concrete and σ Stefan- Boltzmann constant. P_∞ resp. T_∞ denotes outer pressure resp. temperature and finally $T_{\text{en}} = T_{\text{en}}(t)$ gives temperature caused by fire.

Initial conditions: To describe environment for $t = 0$, we prescribe initial conditions

$$P(\mathbf{x}, 0) = P_0 \qquad \text{for } \mathbf{x} \in \Omega, \tag{8}$$
$$T(\mathbf{x}, 0) = T_0 \qquad \text{for } \mathbf{x} \in \Omega, \tag{9}$$

where P_0 and T_0 are pressure and temperature in $t = 0$.

4. Numerical methods

Equations (1)–(3) together with boundary conditions (4)–(7) and with initial conditions (8), (9) form mathematical model. This model is implemented in Matlab, where we use following approach.

For time discretization we use Rothe method. It leads to a system of nonlinear partial differential equations. To solve this we used finite element method in each time step. Basis and test functions are bilinear polynomials as we choose, for spatial discretization, square conforming uniform mesh. Integrals appearing in finite element method are computed by Gaussian quadrature. Finite element method provides system of nonlinear equations, which is solved by Newton's method. Stopping criteria is residual tolerance set to the value 10^{-8}.

5. Example

Let us present results of our model problem. The set Ω is a rectangle $50\,\mathrm{mm} \times 100\,\mathrm{mm}$. Γ_R is left and upper side of the Ω and so Γ_R is right and lower side.

The data of the model were set as follows: $C_w = 4180$ J kg^{-1} °C^{-1}, $\Delta H_{\mathrm{deh}} = 2.44 \cdot 10^{-6}$ J kg^{-1}, $g = 9.81$ m s$^-2$, $\alpha_R = 25$ W m^{-2} °C^{-1}, $\alpha_N = 4$ W m^{-2} °C^{-1}, $\beta_R = 20 \cdot 10^{-9}$ s m^{-1}, $\beta_N = 10 \cdot 10^{-9}$ s m^{-1}, $e = 0.7$, $P_\infty = P_0 = 1330$ Pa, $T_\infty = T_0 = 25\,^\circ\mathrm{C}$.

For thermal conductivity of concrete λ holds, see [2], $\lambda_{\mathrm{low}} \le \lambda \le \lambda_{\mathrm{up}}$, where

$$\lambda_{\mathrm{low}}(T) = 2 - 0.2451\,\frac{T}{100} + 0.0107\left(\frac{T}{100}\right)^2, \quad \lambda_{\mathrm{up}}(T) = 1.36 - 0.136\,\frac{T}{100} + 0.0057\left(\frac{T}{100}\right)^2.$$

In the model was set $\lambda = \frac{\lambda_{\mathrm{low}} + \lambda_{\mathrm{up}}}{2}$.

Following [2], density of concrete ρ_s and specific heat of concrete $C_s(T)$ is:

$$\rho_s(T) = \begin{cases} 2500 & \text{for} \quad 20\,^\circ\mathrm{C} \le T \le 115\,^\circ\mathrm{C}, \\ 2500\left(1 - 0.02\,\frac{T-115}{85}\right) & \text{for} \quad 115\,^\circ\mathrm{C} \le T \le 200\,^\circ\mathrm{C}, \\ 2500\left(0.98 - 0.03\,\frac{T-200}{200}\right) & \text{for} \quad 200\,^\circ\mathrm{C} \le T \le 400\,^\circ\mathrm{C}, \\ 2500\left(0.95 - 0.07\,\frac{T-400}{800}\right) & \text{for} \quad 400\,^\circ\mathrm{C} \le T \le 1200\,^\circ\mathrm{C}\,, \end{cases}$$

and

$$C_s(T) = \begin{cases} 900 & \text{for} \quad 20\,^\circ\mathrm{C} \le T \le 100\,^\circ\mathrm{C}, \\ 800 + T & \text{for} \quad 100\,^\circ\mathrm{C} \le T \le 200\,^\circ\mathrm{C}, \\ 900 + \frac{T}{2} & \text{for} \quad 200\,^\circ\mathrm{C} \le T \le 400\,^\circ\mathrm{C}, \\ 1100 & \text{for} \quad 400\,^\circ\mathrm{C} \le T \le 1200\,^\circ\mathrm{C}\,. \end{cases}$$

Enthalpy of evaporation is given in [9],

$$\Delta H_{\mathrm{evap}}(T) = 2.672 \cdot 10^5\ (374.15 - T)^{0.38} \quad \text{for } T \le 400\,^\circ\mathrm{C}.$$

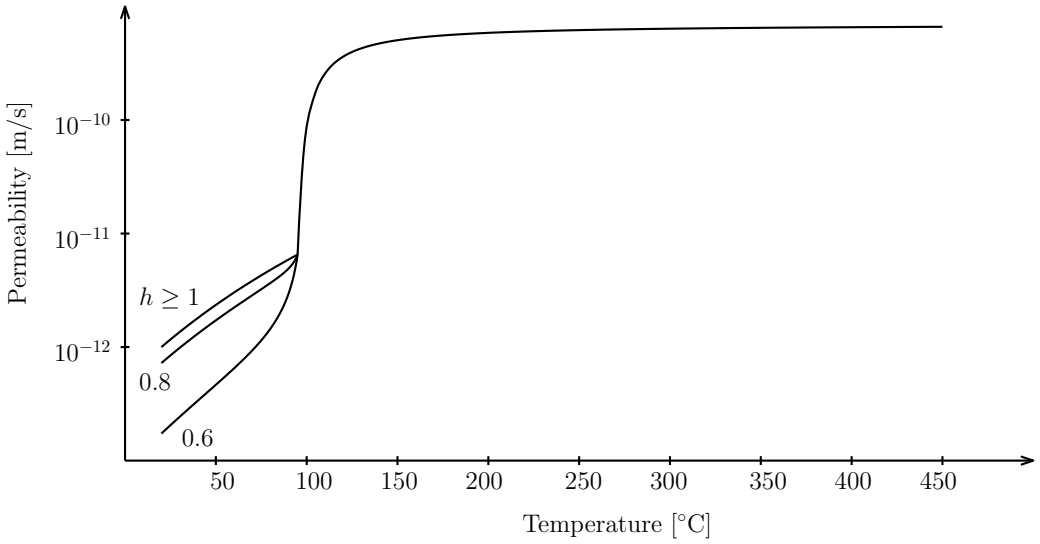


Figure 1: Logarithmic plot of permeability $K\left(T, h(T,P)\right)$ of concrete

As T_{en}, we used ISO curve given by [1], $T_{\mathrm{en}}(t) = T_0 + 345\,\log(480\,t+1)$.

Permeability $K(T,P)$ can be found in [3] and is given by relationship:

$$K(T,h) = \begin{cases} 10^{-12}\left(\alpha + \frac{1-\alpha}{1+\left(\frac{1-h}{0.25}\right)^4}\right) \mathrm{e}^{2\,700\left((T_0+273.15)^{-1}-(T+273.15)^{-1}\right)} & \text{for} \quad T \leq 95\,^{\circ}\mathrm{C},\ h \leq 1, \\ 10^{-12}\,\mathrm{e}^{2\,700\left((T_0+273.15)^{-1}-(T+273.15)^{-1}\right)} & \text{for} \quad T \leq 95\,^{\circ}\mathrm{C},\ h > 1, \\ 10^{-12}\,\mathrm{e}^{2\,700\left((T_0+273.15)^{-1}-(368.15)^{-1}\right)}\,\mathrm{e}^{\frac{T-95}{0.881+0.214(T-95)}} & \text{for} \quad T > 95\,^{\circ}\mathrm{C}, \end{cases}$$

where several auxiliary functions are used. We define $\alpha(T)$ and $h(T,P)$ as

$$\alpha(T) = \left(1 + \frac{19\,(95-T)}{70}\right)^{-1}, \quad h(T,P) = \frac{P}{P_s} = \frac{P}{\mathrm{e}^{23.5771 - \frac{4\,042.9}{(T+273.15)-37.58}}},$$

where P_s is a saturated vapour pressure. Plot of the permeability is on the Figure 1.

Results of the model are on the Fig. 2. Time step is set to 5 sec., number of mesh elements is 20×40.

6. Conclusion

Development of reasonable models for the prediction of behavior of concrete structures is strongly required by the applied research in civil engineering. Practical validation of the models suffers from the lack of data from experiments. Sufficiently general formulation of the problem should be a motivation for further research.

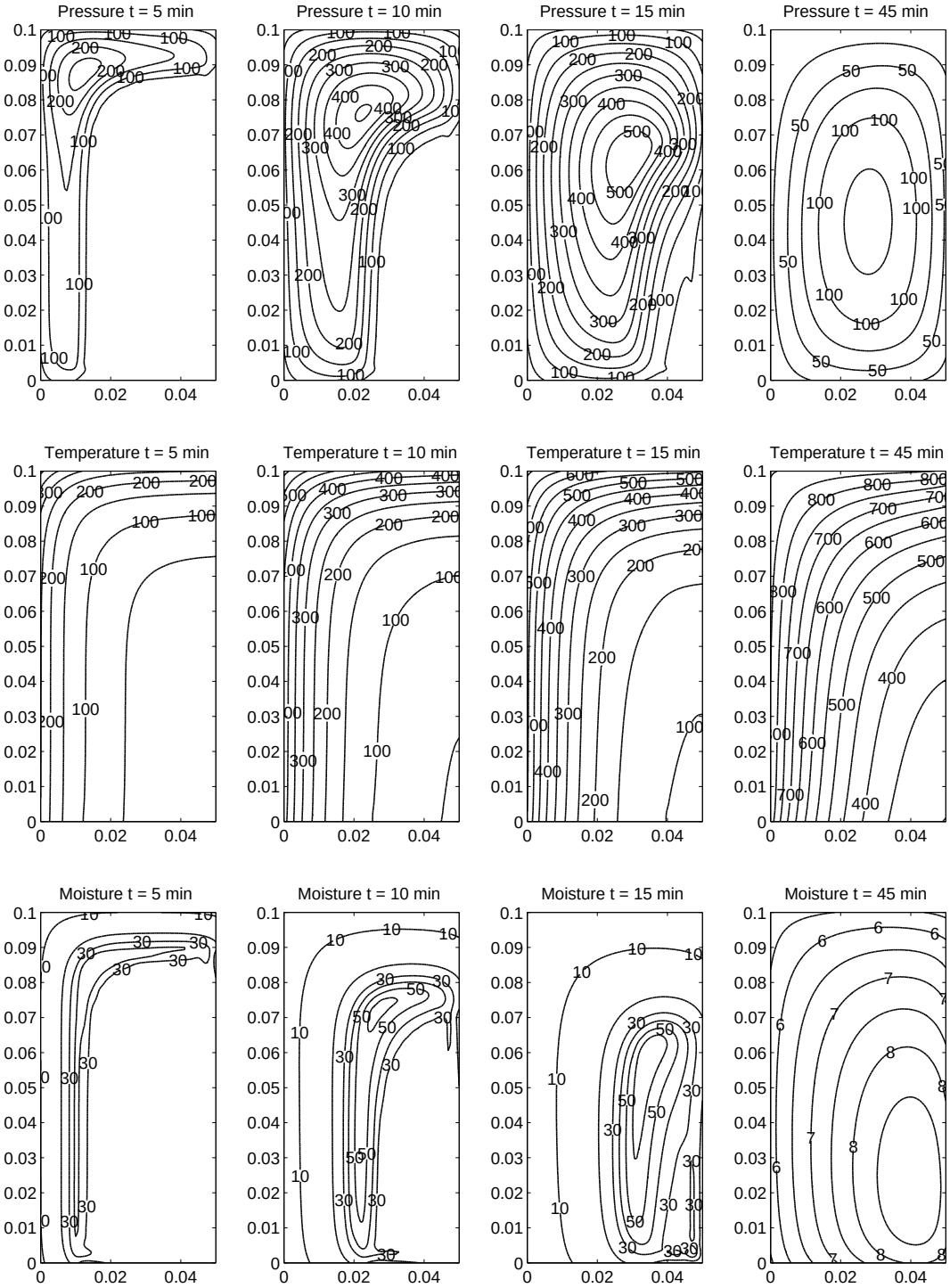


Figure 2: Time development of pressure P [kPa], temperature T [°C] and moisture [kg m^{-3}]. Time step: 5 sek, number of spatial elements: 20×40.

Acknowledgments

This work was supported by Brno University of Technology, grant No. FAST-J-14-2362.

References

[1] ČSN EN 1991-1-2 (730035) eurokód 1: Zatížení konstrukcí - Část 1-2: Obecná zatížení - zatížení konstrukcí vystavených účinkům požáru, 2004.

[2] ČSN EN 1992-1-2 (730035) eurokód 2: Navrhování betonových konstrukcí - Část 1-2: Obecná pravidla - navrhování konstrukcí na účinky požáru, 2006.

[3] Bažant, Z. and Kaplan, M.: *Concrete at high temperatures: material properties and mathematical models.* Longman Group Limited, 1996.

[4] Beneš, M., Štefan, R., and Zeman, J.: Analysis of coupled transport phenomena in concrete at elevated temperatures. Appl. Math. Comput. **219** (2013), 7262–7274.

[5] Dwaikat, M. and Kodur, V.: Hydrothermal model for predicting fire-induced spalling in concrete structural systems. Fire Safety Journal **44** (2009), 425–434.

[6] Gawin, D. and Pesavento, F.: An overview of modeling cement based materials at elevated temperatures with mechanics of multi-phase porous media. Fire Technology **48** (2012), 753–793.

[7] Hertz, K.: *Analyses of concrete structures exposed to fire: U-050.* Department of Buildings and Energy, Technical University of Denmark, 1999.

[8] Ingham, J. P.: Application of petrographic examination techniques to the assessment of fire-damaged concrete and masonry structures. Materials Characterization **60** (2009), 700 – 709. 11th Euroseminar on Microscopy Applied to Building Materials (EMABM).

[9] Moran, M., Shapiro, H., Boettner, D., and Bailey, M.: *Fundamentals of engineering thermodynamics.* John Wiley & Sons, 2010.

[10] Černý, R. and Rovnaníková, P.: *Transport processes in concrete.* Taylor & Francis, 2002.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

IRREGULARITY OF TURING PATTERNS IN THE THOMAS MODEL WITH A UNILATERAL TERM

Vojtěch Rybář, Tomáš Vejchodský

Institute of Mathematics of the Academy of Sciences of the Czech Republic
Žitná 25, Praha 1, 115 67, Czech Republic
{rybar,vejchod}@math.cas.cz

Abstract

In this contribution we add a unilateral term to the Thomas model and investigate the resulting Turing patterns. We show that the unilateral term yields nonsymmetric and irregular patterns. This contrasts with the approximately symmetric and regular patterns of the classical Thomas model. In addition, the unilateral term yields Turing patterns even for smaller ratio of diffusion constants. These conclusions accord with the recent findings about the influence of the unilateral term in a model for mammalian coat patterns [3]. This indicates that the observed effects of the unilateral term are general and apply to a variety of systems.

1. Introduction

Systems of reaction-diffusion equations are widely used to model various phenomena in biology and chemistry. Spatio-temporal ecological models (e.g. predator-pray models), chemical kinetics and tumour growth can serve as examples. In addition, reaction-diffusion systems have successfully explained the spontaneous emergence of skin and coat patters in mammals, fish, gasteropods and others. One of the well-established reaction-diffusion models is the Thomas reaction kinetics model [9]. It has originally been used for modelling of chemical reactions involving oxygen and uric acid. However, Murray in [7] showed that this model can successfully model the formation of coat patterns in mammals.

The mechanism responsible for the creation of spatial patterns is known as the Turing diffusion driven instability [10]. This instability occurs if a spatially homogeneous stationary solution is stable with respect to small spatially homogeneous perturbations and unstable with respect to small spatially heterogeneous perturbations. A new stable and spatially heterogeneous steady state solution can evolve in this case and it is called a pattern. Turing instability is well known and necessary conditions for its emergence are derived, e.g. in [7], under the condition that the corresponding nonlinear terms are smooth.

The main idea of this paper is to consider the Thomas model appended by a non-smooth unilateral term. Reaction-diffusion systems with unilateral terms, mainly

in the form of variational inequalities, have been studied in [1, 4, 5] and several interesting and surprising properties have been reported. For example, there are theoretical studies showing that certain unilateral systems can produce Turing patterns for virtually arbitrary ratio of diffusion coefficients. This is surprising, because the corresponding classical reaction-diffusion system (without any unilateral term) produces Turing patterns only if this ratio is sufficiently away from one.

This motivates us to study the system of reaction-diffusion equations for the evolution of concentrations $u = u(t, x, y)$ and $v = v(t, x, y)$ of two morphogens in the following form:

$$\frac{\partial u}{\partial t} = \Delta u + \gamma(a - u - h(u, v)) \text{ in } (0, T) \times \Omega, \tag{1}$$

$$\frac{\partial v}{\partial t} = d\Delta v + \gamma \left(\alpha b - \alpha v - h(u, v) + \tau(v - \hat{v})^-\right) \text{ in } (0, T) \times \Omega \tag{2}$$

where

$$h(u, v) = \frac{\rho u v}{1 + u + K u^2}.$$

The model parameters $a, b, d, \alpha, \gamma, \tau, K$, and ρ are constants, $\hat{v}$ stands for the second component of the ground state, which is defined below, T denotes the final time, $\Omega \subset \mathbb{R}^2$ is a domain, and $\tau(v - \hat{v})^-$ is the unilateral term which is multiplied by γ in order to make it proportional to the size of the domain Ω in the same manner as the other nonlinear terms. Notice that the negative part is defined as $w^- = \max(0, -w)$. For $\tau = 0$, system (1)–(2) coincides with the original Thomas model. However, in this paper we mainly consider $\tau > 0$ and study the effect of the unilateral term $\tau(v - \hat{v})^-$ on the emerging patterns.

We will couple the model (1)–(2) with zero flux boundary condition

$$\frac{\partial u}{\partial n} = \frac{\partial v}{\partial n} = 0 \text{ on } \partial\Omega, \tag{3}$$

where n stands for the outward unit normal vector to the boundary $\partial\Omega$. The spatially homogeneous steady state solution mentioned above is known as the ground state and it is defined as a pair $\hat{u}, \hat{v} \in \mathbb{R}$, which solves the nonlinear system

$$a - \hat{u} - h(\hat{u}, \hat{v}) = 0 \quad \text{and} \quad \alpha b - \alpha\hat{v} - h(\hat{u}, \hat{v}) = 0.$$

Clearly, the constant functions $u(t, x, y) = \hat{u}$ and $v(t, x, y) = \hat{v}$ form a stationary solution to system (1)–(2) with boundary conditions (3). The component $\hat{v}$ of the ground state is used in (2) to define the unilateral term. Notice that it is nonsmooth exactly at the point $\hat{v}$. The biological motivation for the nonsmooth unilateral term in (2) and its further properties are discussed in the next section.

2. Unilateral terms

A general biological motivation and existing theoretical results for reaction-diffusion systems with unilateral terms are thoroughly discussed in [3]. In this short contribution, we only offer a short overview for the sake of completeness.

System (1)–(2) for concentrations of two morphogens diffusing within a tissue is biologically plausible, because we can expect receptors in the cell membrane that detect the local concentration v of the second morphogen. The cell then reacts in such a way that if the concentration v drops below the threshold value $\hat{v}$, the cell will commence to produce the second morphogen. Similarly, as soon as the concentration v reaches the threshold $\hat{v}$ the cell stops to produce it.

This mechanism is modelled in equation (2) by the unilateral term $\tau(v-\hat{v})^-$. When v is smaller than the threshold $\hat{v}$, the term $(v-\hat{v})^-$ becomes positive and the concentration v starts to increase with the rate $\gamma\tau|v-\hat{v}|$. In other words, the unilateral source term starts to be active. When the concentration v decreases to the level of the threshold $\hat{v}$, the unilateral term $\tau(v-\hat{v})^-$ vanishes and ceases to have any effect in the system.

From both the biological and mathematical point of view it is natural to set the threshold to the value $\hat{v}$ of the ground state. Naturally, the parameter τ governs the intensity of the unilateral term.

If $\tau = 0$ then all nonlinear terms in system (1)–(2) are smooth and the standard linear analysis, see e.g. [7], can be performed to derive the necessary conditions for the Turing instability to occur. In case of system (1)–(2) this analysis restricts the diffusion coefficient d to be sufficiently large, see below. However, recent results [1, 4, 5] surprisingly revealed that this condition on d can be relaxed if certain unilateral terms or conditions are added to the system. This is an interesting feature both mathematically and biologically. Especially, in the light of the common critique of the Turing pattern formation mechanisms, that the diffusion constants of the two morphogens should be similar, because both the morphogens are presumed to be of a similar chemical nature.

The effects of the unilateral term on the resulting patterns have been studied in [3] using a model for generating pigment patterns on coats of leopards and jaguars [2, 6]. Paper [3] concludes that the unilateral term leads to nonsymmetric and irregular patterns and that the patterns appear even for ratios of diffusions violating the condition from the linear analysis. In this contribution, we investigate the Thomas model to see if we can obtain comparable results as in [3]. This would confirm that the conclusions of [3] are more general and do not apply to one specific model only.

3. Numerical results

We solve system (1)–(2) numerically using own finite element solver based on triangular meshes. The Matlab built-in adaptive time-stepping ODE solver `ode15s` is used for the time integration. We use the following set of parameters:

$$a = 150,\ b = 100,\ \alpha = 1.5,\ \gamma = 252,\ K = 0.05,\ \rho = 13. \tag{4}$$

We vary the diffusion coefficient d between 22.5 and 27.5 and the intensity of the unilateral source τ between 0 and 2. The domain is a square $\Omega = (-2, 2)^2$ and the computation is terminated at the final time $T = 4$ as the solution of the system

is already close to the steady state at this point. The ground state for parameter values (4) is approximately $(\hat{u}, \hat{v}) = (37.7380, 25.1588)$. The initial condition is chosen as a small random noise around this ground state. The same initial condition is used for all presented results.

Using these parameter values, we perform a numerical experiment to study the effects of the intensity of the unilateral source τ and the diffusion coefficient d on the resulting Turing patterns. Since both components u and v provide complementary results, we present plots based on v only. Figure 1 shows the resulting Turing patterns for various values of parameters τ and d.

First, we observe the qualitative change of the patterns with growing τ, see the first column in Figure 1. For $\tau = 0$ the pattern consists of close-to-circular spots with similar sizes. These spots are almost symmetrically placed. As the intensity of the unilateral term τ grows, the spots become irregular and gradually more and more elongated. The larger spots seem to be fused from several smaller ones. Starting from the value $\tau = 1$ the pattern is already substantially nonsymmetric and it is qualitatively distinct from the close to regular pattern for $\tau = 0$.

Another outcome of the performed experiment is that the unilateral term enables patterns even for d smaller than the usual linear theory [7] permits. Indeed, if $\tau = 0$ system (1)–(2) contains no unilateral term, the remaining nonlinearities are smooth, and the standard linear analysis of the Turing instability [7] yields the following critical value [8] for the diffusion coefficient d:

$$d_{\text{crit}} = \frac{\det B - b_{12}b_{21} + 2\sqrt{-b_{12}b_{21}\det B}}{b_{11}} \approx 27.027, \tag{5}$$

where

$$B = \begin{bmatrix} b_{11}, b_{12} \\ b_{21}, b_{22} \end{bmatrix} = -\gamma \begin{bmatrix} 1 + \partial h/\partial u, -\partial h/\partial v \\ \partial h/\partial v, \alpha + \partial h/\partial v \end{bmatrix} (\hat{u}, \hat{v}) \approx \begin{bmatrix} 226.7, -1124.5 \\ 478.7, -1502.5 \end{bmatrix}$$

is the Jacobi matrix of system (1)–(2) evaluated at the ground state and the numerical values correspond to (4). The original Thomas model (i.e. the case $\tau = 0$) can exhibit Turing instability only if $d > d_{\text{crit}}$.

We may verify this condition in the first row of Figure 1. The second and subsequent columns of Figure 1 show that as the intensity of the unilateral source τ grows, Turing patterns emerge even for the diffusion coefficient smaller than the critical value (5). In general, this indicates that the additional unilateral term can weaken the condition on the diffusions and enables the emergence of patterns for diffusion coefficients of the two morphogens closer to each other.

