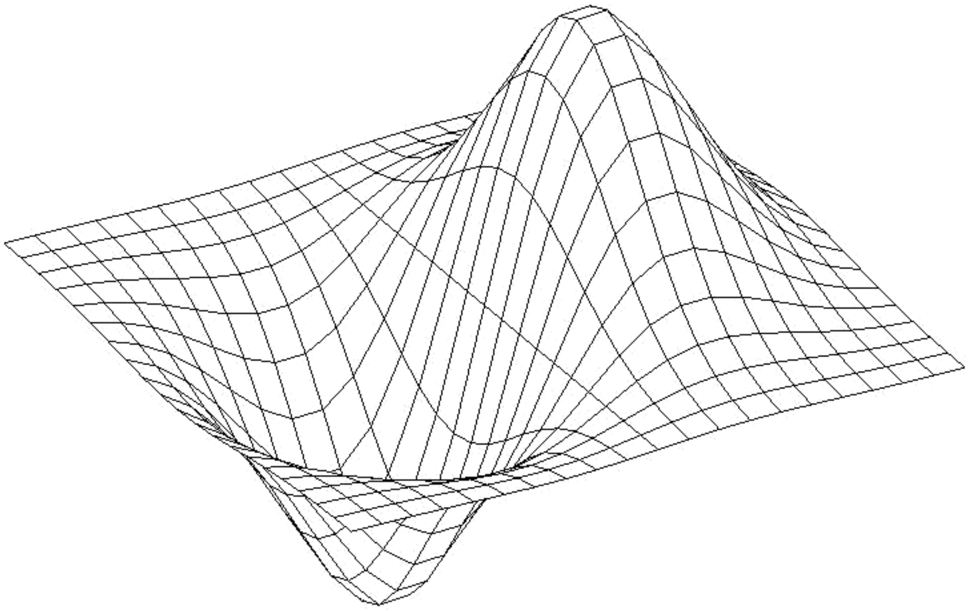
Faculty of Mathematics and Physics, Charles University

Institute of Computer Science, Academy of Sciences

SNA'10

Seminar on Numerical Analysis

Modelling and Simulation
of Challenging Engineering Problems



Winter School

High-Performance and Parallel Computers,
Programming Technologies & Numerical Linear Algebra

Nové Hrady, January 18 – 22, 2010

Programme committee:

Radim Blaheta	Institute of Geonics AS CR, Ostrava
Zdeněk Dostál	VŠB-Technical University, Ostrava
Ivo Marek	Czech Technical University, Prague
Zdeněk Strakoš	Charles University, Prague
Miroslav Rozložník	Institute of Computer Science AS CR, Prague

Organizing committee:

Vlasta Císařová	Conference Center, Nové Hrady
Miroslav Rozložník	Institute of Computer Science AS CR, Prague
Jurjen Duintjer Tebbens	Institute of Computer Science AS CR, Prague
Štěpán Papáček	Institute of Physical Biology, Nové Hrady
Petr Tichý	Institute of Computer Science AS CR, Prague
Miroslav Tůma	Institute of Computer Science AS CR, Prague

Conference secretary:

Hana Bílková	Institute of Computer Science AS CR, Prague

Institute of Computer Science AS CR
ISBN 978-80-87136-07-2

Preface

This series of winter schools started in 2005 (it was coupled with the Seminar on Numerical Analysis going back to 2003). Originally it was meant for a limited group of people around one research project. We are very happy to see how the seed planted a few years ago has grown into a healthy event which attracts more and more students as well as distinguished researchers from the community. Moreover, the scope now covers a large area from modelling through discretization and numerical analysis to computational methods, including optimization and various applications. This is extremely important in particular at the time of overspecialization and perhaps even fragmentation of research fields. Moreover, it demonstrates that there is no division between theory and applications; they must stay together and benefit from each other.

This year our school will become truly international, and the complementary program of contributed presentations and posters represents itself a small conference. With growing number of participants, we will have to reconsider the format of the seminar part in the future, in order to accommodate the number of contributions. We wish to emphasize the school part, as well as give an opportunity for other presentations, using perhaps more than now the convenient form of poster sessions. We wish also to keep our Moravian and Silesian / Bohemian biannual duality, with the Academy of Sciences (ÚGN and ÚI), VSB-TU Ostrava, Czech Technical University (Faculty of Civil Engineering) and Charles University (Faculty of Mathematics and Physics) as the main organizers. We look forward to the SNA hosted in the very friendly environment in the beautiful town Nové Hrady, as well as to the developments in the forthcoming years.