4. Conclusions

This contribution evaluates the effect of the additional unilateral source term in the Thomas reaction-diffusion system. We have observed that patters in systems with sufficiently intensive unilateral term are less regular and symmetric compared

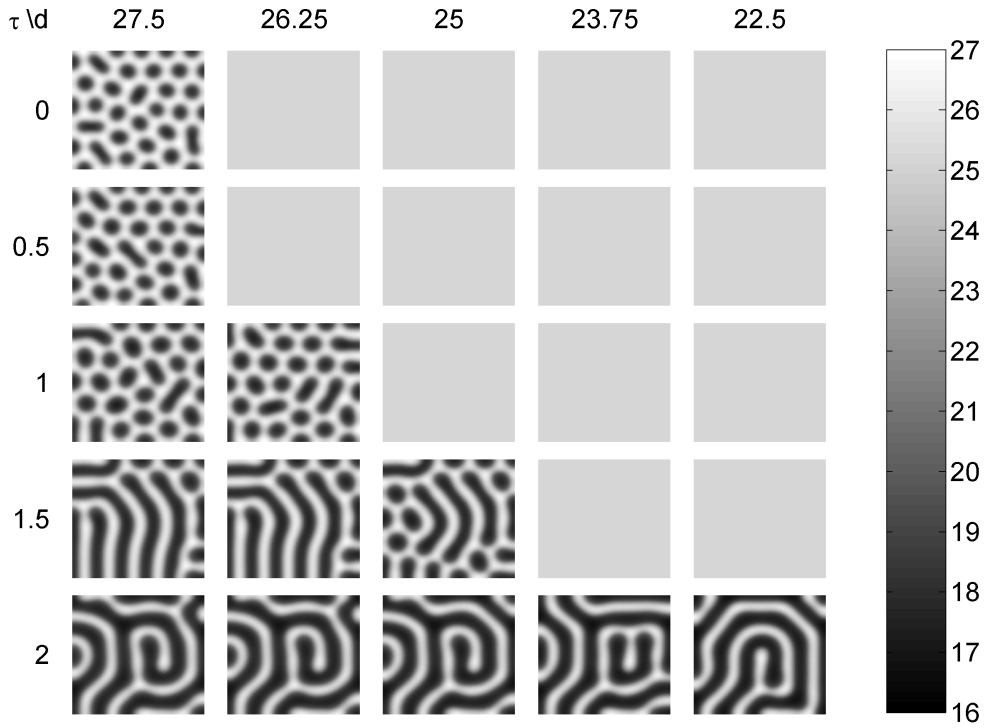


Figure 1: Patterns for various values of the intensity of the unilateral source τ and diffusion coefficient d

to patterns in systems with a weak or no unilateral term. Further, in comparison with classical systems with no unilateral regulation, the unilateral term can enable the emergence of Turing patterns even for those values of the diffusion coefficient d which prevent the Turing instability in the classical systems.

These results accord with conclusions of a more detailed study [3], where a reaction-diffusion model for coat patterns of leopard and jaguar [2, 6] is analysed. Thus, the reported effects of the unilateral source term seem to be more general and valid for more types of reaction-diffusion systems. Beside this, the observed effects verify and illustrate theoretical findings of [4], where a unilateral regulation in terms of variational inequalities is presented.

From the practical point of view, it has been suggested in [3] that the unilateral source term can explain the irregular mutant colouration observed in certain mammals, such as king cheetahs.

Reaction-diffusion systems have been studied for several decades, the corresponding literature is wide and various perspectives are already covered. However, this contribution as well as the paper [3] confirm that there are still aspects, such as the

unilateral source terms that are interesting from both theoretical and practical point of view and that deserve further investigations.

Acknowledgements

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no. 328008. Further, it has been supported by the grant SVV-2014-260106 and by RVO 67985840. This support is gratefully acknowledged.

References

[1] Baltaev, J., Kučera, M., and Väth, M.: A variational approach to bifurcation in reaction-diffusion systems with Signorini type boundary conditions. Appl. Math. **57** (2012), 143–165.

[2] Barrio, R., Varea, C., and Aragón, J.: A two dimensional numerical study of spatial pattern formation in interacting Turing systems. Bull. Math. Biol. **61** (1999), 483–505.

[3] Jaroš, F., Kučera, M., Rybář, V., and Vejchodský, T.: Unilateral regulation breaks regularity of Turing patterns. Preprint arXiv:1502.05371.

[4] Kučera, M. and Väth, M.: Bifurcation for a reaction-diffusion system with unilateral and Neumann boundary conditions. J. Differential Equations **252** (2012), 2951–2982.

[5] Kim, I. S. and Väth, M.: The Krasnoselskii-Quittner formula and instability of a reaction-diffusion system with unilateral obstacles. Dyn. Partial Differ. Equ. **11** (2014), 229–250.

[6] Liu, R. T., Liaw, S. S., and Maini, P. K.: Two-stage Turing model for generating pigment patterns on on the leopard and the jaguar. Phys. Rev. **74** (2006), 011 914.

[7] Murray, J. D.: *Mathematical biology. II. Spatial models and biomedical applications*. Springer-Verlag, New York, 2003.

[8] Nishiura, Y.: Global structure of bifurcating solutions of some reaction-diffusion systems. SIAM J. Math. Anal. **13** (1982), 555–593.

[9] Thomas, D.: Artificial enzyme membranes, transport, memory, and oscillatory phenomena. In: D. Thomas and J. P. Kernevez (Eds.), *Analysis and control of immobilized enzyme systems*, pp. 115–150. North Holland, Amsterdam, 1976.

[10] Turing, A. M.: The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society B **237** (1952), 37–72.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

SMOOTH APPROXIMATION SPACES BASED ON A PERIODIC SYSTEM

Karel Segeth

Institute of Mathematics, Academy of Sciences
Žitná 25, 115 67 Prague 1, Czech Republic
segeth@math.cas.cz

Abstract

A way of data approximation called smooth was introduced by Talmi and Gilat in 1977. Such an approach employs a (possibly infinite) linear combination of smooth basis functions with coefficients obtained as the unique solution of a minimization problem. While the minimization guarantees the smoothness of the approximant and its derivatives, the constraints represent the interpolating or smoothing conditions at nodes. In the contribution, a special attention is paid to the periodic basis system $\exp(-\mathrm{i}kx)$. A 1D numerical example is presented.

1. Introduction

Measurements of the values of a continuous function of one or more independent variables are performed in many branches of science and technology. The data correspond to a finite number of measurement nodes but we need also its extension: the values corresponding to other points in some domain. The way of smooth interpolation [3, 4] is to minimize the L^2 norm of the interpolating function and that of its chosen (possibly all) derivatives. This is a variational problem with constraints represented by the interpolation conditions. An example of a smooth interpolation is the well-known spline interpolation.

We are mostly interested in the case of a single independent variable in the contribution. We generalize the approach of [4], and introduce the problem to be solved and the tools necessary to this aim in Sec. 2. We also quote the general existence theorem for smooth interpolation [3]. We are concerned with the use of basis system $\exp(-\mathrm{i}kx)$ of exponential functions of pure imaginary argument for 1D, 2D, and 3D smooth approximation problems in Sec. 3. In the conclusion, we show and discuss results of numerical experiments to compare the classical interpolation formulae and various kinds of the smooth approximation.

2. Problem of interpolation. Smooth interpolation

Let us have a finite number N of (complex, in general) measured (sampled) values $f_1, f_2, \ldots, f_N \in C$ obtained at N mutually distinct nodes $X_1, X_2, \ldots, X_N \in R^n$. Assume that $f_j = f(X_j)$ are measured values of some continuous function f. The dimension n of the independent variable may be arbitrary. For the sake of simplicity we put $n = 1$ except for Sec. 3 and assume that $X_1, X_2, \ldots, X_N \in \Omega$, where either $\Omega = [a, b]$ is a finite interval or $\Omega = (-\infty, \infty)$.

The *problem of interpolation* is construction of the interpolating function z fulfilling the interpolation conditions

$$z(X_j) = f(X_j), \quad j = 1, \ldots, N. \tag{1}$$

The problem of data interpolation does not have a unique solution. The property (1) of the interpolating function is uniquely formulated by mathematical means but there are also additional conditions on the *subjective perception* of the behavior of the interpolating curve between nodes that can hardly be formalized.

An inner product space is introduced to take into account the additional conditions in the problem of smooth interpolation [3], [4]. Let $\{B_l\}_{l=0}^{\infty}$ be a sequence of nonnegative numbers and let L be the smallest nonnegative integer such that $B_L > 0$ while $B_l = 0$ for $l < L$. Let $\widetilde{\mathcal{W}}$ be a linear vector space of complex functions g continuous together with their derivatives of all orders on the interval Ω.

Put

$$(g, h)_L = \sum_{l=0}^{\infty} B_l \int_{\Omega} g^{(l)}(x)[h^{(l)}(x)]^* \,\mathrm{d}x, \qquad |g|_L^2 = \sum_{l=0}^{\infty} B_l \int_{\Omega} |g^{(l)}(x)|^2 \,\mathrm{d}x, \tag{2}$$

where * denotes the complex conjugate.

If $L = 0$ (i.e. $B_0 > 0$), $g \in \widetilde{\mathcal{W}}$, and the value of $|g|_0$ exists and is finite, then $(g, h)_0 = (g, h)$ has the properties of *inner product* and the expression $|g|_0 = \|g\|$ is *norm* in the normed space W_0.

If $L > 0$ let $P_{L-1} \subset \widetilde{\mathcal{W}}$ be the subspace whose basis $\{\varphi_p\}$ consists of monomials $\varphi_p(x) = x^{p-1}$, $p = 1, \ldots, L$, and $(\varphi_p, \varphi_q)_L = 0$ for $p \neq q$. Using (2), we construct the *quotient space* $\widetilde{\mathcal{W}}/P_{L-1}$ whose zero class is the subspace P_{L-1}. We see that then $(\cdot, \cdot)_L$ and $|\cdot|_L$ represent the inner product and norm in the normed space $W_L = \widetilde{\mathcal{W}}/P_{L-1}$.

For an arbitrary $L \geq 0$, choose a *basis system* of functions $\{g_k\} \subset W_L$, $k = 1, 2, \ldots$, that is complete and orthogonal (in the inner product of W_L), $(g_k, g_m)_L = 0$ for $k \neq m$, $(g_k, g_k)_L = |g_k|_L^2 > 0$. If $L > 0$ then it is, moreover, $(\varphi_p, g_k)_L = 0$ for $p = 1, \ldots, L$, $k = 1, 2, \ldots$. The set $\{\varphi_p\}$ is empty for $L = 0$.

The *problem of smooth interpolation* consists in finding the coefficients A_k and a_p of the expression $z(x) = \sum_{k=1}^{\infty} A_k g_k(x) + \sum_{p=1}^{L} a_p \varphi_p(x)$ such that (1) holds and the quantity $|z|_L^2$ attains its minimum.

Let the sum $R_L(x,y) = \sum_{k=1}^{\infty} g_k(x) g_k^*(y) |g_k|_L^{-2}$, called the *generating function*, converges for all $x, y \in \Omega$. Theorem 1 of [3] states how to obtain the smooth interpolant z in the form

$$z(x) = \sum_{j=1}^{N} \lambda_j R_L(x, X_j) + \sum_{p=1}^{L} a_p \varphi_p(x), \tag{3}$$

where the coefficients λ_j, $j = 1, \ldots, N$, and a_p, $p = 1, \ldots, L$, are the unique solution of a nonsingular system of $N + L$ linear algebraic equations.

3. A choice of basis function system

Recall that we have put $n = 1$. Let the function f to be approximated be periodic in $[0, 2\pi]$. We choose periodic exponential functions of pure imaginary argument for the basis system $\{g_k\}$. The following theorem shows important properties of the system.

Theorem 1. *Let there be an integer $s \geq L$ such that $B_l = 0$ for all $l > s$ in W_L. The system of periodic exponential functions of pure imaginary argument*

$$g_k(x) = \exp(-\mathrm{i}kx), \quad x \in [0, 2\pi], \ k = \ldots, -2, -1, 0, 1, 2, \ldots, \tag{4}$$

is then complete and orthogonal in W_L.

Proof. The orthogonality and completeness of the system $\{g_k\}$ in $H^s(0, 2\pi)$ is proven, e.g., in [1]. The proof for the space W_L is based on the equivalence of norms. □

The range of k implies a minor change in the notation introduced above. For the basis system (4), notice that

$$R_L(x,y) = \sum_{k=-\infty}^{\infty} \frac{g_k(x) g_k^*(y)}{|g_k|_L^2} = \sum_{k=-\infty}^{\infty} \frac{\exp(-\mathrm{i}k(x-y))}{|g_k|_L^2} \tag{5}$$

is the Fourier series in $L^2(0, 2\pi)$ with the coefficients $|g_k|_L^{-2}$, $|g_k|_L^2 = 2\pi \sum_{l=L}^{\infty} B_l k^{2l}$.

Let now the function f to be approximated be nonperiodic on $(-\infty, \infty)$ and $f^{(l)}(\pm\infty) = 0$ for all $l \geq 0$. Let us define the generating function $R_L(x,y)$ as the Fourier transform of the function $|g_k|_L^{-2}$ of a continuous variable k,

$$R_L(x,y) = \int_{-\infty}^{\infty} \frac{\exp(-\mathrm{i}k(x-y))}{|g_k|_L^2} \, \mathrm{d}k, \tag{6}$$

if the integral exists. Using the effect of transition from the Fourier series (5) to the Fourier transform (6), we have transformed the basis functions, enriched their spectrum, and released the requirement of periodicity of f. Moreover, if the integral (6) does not exist, in many instances we can calculate $R_L(x,y)$ as the Fourier transform $\mathcal{F}$ of the generalized function $|g_k|_L^{-2}$ of k.

Choosing now a particular sequence $\{B_l\}$, we complete the definition of the inner product and norm (2) in a particular W_L. Let us thus put $B_l = 0$ for all l with the exception of $B_2 = 1$ (cf. [4]). It means that we have $L = 2$ and minimize the usual L^2 norm of the second derivative of the interpolant (3), $z(x) = \sum_{j=1}^{N} \lambda_j R_2(x, X_j) + a_0 + a_1 x$. We have $|g_k|_2^2 = 2\pi k^4$ and putting $r = |x - y|$, we arrive at

$$R_2(x, y) = \mathcal{F}(1/(2\pi k^4)) = \tfrac{1}{12} r^3, \tag{7}$$

where $\mathcal{F}$ denotes the integral Fourier transform or the Fourier transform of a generalized function [2]. It is easy to find out that this version of smooth approximation is, in fact, the well-known *cubic spline interpolation*.

There are further practical examples of smooth approximation where the integral generating function R_L can be calculated with the help of the Fourier transform.

We can generalize the smooth interpolation procedure of Sec. 2 to R^n, n being a positive integer. We do not introduce the notation in R^n in detail but will you keep in mind that all the derivatives are partial now. We choose the system of periodic exponential functions $g_k(x) = \exp(-\mathrm{i}k \cdot x)$ of pure imaginary vector argument, which can be proven to be complete and orthogonal in W_L, and put r equal to the Euclidean norm of $x - y$.

Let $n = 2$. In the definition of inner product in W_L, we put $L = 2$ and construct analog of a spline in two dimensions. The interpolant has the form $z(x) = \sum_{j=1}^{N} \lambda_j R_2(x, X_j) + a_0 + a_1 x_1 + a_2 x_2$ and it is $|g_k|_2^2 = 2\pi(k_1^2 + k_2^2)^2$. We arrive at $R_2(x, y) = \mathcal{F}(1/(2\pi(k_1^2 + k_2^2)^2)) = C_2 r^2 \ln r + C_2' r^2$, where C_2, C_2' are constants [2].

Let $n = 3$. With the same choice $L = 2$ we construct analog of a spline in three dimensions. The interpolant has the form $z(x) = \sum_{j=1}^{N} \lambda_j R_2(x, X_j) + a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3$ and it is $|g_k|_2^2 = 2\pi(k_1^2 + k_2^2 + k_3^2)^2$. We have $R_2(x, y) = \mathcal{F}(1/(2\pi(k_1^2 + k_2^2 + k_3^2)^2)) = C_3 r$, where C_3 is a constant [2].

For $n = 1$, we will also consider another interesting choice of $\{B_l\}$ with the system (4). Putting $L = 0$, $r = |x - y|$ and, in particular, $B_l = D^{2l}/(2l)!$, $D = \frac{1}{3}$, we calculate [4]

$$R_0(x, y) = \frac{1}{2D \cosh(\pi r/(2D))}. \tag{8}$$

4. Computational comparison

To present results of numerical experiments we use two complete and orthogonal systems $\{g_k\}$ in W_L. We assume that the function to be interpolated is not periodic.

(i) Exponential functions of pure imaginary argument (4) {dashed line} with the generating function (7), $L = 2$, $B_2 = 1$, i.e. cubic spline interpolation.

(ii) The same functions (4) {dashed line} with the generating function (8).

(iii) Orthonormalized monomials {dotted line}. The system of monomials $h_k(x) = x^k$, $k = 0, 1, 2, \ldots$, is orthonormalized numerically on $(-1, 1)$ by the Gram-Schmidt procedure with respect to the inner product $(g, h)_0$. We use $L = 0$ and B_l the same as in (ii). $R_0(x, y)$ is evaluated numerically.

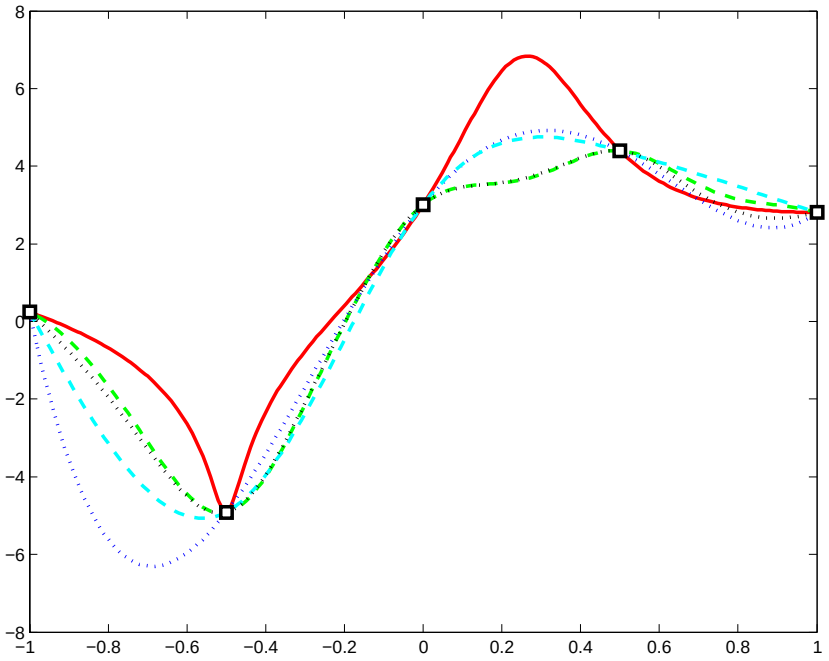


Figure 1: $N = 5$, "pole" $x = 0.25$ is not an interpolation node. Curves at $x = 0.80$ from top to bottom: (i), (ii), true, (iii), (iv)

Next two interpolation methods are classical.

(iv) Polynomial interpolation {dotted line}.

(v) Rational interpolation {dash-dot line}.

The interpolated function

$$f(x) = \ln(\tfrac{1}{100}(x + \tfrac{1}{2})^2 + 10^{-5}) + \frac{6}{1 + 16(x - \tfrac{1}{4})^2} + 6 \tag{9}$$

has "almost a singularity" at $x = -\frac{1}{2}$ and "almost a pole" at $x = \frac{1}{4}$. The smooth as well as classical interpolation of the function (9) has been constructed in several equidistant grids of N nodes on $[-1, 1]$. Some very inaccurate results (obtained e.g. by the polynomial interpolation of high degree) are omitted in some of the following graphs. In the figures, the solid line represents the true solution, i.e. the function (9). The results of interpolation are in Fig. 1 and 2. They show some qualitative behavior of the results but the quantitative properties can hardly be seen.

5. Conclusion

Since the extent of this contribution is limited we presented only a single example. It would not be fair to draw principal conclusions from it. The computation shows that the smooth interpolation is a competitive method. The L_∞ error of all the methods used, except for error of the polynomial interpolation, decreases as N

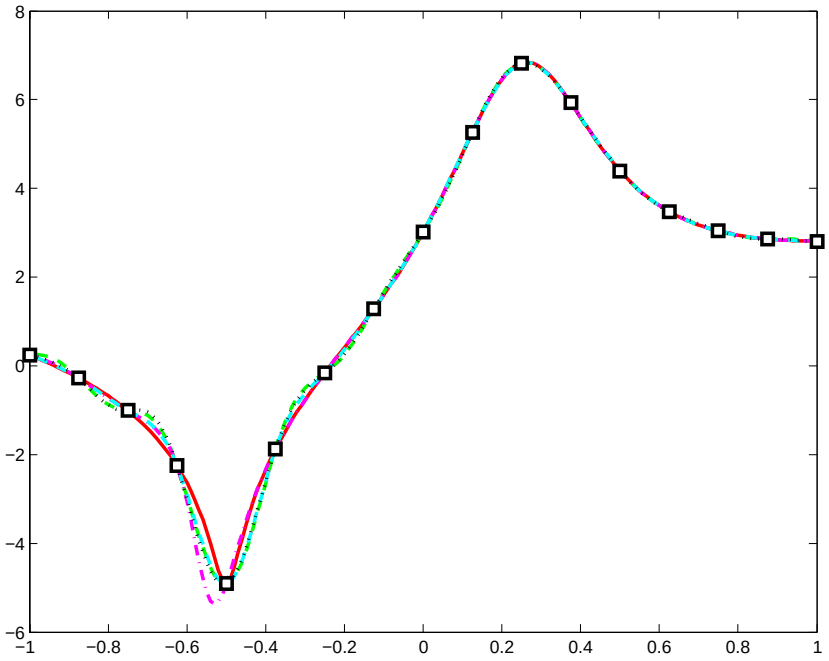


Figure 2: $N = 17$. Curves at $x = -0.55$ from top to bottom: true, (i) identical to (ii) and (iii), (v)

increases. Nevertheless, we should keep in mind that the only ultimate interpolation conditions are the values at nodes.

The case of $n > 1$ is much more interesting and makes many important applications possible. The interpolation nodes can be arbitrarily placed in the plane or space and large sets of data measured can be handled. There are also several further choices of the sequence $\{B_l\}$ that lead to a smooth approximating function possessing some "physical properties" like the cubic spline.

Acknowledgement

This work was supported by project RVO 67985840 and by grant 101/14-02067S of the Czech Science Foundation.

References

[1] Gilliam, D.: Introduction to Sobolev spaces on the circle. 2006. Available from `texas.math.ttu.edu/~gilliam/f06/m5340_f06/sobolev_sp_circle.pdf`

[2] Kreĭn, S. G. (Ed.): *Functional analysis.* (Russian.) Nauka, Moskva, 1964.

[3] Segeth, K.: Some computational aspects of smooth approximation. Computing **95** (2013), S695–S708.

[4] Talmi, A. and Gilat, G.: Method for smooth approximation of data. J. Comput. Phys. **23** (1977), 93–123.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

SOLUTION OF MECHANICAL PROBLEMS IN FRACTURED ROCK WITH THE USER-DEFINED INTERFACE OF COMSOL MULTIPHYSICS

Ilona Škarydová[1], Milan Hokr[2]

[1] Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
ilona.skarydova@tul.cz
[2] Institute for Nanomaterials, Advanced Technologies and Innovation,
Technical University of Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
milan.hokr@tul.cz

Abstract

This paper presents the main concept and several key features of the user-defined interface of COMSOL Java API for the solution of mechanical problems in fractured rock. This commercial computational system based on FEM has yet to incorporate fractures in mechanical problems.

Our aim is to solve a 2D mechanical problem with a fracture which is defined separately from finite-element discretization and the fracture properties are included through the constitutive laws. This will be performed based on the principles of the extended finite element method as a way of fracture description, enrichment functions for rock elements containing fractures, etc.

We present an approach to describing a simple mechanical problem in COMSOL Java API together with a proposal of a solution method, and we also demonstrate the potential of COMSOL Java API for solving more complicated problems with fractures.

1. Introduction

For many applications it is important to evaluate the effects of fractures in rock on its mechanical properties and the effects of stress on fracture opening/closure. An example which is also the context of our work is the concept of the geological disposal of spent nuclear fuel, with three protective barriers (copper/steel containers, bentonite, and a stable rock massif). Safety analysis requires an understanding and prediction of the complex thermo-hydro-mechanical-chemical processes and the current engineering software can sometimes lack the required detail.

Despite the fact that a number of methods for solving problems of fractured rock mechanics (DEM [2], XFEM [7], FEM with special fracture elements [5]) are implemented in various software packages, most are too specialized and have complicated coupling with other important processes.

In this paper we present a methodology for solving specialized problems using a commercially available code programming interface, combining the advantages of established supported code with the freedom of an open-source project. The use of Java API in COMSOL Multiphysics software is presented here. Firstly, we describe the implementation process of a simple plane strain problem, then we highlight several advanced functions and other required features (constitutive laws, XFEM principles) of COMSOL Java API.

2. Description of a plane strain problem

The approach will be described using a basic 2D plane strain problem which is represented by Hooke's law

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}, \tag{1}$$

where σ_x, σ_y and τ_{xy} are normal and shear components of the stress tensor, E is Young's modulus, ν is Poisson's ratio, ε_x, ε_y and γ_{xy} are normal and shear components of the strain tensor and the equilibrium equations are written as

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + k_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + k_y &= 0, \end{aligned} \tag{2}$$

where k_x and k_y are components of a force vector.

The dependence of the strain tensor components on components of the displacement vector $\mathbf{u} = [u, v]^T$ for a small deformation is expressed by

$$\varepsilon_x = \frac{\partial u}{\partial x} \qquad \varepsilon_y = \frac{\partial v}{\partial y} \qquad \varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) = \frac{1}{2}\gamma_{xy}. \tag{3}$$

If we substitute Hooke's law (1) together with the strain-displacement relation (3) into the equilibrium equations (2), we get the form

$$\begin{aligned} \frac{\partial}{\partial x}\left[\left(\lambda + 2\mu\right)\frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y}\right] + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + k_x &= 0 \\ \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \frac{\partial}{\partial y}\left[\lambda \frac{\partial u}{\partial x} + \left(\lambda + 2\mu\right) + \frac{\partial v}{\partial y}\right] + k_y &= 0, \end{aligned} \tag{4}$$

where λ and μ are so-called Lamé coefficients, which can be derived from Young's modulus and Poisson's ratio. This form can be used for specific expression of solved problem in a "General form PDE" in COMSOL Multiphysics.

3. Computational tool

The computational system COMSOL Multiphysics based on FEM was chosen as an appropriate computational tool. Despite the fact that features for calculating with fractures in mechanical problems are not included in it, COMSOL provides sufficient variability for their implementation. In addition it has proven to be a suitable tool for solving coupled processes.

We use a special interface of COMSOL Multiphysics called the Java application programming interface (COMSOL Java API, [1]), which provides access to special extended features and functions that are not available from commonly used graphical user interface (GUI).

The basic model is possible to define and export from GUI (model.java) or directly define in the integrated development environment Eclipse, [3] using the appropriate commands. Then the model in Java code is processed in Eclipse. This environment also allows new results to be directly exported in different formats (.jpg, .png, .txt, .cls) or the access and connection with the GUI of COMSOL Multiphysics. A diagram of the approach to the solution using COMSOL is summarized in Figure 1.

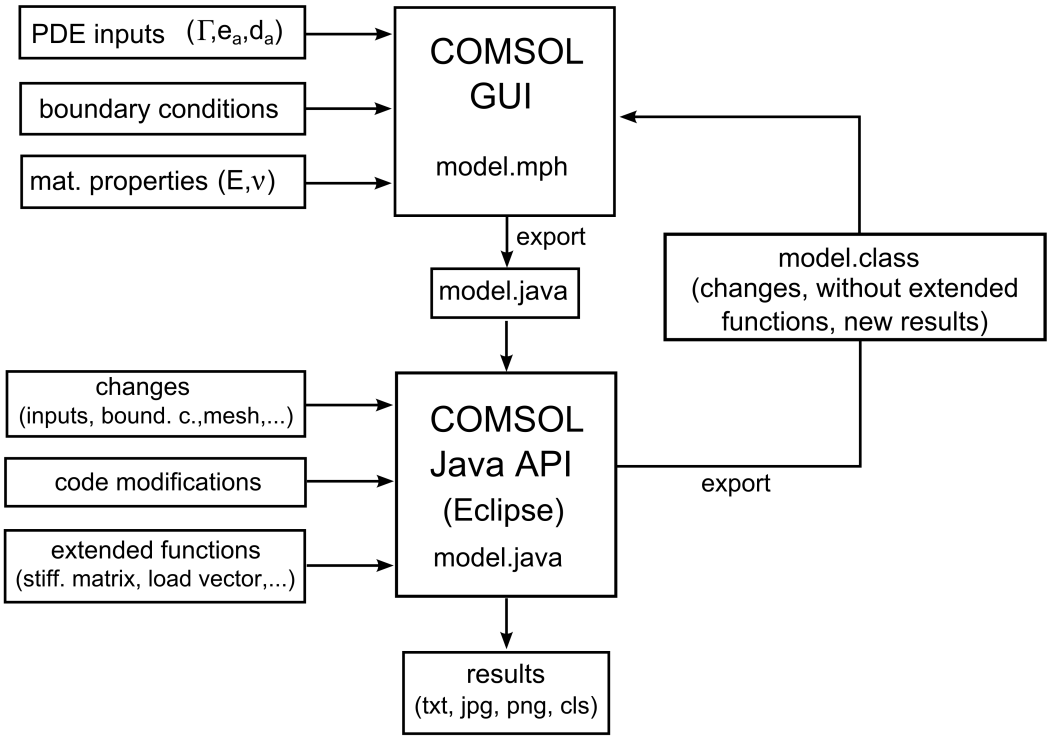


Figure 1: Approach of the solution using COMSOL Java API, solved using a user-defined PDE with Hooke's law (1) as a constitutive relation

The built-in physical interface which describes the physical process using a partial differential equation is replaced by a "Mathematical module" with a "General form PDE" interface. Within this module it is possible to apply a user-defined partial differential equation through the individual coefficients. The form of the partial differential equation is specified as

$$e_a \frac{\partial^2 \mathbf{u}}{\partial t^2} + d_a \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \Gamma = f \qquad \mathbf{u} = [u, v]^T, \tag{5}$$

where $\mathbf{u}$ is a displacement vector, e_a is a mass coefficient, d_a is a damping coefficient and matrix Γ can be expressed by

$$\begin{aligned} \Gamma_{1x} = (\lambda + 2\mu)\frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y}, \qquad \Gamma_{1y} = \mu\left(\frac{\partial u}{\partial y} + \lambda \frac{\partial v}{\partial x}\right), \\ \Gamma_{2x} = \Gamma_{1y} \qquad \Gamma_{2y} = \lambda \frac{\partial u}{\partial x} + (\lambda + 2\mu)\frac{\partial v}{\partial y}\Big), \end{aligned} \tag{6}$$

which is taken from (4). For a steady-state problem the last divergence term on the left side of the equation (5) and $e_a = 0, d_a = 0$ are considered.

The results of the model can be displayed using exported model.class in the GUI of COMSOL Multiphysics or in a simply created graphical interface in COMSOL Java API. Unfortunately, exported model with Java modifications is unable to store results for recomputed solution in the GUI (extended functions are not included in GUI).

4. Perspectives for fracture mechanics implementation

The previous section described the elementary plane strain problem and how to define and solve it in COMSOL Java API with a user-defined PDE. The following section specifies other important aspects and features (i.e. fractures with a predetermined fixed position and their influence on the elastic properties of rock, constitutive laws, special functions of COMSOL Java API, use of certain XFEM principles) which are necessary for defining the problem with the fractures.

4.1. Constitutive laws

Constitutive laws are special empirical or theoretical formulas which express the behaviour of a material under a general load. They are important for describing the rock-matrix behaviour and also for expressing the influence of a fracture on rock mass properties. A large number of constitutive laws are referred to in [6]: different relations are proposed for rock and fractures. Constitutive laws are often described in terms of "strength-form" (they describe the limit stress when the failure occurs) but for our purpose the stress-strain relation is more suitable.

The simplest and the most popular formula for expressing elastic rock behaviour is the above-mentioned Hooke's law (1) with two independent material parameters

for an isotropic material. The elastic/plastic behaviour of the rock can be described using the Mohr-Coulomb or Hoek-Brown constitutive laws, which assume that the failure occurs under the highest shear stress.

The behaviour of the fractures can be described for example by empirical criteria (e.g. the Mohr-Coulomb criterion, Goodman's law, the Barton-Bandis criterion) or theoretical models (the Amadei-Saeb or Plesha model). For example, the Barton-Bandis model is an empirical constitutive model requiring JRC (joint roughness coefficient) and JCS (joint wall compressive strength) parameters.

4.2. Fractures and their representation

Fractures and their representation in the model are the next important part of the implementation. COMSOL Multiphysics does not have any built-in approach to solving mechanical problems with fractures. Thus, fractures have to be controlled externally in Java code.

Fractures are represented by lines in the 2D case and are defined separately (they are not dependent on the computational mesh). A similar approach is used in the XFEM family of methods [4]. Many of them use the so-called level-set method [8], which does not require curve parameterization and is also more suitable for problems with moving interfaces or growing fractures.

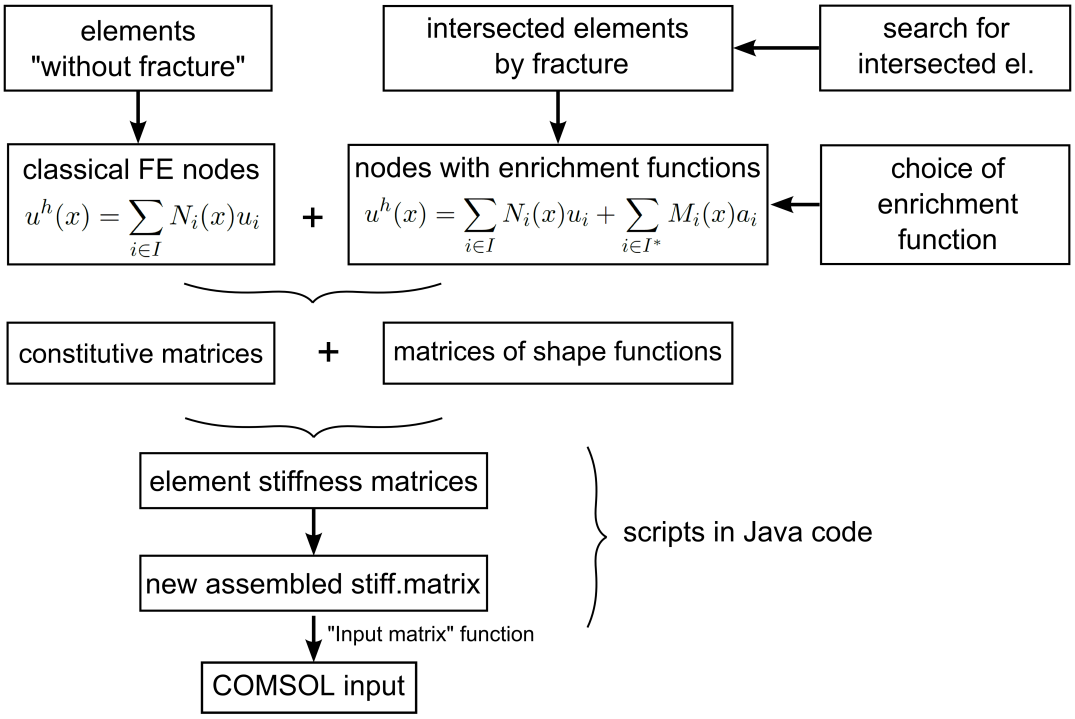


Figure 2: XFEM solution for fractures in COMSOL Java API

4.3. Implementation and related problems

We can then apply several principles of the Extended Finite Element method ([7, 4]) to the problem with fractures. XFEM is a numerical method based on a generalization of FEM and it enables a local enrichment of approximation spaces (hence discontinuous behaviour in a small part of the domain). In this way it is possible to define both weak and strong discontinuities (discontinuity in the stress and strain field or discontinuity in the displacement field, respectively).

For the implementation of XFEM it is necessary to use special functions available in COMSOL Java API (see Figure 2). One of these functions is the "Input matrix", which enables the input of an externally assembled matrix or vector (stiffness, mass or damping matrix and load vector) in a sparse form. One disadvantage is that the matrices are not stored in the model, which has to be processed externally.

The next problem, which is necessary to deal with it, is the detection of intersection elements (elements of the rock matrix which are intersected by the fracture). This feature can be solved by code in COMSOL Java API using the known position of the fracture and the positions of the individual elements from the exported mesh file.

5. Conclusions

We have shown that the COMSOL Java API can be conveniently used for testing or applying new modelling concepts and numerical schemes, which can be of interest to the wider community.

The example of including fractures to the elasticity problem has been presented on a conceptual level, with its implementation planned for future work.

Acknowledgements

This work was supported by the project LO1201 through the financial support of the Ministry of Education, Youth and Sports in the framework of the targeted support of the National Programme for Sustainability I, by the Ministry of Education of the Czech Republic within the SGS project no. 21066/115 on the Technical University of Liberec and by the Ministry of Industry and Trade of the Czech Republic within the project FR TI3/579.

References

[1] COMSOL AB, Stockholm, Sweden: *Comsol Java API reference guide*, 1^{st} ed.,2012. Version 4.3a.

[2] Cundall, P. A. and Strack, O. D. L.: A discrete numerical model for granular assemblies. Géotechnique **29** (1979), 47–65.

[3] Eclipse: integrated development environment. `<https://www.eclipse.org/>`. Accessed: 2014-09-10.

[4] Fries, T. P. and Belytschko, T.: The extended/generalized finite element method: An overview of the method and its applications. Internat. J. Numer. Methods Engrg. **84.3** (2010), 253–304.

[5] Goodman, R. E.: *Methods of geological engineering in discontinuous rocks*. West Publishing Company, San Francisco, CA, 1976.

[6] Jing, L. and Stephansson, O.: *Fundamentals of discrete element methods for rock engineering: theory and applications*. Elsevier, Amsterdam, 2007.

[7] Moës, N., Dolbow, J., and Belytschko, T.: A finite element method for crack growth without remeshing. Internat. J. Numer. Methods Engrg. **46** (1999), 131–150.

[8] Osher, S. J. and Fedkiw, R. P.: *Level set methods and dynamic implicit surfaces*. Springer-Verlag, New York, 2002.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

NUMERICAL SIMULATION OF FREE-SURFACE FLOWS WITH SURFACE TENSION

Petr Sváček

Czech Technical University, Faculty of Mechanical Engineering
Dep. of Technical Mathematics, Karlovo nám. 13, Praha 2, Czech Republic
Petr.Svacek@fs.cvut.cz

Abstract

This paper focuses on the mathematical modelling and the numerical approximation of the flow of two immiscible incompressible fluids. The surface tension effects are taken into account and mixed boundary conditions are used. The weak formulation is introduced, discretized in time, and the finite element method is applied. The free surface motion is treated with the aid of the level set method. The numerical results are shown.

1. Introduction

The mathematical modelling of two-phase flows with the consideration of the free surface motion influenced by the surface tension is addressed in various scientific as well as technical applications. Such a problem is important both from the mathematical modelling point of view and also from the technical practice. Particularly, its numerical approximation is very challenging task, see among others [1], [2] or [3]. The approximation of the surface tension naturally can play a key role here.

In this paper, we consider the two-dimensional flow of two immiscible fluids, the problem is mathematically described and the variational formulation is introduced. For the discretization the finite element(FE) method is used. The free surface motion is realized using the level set method, cf. [7] or [5]. In the case of high surface tension, a modification of the standard FE method is required to avoid the spurious currents, see [6] or [1]. For the verification of the implemented method a benchmark problem is solved, cf. [3].

2. Mathematical description

Let us consider the computational domain $\Omega \subset \mathbb{R}^2$ with the Lipschitz continuous boundary $\partial\Omega$ with its mutually disjoint parts Γ_W, Γ_S, Γ_O. The domain is occupied at time t by two immiscible fluids, i.e. $\Omega = \Omega^A_{(t)} \cup \Omega^B_{(t)}$, the fluid A occupies $\Omega^A_{(t)}$ and the fluid B occupies $\Omega^B_{(t)}$, see Fig. 1. The interface between the two fluids is denoted

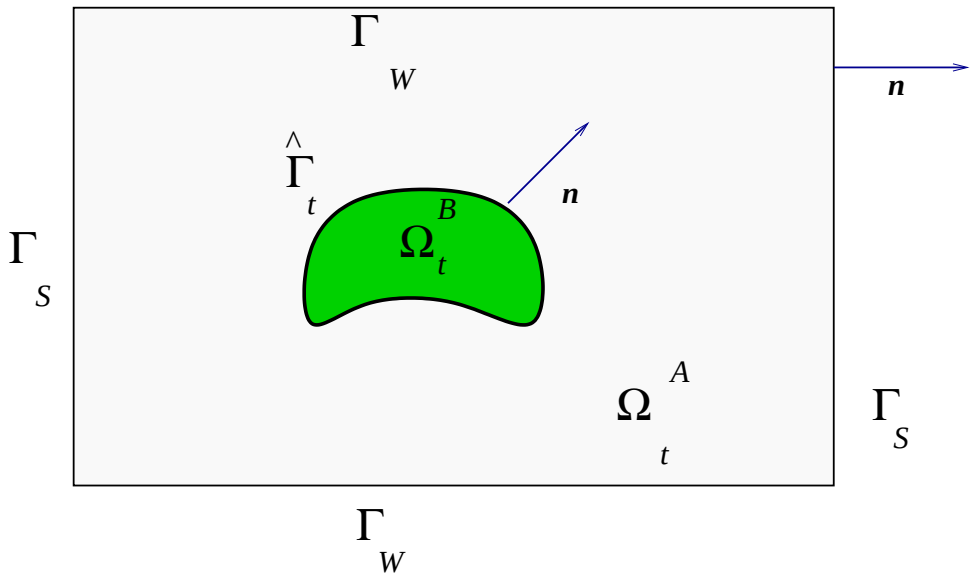


Figure 1: The computational domain Ω, its sub-domains $\Omega^A_{(t)}$ and $\Omega^B_{(t)}$, the interface $\hat{\Gamma}_t$ and the normal vector.

by $\hat{\Gamma}_t = \partial\Omega^A_{(t)} \cap \partial\Omega^B_{(t)}$. Further, we denote by $\Gamma^k_{W,t} = \Gamma_W \cap \partial\Omega^k_{(t)}$, $\Gamma^k_{S,t} = \Gamma_S \cap \partial\Omega^k_{(t)}$ and $\Gamma^k_{O,t} = \Gamma_O \cap \partial\Omega^k_{(t)}$ for $k = A$ or $k = B$.

The flow of the fluid A in the domain $\Omega^A_{(t)}$ is described by the incompressible system of Navier-Stokes equations

$$\frac{\partial\left(\rho^A \boldsymbol{u}^A\right)}{\partial t} + \rho^A(\boldsymbol{u}^A \cdot \nabla)\boldsymbol{u}^A - \nabla \cdot \boldsymbol{\sigma}^A = \rho^A \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u}^A = 0, \tag{1}$$

where ρ^A denotes the constant fluid A density, $\boldsymbol{u}^A = \boldsymbol{u}^A(x,t)$ is its flow velocity defined for $x \in \Omega^A_{(t)}$ and $t \in [0,T)$, and $\boldsymbol{\sigma}^A$ is the Cauchy stress tensor given by $\boldsymbol{\sigma}^A = -p^A I + \mu^A(\nabla \boldsymbol{u}^A + \nabla^T \boldsymbol{u}^A)$, where $p^A = p^A(x,t)$ is the pressure and μ^A is the viscosity coefficient. Similarly, the flow of the fluid B in the domain $\Omega^B_{(t)}$ is governed by

$$\frac{\partial\left(\rho^B \boldsymbol{u}^B\right)}{\partial t} + \rho^B(\boldsymbol{u}^B \cdot \nabla)\boldsymbol{u}^B - \nabla \cdot \boldsymbol{\sigma}^B = \rho^B \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u}^B = 0, \tag{2}$$

where ρ^B denotes the constant fluid B density, $\boldsymbol{u}^B = \boldsymbol{u}^B(x,t)$ is its flow velocity defined for $x \in \Omega^B_{(t)}$ and $t \in [0,T)$, and $\boldsymbol{\sigma}^B$ is the Cauchy stress tensor given by $\boldsymbol{\sigma}^B = -p^B I + \mu^B(\nabla \boldsymbol{u}^B + \nabla^T \boldsymbol{u}^B)$, where $p^B = p^B(x,t)$ is the pressure and μ^B is the viscosity coefficient. In eqs. (1-2) $\boldsymbol{f}$ denotes the gravitational acceleration (acting in the negative x_2 direction).

The motion of both fluids is then driven by the continuity equation

$$\frac{\partial \rho}{\partial t} + (\boldsymbol{u} \cdot \nabla)\rho = 0. \tag{3}$$

The domains $\Omega^A_{(t)}$ and $\Omega^B_{(t)}$ are then implicitly determined by the equations $\rho = \rho^A$ and $\rho = \rho^B$, respectively.

The initial conditions at time $t = 0$ are given $\boldsymbol{u}^A(x,0) = 0$, $\rho(x,0) = \rho^A$ for $x \in \Omega^A_{(0)}$ and $\boldsymbol{u}^B(x,0) = 0$, $\rho(x,0) = \rho^B$ for $x \in \Omega^B_{(0)}$. On the interface the following boundary conditions are specified on $\hat{\Gamma}_t$

$$a) \qquad \boldsymbol{u}^A = \boldsymbol{u}^B, \qquad b) \qquad \sigma^A \cdot \boldsymbol{n} - \sigma^B \cdot \boldsymbol{n} = \gamma\kappa\boldsymbol{n}, \tag{4}$$

where γ is the surface tension coefficient, κ denotes the curvature of the interface Γ_I and $\boldsymbol{n}$ here denotes the normal to the Γ_I pointing into $\Omega^B_{(t)}$. On the boundary $\partial\Omega$ the following boundary conditions are prescribed

$$\begin{array}{lllll}
a) & \boldsymbol{u}^A = 0 & \text{on } \Gamma^A_{W,t}, & \boldsymbol{u}^B = 0 & \text{on } \Gamma^B_{W,t}, \\
b) & \boldsymbol{u}^A \cdot \boldsymbol{n} = 0, \frac{\partial(\boldsymbol{u}^A \cdot \boldsymbol{t})}{\partial \boldsymbol{n}} = 0 & \text{on } \Gamma^A_{S,t}, & \boldsymbol{u}^B \cdot \boldsymbol{n} = 0, \frac{\partial(\boldsymbol{u}^B \cdot \boldsymbol{t})}{\partial \boldsymbol{n}} = 0 & \text{on } \Gamma^B_{S,t}, \\
c) & \boldsymbol{\sigma}^A \cdot \boldsymbol{n} = 0 & \text{on } \Gamma^A_{O,t}, & \boldsymbol{\sigma}^B \cdot \boldsymbol{n} = 0 & \text{on } \Gamma^B_{O,t},
\end{array} \tag{5}$$

where $\boldsymbol{n}$ denotes the unit outward normal to the boundary of Ω, and $\boldsymbol{t}$ is the unit tangent vector to the boundary of Ω.