On behalf of the Programme and Organizing Committee of SNA 2010

Zdeněk Strakoš, Miro Rozložník

Contents

Winter school lectures

V. Dolejší
Solution of linear algebra systems arising from the compressible Navier-Stokes equations

P. Jiránek, Z. Strakoš, M. Vohralík
A posteriori error estimates and stopping criteria for iterative solvers

A. Miedlar
Adaptive methods for PDE-eigenvalue problems

T. Levitner, Š. Timr, J. Urban, J. Vaněk, I. Khrytankova, D. Štys, D. Štys jr., P. Císař, T. Náhlík
Cell monolayer development as stochastic causal system governed by underlying non-linear dynamic

P. Tichý
On efficient numerical approximation of the scattering amplitude

Multiscale modelling with Schwarz iterative methods

O. Axelsson, R. Blaheta, V. Sokol

Institute of Geonics AS CR, v.v.i., Ostrava, Czech Republic

1 Introduction

This paper considers the problem of finite element analysis of heterogeneous materials, especially the case when samples with deterministically or stochastically given microstructure with scale ε comparable with the mesh size h are tested numerically to evaluate the macroscale behaviour. This analysis is computationally expensive because a fine dicsretization is used to capture the (heterogeneity of) the microstructure and the solved problem involves the coefficient jumps. As a consequence, the algebraic systems arising from the discretisation become ill–conditioned and convergence of standard iterative methods deteriorates with oscillations of coefficients.

An illustration of this behaviour can be found in [6], where we perform numerical testing (homogenization) of mechanical properties of coal–resin geocomposites with the aid of GEM software [7] using Schwarz–type parallel solvers.

In this paper, we shall consider another example of numerical testing of Darcy flow in heterogeneous media. The microstructure in this example is generated stochastically, which allows to investigate the influence of heterogeneity onto the convergence behaviour of iterative solvers more systematically. The example was considered already in [5], where we investigated mixed formulation of the problem and iterative solution by MINRES with an augmented–Lagrangian–Schwarz preconditioning. See also [2], where this convergence behaviour is investigated theoretically.

Here we focus on standard (primal) formulation of the Darcy flow problem and behaviour of Schwarz–type methods in the case of heterogeneous media.

2 Model problem and strip-like domain decomposition

Let us consider a model problem of saturated Darcy flow through a sample area (volume) $\Omega = \langle\, 0,\, 1\rangle \times \langle\, 0,\, 1\rangle$. The flow is described by the equations

$$\begin{aligned} \nabla \cdot v &= f, \quad v = -k\nabla u \qquad \text{in} \quad \Omega\,, && (1)\\ v \cdot n &= 0 \quad \text{on } \Gamma_v = \{x \in \partial\Omega :\ x_2 = 0 \text{ or } x_2 = 1\}\,, && (2)\\ u = 1 &\quad \text{on} \quad \Gamma_{u1} = \{x \in \partial\Omega :\ x_1 = 0\}\,, && (3)\\ u = 0 &\quad \text{on} \quad \Gamma_{u2} = \{x \in \partial\Omega :\ x_1 = 1\}\,. && (4) \end{aligned}$$

The stochastic character is given by the permeability coefficient k . We shall assume that it is a random field such that

$$z(x) = \ln k(x) \in N(0, \sigma^2) \quad \text{for any} \quad x \in \Omega\,,$$

which means that $z(x)$ has normal distribution with the mean $\mu = 0$ and the variance σ^2 . This variance will be a parameter for testing robustness of iterative solvers. We could also require a correlation for the random field $k(x)$, see [5]. But for the purpose of testing the solvers, we skip this requirement, which implementation needs an extra effort.

We assume that the problem (1) – (4) is discretized by means of linear triangular finite elements. For simplicity, we assume that the domain Ω is divided into rectangular $h \times h$ elements which are consequently divided by diagonals into triangular elements. The continuous, piecewise linear functions on the given triangulation $\mathcal{T}_h$ and zero on $\Gamma_{u1} \cup \Gamma_{u2}$ create the FE space V_h. Using the standard nodal basis in V_h, we can derive the FE system

$$Au = b. \tag{5}$$

Schwarz-type methods use overlapping decomposition of the domain Ω. In this paper, we consider division of Ω into strips Ω_i^0 aligned with the FE grid. These strips are further extended to Ω_i^δ (symmetrically, if possible) by one or more layer of fine grid elements, see Fig. 1.

Note that strip–like subdomains are advantageous for efficient solution of subproblems by direct solvers (using the advantage of small bandwidth) and simple communication pattern given by at most two neighbours for each subdomain. Strips can be generalized to layers in 3D. They are very natural for FE software oriented to structured grids (as GEM software [7]) but analogous decomposition can be also defined for unstructured grids [8].