3. Variational formulation

In order to introduce the weak formulation, we start with the definition of the function space $Q = L^2(\Omega)$ for the pressure and $\boldsymbol{V}$ the function space for the velocity, where $\boldsymbol{V} = \{\boldsymbol{v} \in \boldsymbol{H}^1(\Omega) : \boldsymbol{v} = 0 \text{ on } \Gamma_W, \boldsymbol{v} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_S\}$. Now, let us take the test function $\boldsymbol{v} \in \boldsymbol{V}$ and multiply the first equations in (1-2) by $\boldsymbol{v}$, integrate over Ω, use Green's theorem, apply the boundary conditions (5b-c) and use the interface condition (4b). We get

$$\begin{aligned}
&\int_{\Omega^A_{(t)}} \rho^A \left(\frac{\partial \boldsymbol{u}^A}{\partial t} + (\boldsymbol{u}^A \cdot \nabla)\boldsymbol{u}^A \right) \cdot \boldsymbol{v} + \boldsymbol{\sigma}^A \cdot (\nabla \boldsymbol{v}) \,\mathrm{d}x - \int_{\Omega^A_{(t)}} \rho^A \boldsymbol{f} \cdot \boldsymbol{v} \,\mathrm{d}x \quad + \\
&\int_{\Omega^B_{(t)}} \rho^B \left(\frac{\partial \boldsymbol{u}^B}{\partial t} + (\boldsymbol{u}^B \cdot \nabla)\boldsymbol{u}^B \right) \cdot \boldsymbol{v} + \boldsymbol{\sigma}^B \cdot (\nabla \boldsymbol{v}) \,\mathrm{d}x - \int_{\Omega^B_{(t)}} \rho^B \boldsymbol{f} \cdot \boldsymbol{v} \,\mathrm{d}x \quad = \quad \int_{\hat{\Gamma}_t} \gamma\kappa\boldsymbol{n} \cdot \boldsymbol{v} \,\mathrm{d}S.
\end{aligned} \tag{6}$$

Formulation (6) can be written in a more compact form using the Heaviside function $H(x,t)$ defined as $H(x,t) = 1$ for $x \in \Omega^A_{(t)}$, $H(x,t) = 0$ for $x \in \Omega^B_{(t)} \cup \hat{\Gamma}_t$. The density and the viscosity functions then are defined by $\rho(x,t) = \rho^A H(x,t) + (1 - H(x,t))\rho^B$ and $\mu(x,t) = \mu^A H(x,t) + (1 - H(x,t))\mu^B$, respectively. Further, the functions $\boldsymbol{u} = \boldsymbol{u}(x,t)$ and $p = p(x,t)$ can be defined by

$$\boldsymbol{u}(x,t) = \begin{cases} \boldsymbol{u}^A(x,t) & \text{for } x \in \overline{\Omega^A_{(t)}}, \\ \boldsymbol{u}^B(x,t) & \text{for } x \in \overline{\Omega^B_{(t)}}, \end{cases} \qquad p(x,t) = \begin{cases} p^A(x,t) & \text{for } x \in \overline{\Omega^A_{(t)}} \setminus \hat{\Gamma}_t, \\ p^B(x,t) & \text{for } x \in \overline{\Omega^B_{(t)}} \setminus \hat{\Gamma}_t. \end{cases}$$

Using this notation, the equation (6) then can be written as

$$\int_\Omega \rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) \cdot \boldsymbol{v} + \boldsymbol{\sigma} \cdot (\nabla \boldsymbol{v}) \,\mathrm{d}x = \int_{\hat{\Gamma}_t} \gamma\kappa\boldsymbol{n} \cdot \boldsymbol{v} \,\mathrm{d}S + \int_\Omega \rho \boldsymbol{f} \cdot \boldsymbol{v} \,\mathrm{d}x, \tag{7}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor given by $\boldsymbol{\sigma} = -pI + \mu(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u})$. Using the Dirac delta function $\delta_{\hat{\Gamma}_t}$ of the interface $\hat{\Gamma}_t$ the equation (7) can be written in the form

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma} = \rho \boldsymbol{f} + \gamma \kappa \boldsymbol{n} \delta_{\hat{\Gamma}_t}. \tag{8}$$

Surface tension. In order to treat the surface tension term, we start with its weak reformulation. Let us define the tangent derivative ∇_Γ as $\nabla_\Gamma g = \nabla g - (\boldsymbol{n} \cdot \nabla g)\boldsymbol{n}$ and the Laplace-Beltrami operator $\triangle_\Gamma = \nabla_\Gamma \cdot \nabla_\Gamma$. Now, using the relation $\kappa \boldsymbol{n} = \triangle_\Gamma \boldsymbol{x}$ and applying the integration by parts on $\hat{\Gamma}_t$ we get

$$\int_{\hat{\Gamma}_t} \gamma \kappa \boldsymbol{n} \cdot \boldsymbol{v} \, \mathrm{d}S = -\int_{\hat{\Gamma}_t} \gamma (\nabla_\Gamma \boldsymbol{x}) \cdot (\nabla_\Gamma \boldsymbol{v}) \, \mathrm{d}S, \tag{9}$$

where for the sake of simplicity it was assumed that $\hat{\Gamma}_t$ is a closed curve.

Level set equation. Furthermore, to treat the motion of the free surface $\hat{\Gamma}_t$ the *level set* method is applied. First, the initial condition for the level set function $\phi = \phi(x,t)$ is prescribed by $\phi(x,0) = \mathrm{dist}(x, \hat{\Gamma}_0) > 0$ for $x \in \Omega^A_{(0)}$, $\phi(x,0) = -\mathrm{dist}(x, \hat{\Gamma}_0) < 0$ for $x \in \Omega^B_{(0)}$, and $\phi(x,0) = 0$ for $x \in \hat{\Gamma}_0$. The motion of the interface $\hat{\Gamma}_t$ is then realized by forcing the function ϕ to solve the equation

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0, \tag{10}$$

which guarantees that the interface is moving with the velocity $\boldsymbol{u}$. Now, the Heaviside function $H(x,t)$ is defined using the sign of the level set function $\phi(x,t)$. Taking into account the level set equation (10) and the definition of the function $\rho(x,t)$, the continuity equation (3) is formally satisfied.

4. Numerical approximation

Flow step. For simplicity, let us consider the equidistant partition of the time interval $[0,T)$ given by $t_n = n\Delta t$, where $n = 0, 1, \ldots, N$ and $\Delta t = T/N$. Let us denote by $\boldsymbol{u}^{(n)}$, $p^{(n)}$, $\phi^{(n)}$, ρ^n and μ^n approximations of the velocity, the pressure the level set function, the density and the viscosity at the time instant t_n, respectively. Let us approximate the time derivative by the backward Euler formula, i.e.

$$\frac{\partial \boldsymbol{u}}{\partial t}|_{t=t_{n+1}} \approx \frac{\boldsymbol{u}^{(n+1)} - \boldsymbol{u}^{(n)}}{\Delta t}, \qquad \frac{\partial \phi}{\partial t}|_{t=t_{n+1}} \approx \frac{\phi^{(n+1)} - \phi^{(n)}}{\Delta t}.$$

Let us assume that $\boldsymbol{u}^{(n)}$, $p^{(n)}$, $\phi^{(n+1)}$, μ^{n+1} and $\rho^{(n+1)}$ are already known. Then the time discretized weak formulation of (8) reads: Find $\boldsymbol{u} = \boldsymbol{u}^{n+1} \in \boldsymbol{V}$ and $p = p^{n+1} \in Q$ such that

$$\begin{aligned} &\int_\Omega \rho^{n+1}(x) \left(\frac{\boldsymbol{u} - \boldsymbol{u}^n}{\Delta t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) \cdot \boldsymbol{v} - p(\nabla \cdot \boldsymbol{v}) + \mu^{n+1}(x) \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} \, \mathrm{d}x \\ &\quad + \int_\Omega (\nabla \cdot \boldsymbol{u}) q \, \mathrm{d}x = -\int_{\hat{\Gamma}^{n+1}} \gamma (\nabla_\Gamma \boldsymbol{x}) \cdot (\nabla_\Gamma \boldsymbol{v}) \, \mathrm{d}S + \int_\Omega \rho^{n+1}(x) \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}x \end{aligned} \tag{11}$$

holds for all $\boldsymbol{v} \in \boldsymbol{V}$ and $q \in Q$. In the practical computations we assume that the domain Ω is a polygonal and the spaces $\boldsymbol{V}$ and Q are approximated by the FE subspaces $\boldsymbol{V}_h$ and Q_h defined over an admissible triangulation $\mathcal{T}_h$, respectively. For the approximation the well-known Taylor-Hood FE are used, i.e. the velocity is sought in the space $\boldsymbol{V}_h = [H_h]^2 \subset \boldsymbol{V}$, where

$$H_h = \{\phi \in C(\overline{\Omega}); \phi|_K \in P_2(K) \text{ for each } K \in \mathcal{T}_h\}, \tag{12}$$

where $P_k(K)$ denotes the space of all polynomials on K of degree less or equal to k. Next, the pressure (as well as the level set function) is approximated in the space

$$Q_h = \left\{\phi \in C(\overline{\Omega}) : \ \phi|_K \in P_1(K) \text{ for each } K \in \mathcal{T}_h\right\}. \tag{13}$$

The discrete flow problem then reads: Find $\boldsymbol{u}_h = \boldsymbol{u}_h^{n+1} \in \boldsymbol{V}_h$ and $p_h = p_h^{n+1}$ such that equation (11) holds for any test function $\boldsymbol{v} := \boldsymbol{v}_h \in \boldsymbol{V}_h$ and $q := q_h \in Q_h$. In order to treat the discontinuity of the pressure due to the presence of the surface tension the extended finite element method (XFEM) is applied, see e.g. [6].

Extended finite element method. The XFEM enlarges the original FE space Q_h using the localization of an enrichment function. For the localization the original base functions of Q_h are used, i.e. we denote the index set $\mathcal{J} = \{1, \dots, n\}$, $n = \dim Q_h$ and the mesh nodes by $\boldsymbol{x}_j$, $j \in \mathcal{J}$. The nodal base functions are then denoted by $q_i \in Q_h$, $i \in \mathcal{J}$ and satisfy $q_i(\boldsymbol{x}_j) = \delta_{ij}$. The $\mathcal{J}'$ is the subset of all the neighbours of the interface $\hat{\Gamma}_t$, i.e. $\mathcal{J}' = \{j \in \mathcal{J} : \operatorname{supp} q_j \cap \hat{\Gamma}_t \neq \emptyset\}$. We shall use the discontinuous enrichment function $H_\Gamma(\boldsymbol{x})$ given as the Heaviside function $H_\Gamma(\boldsymbol{x}) = H(\boldsymbol{x}, t_{n+1})$. Now, the enrichment of the space Q_h is made using the discontinuous base functions q_j^{xfe} defined by $q_j^{xfe}(\boldsymbol{x}) = q_j(\boldsymbol{x})\,(H_\Gamma(\boldsymbol{x}) - H_\Gamma(\boldsymbol{x}_j))$. Here, $H_\Gamma(\boldsymbol{x}_j)$ can be left out from the right hand side as this only adds a constant multiple of the continuous base function $q_j(\boldsymbol{x})$. On the other hand, this term makes the function $q_j^{xfe}(\boldsymbol{x})$ being zero at every node $\boldsymbol{x}_i$, $i \in \mathcal{J}$ and also makes the support of $q_j^{xfe}(\boldsymbol{x})$ localized only to the elements containing the interface $\hat{\Gamma}_t$, which simplifies the practical discretization of the problem. The FE space Q_h is then replaced by the extended FE space $Q_h^{xfe} = Q_h \oplus \operatorname{span}\{q_j^{xfe} : j \in \mathcal{J}'\}$.

Level set step and coupled problem. Eq. (10) is time discretized, weakly formulated and the standard Galerkin FE method is employed, leading to the discrete system

$$\mathbb{M}(\Phi^{(n+1)} - \Phi^{(n)}) + \Delta t\, \mathbb{K}\Phi^{(n+1)} = 0, \tag{14}$$

where $\mathbb{M}$ is the consistent mass matrix, the matrix $\mathbb{K}$ represents the convection and $\Phi^{(k)} = \left(\phi^{(k)}(\boldsymbol{x}_i)\right)_{i \in \mathcal{J}}$ denotes the nodal values of the level set function. In order to obtain a stable scheme, the algebraic flux corrections can be applied, see [4]. Nevertheless, in the considered case of a continuous level set function φ, this is mostly equivalent to the Galerkin method (at least for a limited time period). It is also known, that for the level set method a re-initialization step is needed to

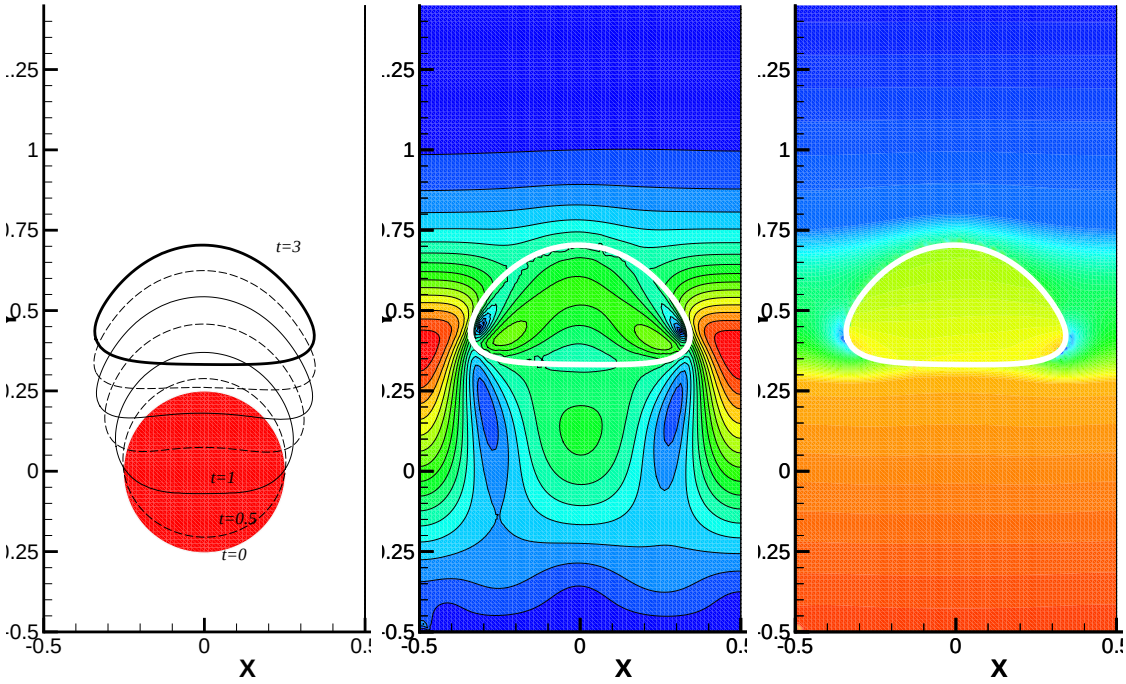


Figure 2: The result of the rising bubble case: The shape of the interface at time instant $t \in \{0, 0.5, 1, 1.5, 2, 3\}$ (on the left), the velocity magnitude isolines (middle), and the pressure isolines (on the right).

maintain the distance like property, see also [3]. Thus we simply use the Galerkin FE approximations and perform the re-initialization step every 5-40 iterations.

The solution of the coupled problem is then performed by the de-coupled algorithm: Assume that the approximations of $\boldsymbol{u}^n$, p^n, ϕ^n, ρ^n, μ^n and $\hat{\Gamma}^{(n)}$ are already known.

I. Solve (14) using the flow velocity $\boldsymbol{u}^n$ to determine ϕ^{n+1}. Perform the re-initialization if needed.

II. Using the approximation ϕ^{n+1} determine ρ^{n+1}, μ^{n+1} and $\hat{\Gamma}^{n+1}$.

III. Solve (11) for approximation of flow velocity $\boldsymbol{u}^{n+1}$ and p^{n+1}.

IV. Set n:= n + 1 and go to I.

5. Numerical results

The numerical results are shown for the case of a rising bubble considered in [3], where the following values were used $\rho^A = 1000\,\mathrm{kg\,m^{-3}}$, $\rho^B = 100\,\mathrm{kg\,m^{-3}}$, $\mu_A = 10\,\mathrm{Pa\,s}$, $\mu_B = 1\,\mathrm{Pa\,s}$, $\boldsymbol{f} = (0, -0.98)\,\mathrm{m\,s^{-2}}$ and $\gamma = 24.5\,\mathrm{N/m}$. The height of the computational domain is $H = 2\,\mathrm{m}$ and width is $W = 1\,\mathrm{m}$. The fluid B is originally

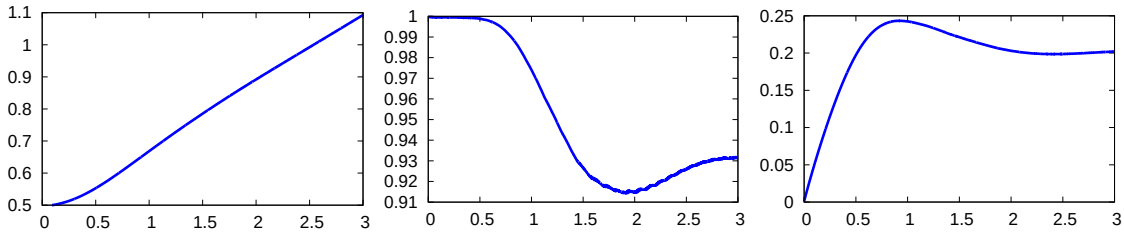


Figure 3: The quantitative results for the rising bubble case: The graphs of the center of mass T_y, the circularity C and the rise velocity V from the left to the right, respectively.

located in the circle of the diameter 0.5 m, whose center is displaced by 0.5 m up from the bottom of the domain. The boundary Γ_W contains the bottom and top of the domain, whereas Γ_S includes the rest of the boundary (i.e. $\Gamma_O = \emptyset$). Due to the gravity force, the fluid B with the lower density starts to rise, which also leads to a shape deformation. However, after some time the fluid B - due to the high value of the surface tension - develops a more stable shape, which keeps rising undeformed, see Fig. 2. The computations were performed on a triangular mesh with an equidistant partition and the spatial step $h = 1/40$ (the coarsest mesh used in [3]). The time step used in the computation was $\Delta t = 0.002$. The motion of the domain $\Omega^B_{(t)}$ with the area $\mathcal{A}(t)$ was tracked in terms of the $y-$coordinate of the center of mass $T_y(t) = \int_{\Omega^B_{(t)}} x_2 \,\mathrm{d}x/\mathcal{A}(t)$, the circularity defined by $C(t) = 2\sqrt{\pi\mathcal{A}(t)}/\int_{\partial\Omega^B_{(t)}} 1 \,\mathrm{d}S$ and the rise velocity $V = \int_{\Omega^B_{(t)}} u_2 \,\mathrm{d}x/\mathcal{A}(t)$. In order to verify the presented numerical method the values of T_y, C and V were computed at every time instant. The graphs of T_y, C and V in dependence on time shown in Figure 3 agrees well with the results in [3]. The quantitative comparison of the referenced values presented in [3] is shown in Table 1, where $T_y(3)$ is the mass center location at time $t = 3\,\mathrm{s}$, $C_{\min}$ denotes the minimal circularity, $V_{\max}$ denotes the maximal rise velocity, $t(C = C_{\min})$ and $t(V = V_{\max})$ are the time instants of their occurrence, respectively.

6. Conclusion

The detailed mathematical description of the motion of two immiscible fluids flow was presented, where the surface tension was approximated using its weak reformulation. The first order time discretization was used and the finite element method was used for the space discretization. The XFEM was employed to capture correctly the discontinuity of the pressure along the surface caused by the surface tension. The solution of the flow problem was coupled with the FEM applied for solution of the transport equation for the level set function. The decoupled strategy was used for the solution of the coupled problem. The presented numerical method was applied for approximation of the benchmark [3]. The data from the numerical simulations shows very good agreement with the reference values even though here only the first

	$T_y(3)$	$C_{\min}$	$t(C = C_{\min})$	$V_{\max}$	$t(V = V_{\max})$
ref. [3]	1.0813	0.9013	1.9041	0.2417	0.9213
present study	1.0801	0.9025	1.898	0.2421	0.92

Table 1: The quantitative results for the rising bubble case: the comparison of the computed and the reference quantities.

order in time discretization was used. The obtained numerical results verify the applied numerical method and its usability for approximation of flows influenced by the surface tension.

Acknowledgements

This work was supported by grant No. 13-00522S of the Czech Science Foundation.

References

[1] Barrett, J. W., Garcke, H., and Nürnberg, R.: Eliminating spurious velocities with a stable approximation of viscous incompressible two-phase Stokes flow. Compu. Methods Appl. Mech. Engrg. **267** (2013), 511–530.

[2] Hirt, C. W. and Nichols, B. D.: Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. **39** (1981), 201–225.

[3] Hysing, S. et al.: Quantitative benchmark computations of two-dimensional bubble dynamics. Internat. J. Numer. Methods Fluids **60** (2009), 1259–1288.

[4] Kuzmin, D.: On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. J. Comput. Phys. **219**(2), 2006, 513–531.

[5] Ransau, S. R.: Solution methods for incompressible viscous free surface flows: A literature review. Tech. Rep. 3/2002, Norwegian University of Science and Technology, Trondheim, 2002.

[6] Sauerland, H. and Fries, T. P.: The stable XFEM for two-phase flows. Comput. & Fluids **87** (2013), 41–49.

[7] Sethian, J. A.: *Level set methods and fast marching methods.* Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, Cambridge, U.K., 1999, 2^{nd} edn.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

COMPUTATIONAL APPROACHES TO SOME INVERSE PROBLEMS FROM ENGINEERING PRACTICE

Jiří Vala

Brno University of Technology, Faculty of Civil Engineering
602 00 Brno, Veveří 331/95, Czech Republic
vala.j@fce.vutbr.cz

Abstract

Development of engineering structures and technologies frequently works with advanced materials, whose properties, because of their complicated microstructure, cannot be predicted from experience, unlike traditional materials. The quality of computational modelling of relevant physical processes, based mostly on the principles of classical thermomechanics, is conditioned by the reliability of constitutive relations, coming from simplified experiments. The paper demonstrates some possibilities of computational identification of such relations, namely for heat and mass transfer, coming from original experimental and numerical results obtained at the Brno University of Technology, in selected engineering applications.

1. Introduction

The analysis of inverse problems is a relatively new interdisciplinary field of knowledge, connecting several theoretical and experimental branches: i) theory of ordinary and partial differential equations, ii) development of robust and effective computational algorithms, coming from the least squares, conjugate gradients, etc. approaches – cf. [8], iii) handling unstable and ill-posed problems, needing construction of artificial regularizers, as discussed in [15], p. 26, iv) transparent physical analysis, taking into account the most significant processes in engineering problems, namely those motivated by the development of structures and technologies, working with advanced materials, whose properties, because of their complicated microstructure, cannot be predicted from experience, unlike traditional materials, v) design of experiments for reliable identification of mechanical, thermal, moisture, etc. characteristics of such materials.

However, the general conception of inverse problems covers problems in nondestructive testing, seismic exploration, remote sensing, radio- and tomography, discussed in [15], p. 192, as well as the determination of an unknown source in the heat equation thanks to some overdetermined values of temperature and heat fluxes like [36]. In this paper we shall pay attention to the shorter list of inverse problems:

from certain balance laws from classical thermomechanics, supplied by constitutive relations, we shall try to determine the unknown or uncertain values of engineering macroscopic characteristics occurring in such relations, thanks to some overdetermined data, obtained by some well-advised experiments.

2. Physical and engineering considerations

Respecting the standard notation of Lebesgue, Sobolev, Bochner, etc. (abstract) function spaces by [25], p. 14, we shall start with a model problem from classical thermomechanics: the conservation of a scalar quantity $u \in L^2(I,V)$ with $V = W^{1,2}(\Omega)$ on certain domain Ω in the Euclidean space R^3 with the boundary Γ supplied by the Cartesian coordinates $x = (x_1, x_2, x_3)$, and on some finite time interval $I = [0, \varsigma]$, bounded by a constant ς, can be expressed, following [4], p. 9, in the form

$$\dot{\varepsilon}(u) + \nabla \cdot \eta(u) = f \qquad \text{on } I \times \Omega\,; \tag{1}$$

dot symbols (here and later everywhere) refer to derivatives with respect to $t \in I$, $f \in L^2(I,H)$ with $H = L^2(\Omega)$ refers to some volume source and $\eta : L^2(I,V) \to L^2(I,V)$ and $\varepsilon : W^{1,2}(I,H) \to W^{1,2}(I,H)$ are certain material-dependent mappings; for the example of conservation of energy with u taken as (absolute) temperature, thermal fluxes $\eta(u)$ and enthalpic (evolutionary) terms $\varepsilon(u)$ see [25], p. 252. Let us assume that Ω is sufficiently smooth to guarantee the validity of Sobolev imbedding, trace and similar theorems by [25], p. 16, needed also in the Gelfand triple by [25], p. 190; more general geometrical configurations could be studied (overcoming a lot of technical difficulties) following [21], p. 62, 222 and 385. Let Γ be decomposed to some disjoint parts Γ_c and Γ_i; consequently we are able to formulate the boundary conditions of the Neumann type

$$\eta(u) \cdot \nu = g \qquad \text{on } I \times \Gamma_c \tag{2}$$

utilizing the (formally) outward unit normal $\nu(x) = (\nu_1(x), \nu_2(x), \nu_3(x))$ on Γ, and those of the Robin type

$$\eta(u) \cdot \nu = \psi(u, u_a) \qquad \text{on } I \times \Gamma_i\,; \tag{3}$$

here we need to know some ambient values $u_a \in L^2((I, L^4(\Gamma_c))$, together with a new (material) interface-dependent mapping $\psi : L^2(I, V \times L^4(\Gamma_c)) \to L^2(I, L^2(\Gamma_i))$. We shall consider the initial $u(.,0) = 0$ on Ω here only; it can be verified that any equilibrium initial condition can be converted to this form.

The much-favoured engineering linearizations of mappings included in (1), (2) and (3) (prime symbols refer to derivatives by the following variables) are $\dot{\varepsilon}(u) = \varepsilon'(u)\dot{u} \approx \kappa\dot{u}$ with some $\kappa \in L^\infty(\Omega)$, $\eta(u) = -\nabla\beta(u) = -\beta'(u)\nabla(u) \approx -\lambda\nabla u$ (in the Fourier, Fick, ... "laws") with some $\lambda \in L^\infty(\Omega)$ and $\psi(u, u_a) \approx \gamma(u - u_a)$ with some $\psi : L^2(I, V \times L^4(\Gamma_c)) \to L^2(I, L^2(\Gamma_i))$. Let us notice that even the existence of some $\beta(u)$ represents an additional assumption: it forces the zero rotation

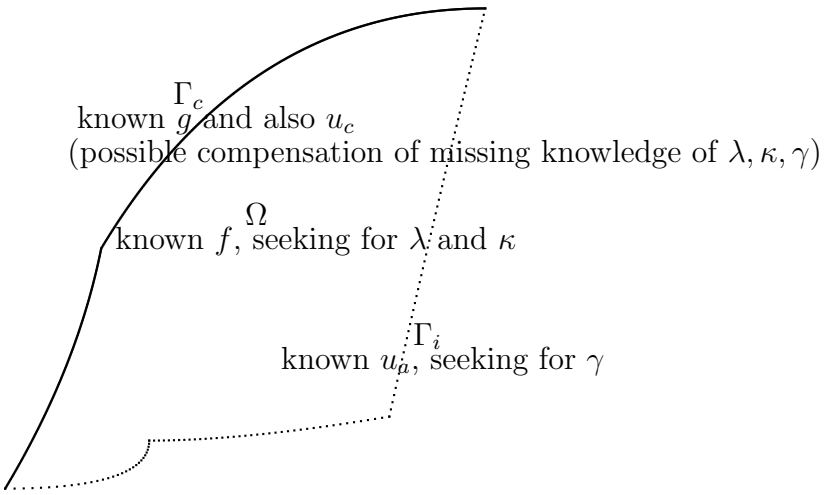


Figure 1: A simplified scheme of geometrical configuration for a model problem.

of $\eta(u)$. Moreover, such scalar characteristics are admissible just for (macroscopically) isotropic media; in more general cases matrix characteristics are necessary.

Fig. 1 shows the above sketched geometrical configuration. To enable the effective analysis with some unknown or uncertain characteristics, some $u_c \in L^2(I, L^2(\Gamma_c))$ is prescribed, too, considered to coincide with the traces of u.

For simplicity, we shall introduce the following notation of scalar products in $L^2(I, X)$, with (generalized) functions ϕ and $\widetilde{\phi}$ from corresponding spaces, i. e. $X = L^2(\Omega)$, $X = L^2(\Omega)^3$, $X = L^2(\Gamma)$,

$$(\phi, \widetilde{\phi}) = \int_I \int_\Omega \phi(x)\widetilde{\phi}(x)\,\mathrm{d}x\,\mathrm{d}t\,, (\nabla\phi, \nabla\widetilde{\phi}) = \int_I \int_\Omega \nabla\phi(x) \cdot \nabla\widetilde{\phi}(x)\,\mathrm{d}x\,\mathrm{d}t\,,$$

$$\langle \phi, \widetilde{\phi} \rangle = \int_I \int_\Gamma \phi(x)\widetilde{\phi}(x)\,\mathrm{d}s(x)\,\mathrm{d}t\,,$$

with $s(x)$ in the sense of Hausdorff measure on Γ; $\langle \phi, \widetilde{\phi} \rangle_i$, $\langle \phi, \widetilde{\phi} \rangle_c$ will denote the same as $\langle \phi, \widetilde{\phi} \rangle$, with Γ_i, Γ_c instead of Γ. Such scalar products are available because X are still Hilbert spaces; some appropriate dualities can be considered instead of them in more general considerations.

The significance of particular physical (and chemical and other) processes depends on engineering applications. In particular, in civil engineering the following processes come into consideration: i) heat transfer (conduction, convection, radiation), ii) air flow, iii) moisture redistribution in porous media, iv) salt and contaminant transport, v) chemical reactions (maturing silicate mixtures, carbonation, ...), vi) phase changes (including those in advanced phase change materials), vii) mechanical deformation (elasticity, plasticity, creep, damage, ...). The above sketched thermomechanical approach generates the balance conditions for a) mass (continuity equations) - with variable density, b) (linear and angular) momentum (Navier - Stokes equations, formulated for various continuum models: by Boltzmann, Cosserat, etc.)

- with variable velocity components (in some reference geometrical configuration), c) energy (Fourier equation) - with variable temperature, d) (semi-)empirical constitutive laws for remaining quantities, separately for particular phases. Bridging between micro- and macrostructure could be performed using some periodic homogenization approach, e. g. the two-scale convergence by [7], or its non-periodic (much more complicated) generalization by [12]; nevertheless, most engineering approaches rely on the mixture theory. Such "multiphysical" analysis dates back to the simple Luikov model, presented in [20], of the simultaneuos heat and moisture transfer, coming to the system of 2 equations of evolution

$$\dot{\tau} = \Delta\tau + \mathcal{K}\dot{\omega}\,, \qquad \dot{\omega} = \mathcal{L}\Delta\omega + \mathcal{L}\mathcal{P}\Delta\tau$$

for 2 unknown functions: the temperature $\tau(x,t)$ and the moisture content $\omega(x,t)$; 3 material characteristics (positive constants) $\mathcal{L}$, $\mathcal{P}$, $\mathcal{K}$ are well-known as Luikov, Posnov and Kossovich numbers. Its slight generalization works with the corresponding fluxes

$$\eta_\tau = (.)\nabla\tau + (.)\nabla\omega\,, \qquad \eta_\omega = (.)\nabla\tau + (.)\nabla\omega$$

and the deeper analysis of material characteristics in all (.) positions; then the first equations handles the so-called Dufour effect, the second equation the so-called Soret one. Much more generalized computational models have been supported by the computer hardware and software development in the last decades: e. g. the model of maturing concrete mixture from [33], referring to the approach of [14], contains 20 equations of evolution, coming from the conservation of mass, momentum and energy related to 4 phases, supplied by appropriate algebraic constitutive relations; the hydration degree, driving the fraction of particular phases must be evaluated from an auxiliary ordinary differential equation.

3. Experimental settings

Unlike complicated advanced "multiphysical" models for direct deterministic calculations, all identification procedures try to arrange necessary measurements under very special conditions, i) to remove or suppress most other influences disturbing a separate physical process by (1), ii) to simplify the geometrical configuration to reduce the complexity of the mathematical and computational analysis, e. g. by the reduction of dimension, thanks to various symmetries, iii) to have a chance to perform some reasonable a posteriori uncertainty analysis. An example of such simple inexpensive measurement equipment for the identification of the thermal conductivity λ and of the thermal capacity (related to unit volume) κ, assuming $\gamma = 0$, is shown on Fig. 2. The controlled heat flux, accompanied by the temperature recorder, supplies all information, needed by Fig. 1. Moreover, for sufficiently large plates the one-dimensional simplification (at least for the first estimate of λ and κ) by [27] is available. However, the proper analysis in R^3 leads to rather complicated relations: even in the case of exploitation of analytic integrals by [3], p. 193, their numerical evaluation may be not quite easy.

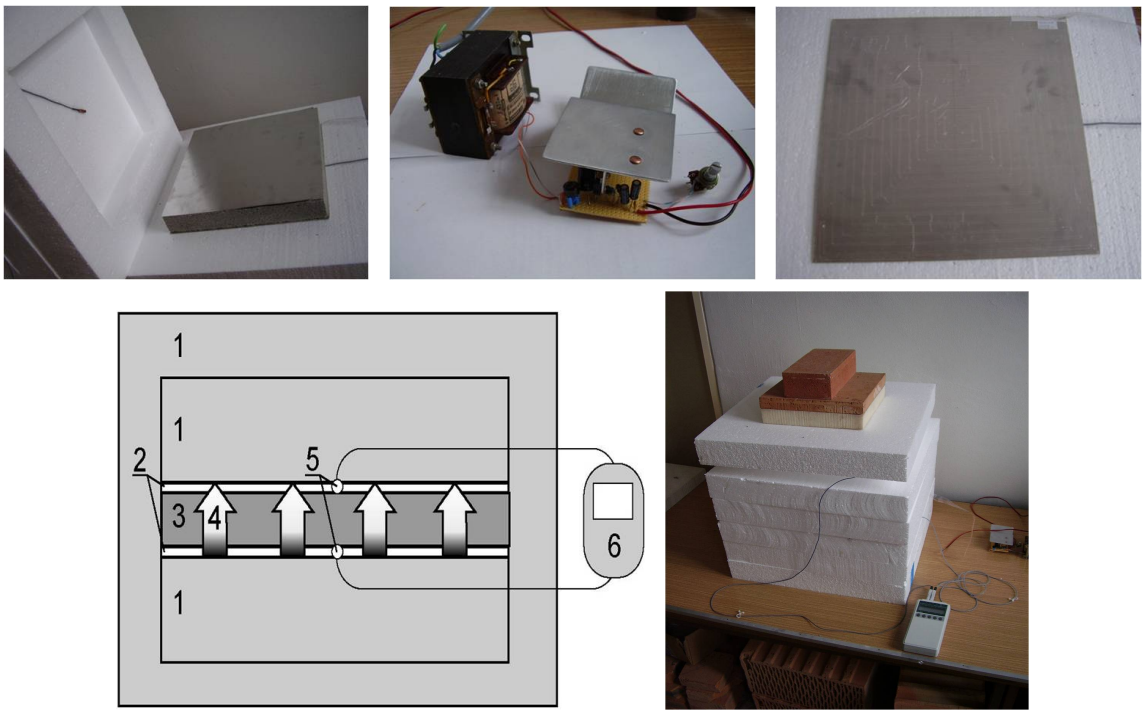


Figure 2: Measurement equipment of the hot-plate type: 1 massive polystyrene insulation layer, 2 couple of aluminium plates: lower heated, upper cold, 3 sample with unknown λ and κ, 4 direction of controlled heat fluxes, 5 temperature sensor(s), 6 temperature recorder.

Other technical solutions of measurement systems than the just presented hot-plate one are known as the hot-ball and hot-wire ones – see [1]. The hot-ball approach works with a sufficiently small heated metal ball, utilizing the spherical coordinates for all computational evaluations, the hot-wire one with a very thin and long heated metal wire, utilizing the cylindric coordinates. In some laboratory settings, namely under hard conditions, as for the testing of fire-clay brickworks, or for the alternative design of powdery insulation materials at high temperature and in vacuum, as an important component of certain heat production and storage system based on sunlight and optical fibers, some modifications are needed, in particular the (nearly) ideal thin hot wire has to be replaced by some massive hollow (ceramic or metal) cylinder, as shown on Fig. 3; for more details see [16].

Especially in the case of elevated or high temperature, in maturing concrete mixtures, during the fire simulation, etc., the factors λ and κ are not constant; as an illustrative example, the lower part of Fig. 3 shows λ for selected powdery insulations (aerogel, perlite, crashed fire clay and certain experimental nano-particles-based material) as a (not very rapidly) increasing function of temperature. Relevant experiments can be organized in several steps at some discrete environmental temperature levels; the contribution of additional thermal fluxes generated by the measurement equipment can be considered as negligible. However, such approach is not practicable in the case of the capillary transfer coefficient λ where (1) describes the conservation

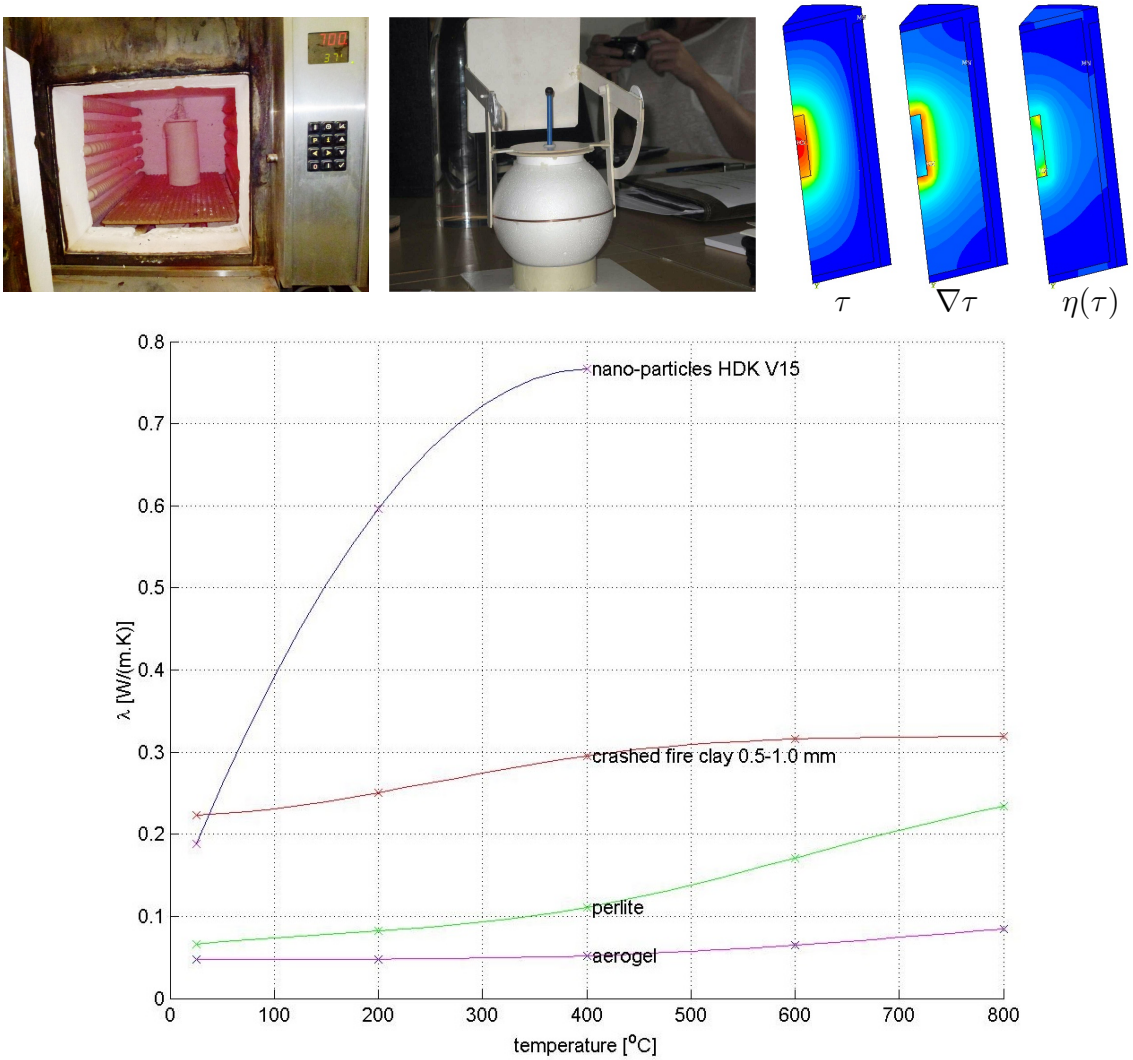


Figure 3: From the left: i) high-temperature cylindric measurement equipment, ii) small model of the thermal accumulator, iii) results of supporting ANSYS-based computations for the evaluation procedure on a cylindric segment, due to service wires. Lower graph: temperature dependence of the thermal conductivity for selected types of powdery insulations.

of moisture mass in some porous material structure ($\kappa = 1$ can be set without loss of generalization) because all experiments show strong dependence of such coefficient on the moisture volume fraction u, thus the tricks with simple functions (like the preceding case) are not adequate. Moreover, to prevent the lack of input data for the identification procedure, the knowledge of values u is needed on Ω or its substantial part, not only on its boundary. Consequently no direct and nondestructive measurements are available; a reasonable compromise may be the indirect measurement exploiting the microwave technique, based on the difference between (relative) electric permittivity and/or magnetic permeability of water and air in pores, as sketched on Fig. 4; for more details on laboratory measurements including calibration techniques see [26].

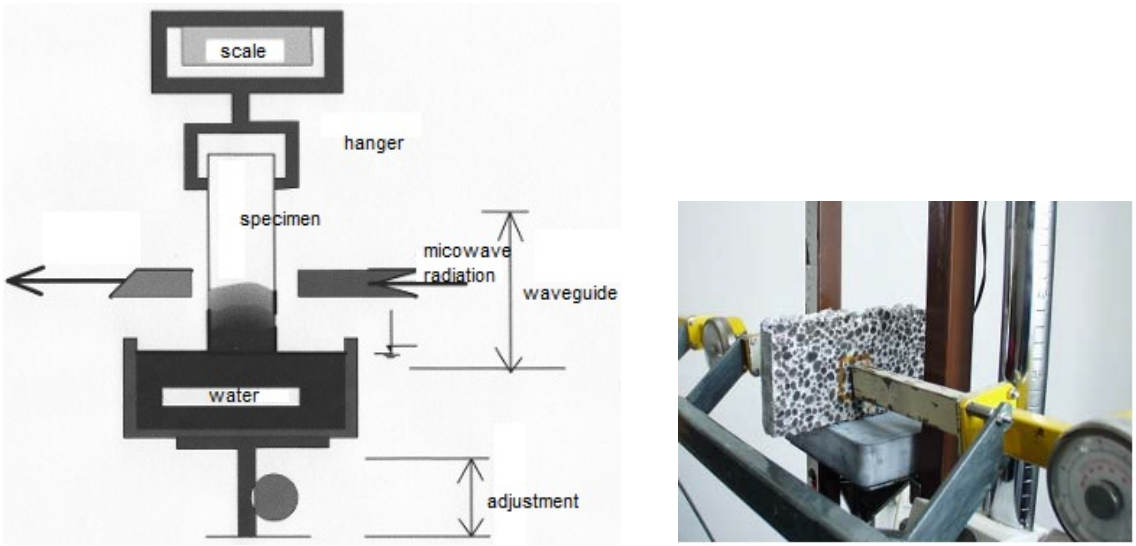


Figure 4: Indirect nondestructive microwave measurement of water content in porous material structure for the identification of the capillary transfer coefficient.

4. Linear and quasilinear problems

As evident from the previous section, we shall work with the set of (in general a priori unknown) characteristics $\vartheta = (\gamma, \lambda, \kappa)$ in appropriate admissible sets of (usually positive) functions.

Following [35] and [17], according to [5], p. 135, let us introduce two functionals

$$F(\vartheta, u, v) = (\kappa \dot{u}, v) + (\lambda \nabla u, \nabla v) + \langle \gamma, uv \rangle_i - (f, v) - \langle g, v \rangle_c - \langle \gamma, u_a v \rangle_i \,,$$

$$G(u) = \frac{1}{2} \langle w, (u - u_c)^2 \rangle_c \,,$$

supplied by certain weight $w \in L^2(\Gamma_c)$, defined for arbitrary $t \in I$, for $u, v \in L^2(I, V)$; consequently $u, v \in L^2(I, L^4(\Gamma))$ and $uv \in L^2(I, L^2(\Gamma))$. This requires the application of the trace theorem; moreover the Sobolev theorem on (compact) imbedding, the Friedrichs - Poincaré inequality, the Lax - Milgram theorem (and its generalizations), the properties of Rothe sequences of abstract functions (continuous and discrete Gronwall lemma, Gelfand imbedding, ...), the Aubin - Lions lemma for abstract functions, etc. (cf. [25] and [13]), are needed in the complete proofs of the following propositions.

Now we are ready to formulate a) a direct model problem, b) a sensitivity one and c) an adjoint one, useful namely in linearized considerations, including those with slightly variable material characteristics (due to the motivation from the preceding section). Such formulations will be useful for the design of a general algorithm for the analysis of an inverse problem, i. e. the problem of identification of $\vartheta = (\gamma, \lambda, \kappa)$ here. Some particular cases may occur in the literature typically: e. g. [17] takes variable γ only, moreover in the steady-state case.

4.1. A direct problem

The weak formulation of a direct problem reads: for some fixed β and $u_0 = 0$ find u such that $F(\vartheta, u, v) = 0$ for any v, i. e.

$$(\kappa \dot{u}, v) + (\lambda \nabla u, \nabla v) + \langle \gamma, (u - u_a)v \rangle_i = (f, v) + \langle g, v \rangle_c \,,$$

valid for any $t \in I$ (here and in all analogous situations). Its strong formulation comes from the obvious application of the Green - Ostrogradskiĭ theorem

$$(\kappa \dot{u} - \nabla \cdot (\lambda \nabla u) - f, v) = \langle \gamma(u_a - u) - \lambda \nabla u \cdot \nu, v \rangle_i + \langle g - \lambda \nabla u \cdot \nu, v \rangle_c \,.$$

However, the reverse application of the same theorem is possible, too; e. g. for the fundamental solution $v_*(x, y) = -1/(4\pi|x-y|)$ of the equation $\Delta v_*(x, y) = 4\pi\delta(x-y)$ locally for $y \in \Omega$ instead of $v(x, t)$ with fixed $t \in I$ we obtain

$$(\kappa \dot{u}, v) - (\beta(u), \Delta v) = (f, v) + \langle \gamma, (u - u_a)v \rangle_i + \langle g, v \rangle_c - \langle \beta(u), \nabla v \cdot \nu \rangle$$

Let us remind that generalized initial conditions are also available: there is sufficient to take $f - f_0$, $g - g_0$, $u_a - u_{a0}$ and $u - u_0$ instead f, g, u_a and u where all zero indices refer to values in $t = 0$; the same could be done for sensitivity and adjoint problems, too.