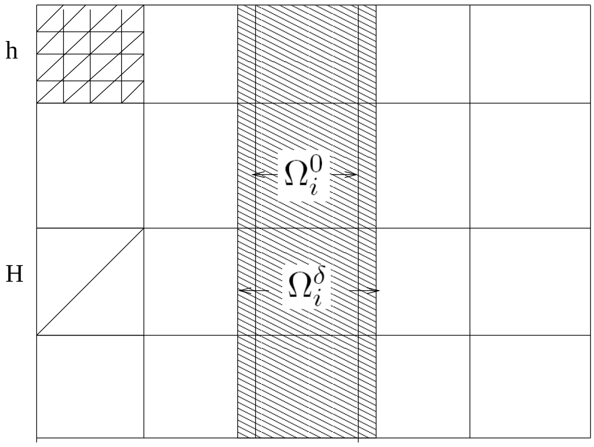


Figure 1: Strip-like decomposition. Note that Ω_i^δ has a minimal symmetric h-extension here leading to overlap $\delta = 2h$, 2h-extension is here maximal not increasing number of neighbours.

Using the overlapping decomposition Ω_k^δ and in case of need also a coarse triangulation $\mathcal{T}_H$ with the corresponding FE space $V_H \equiv V_0$, it is possible to construct many variants of Schwarz–type methods, see [4], [11]. These methods use decompositions of the FE space

$$\begin{aligned} V_h &= V_1 + \ldots + V_k && \text{one-level decomposition,} \\ V_k &= \{v \in V_h,\ v \equiv 0 \quad \text{in} \quad \Omega \setminus \Omega_k^\delta\}\,, \\ V_h &= V_0 + V_1 + \ldots + V_k && \text{two-level decomposition.} \end{aligned}$$

The simplest and most commonly used additive Schwarz preconditioning methods use preconditioners B_{AS} of one-level or two-level type,

$$B_{AS}^{(1)} = \sum_{k=1}^{m} R_k^T A_k^{-1} R_k, \quad B_{AS}^{(2)} = \sum_{k=0}^{m} R_k^T A_k^{-1} R_k.$$

For $k = 1, \ldots, m$, R_k is the restriction which selects degrees of freedoms lying in $\bar{\Omega}_k^\delta \setminus (\Gamma_u \cup \Gamma_k^0)$, where $\Gamma_k^0 = \partial\Omega_k^\delta \cap \Omega$ is the inner boundary of Ω_k^δ, A_k is the FE matrix corresponding to subproblem on Ω_k^δ with homogeneous Dirichlet boundary conditions on Γ_k^0. For two-level preconditioner, R_0^T represents the prolongation induced by imbedding $V_H \subset V_h$ and A_0 is the FE matrix for V_H.

3 Analysis of the Schwarz methods

The analysis of the Schwarz methods for SPD problems usually uses the Lion's lemma ([4],[11]) and investigation of two properties

$$\text{(P1)} \quad \exists K_0 > 0 \ \forall v \in V_h \ \exists v_k \in V_k\,, \ v = v_1 + \ldots + v_m \ : \sum \parallel v_k \parallel_a^2 \leq K_0 \parallel v \parallel_a^2$$

$$\text{(P2)} \quad \exists K_1 > 0 \ \forall v \in V_h \ \forall v_k \in V_k\,, \ v = v_1 + \ldots + v_m \ : \parallel v \parallel_a^2 \leq K_1 \sum \parallel v_k \parallel_a^2$$

where $\parallel v \parallel_a^2 = \sqrt{a(v,v)}$, a is the SPD bilinear form from the variational formulation of the problem $(1) - (4)$. The constants can be used for estimation of the convergence or the condition number of preconditioned systems, e.g.

$$\mathrm{cond}(B_{AS}A) \leq K_0 K_1.$$

The estimation of K_1 is easy. For our strip decomposition, we can take $K_1 = 3$ for one-level decomposition and $K_1 = 4$ for two-level decomposition. The investigation of stability constant K_0 usually use a decomposition of unity

$$1 = \sum_{k=1}^{m} \theta_k, \quad \mathrm{supp}(\theta_k) \subset \Omega_k^\delta, \quad \parallel \theta_k \parallel_\infty \leq 1, \quad \parallel \nabla\theta_k \parallel_\infty = O(\delta^{-1})$$

and construct the decomposition of $v \in V_h$, $v = \sum v_k$, $v_k \in V_k$ with

$$\begin{aligned} v_k &= \Pi_h(\theta_k v) && \text{for one-level decomposition,} \\ v_0 &= Qv, \quad v_k = \Pi_h(\theta_k(v - v_0)) && \text{for two-level decomposition,} \end{aligned}$$