4.2. A sensitivity problem

The weak formulation of a sensitivity problem reads: for some fixed ϑ, $\widetilde{\vartheta}$ (expressing some change of ϑ) and $u_0 = 0$ find $\widetilde{u}$ such that $DF(\vartheta, u, v, \widetilde{\vartheta}, \widetilde{u}, o) = 0$, with o referring to zero-valued functions, for any v, i. e.

$$(\kappa \dot{\widetilde{u}}, v) + (\lambda \nabla \widetilde{u}, \nabla v) + \langle \gamma, \widetilde{u} v \rangle_i = \langle \widetilde{\gamma}, (u_a - u)v \rangle_i - (\widetilde{\lambda} \nabla u, \nabla v) - (\widetilde{\kappa} \dot{u}, v) \,.$$

Its strong formulation comes from the obvious application of the Green - Ostrogradskiĭ theorem

$$(\kappa \dot{\widetilde{u}} - \nabla \cdot (\lambda \nabla \widetilde{u}) - \nabla \cdot (\widetilde{\lambda} \nabla u), v)$$

$$= \langle \widetilde{\gamma}(u_a - u) - \gamma \widetilde{u} - \lambda \nabla \widetilde{u} \cdot \nu - \widetilde{\lambda} \nabla u \cdot \nu, v \rangle_i - \langle \lambda \nabla \widetilde{u} \cdot \nu + \widetilde{\lambda} \nabla u \cdot \nu, v \rangle_c \,.$$

The reverse application of the Green - Ostrogradskiĭ theorem gives here

$$(\kappa \dot{\widetilde{u}}, v) - (\beta(\widetilde{u}) + \widetilde{\beta}(u), \Delta v) + \langle \gamma, \widetilde{u} v \rangle_i = \langle \widetilde{\gamma}, (u_a - u)v \rangle_i - \langle \widetilde{\kappa} \dot{u}, v \rangle - \langle \beta(\widetilde{u}) + \widetilde{\beta}(u), \nabla v \cdot \nu \rangle \,.$$

4.3. An adjoint problem

The weak formulation of an adjoint reads: for some fixed ϑ and $u_\varsigma = 0$ find v such that $DF(\vartheta, u, v, o, \widetilde{u}, o) = DG(u, \widetilde{u})$ for u coming from a direct problem and for any $\widetilde{u}$, i. e.

$$-(\kappa \widetilde{u}, \dot{v}) + (\lambda \nabla \widetilde{u}, \nabla v) + \langle \gamma, \widetilde{u} v \rangle_i$$

Its strong formulation comes from the obvious application of the Green - Ostrogradskiĭ theorem

$$-(\widetilde{u}, \kappa\dot{v} + \nabla\cdot(\lambda\nabla v)) = \langle\widetilde{u}, w(u-u_c) - \lambda\nabla v\cdot\nu,\rangle_c - \langle\widetilde{u}, \gamma v + \lambda\nabla v\cdot\nu\rangle_i\,.$$

The reverse application of the Green - Ostrogradskiĭ theorem gives here

$$-(\widetilde{u}, \kappa\dot{v}) - (\Delta\beta(\widetilde{u}), v) + \langle\gamma, \widetilde{u}v\rangle_i = \langle w, (u-u_c)\widetilde{u}\rangle_c - \langle\nabla\beta(\widetilde{u})\cdot\nu, v\rangle\,.$$

Combining $\widetilde{u}$ from a sensitivity and v from an adjoint problem, we receive

$$\langle w, (u-u_c)\widetilde{u}\rangle_c = \langle\widetilde{\gamma}, (u_a-u)v\rangle_i - (\widetilde{\lambda}\nabla u, \nabla v) - (\widetilde{\kappa}\dot{u}, v)\,;$$

thus it is natural to introduce a new functional

$$J(\vartheta) = \int_I G(u)\mathrm{d}t\,.$$

4.4. Computational algorithms

For simplicity of notation, let us set $J_*(\gamma) = J(\vartheta)$ here, in particular with $\vartheta = (\gamma, o, o)$; the analogous derivation of the general case is left to the (very patient) reader. Then we shall need some reasonable estimate γ^0 for the construction of iterations γ^k with $k \in \{1, 2, \ldots\}$, the evaluation of gradients $\mathcal{G}^k = (u^k(\gamma^k) - u_a)v^k$ and differentials $DJ_*(\gamma^k, \widetilde{\gamma}^k) = \langle\widetilde{\gamma}^k, \mathcal{G}^k\rangle_i$, $D^2J_*(\gamma^k, \widetilde{\gamma}^k, \widetilde{\gamma}^k) = \langle w, \widetilde{u}(\gamma^k, \widetilde{\gamma}^k)^2\rangle_c$. The conjugate gradient algorithm, following [2], can be expressed in the form

$$\gamma^{k+1} = \gamma^k + a^k\widetilde{\gamma}^k\,,$$

$$\widetilde{\gamma}^k = b^k\widetilde{\gamma}^{k-1} - \mathcal{G}^k\,, \quad \text{in particular } \widetilde{\gamma}^0 = 0 \;\; (b^1 \text{ is not needed})\,;$$

a^k come from the minimum line search with the result

$$a^k = -DJ_*(\gamma^k, \widetilde{\gamma}^k)/D^2J_*(\gamma^k; \widetilde{\gamma}^k; \widetilde{\gamma}^k)\,,$$

whereas b^k are generated by the Fletcher - Reeves formula

$$b^k = \langle\mathcal{G}^k, \mathcal{G}^k\rangle_i/\langle\mathcal{G}^{k-1}, \mathcal{G}^{k-1}\rangle_i\,,$$

the Dai - Yuan formula

$$b^k = \langle\mathcal{G}^k, \mathcal{G}^k\rangle_i/\langle\widetilde{\gamma}^{k-1}, \mathcal{G}^k - \mathcal{G}^{k-1}\rangle_i\,,$$

or some similar one; for the discussion of suitable choice of such formulae see [23] and [29]. Especially for an assumed constant γ on Γ_i this degenerates to the classical Newton algorithm.

Now the complete computational strategy depends on the choice of number of iterations for γ^k, λ^k and κ^k separately. However, λ^k and κ^k, defined on Ω, may suffer from the lack of data, namely in the case of their rather rich admissible sets; therefore some modification of this approach could be needed. Certain remedy will be recommended in the sixth section.

5. Stochastic generalizations

To obtain $J(\vartheta) \approx 0$ in the previous section is quite not realistic; this depends not only on the quality, efficiency and robustness of the above presented purely deterministic algorithm, but also on the stochastic character of data, influence of disturbing physical processes and measurement imprecisions. However, sources of such errors cannot be distinguished, which restrains the validity of identification results; moreover, most technical standards on the laboratory testing require to submit some uncertainty analysis. Thus it could be useful to generalize all deterministic formulations to stochastic ones, although a lot of difficulties, including that in the mathematical verification (as the absence of simple imbedding and similar theorems for proofs), must be expected.

In general, instead of the spaces of abstract functions of the type $L^2(I, \mathcal{S})$ with S taken as V, $L^2(\Omega)$, etc., we are able, following [35], to define the spaces $L^2(\Theta, I, \mathcal{S})$ where Θ refers to a space of elementary events, supplied with some σ-algebra and some probability measure P. Our optimization functional then obtains a new parameter $\theta \in \Theta$, i. e.

$$J_*(\gamma) = \frac{1}{2} \int_\Theta \int_I \int_{\Gamma_c} w(x,\theta)(u(x,t,\theta) - u_c(x,t,\theta))^2 \,\mathrm{d}s(x)\,\mathrm{d}t\,\mathrm{d}P\,.$$

Various approaches to the mimimization of such (or similar) functional can be then found in the literature, e. g. i) [22] applies the Karhunen - Loève spectral expansion, or, alternatively, the expansion based on the Hermitean polynomial chaos, which leads to the stochastic finite element technique, ii) [34] prefers the Bayesian approach, with Markov chains and Monte Carlo simulations, iii) a quite different algorithm comes from the Sobol sensitivity analysis by [19], relying on Monte Carlo simulations again. Nevertheless, the common drawbacks of such analysis, in addition to the above mentioned difficulties in functional and numerical analysis, are numerous artificial regularization tricks, as the Tikhonov regularization by [36], absence of appropriate software tools oriented to engineering applications and exceedingly time-consuming and expensive computations.

6. Nonlinear problems

Regardless of the formal similarity of mass and energy balance equaitions, as well as of the linearized Fourier and Fick constitutive equations, typical material characteristics for diffusion of liquid water, water vapour and various contaminants are much more complicated than those from the heat transfer with dominated conduction, discussed in [31] – all results depend on material microstructure (not only on such macroscopic characteristics as volume fraction of pores) significantly, diffusion is typically not quite reversible, etc. Consequently the approach from the fourth section do not lead to any credible results for engineering simulations. As a motivation from an useful modification of such approach, we shall come from the experimental tool sketched on Fig. 4.

Let us start, following [25], p. 253, with some useful transforms and substitutions, namely with the enthalpic and Kirchhoff transformations by [31] (for various right-hand sides)

$$\kappa(u)\dot{u} - \nabla \cdot (\lambda(u)\nabla u) = \dots$$

$$\hat{\kappa}(u(r)) = \int_0^r \kappa(\rho)\,\mathrm{d}\rho\,, \qquad \hat{\lambda}(u(r)) = \int_0^r \lambda(\rho)\,\mathrm{d}\rho\,, \qquad \beta(u) = \hat{\lambda}(\hat{\kappa}^{-1}(u))\,,$$

consequently

$$\dot{U} - \Delta\beta(U) = \dots$$

for the (adroitly defined) enthalpy $U = \widehat{\kappa}(u)$. For simplicity, in all remaining considerations we shall take only $\kappa(u) = 1$, zero f and empty Γ_i.

For an effective computation, the natural requirements are: i) $u \approx u_*$ on some set $\Omega_* \subseteq \Omega$ with $\mathrm{meas}(\Omega_*) > 0$, with measured u_*-values, to avoid lack of data, ii) introduction of

$$G(u) = \frac{1}{2}(u - u_*, w(u - u_*))$$

with some weight $w \in L^\infty(\Omega_*)$, zero-valued outside Ω_* iii) local estimates of $\beta(.)$ or $\lambda(.)$, coming from the direct formulation. For sufficiently smooth $\beta(.)$ we are then able to perform obvious conversions

$$\nabla\beta(u) = \beta'(u)\nabla(u) = \lambda(u)\nabla u\,,$$

$$\Delta\beta(u) = \nabla \cdot \nabla\beta(u) = \nabla \cdot (\lambda(u)\nabla u) = \lambda'(u)\nabla u \cdot \nabla u + \lambda(u)\Delta u\,.$$

The weak formulation of a direct problem reads: for some fixed ϑ and $u_0 = 0$ find u such that $F(\beta, u, v) = 0$ for any v, i. e.

$$(\dot{u}, v) + (\nabla\beta(u), \nabla v) = \langle g, v\rangle\,.$$

Its strong formulation comes from the obvious application of the Green - Ostrogradskiĭ theorem

$$(\dot{u} - \Delta\beta(u), v) = \langle g - \nabla\beta(u) \cdot \nu, v\rangle\,, \tag{4}$$

or from its alternative form (with λ instead of β)

$$(\dot{u} - \lambda'(u)\nabla u \cdot \nabla u - \lambda(u)\Delta u, v) = \langle g - \nabla\beta(u) \cdot \nu, v\rangle\,. \tag{5}$$

The analysis of solvability of (4) can be done by [25], p. 239. The analogous (not quite general) analysis of (5) in [24] needs non-trivial regularity results from [13] and auxiliary lemmas from [10]. The reverse application of the Green - Ostrogradskiĭ theorem gives here

$$(\dot{u}, v) - (\beta(u), \Delta v) = \langle g, v\rangle - \langle \beta(u), \nabla v \cdot \nu\rangle\,.$$

Most authors do not distinguish between u and u_* at all, inserting u_* (if available) instead of u into all calculations. To identify a function $\beta(u)$, some decomposition (finite-dimensional in practical calculations) is needed. The standard one is $\beta(u) = c_i\beta_i(u)$ where the sum over $i \in \{1, 2, \ldots\}$ (due to the Einstein summation rule) is considered for prescribed functions $\beta_i(u)$ and unknown real coefficients c_i. However, [6], p. 62, develops another approximate method where $M_i = \{(x,t) \in \Omega \times I : \overline{\lambda}_{i-1} \leq \lambda(u(x,t)) \leq \overline{\lambda}_i\}$, $\lambda_i(u) = (\overline{\lambda}_{i-1} + \overline{\lambda}_i)/2$ and $c_i = \mathrm{meas}(M_i)$, utilizing a priori given constants $\overline{\lambda}_0, \overline{\lambda}_1, \ldots$, as a basis for the *double integration method* by [9].

Some explicit formulae for the evaluation of $\lambda(u)$ can be found in the literature, coming from the one-dimensional simplification on a half-line (for a theoretically infinite sample). The most celebrated result, based on the Boltzmann - Matano transformation $y = x/(2\sqrt{t})$ (generating an ordinary differential problem in y), is

$$\lambda(u(x,t)) = \frac{1}{2tu'_x(x,t)} \int_x^\infty \xi u'_\xi(\xi,t)\,\mathrm{d}\xi\,; \tag{6}$$

for various modifications of this formula and for the historical remarks see [18]. As shown in [32] (including an original software code in MATLAB), infinite integrals in (6) can be removed for the prescribed boundary flux g (from direct measurements) with the result

$$\lambda(u(x,t)) = \frac{1}{u'_x(x,t)} \left(\int_0^x \dot{u}(\xi,t)\,\mathrm{d}\xi - g(t) \right) .$$

Another modification of (6)

$$\lambda(u(x,t)) = -\frac{1}{u'_x(x,t)} \int_x^\infty \dot{u}(\xi,t)\,\mathrm{d}\xi$$

is presented as the *third integration method* in [28].

General estimates of $\beta(.)$ or $\lambda(.)$ from three-dimensional experimental data are more delicate, utilizing some (numerically unpleasant) Dirac distributions $\delta(.)$ in most cases. The *second integration method* by [28] comes from the equation of type

$$(\dot{u} - \lambda'(u)\nabla u \cdot \nabla u - \lambda(u)\Delta u, v) = \ldots$$

for $v = \delta(x - \xi)\delta(t - \iota)$, $\xi \in \Omega$ and $\iota \in I$. Consequently

$$\lambda'(u)\nabla u \cdot \nabla u + \lambda(u)\Delta u = \dot{u}$$

remains on $\Omega \times I$; this can be solved (unlike a direct nonlinear problem) as one linear ordinary differential equation. The *first integration method* by [28] considers

$$(\dot{u}, v) - (\beta(u), \Delta v) = \ldots$$

for $v(x,t) = v_*(x,\xi)\delta(t - \iota)$; the integration then gives

$$\beta(u(x,t)) = -\frac{1}{4\pi} \int_\Omega \frac{\dot{u}(\xi,t)}{|x - \xi|}\,\mathrm{d}\xi$$

locally. In the above announced *double integration method* it is sufficient to choose $v = \delta(x-\xi)\delta(t-\iota)$ with $\xi \in \Omega$ and $\iota \in I$ in

$$(\dot{u} - \nabla \cdot (\lambda(u)\nabla(u)), v) = \dots ;$$

however, M_i for $i \in \{1, 2, \dots\}$ must be (approximately) detected from the analysis of isohypersufaces $u(x,t)$, consequently the integration over $\Omega \times I$ is needed to determine c_i (which is extremely expensive for any two- ore more-dimensional case). An alternative approach of [6], p. 67, then relies on some special genetic algorithms; for still other alternative optimization approaches cf. [8].

Let us consider $c = (c_1, c_2, \dots)$ (and later also $\widetilde{c} = (\widetilde{c}_1, \widetilde{c}_2, \dots)$). A direct, sensitivity and adjoint problem can be now formulated similarly to the fourth section here; we shall present the weak formulations only. For a direct problem this reads: for some fixed $c = (c_1, c_2, \dots)$ and for $u_0 = 0$ find u such that $F(c, u, v) = 0$ for any v, i. e.

$$(\dot{u}, v) + (\nabla \beta_i(u), \nabla v) c_i = \langle g, v \rangle .$$

For a sensitivity problem this reads: for some fixed c and $\widetilde{c}$ and for $u_0 = 0$ find $\widetilde{u}$ such that $DF(c, u, v, \widetilde{c}, \widetilde{u}, o) = 0$ for any v, i. e.

$$(\dot{\widetilde{u}}, v) + (\nabla \beta_i(\widetilde{u}), \nabla v) c_i = (\nabla \beta_i(u), \nabla v) \widetilde{c}_i .$$

For an adjoint problem this reads: for some fixed c and for $u_\varsigma = 0$ find v such that $DF(c, u, v, o, \widetilde{u}, o) = DG(u, \widetilde{u})$ for u from a direct problem and for any $\widetilde{u}$, i. e.

$$-(\widetilde{u}, \dot{v}) + (\nabla \beta_i(\widetilde{u}), \nabla v) c_i = (w(u - u_*), \widetilde{u}) .$$

Combining $\widetilde{u}$ from a sensitivity and v from an adjoint problem, we receive

$$(\nabla \beta_i(u), \nabla v) \widetilde{c}_i = (w(u - u_*), \widetilde{u}) ;$$

$J(c) = G(u)$ can be introduced.

The conjugate gradient algorithm, starting from certain initial estimate c^0 of c, works with iterations c^k for $k \in \{1, 2, \dots\}$, gradients $\mathcal{G}^k = (u^k(c^k) - u_*) v^k$ and differentials $DJ_*(c^k, \widetilde{c}^k) = (\widetilde{c}^k, \mathcal{G}^k)$, $D^2 J_*(c^k, \widetilde{c}^k, \widetilde{c}^k) = (w\widetilde{u}(c^k, \widetilde{c}^k), \widetilde{u}(c^k, \widetilde{c}^k))$. This leads to the algorithm

$$c^{k+1} = c^k + a^k \widetilde{c}^k ,$$

$$\widetilde{c}^k = b^k \widetilde{c}^{k-1} - \mathcal{G}^k , \quad \text{in particular } \widetilde{c}^0 = 0 \ \ (b^1 \text{ is not needed})$$

again; here

$$a^k = -DJ_*(c^k, \widetilde{c}^k) / D^2 J_*(c^k; \widetilde{c}^k; \widetilde{c}^k) , \qquad b^k = (w\mathcal{G}^k, \mathcal{G}^k) / (w\mathcal{G}^{k-1}, \mathcal{G}^{k-1}),$$

with possible alternatives for the evaluation of b^k again.

7. Conclusion

The increase of requirements from engineering practice to reliable analysis of inverse problems, namely on identification of material characteristics in thermodynamical applications, discussed in this paper, due to advanced materials, structures and technologies, seem to be faster than the progress in analysis of existence of their (unique) solutions, of (global) convergence of sequences of approximate solutions in finite-dimensional spaces, etc. Even the variety of (often ad hoc) computational algorithms documents the absence of a general, inexpensive and robust one, working for a large class of experimental settings. Clearly this is a strong motivation for further research – maybe following the way predicted by [11]: i) overreaction to immature technology (naive euphoria), ii) frustration (cynicism), iii) true user benefits (realistic expectation), with certain asymptote of reality.

Acknowledgements

This work was supported by the project No. FAST-S-14-2346 of the specific university research at Brno University of Technology.

References

[1] André, S., Rémy, B., Pereira F.R., and Cella, N.: Hot wire method for the thermal characterization of materials: inverse problem application. *Engenharia Térmica* **4** (2003), 55–64.

[2] Axelsson, O.: A generalized conjugate gradient least square method. *Numerische Mathematik* **51** (1987), pp. 23–29.

[3] Barták, J., Herrmann, L., Lovicar, V., and Vejvoda, O.: *Partial Differential Equations of Evolution*. Ellis Horwood, Chichester, 1991.

[4] Bermúdez de Castro, A.: *Continuum thermomechanics*. Birkhäuser, Basel, 2005.

[5] Bochev, P.B., and Gunzburger, M.D. *Least-Squares Finite Element Methods*. Springer, New York, 2009.

[6] Černý, R., et al.: *Complex System of Methods for Directed Design and Assessment of Functional Properties of Building Materials: Assessment and Synthesis of Analytical Data and Construction of the System*. Czech Technical University, Prague, 2010.

[7] Cioranescu, D., and Donato, P.: *An Introduction to Homogenization*. Oxford University Press, Oxford, 1999.

[8] Colaço, M.J., Orlande, H.R.B., and Dulikravich, G.S.: Inverse and optimization methods in heat transfer. *Journal of the Brazilian Society of Mechanical Science and Engineering* **28** (2006), 1–24.

[9] Drchalová, J., and Černý, R.: Non-steady-state methods for determining the moisture diffusivity of porous materials. *International Communications in Heat and Mass Transfer* **25** (1998), 109–116.

[10] Feireisl, E., Petzeltová, H., and Simondon, F.: Admissible solutions for a class of nonlinear parabolic problems with non-negative data. *Proceedings of the Royal Society in Edinburgh*, Section A – Mathematics, **131** (2001), 857–883.

[11] Fish, J.: Multiscale computations: boom or bust. *IACM Expressions* **22** (2008), 4–7.

[12] Franců, J., and Svanstedt, N.: Some remarks on two-scale convergence and periodic unfolding. *Applications of Mathematics* **57** (2012), 359–375.

[13] Fučík, S., and Kufner, A.: *Nonlinear Differential Equations*. Elsevier, Amsterdam, 1980.

[14] Gawin, D., Pesavento, P., and Schrefler, B.A. Hygro-thermo-chemo-mechanical modelling of concrete at early ages and beyond. Part I: hydration and hygro-thermal phenomena. Part II: shrinkage and creep of concrete. *International Journal for Numerical Methods in Engineering* **67** (2006), 299331 and 332–363.

[15] Isakov, V.: *Inverse Problems for Partial Differential Equations*. Springer, New York, 2006.

[16] Jarošová, P., Štšastník, S., and Vala, J.: Identification of thermal conductivity of powdery insulation materials. *Advanced Materials Research* **2** (2014), 333–339.

[17] Jin, B., and Zou, J.: Inversion of Robin coefficient by a spectral stochastic finite element approach. *Journal of Computational Physics* **227** (2008), 3282-3306.

[18] Kailasam, S.K., Lacombe, L.C., and Glicksman, M.E.: Evaluation of the methods for calculating the concentration-dependent diffusivity in binary systems. *Metallurgical and Materials Transactions A* **30** (1999), 2605–2610.

[19] Kala, Z.: Sensitivity analysis of steel plane frames with initial imperfections. *Engineering Structures* **33** (2011), 2342–2349.

[20] Luikov, A.V.: *Heat and mass transfer in capillary-porous bodies*. Pergamon, Oxford, 1966.

[21] Maz'ya, V.G.: *Prostranstva S. L. Soboleva*. Izdateĭstvo Leningradskogo universiteta, Leningrad (St. Petersburg), 1985. (In Russian.)

[22] Narayanan V.A.B., and Zabaras, N.: Stochastic inverse heat conduction using a spectral approach. *International Journal for Numerical Methods in Engineering* **60** (2004), 1569–1593.

[23] Ng, K.W., and Rohanin, A.: Modified Fletcher - Reeves and Dai - Yuan conjugate gradient method for solving optimal control problem of monodomain model. *Applied Mathematics* **3** (2012), 864–872.

[24] Rincon, M.A., Límaco, J., and Liu, I.-S.: Existence and uniqueness of solutions of a nonlinear heat equation. *Tendências em Matemática Aplicada e Computacional* **6** (2005), 273–284.

[25] Roubíček, T.: *Nonlinear Partial Differential Equations with Applications*. Birkhäuser, Basel, 2005.

[26] Škramlik, J., Novotný, M., and Šuhajda, K.: The moisture in capillaries of building materials. *DPC Journal of Civil Engineering and Architecture* **2** (2012), 1536–1543.

[27] Šťastník, S., Vala, J., and Kmínová, H.: Identification of thermal technical characteristics from the measurement of non-stationary heat propagation in porous materials. *Kybernetika* **43** (2007), 561–576.

[28] Stenlund, H.: Three Methods for Solution of Concentration Dependent Diffusion Coefficient. *Visilab Signal Technologies*, 2004; available at www.visilab.fi/nonlinear_diffusion.pdf.

[29] Sun, J., and Zhang, J.: Global convergence of conjugate gradient method without line search. *Annals of Operations Research* **103** (2001), 161–173.

[30] 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.

[31] Vala, J.: On the computational identification of temperature-variable characteristics of heat transfer. *Proceedings of International Conference Applications of Mathematics (in honor of the 70th birthday of Karel Segeth)* in Prague, 215–224. Mathematical Institute AS CR, Prague, 2013.

[32] Vala, J., and Jarošová, P.: Identification of the capillary conduction coefficient from experimental data. *Forum Statisticum Slovacum* **9** (2013), 250–255.

[33] 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* **65** (2011), 111–114.

[34] Wan, W., and Zabaras, N.: A Bayesian approach to multiscale inverse problems using the sequential Monte Carlo method. *Inverse Problems* **27** (2011), 105004/1–25.

[35] Zabaras, N.: Inverse problems in heat transfer. In: Minkowycz, W.J., Sparrow, E.M., and Murthy, J.S. (Eds.), *Handbook on Numerical Heat Transfer*, Chap. 17. J. Wiley & Sons, Hoboken, 2004.

[36] Zhao, Z., Xie, O., Meng, Z., and You, L.: Determination of an unknown source in the heat equation by the method of Tikhonov regularization in Hilbert scales. Journal of Applied Mathematics and Physics **2** (2014), 10–17.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

ON RUNGE–KUTTA, COLLOCATION AND DISCONTINUOUS GALERKIN METHODS: MUTUAL CONNECTIONS AND RESULTING CONSEQUENCES TO THE ANALYSIS

Miloslav Vlasák, Filip Roskovec

Charles University in Prague, Faculty of Mathematics and Physics
Sokolovská 83, Prague 8, Czech Republic
vlasak@karlin.mff.cuni.cz, roskovec@gmail.com

Abstract

Discontinuous Galerkin (DG) methods are starting to be a very popular solver for stiff ODEs. To be able to prove some more subtle properties of DG methods it can be shown that the DG method is equivalent to a specific collocation method which is in turn equivalent to an even more specific implicit Runge–Kutta (RK) method. These equivalences provide us with another interesting view on the DG method and enable us to employ well known techniques developed already for any of these methods. Our aim will be proving the superconvergence property of the DG method in Radau quadrature nodes.

1. Introduction

The Discontinuous Galerkin (DG) method, either as space or time discretization, starts to play an important role in problems, where robust and highly efficient solvers are needed. Such a method enables a user to fully exploit adaptivity with higher order approximation and still it remains very robust.

The DG time discretizations are usually analyzed by similar means as the finite element method, see e.g. [9]. In such a way we obtain L^∞ estimates of order s for $s-1$ degree polynomial approximation. But numerical experiments often show better behaviour of the discrete solution in the nodes of Radau quadrature and especially in the endpoints of intervals. This phenomenon is usually called superconvergence.

Our aim will be showing some ideas how the possible analysis of superconvergence can be carried out in this case. In our approach we will focus on the Radau quadrature variant, where the integrals from the classical DG discretization are replaced by (right) Radau quadrature of suitable order, i.e. the quadrature preserves linear terms. As a first step we will show generally that this Radau quadrature variant of the DG method is equivalent to the well known Radau IIA Runge–Kutta (RK) method in Radau quadrature nodes. Then it is possible to use classical results developed for implicit RK methods to achieve superconvergence error estimates. In this part we will be mainly focused on stiff, linear problems.

2. ODE and discretizations

Let us assume the following ODE

$$\begin{aligned} y'(t) &= f(t, y(t)), \quad \forall t \in (0, T), \\ y(0) &= \alpha. \end{aligned} \tag{1}$$

Let us assume $t_m = m\tau$ be an equidistant partition of $(0, T)$ with time step τ. We introduce several one–step methods:

Runge–Kutta methods: Let $a_{i,j}$, b_i, c_i, $i, j = 1, \ldots, s$ be suitable coefficients. Then we call the sequence y^m satisfying $y^0 = \alpha$

$$\begin{aligned} g_i^m &= y^{m-1} + \tau \sum_{j=1}^{s} a_{i,j} f(t_{m-1} + \tau c_j, g_j^m), \quad \forall i = 1, \ldots, s, \\ y^m &= y^{m-1} + \tau \sum_{i=1}^{s} b_i f(t_{m-1} + \tau c_i, g_i^m) \end{aligned} \tag{2}$$

the RK solution of (1) approximating values $y(t_m)$.

Collocation methods: Let c_i, $i = 1, \ldots, s$ be suitable coefficients. Let $y^0 = \alpha$. In every step we construct polynomial p of degree at most s such that

$$\begin{aligned} p(t_{m-1}) &= y^{m-1}, \\ p'(t_{m-1} + \tau c_i) &= f(t_{m-1} + \tau c_i, p(t_{m-1} + \tau c_i)), \quad \forall i = 1, \ldots, s. \end{aligned} \tag{3}$$

Then we put $y^m = p(t_m)$. We call the resulting sequence the collocation solution of (1) approximating values $y(t_m)$.

Discontinuous Galerkin method: Let us denote $I_m = (t_{m-1}, t_m)$. Let us define the space

$$S^\tau = \{v \in L^2(0, T) : v|_{I_m} \in P^{s-1}\}, \tag{4}$$

where P^{s-1} is a space of polynomials of degree $s - 1$. Since the functions from S^τ are discontinuous in general in nodes of the partition, we denote the limit at nodes $v_\pm^m = v(t_m \pm)$ and the jump $\{v\}_m = v_+^m - v_-^m$. We call $u \in S^\tau$ the DG solution of (1) if $u_-^0 = \alpha$ and

$$\int_{I_m} u'(t)v(t)dt + \{u\}_{m-1} v_+^{m-1} = \int_{I_m} f(t, u(t))v(t)dt, \quad \forall v \in S^\tau, \forall m. \tag{5}$$

For comparison with previous methods we focus mainly on endpoints of intervals: $u_-^m \approx y(t_m)$.

Radau discontinuous Galerkin method: Let $r \in P^s$ be the (right) Radau polynomial, i.e. $r(0) = 1$, $r(1) = 0$, and for $s \geq 2$ let r be orthogonal to the polynomial space P^{s-2}. We can define the (right) Radau quadrature by

$$\int_0^1 F(t)dt \approx Q[F(t)] = \sum_{i=1}^{s} w_i F(x_i), \tag{6}$$

where x_i are roots of r and w_i are chosen in such a way that the resulting quadrature is accurate for polynomials P^{2s-2}. Similarly we can define the Radau quadrature $Q_m[.]$ and Radau polynomial r_m on I_m. We can define the Radau DG solution of (1) by replacing integrals by Radau quadratures in (5)

$$Q_m[u'(t)v(t)] + \{u\}_{m-1}v_+^{m-1} = Q_m[f(t,u(t))v(t)], \quad \forall v \in S^\tau, \forall m. \tag{7}$$

3. The Radau discontinuous Galerkin method is a Runge–Kutta method

In fact we want to show this in two steps. First, when the coefficients c_i of the collocation method are chosen as Radau quadrature nodes, then there is the following relation between the collocation polynomial p and Radau DG solution u

$$p(t) = u(t) - \{u\}_{m-1}r_m(t). \tag{8}$$

From this it follows that $p(t_{m-1} + \tau c_i) = u(t_{m-1} + \tau c_i)$, since $r_m(t_{m-1} + \tau c_i) = 0$. Since $c_s = 1$ we gain the correspondence of the collocation solution and the Radau DG solution at t_m, i.e. $y^m = p(t_m) = u_-^m$.

Lemma 1. *Let $p \in P^s$ be the collocation polynomial on I_m associated to the collocation method with coefficients c_i chosen as Radau quadrature nodes, $u \in P^{s-1}$ be the Radau DG solution on I_m and $r_m \in P^s$ be the (right) Radau polynomial on I_m. Then (8) holds.*

The proof follows the ideas from [7], where a similar case (continuous Galerkin and Gauss quadrature) is considered.

Proof. Let $u \in P^{s-1}$ be the Radau DG solution on I_m. We need to verify (3).

$$p(t_{m-1}) = u|_{I_m}(t_{m-1}) - \{u\}_{m-1}r_m(t_{m-1}) = u_+^{m-1} - (u_+^{m-1} - u_-^{m-1}) = u_-^{m-1}. \tag{9}$$

We denote $\ell_{m,i}$ the Lagrange interpolation basis function

$$\ell_{m,i}(t) = \prod_{j \neq i} \frac{t - t_{m-1} - \tau c_j}{\tau(c_i - c_j)}. \tag{10}$$

We can use $\ell_{m,i}$ as test functions in (7) and we obtain

$$w_i u'(t_{m-1} + \tau c_i) + \{u\}_{m-1}\ell_{m,i}(t_{m-1}) = w_i f(t_{m-1} + \tau c_i, u(t_{m-1} + \tau c_i)). \tag{11}$$

Now it is sufficient to show that $w_i r'_m(t_{m-1} + \tau c_i) = -\ell_{m,i}(t_{m-1})$. Since the product $\ell_{m,i}r'_m \in P^{2s-2}$, Radau quadrature for such a term is exact and we obtain

$$\begin{aligned} w_i r'_m(t_{m-1} + \tau c_i) &= Q_m[\ell_{m,i}(t)r'_m(t)] = \int_{I_m} \ell_{m,i}(t)r'_m(t)dt \qquad (12)\\ &= \ell_{m,i}(t_m)r_m(t_m) - \ell_{m,i}(t_{m-1})r_m(t_{m-1}) - \int_{I_m} \ell'_{m,i}(t)r_m(t)dt = -\ell_{m,i}(t_{m-1}), \end{aligned}$$

since $r_m(t_m) = 0$, $r_m(t_{m-1}) = 1$ and r_m is orthogonal to P^{s-2} on I_m. □

The second step is that every collocation method is equivalent to a suitable RK method.

Lemma 2. *Let the RK coefficients be chosen in the following way*

$$a_{i,j} = \int_0^{c_i} \ell_j(t)dt, \quad \forall i,j = 1,\dots,s, \tag{13}$$

$$b_i = \int_0^1 \ell_i(t)dt, \quad \forall i = 1,\dots,s, \tag{14}$$

where ℓ_i is the Lagrange interpolation basis function

$$\ell_i(t) = \prod_{j\neq i} \frac{t-c_j}{c_i-c_j}. \tag{15}$$

Then the values g_i^m, $i = 1,\dots,s$ and y^m produced by such a RK method are equal to the values $p(t_{m-1}+\tau c_i)$, $i = 1,\dots,s$ and y^m produced by the collocation method with the same coefficients c_i.

Proof. The proof can be found in [4] or [10]. □

Now from Lemma 1 and Lemma 2 we can see that the values produced by the Radau DG method in Radau quadrature nodes are equal to the values produced by a suitable RK method. Such a RK method is the well known Radau IIA RK method.

4. Analysis of the Radau IIA Runge–Kutta method

Now, we shall turn our focus on numerical analysis of linear problems

$$y'(t) = By(t) + f(t), \quad \forall t \in (0,T). \tag{16}$$

To do so, we shall focus on Dalquist's equation $y'(t) = \lambda y(t)$ with the exact solution $y(t_m) = e^{\tau\lambda}y(t_{m-1})$. For the purpose of analysis we assume $\mathrm{Re}\lambda \leq 0$, i.e. stable behaviour of the solution. Rewriting (2) in a vector–matrix formulation we obtain

$$g^m = y^{m-1}\mathbf{1} + \tau\lambda A g^m, \tag{17}$$

$$y^m = y^{m-1} + \tau\lambda b^T g^m, \tag{18}$$

where vector $\mathbf{1} = (1,\dots,1)^T$, matrix A and vectors b and g^m are formed by entries $a_{i,j}$, b_i and g_i^m. Eliminating inner stages g_i^m we obtain $y^m = R(\tau\lambda)y^{m-1}$, where

$$R(z) = 1 + zb^T(I-zA)^{-1}\mathbf{1} = \frac{\det(I-zA+zb^T\mathbf{1})}{\det(I-zA)}. \tag{19}$$

Following [6, Theorem 3.11] we can see that $R(z)$ is in the case of Radau IIA RK method the "subdiagonal" $(s-1,s)$–Padé approximation satisfying

$$\exp(z) - R(z) = O(z^{2s}). \tag{20}$$

Moreover, following results from [6, Chapter IV.4] we can conclude that $R(z)$ is also A–stable, i.e. $|R(z)| \leq 1$ for any $\mathrm{Re} z \leq 0$. We define the local error

$$\rho^m = y(t_m) - R(\tau\lambda)y(t_{m-1}) = (\exp(\tau\lambda) - R(\tau\lambda))y(t_{m-1}). \tag{21}$$

From (20) we can see that $|\rho^m| \leq C\tau^{2s} \max |y^{(2s)}|$. Then the error analysis follows easily from the stability of $R(z)$

$$|e^m| = |y(t_m) - y^m| = |\rho^m + R(\tau\lambda)e^{m-1}| \leq \dots \tag{22}$$
$$\dots \leq |R(\tau\lambda)|^m |e^0| + \sum_{i=1}^{m} |R(\tau\lambda)|^{m-i} |\rho^i| \leq |e^0| + T\frac{1}{\tau} \max_i |\rho^i|.$$

Assuming $e^0 = 0$ we gain global error estimate $e^m = O(\tau^{2s-1})$.

This result can be extended to the multidimensional case $y'(t) = By(t)$, where B is a matrix (or operator on Banach spaces in general) satisfying $\mathrm{Re}\,\langle By, y\rangle \leq 0$. The extension remains almost the same as the scalar case with the only difficulty arising from the question whether $\|R(\tau B)\| \leq 1$. The answer to this question is positive. The proof of the matrix case can be found in [6, Theorem 11.2]. The proof of the general operator case can be found in [8].

Now, we shall come back to equation (16). Unfortunately, the extension of previous results is not straightforward. According to [1] it is necessary to assume an additional assumption, otherwise the so-called order reduction phenomena occur.

Theorem 3. *Let y be the exact solution of (16) with operator B satisfying $Re\,\langle Bv, v\rangle \leq 0$. Let*

$$y^{(k)} \in Dom(B^{2s-k}), \quad \forall k = s+1, \dots, 2s. \tag{23}$$

Then the Radau IIA RK solution y^m converges with order $2s-1$, i.e. $\|y(t_m) - y^m\| = O(\tau^{2s-1})$.

Proof. The proof can be found in [1]. □

We should mention that in the previous case $f = 0$, the additional assumption (23) was automatically satisfied for solutions with bounded derivatives, i.e. $y^{(2s)}$ bounded. For ODEs coming from PDE discretizations in space assumption (23) can be reformulated as some kind of regularity and compatibility conditions on data. In usual context of weakly formulated PDEs these conditions are considered unnatural. Assumption (23) is necessary to achieve order $2s$, but it can be relaxed to obtain reduced orders, still higher than s. For assumptions needed to obtain order $s+1$ see e.g. [3].

Up to now we have analyzed the error in the nodes t_m only. From [2] and [5] follows the local error estimate for internal stages g_i^m of implicit RK methods. In the case of Radau IIA RK method we obtain order $s+1$ there. Together with global error estimates at t_m at least of order $s+1$ we get also global error estimates at Radau quadrature nodes of order $s+1$.

Acknowledgements

This work was supported by grant No. 13-00522S of the Czech Science Foundation. The first author is a junior researcher of the University centre for mathematical modelling, applied analysis and computational mathematics (Math MAC)

References

[1] Brenner, P., Crouzeix, M., and Thomée, V.: Single step methods for inhomogeneous linear differential equations in Banach spaces. RAIRO **16**(1) (1982), 5–26.

[2] Dekker, K.: Error bounds for the solution to the algebraic equation in Runge-Kutta methods. BIT **24** (1984), 347–356.

[3] Frank, R., Schneid, J., and Ueberhuber, C. W.: Order results for impicit Runge-Kutta methods applied to stiff systems. SIAM J. Numer. Anal. **22**(3) (1985), 515–534.

[4] Guillou, A. and Soulé, J. L.: La résolution numérique des problemes différentiels aux conditions initiales par des méthodes de collocation. R.I.R.O. **R-3** (1969), 17–44.

[5] Hairer, E., Norsett, S. P., and Wanner, G.: *Solving ordinary differential equations I, Nonstiff problems*. Springer Verlag, 2000.

[6] Hairer, E. and Wanner, G.: *Solving ordinary differential equations II, Stiff and differential-algebraic problems*. Springer Verlag, 2002.

[7] Hulme, B. L.: One step piecewise polynomial Galerkin methods for initial value problems. Math. Comp. **26** (1972), 415–424.

[8] von Neumann, J.: Eine Spektraltheorie für allgemeine Operatoren eines unitären Reumes. Math. Nachrichten. **4** (1951), 258–281.

[9] Thomée, V.: *Galerkin finite element methods for parabolic problems. 2nd revised and expanded ed.*. Springer Verlag, Berlin, 2006.

[10] Wright, K.: Some relationship between implicit Runge-Kutta collocation and Lanczos τ methods and their stability properties. BIT **10** (1969), 217–227.

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

A MODIFIED LIMITED-MEMORY BNS METHOD FOR UNCONSTRAINED MINIMIZATION DERIVED FROM THE CONJUGATE DIRECTIONS IDEA

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
[2]Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

Abstract

A modification of the limited-memory variable metric BNS method for large scale unconstrained optimization of the differentiable function $f : \mathcal{R}^N \to \mathcal{R}$ is considered, which consists in corrections (based on the idea of conjugate directions) of difference vectors for better satisfaction of the previous quasi-Newton conditions. In comparison with [11], more previous iterations can be utilized here. For quadratic objective functions, the improvement of convergence is the best one in some sense, all stored corrected difference vectors are conjugate and the quasi-Newton conditions with these vectors are satisfied. The algorithm is globally convergent for convex sufficiently smooth functions and our numerical experiments indicate its efficiency.

1. Introduction

The BNS method (see [3]) belongs to the variable metric (VM) or quasi-Newton (QN) line search iterative methods, see [9], [10]. 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 $x_{k+1} = x_k + s_k$, $s_k = t_k d_k$, $k \geq 0$, where usually the direction vector $d_k \in \mathcal{R}^N$ is $d_k = -H_k g_k$, matrix H_k is symmetric positive definite and a stepsize $t_k > 0$ is chosen in such a way that

$$f_{k+1} - f_k \leq \varepsilon_1 t_k g_k^T d_k, \qquad g_{k+1}^T d_k \geq \varepsilon_2 g_k^T d_k, \qquad k \geq 0 \tag{1}$$

(the Wolfe line search conditions, see [10]), where $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$; typically H_0 is a multiple of I and H_{k+1} is obtained from H_k by a VM update to satisfy the QN condition (see [9]) $H_{k+1} y_k = s_k$, $y_k = g_{k+1} - g_k$, $k \geq 0$.

Among VM methods, the BFGS method, see [9], [10], belongs to the most efficient; it preserves positive definite VM matrices and can be easily modified for large-scale optimization; the BNS and L-BFGS (see [5], [6] - subroutine PLIS) methods represent its well-known limited-memory adaptations. In every iteration, we repeatedly update an initial approximation of the inverse Hessian matrix $\zeta_k I$, $\zeta_k > 0$, by the BFGS method, using $\tilde{m} + 1$ couples of vectors $(s_{k-\tilde{m}}, y_{k-\tilde{m}}), \ldots, (s_k, y_k)$ successively (without forming approximations of the inverse Hessian matrix explicitly),

where $\tilde{m}=\min(k, m-1)$ and $m>1$ is a given parameter. In the case of the BNS method, the direction vector can be calculated without computing matrix H_+, see [3], by

$$-H_+g_+ = -\zeta g_+ - S\Big[U^{-T}\Big((D+\zeta Y^TY)U^{-1}S^Tg_+ - \zeta Y^Tg_+\Big)\Big] + Y\Big[\zeta U^{-1}S^Tg_+\Big], \quad (2)$$

(we often omit index k and replace indices $k+1, k-1$ by the symbols $+, -$ for simplification), where for $k \geq 0$ we denote $b_k = s_k^Ty_k$ and $S_k = [s_{k-\tilde{m}}, \ldots, s_k]$, $Y_k = [y_{k-\tilde{m}}, \ldots, y_k]$, $D_k = \mathrm{diag}[b_{k-\tilde{m}}, \ldots, b_k]$, $(U_k)_{i,j} = (S_k^TY_k)_{i,j}$ for $i \leq j$, $(U_k)_{i,j} = 0$ otherwise (an upper triangular matrix).

The concept of conjugacy plays an important role in optimization methods based on quadratic models, see e.g. [10]. We generalize the approach presented in [11], using vectors from more previous iterations to correct vectors s, y. Unlike [11], we use the BNS concept to calculate the direction vector, since then the increase in the number of required arithmetic operations can be relatively small. We use corrected quantities $\tilde{s}_k, \tilde{y}_k, \tilde{b}_k, \tilde{H}_k$, $k \geq 0$, defined by $\tilde{s}_0 = s_0$, $\tilde{y}_0 = y_0$, $\tilde{b}_0 = b_0$, $\tilde{H}_0 = I$ and

$$\tilde{s}_k = s_k + \hat{\underline{S}}_k\sigma_k, \quad \tilde{y}_k = y_k + \hat{\underline{Y}}_k\eta_k, \quad \tilde{b}_k = \tilde{s}_k^T\tilde{y}_k, \quad k > 0, \quad (3)$$

where matrices $\hat{\underline{S}}_k, \hat{\underline{Y}}_k$ contain some columns of $\tilde{\underline{S}}_k = [\tilde{s}_{k-\tilde{m}}, \ldots, \tilde{s}_{k-1}]$, $\tilde{\underline{Y}}_k = [\tilde{y}_{k-\tilde{m}}, \ldots, \tilde{y}_{k-1}]$ (we denote a set of indices i of these selected vectors $\tilde{s}_i, \tilde{y}_i$ by $\underline{\mathcal{I}}_k$ and $\mathcal{I}_k = \underline{\mathcal{I}}_k \cup \{k\}$; it can be $\underline{\mathcal{I}}_k = \emptyset$, in which case we set $\tilde{s}_k = s_k$, $\tilde{y}_k = y_k$, $\tilde{b}_k = b_k$) and σ_k, η_k are chosen in such a way that $\tilde{b}_k > 0$. Positive definite matrix $\tilde{H}_+$ is obtained by analogy to H_+, using corrected difference vectors. Note that matrix $\tilde{H}_+$ satisfies the QN condition $\tilde{H}_+\tilde{y} = \tilde{s}$ and that the direction vector $\tilde{d}_+ = -\tilde{H}_+g_+$ (and consequently, also an auxiliary vector $\tilde{Y}^T\tilde{H}_+g_+$) can be calculated by analogy to (2).

In Section 2 we investigate the BFGS update with corrected difference vectors

$$\ddot{H}_+ = (1/\tilde{b})\tilde{s}\tilde{s}^T + \tilde{V}\ddot{H}\tilde{V}^T, \quad \tilde{V} = I - (1/\tilde{b})\tilde{s}\tilde{y}^T, \quad (4)$$

where $\ddot{H}$ is any symmetric positive definite matrix, and discuss the choice of parameters σ, η. In Section 3 we show properties of $\ddot{H}_+$ and a role of unit stepsizes for quadratic functions. Application to the corrected BNS method 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. We will denote the Frobenius matrix norm by $\|\cdot\|_F$, the spectral matrix norm by $\|\cdot\|$ and the Euclidean vector norm by $|\cdot|$. Details and proofs of assertions can be found in [13].

2. Derivation of the method

Assuming that set $\underline{\mathcal{I}}$ is non-empty, we will investigate the influence of the correction parameters σ, η on properties of matrix $\ddot{H}_+$, given by (4). For our purpose, the satisfaction of the QN conditions $\ddot{H}_+\hat{Y} = \hat{S}$, $\hat{S} = [\hat{\underline{S}}, \tilde{s}]$, $\hat{Y} = [\hat{\underline{Y}}, \tilde{y}]$, plays a crucial role. We will suppose that the auxiliary QN conditions $\ddot{H}\hat{\underline{Y}} = \hat{\underline{S}}$ are satisfied (thus matrix $\hat{\underline{S}}^T\hat{\underline{Y}} = \hat{\underline{Y}}^T\ddot{H}\hat{\underline{Y}}$ is symmetric) and give a technique which guarantees the satisfaction of these conditions for a suitable matrix $\ddot{H}$. We denote $\ddot{B} = \ddot{H}^{-1}$, $\ddot{B}_+ = \ddot{H}_+^{-1}$, $\ddot{a} = \tilde{y}^T\ddot{H}\tilde{y}$.

The following lemma shows that, under some assumptions, conditions $\ddot{H}_+\tilde{y}_i=\tilde{s}_i$ are equivalent to the conjugacy of vector $\tilde{s}$ with vectors $\tilde{s}_i$ with respect to $\ddot{B}, \ddot{B}_+$, i.e. $\tilde{s}^T\ddot{B}\tilde{s}_i = \tilde{s}^T\ddot{B}_+\tilde{s}_i = 0$, $i \in \underline{\mathcal{I}}$, or $\underline{\hat{S}}^T\tilde{y} = \underline{\hat{Y}}^T\tilde{s} = 0$; these equations can be easily solved.

Lemma 1. *Let $\ddot{H}$ be any symmetric positive definite matrix satisfying $\ddot{H}\underline{\hat{Y}} = \underline{\hat{S}}$, matrix $\ddot{H}_+$ be given by (4) and let $\tilde{b} > 0$. Then $\ddot{H}_+$ is symmetric positive definite. If vectors $\tilde{s}$, $\ddot{H}\tilde{y}$ are linearly independent then $\ddot{H}_+\hat{Y}=\hat{S}$ if and only if $\underline{\hat{S}}^T\tilde{y}=\underline{\hat{Y}}^T\tilde{s}=0$.*

Lemma 2. *Let matrix $\underline{\hat{S}}^T\underline{\hat{Y}}$ be nonsingular. Then the unique solution (σ,η) to $\underline{\hat{S}}^T\tilde{y}=\underline{\hat{Y}}^T\tilde{s}=0$ is (σ^*,η^*), where $\sigma^* = -\left(\underline{\hat{Y}}^T\underline{\hat{S}}\right)^{-1}\underline{\hat{Y}}^T s$, $\eta^* = -\left(\underline{\hat{S}}^T\underline{\hat{Y}}\right)^{-1}\underline{\hat{S}}^T y$.*

Theorem 1 shows the variational characterizations of the choice $\sigma = \sigma^*$, $\eta = \eta^*$ also for non-quadratic functions, see also Theorem 3. Assumptions of Theorem 2 give our simple strategy for choosing matrices $\hat{S}, \hat{Y}$, which guarantees the satisfaction of the QN conditions $\tilde{H}_{k+1}\hat{Y}_k = \hat{S}_k$ and the corresponding auxiliary QN conditions.

Theorem 1. *Let $\tilde{b} > 0$ for $(\sigma,\eta) = (\sigma^*,\eta^*)$, matrix $\underline{\hat{S}}^T\underline{\hat{Y}}$ be nonsingular, matrices $\ddot{H}, \ddot{H}_+$ satisfy the same assumptions as in Lemma 1 and define $\mathcal{S}(\underline{\hat{S}},\underline{\hat{Y}}) = \{(\sigma,\eta):$ $\underline{\hat{S}}^T\tilde{y}=\underline{\hat{Y}}^T\tilde{s}\}$. If we have any symmetric positive definite matrix $\ddot{G}$ such that $\ddot{G}\underline{\hat{S}}=\underline{\hat{Y}}$ and $\ddot{G}(s+\underline{\hat{S}}\ddot{\sigma}) = y + \underline{\hat{Y}}\ddot{\eta}$ for some $(\ddot{\sigma},\ddot{\eta}) \in \mathcal{S}(\underline{\hat{S}},\underline{\hat{Y}})$, then within $(\sigma,\eta) \in \mathcal{S}(\underline{\hat{S}},\underline{\hat{Y}})$, values $\|\ddot{G}^{1/2}\ddot{H}_+\ddot{G}^{1/2} - I\|_F^2$ and $\tilde{b}$ are minimized by the choice $\sigma = \sigma^*$, $\eta = \eta^*$.*

Theorem 2. *Suppose that each set $\underline{\mathcal{I}}_k$, $k > 0$, is chosen in such a way that $\underline{\mathcal{I}}_k \subset \mathcal{I}_{k-1}$, $\tilde{b}_k > 0$ and $\underline{\hat{S}}_k^T\tilde{y}_k = \underline{\hat{Y}}_k^T\tilde{s}_k = 0$ in case that $\underline{\mathcal{I}}_k \neq \emptyset$. Then for $k > 0$: $\tilde{s}_i^T\tilde{y}_j = \tilde{y}_i^T\tilde{s}_j = 0$, $i \in \underline{\mathcal{I}}_k$, $i < j \leq k$, the QN conditions $\tilde{H}_{k+1}\hat{Y}_k = \hat{S}_k$ are satisfied and the auxiliary QN conditions $\ddot{H}_k\underline{\hat{Y}}_k = \underline{\hat{S}}_k$ are satisfied for $\underline{\mathcal{I}}_k \neq \emptyset$ with those matrices $\ddot{H}_k$ by the BFGS updating (4) of which we get matrices $\tilde{H}_{k+1} = \ddot{H}_{k+1}$.*

The first assertion of the theorem implies that all matrices $\hat{S}^T\hat{Y}$ are diagonal and thus many results can be simplified. E.g. vectors σ^*, η^* have components $-s^T\tilde{y}_i/\tilde{b}_i, -\tilde{s}_i^T y/\tilde{b}_i$, $i \in \underline{\mathcal{I}}$, and a damage of the QN condition with non-corrected vectors caused by our corrections and value $\tilde{b}$ for $(\sigma,\eta) = (\sigma^*,\eta^*)$ can be written:

$$(\ddot{H}_+y-s)^T\ddot{B}_+(\ddot{H}_+y-s) = b\sum\nolimits_{i\in\underline{\mathcal{I}}}(\tilde{s}_i^Ty - s^T\tilde{y}_i)^2/(b\tilde{b}_i),\ \tilde{b} = b - \sum\nolimits_{i\in\underline{\mathcal{I}}} s^T\tilde{y}_i\,\tilde{s}_i^Ty/\tilde{b}_i. \quad (5)$$

3. Results for quadratic functions

Here we suppose that f is a quadratic function with a symmetric positive definite Hessian G and $\eta_k = \sigma_k$, $k > 0$, which yields $\tilde{y}_k = G\tilde{s}_k$, as for non-corrected vectors. The following lemma and theorem show that for the choice $\sigma = \sigma^*$, the improvement of convergence is the best in some sense for linearly independent direction vectors.

Lemma 3. *Let f be a quadratic function $f(x) = \frac{1}{2}(x-\bar{x})^TG(x-\bar{x})$, $\bar{x} \in \mathcal{R}^N$, with a symmetric positive definite matrix G and all columns of $[\underline{\tilde{S}}, s]$ be linearly independent. Then for any selection of $\underline{\hat{S}}$, $\underline{\hat{Y}}$ from $\underline{\tilde{S}}$, $\underline{\tilde{Y}}$, matrix $\underline{\hat{S}}^T\underline{\hat{Y}}$ is symmetric positive definite, value σ^* is well defined by Lemma 2 and $\tilde{b} > 0$ for any $\sigma = \eta$.*

Theorem 3. *Let $\ddot{H}$ be any symmetric positive definite matrix satisfying $\ddot{H}\underline{\hat{Y}} = \underline{\hat{S}}$ and suppose that $\sigma = \eta$ and that the assumptions of Lemma 3 are satisfied. Then $\tilde{b} > 0$ and the choice $\sigma = \sigma^*$ implies $\ddot{H}_+ y = s$ and minimizes values $\tilde{b}$ and $\|G^{1/2}\ddot{H}_+ G^{1/2} - I\|_F$ as a function of σ, where matrix $\ddot{H}_+$ is defined by update (4) of $\ddot{H}$.*

Theorem 4 describes a situation when the case $\sigma = \sigma^*$ occurs in all iterations. Comparing these results with those given in [11] (Theorem 3.2) for the unit stepsizes, we see that they are similar. Theorem 5 gives an interesting explanation.

Theorem 4. *Let the assumptions of Lemma 3 be satisfied with the columns of every matrix $[\underline{\tilde{S}}_k, s_k]$ linearly independent and let always $\underline{\hat{S}}_k = \underline{\tilde{S}}_k$, $\underline{\hat{Y}}_k = \underline{\tilde{Y}}_k$, $\sigma_k = \sigma_k^*$, $k > 0$. Then all columns of $\tilde{S}_k$ are G-conjugate, i.e. matrices $\tilde{S}_k^T \tilde{Y}_k$ are diagonal and all QN conditions $\tilde{H}_{k+1}\tilde{Y}_k = \tilde{S}_k$, $\ddot{H}_k \underline{\tilde{Y}}_k = \underline{\tilde{S}}_k$, are satisfied, with those matrices $\ddot{H}_k$ by the BFGS updating (4) of which we get matrices $\tilde{H}_{k+1} = \ddot{H}_{k+1}$, $k > 0$.*

Theorem 5. *Let $\tilde{H}, \tilde{H}_+$ be symmetric positive definite matrices satisfying $\tilde{H}\underline{\hat{Y}} = \underline{\hat{S}}$, $\tilde{H}_+ \underline{\hat{Y}} = \underline{\hat{S}}$, $d = -\tilde{H}g$, $d_+ = -\tilde{H}_+ g_+$, $\sigma = \eta$, $t = 1$ and the assumptions of Lemma 3 be satisfied. Then $\underline{\hat{S}}^T y_+ = \underline{\hat{Y}}^T s_+ = 0$, i.e. all columns of $\underline{\hat{S}}$ are G-conjugate with s_+.*

4. Implementation

It is important to say that not all vectors $\tilde{s}_i, \tilde{y}_i$, $i \in \underline{\mathcal{I}}$, are suitable as correction vectors. Principally, we do not use vectors $\tilde{s}_i, \tilde{y}_i$, $k - \tilde{m} \le i < k$, $k > 0$, for the correction process (i.e. we decide that $i \notin \underline{\mathcal{I}}_k$) if $\tilde{b}_k \le 0$, if resultant values $b_k/\tilde{b}_k$, $b_k/\ddot{a}_k$, $b_k/\tilde{s}_k^T \ddot{B}_k \tilde{s}_k$ or $(\tilde{s}_i^T y_k - s_k^T \tilde{y}_i)^2/(b_k \tilde{b}_i)$ (see (5)) are too great or if $i \notin \mathcal{I}_{k-1}$, see Theorem 2.

In order to prove global convergence, we also exclude index i from $\underline{\mathcal{I}}$ if values $|\tilde{s}_i|/|s_i|$, $|\tilde{y}_i|/|y_i|$ are too great. Note that these values were rarely greater than 50 in our numerical experiments with $N = 5000$. Further, Theorem 5 indicates that an influence of the second and further correction vectors can be small. Thus for $i < k-1$, $k > 0$, we should not correct if a benefit of corrections is negligible, see [13] for details.

Algorithm 1 (without indices elimination details and stopping criteria)

Data: A number $m > 1$ of VM updates per iteration, line search and correction parameters and a maximum number of correction vectors $n \in [0, m-1]$.

Step 0: Initiation. Choose starting point $x_0 \in \mathcal{R}^N$, define starting matrix $\tilde{H}_0 = I$ and direction vector $d_0 = -g_0$ and initiate iteration counter k to zero.

Step 1: Line search. Set $\tilde{m} = \min(k, m-1)$. Compute $x_{k+1} = x_k + t_k d_k$, where t_k satisfies (1), $g_{k+1} = \nabla f(x_{k+1})$, $s_k = t_k d_k$, $y_k = g_{k+1} - g_k$, $b_k = s_k^T y_k$, $\zeta_k = b_k / y_k^T y_k$. If $k = 0$ set $\tilde{s}_k = s_k$, $\tilde{y}_k = y_k$, $\tilde{b}_k = \tilde{s}_k^T \tilde{y}_k$, $\mathcal{I}_k = \{0\}$, $\tilde{S}_k = [\tilde{s}_k]$, $\tilde{Y}_k = [\tilde{y}_k]$, $\tilde{S}_k^T \tilde{Y}_k = [\tilde{s}_k^T \tilde{y}_k]$, $\tilde{Y}_k^T \tilde{Y}_k = [\tilde{y}_k^T \tilde{y}_k]$, compute $\tilde{S}_k^T g_{k+1}, \tilde{Y}_k^T g_{k+1}$ and go to Step 5. Compute $\underline{\tilde{S}}_k^T g_{k+1}$, $\underline{\tilde{Y}}_k^T g_{k+1}$, $\underline{\tilde{Y}}_k^T s_k = -t_k \underline{\tilde{Y}}_k^T \tilde{H}_k g_k$, $\underline{\tilde{S}}_k^T y_k = \underline{\tilde{S}}_k^T g_{k+1} - \underline{\tilde{S}}_k^T g_k$ and $\underline{\tilde{Y}}_k^T y_k = \underline{\tilde{Y}}_k^T g_{k+1} - \underline{\tilde{Y}}_k^T g_k$.

Step 2: Elimination of indices. Set $\underline{\mathcal{I}}_k = \{i \in \mathcal{I}_{k-1} : i \ge k - n\}$. Eliminate non-suitable indices from $\underline{\mathcal{I}}_k$. If $\underline{\mathcal{I}}_k = \emptyset$ go to Step 4, otherwise form matrices $\underline{\hat{S}}_k, \underline{\hat{Y}}_k$.

Step 3: Correction. Compute $(\sigma_k)_i = -s_k^T \tilde{y}_i / \tilde{b}_i$, $(\eta_k)_i = -\tilde{s}_i^T y_k / \tilde{b}_i$ for $i \in \underline{\mathcal{I}}_k$ and $\tilde{s}_k, \tilde{y}_k, \tilde{b}_k$ by (3). Set $\mathcal{I}_k = \underline{\mathcal{I}}_k \cup \{k\}$.

Step 4: Matrix updating. Similarly as in [3] form matrices $\tilde{S}_k$, $\tilde{Y}_k$, $\tilde{S}_k^T\tilde{Y}_k$, $\tilde{Y}_k^T\tilde{Y}_k$.

Step 5: Direction vector. Compute $d_{k+1} = -\tilde{H}_{k+1}g_{k+1}$ by the BNS method with vectors $(\tilde{s}_{k-\tilde{m}}, \tilde{y}_{k-\tilde{m}}), \ldots, (\tilde{s}_k, \tilde{y}_k)$ and an auxiliary vector $\tilde{Y}_k\tilde{H}_{k+1}g_{k+1}$, see Section 1. Set $k:=k+1$. If $k \geq m$ delete the first column of $\tilde{S}_{k-1}$, $\tilde{Y}_{k-1}$ and the first row and column of $\tilde{S}_{k-1}^T\tilde{Y}_{k-1}$, $\tilde{Y}_{k-1}^T\tilde{Y}_{k-1}$ to form matrices $\underline{\tilde{S}}_k, \underline{\tilde{Y}}_k, \underline{\tilde{S}}_k^T\underline{\tilde{Y}}_k, \underline{\tilde{Y}}_k^T\underline{\tilde{Y}}_k$. Go to Step 1.

5. Global convergence

Assumption 1. *The objective function $f : \mathcal{R}^N \to \mathcal{R}$ is bounded from below and uniformly convex with bounded second-order derivatives (i.e. $0 < \underline{G} \leq \underline{\lambda}(G(x)) \leq \overline{\lambda}(G(x)) \leq \overline{G} < \infty$, $x \in \mathcal{R}^N$, where $\underline{\lambda}(G(x))$ and $\overline{\lambda}(G(x))$ are the lowest and the greatest eigenvalues of the Hessian matrix $G(x)$).*

Theorem 6. *If the objective function f satisfies Assumption 1, Algorithm 4.1 generates a sequence $\{g_k\}$ that satisfies $\lim_{k\to\infty}|g_k|=0$ or terminates with $g_k=0$ for some k.*

6. Numerical experiments

We demonstrate the influence of vector corrections on the number of evaluations and computational time, using the following collections of test problems: **Test 11** [8] (55 modified problems from CUTE collection [2] with $N = 1000 - 5000$, computed repeatedly ten times), test from [1], termed **Test 12** here, 73 problems, $N = 10000$, **Test 25** [7] (68 problems), $N = 10000$. The source texts and the corresponding reports can be downloaded from `camo.ici.ro/neculai/ansoft.htm` (Test 12) and `www.cs.cas.cz/luksan/test.html` (Tests 11 and 25).

Table 1 contains the total number of function evaluations (NFV) and the total computational time in seconds (Time) for the following limited-memory methods: L-BFGS [5], method from [11] and new Algorithm 1 for $n = 2, 4$, all implemented in the system UFO [12]. We have used $m=5$ and the final precision $\|g(x^\star)\|_\infty \leq 10^{-6}$.

Method	Test 11		Test 12		Test 25	
	NFV	Time	NFV	Time	NFV	Time
L-BFGS	80539	10.361	119338	50.88	502966	429.01
Alg. 4.1 in [11]	64395	9.614	67619	32.61	325441	318.71
Alg. 1, $n=2$	62770	8.795	67372	31.06	302908	302.62
Alg. 1, $n=4$	64127	8.977	66403	30.77	308847	298.05

Table 1: Comparison of the selected methods

For Test 25, we also compare these methods by using performance profiles [4]. Value $\rho_M(0)$ is the percentage of the test problems for which method M is the best and value $\rho_M(\tau)$ for τ large enough is the percentage of the problems that method M can solve. Performance profiles show the relative efficiency and reliability of the methods: the higher is the particular curve, the better is the corresponding method.

Acknowledgements

This work was supported by the Grant Agency of the Czech Republic, project No. 13-06684S, and the Institute of Computer Science of the AS CR (RVO: 67985807).

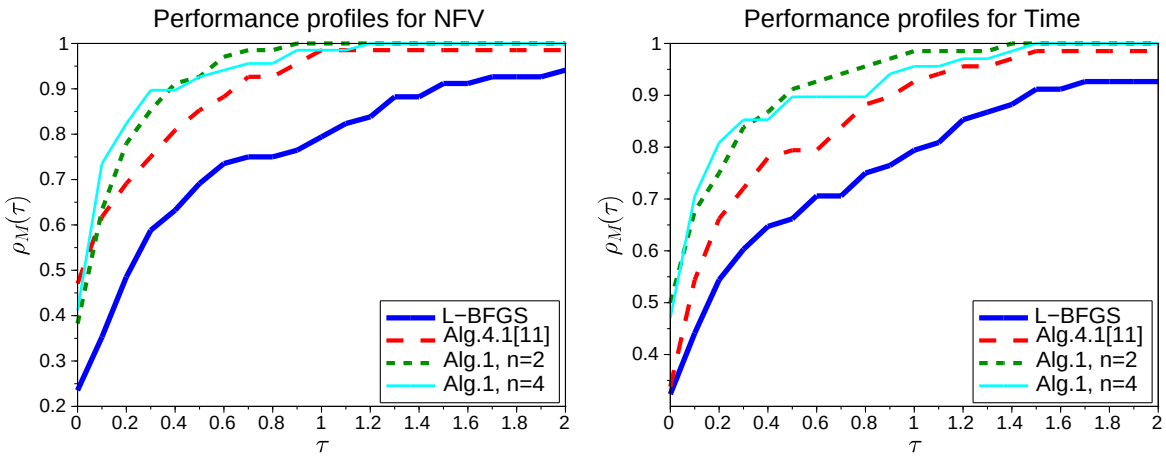


Figure 1: Comparison of $\rho_M(\tau)$ for Test 25 (68 problems) and various methods.

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] Byrd, R. H., Nocedal, J., and Schnabel, R. B.: Representation of quasi-Newton matrices and their use in limited memory methods. Math. Program. **63** (1994), 129–156.

[4] Dolan, E. D. and Moré, J. J.: Benchmarking optimization software with performance profiles. Math. Program. **91** (2002), 201–213.

[5] Liu, D. C. and Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45** (1989), 503–528.

[6] Lukšan, L., Matonoha, C., and Vlček, J.: Algorithm 896: LSA – Algorithms for Large-Scale Optimization. ACM Trans. Math. Software **36** (2009), 16:1–16:29.

[7] Lukšan, L., Matonoha, C., and Vlček, J.: Sparse test problems for unconstrained optimization. Report V-1064, ICS AS CR, Prague, 2010.

[8] Lukšan, L., Matonoha, C., and Vlček, J.: Modified CUTE problems for sparse unconstrained optimization. Report V-1081, ICS AS CR, Prague, 2010.

[9] Lukšan, L. and Spedicato, E.: Variable metric methods for unconstrained optimization and nonlinear least squares. J. Comput. Appl. Math. **124** (2000), 61–95.

[10] Nocedal, J. and Wright S. J.: *Numerical optimization*. Springer-Verlag, New York, 1999.

[11] Vlček, J. and Lukšan, L.: A conjugate directions approach to improve the limited-memory BFGS method. Appl. Math. Comput. **219** (2012), 800–809.

[12] Lukšan, L., Tůma, M., Vlček, J., Ramešová, N., Šiška, M., Hartman, J., and Matonoha, C.: UFO 2013. Interactive System for Universal Functional Optimization. Report V-1191, ICS AS CR, Prague, 2013, `www.cs.cas.cz/luksan/ufo.html`.

[13] Vlček, J. and Lukšan, L.: A modified limited-memory BNS method for unconstrained minimization based on the conjugate directions idea. Report V-1203, ICS ASCR, Prague, 2014, `www.cs.cas.cz/vlcek/reports.html`.

List of Participants

Monika Balázsová, b.moncsi@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Stanislav Bartoň, barton@mendelu.cz
Ústav techniky a automobilové dopravy, Agronomická fakulta, Mendelova Univerzita v Brně

Petr Bauer, bauer@it.cas.cz
Ústav termomechaniky AV ČR, v. v. i., Praha

Václav Bittner, vbittner@seznam.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

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 v Bratislave, Slovenská republika

Vít Dolejší, dolejsi@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Jiří Eckstein, jiri.eckstein@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Jiří Egermaier, jirieggy@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Cyril Fischer, fischerc@itam.cas.cz
Ústav teoretické a aplikované mechaniky AV ČR, v. v. i., Praha

Martin Hanek, martin-hanek@centrum.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Hana Horníková, hhornik@students.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Jaroslav Hron, hron@karlin.mff.cuni.cz
Matematický ústav Univerzity Karlovy, Matematicko-fyzikální fakulta UK v Praze

Petra Jarošová, jarosova.p@fce.vutbr.cz
Ústav pozemního stavitelství, Fakulta stavební VUT v Brně

Pavel Karban, karban@kte.zcu.cz
Katedra teoretické elektrotechniky, Fakulta elektrotechnická, Západočeská univerzita v Plzni

Radka Keslerová, keslerov@marian.fsik.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Vladimír Klement, wlada@post.cz
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská ČVUT v Praze

Roman Knobloch, roman.knobloch@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Lukáš Korous, korous@rice.zcu.cz
Katedra teoretické elektrotechniky, Fakulta elektrotechnická, Západočeská univerzita v Plzni

Tomáš Kozubek, tomas.kozubek@vsb.cz
Národní superpočítačové centrum IT4Innovations, Vysoká škola báňská - Technická univerzita Ostrava

Jiří Krček, jiri.krcek@vsb.cz
Katedra matematiky a deskriptivní geometrie, Vysoká škola báňská – Technická univerzita Ostrava

Michal Křížek, krizek@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Jaroslav Kruis, kruis@fsv.cvut.cz
Katedra mechaniky, Fakulta stavební ČVUT v Praze

Lukáš Krupička, luk.krupicka@gmail.com
Katedra matematiky, Fakulta stavební ČVUT v Praze

Václav Kučera, vaclav.kucera@email.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Pavel Kůs, pkus@rice.zcu.cz
Katedra teoretické elektrotechniky, Fakulta elektrotechnická, Západočeská univerzita v Plzni

Daniel Langr, daniel.langr@fit.cvut.cz
Katedra počítačových systémů, Fakulta informačních technologií ČVUT v Praze

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

Ctirad Matonoha, matonoha@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Karel Mikeš, karel.mikes.1@fsv.cvut.cz
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 informatiky a aplikované matematiky, Moravská vysoká škola Olomouc

Štěpán Papáček, spapacek@frov.jcu.cz
Škola komplexních systémů, Fakulta rybářství a ochrany vod, Jihočeská univerzita v Českých Budějovicích

Petr Pařík, parik@it.cas.cz
Ústav termomechaniky AV ČR, v. v. i., Praha

Jan Pech, jpech@it.cas.cz
Ústav termomechaniky AV ČR, v. v. i., Praha

Michal Petřík, pe3k.michal@gmail.com
Ústav techniky a automobilové dopravy, Agronomická fakulta, Mendelova Univerzita v Brně

Lukáš Pospíšil, lukas.pospisil@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, VŠB - Technická univerzita Ostrava

Jan Přikryl, prikryl@utia.cas.cz
Ústav teorie informace a automatizace AV ČR, v. v. i., Praha

Petr Přikryl, prikryl@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Petra Rozehnalová, rozehnalova.petra@gmail.com
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Vojtěch Rybář, rybar@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Karel Segeth, segeth@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Ivan Šimeček, ivan.simecek@fit.cvut.cz
Katedra počítačových systémů, Fakulta informačních technologií ČVUT v Praze

Martina Šimůnková, martina.simunkova@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Jakub Šístek, sistek@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Ilona Škarydová, ilona.skarydova@tul.cz
Ústav nových technologií a aplikované informatiky, Fakulta mechatroniky, informatiky a mezioborových studií, Technická univerzita v Liberci

Pavel Šolín, pavel@nclab.com
Department of Mathematics and Statistics, University of Nevada, Reno, USA; Ústav termomechaniky AV ČR, v. v. i., Praha

Ivan Soukup, ivan.soukup@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Jan Šourek, sourekj@students.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Petr Sváček, Petr.Svacek@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Eva Turnerová, turnerov@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Jiří Vala, Vala.J@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Tomáš Vejchodský, vejchod@math.cas.cz
University of Oxford, GB; Matematický ústav AV ČR, v. v. i., Praha

Miloslav Vlasák, vlasak@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Jan Vlček, vlcek@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Jan Zítko, zitko@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Andrej Živčák, andrej.zivcak@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Andrea Živčáková, andrea.zivcakova@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze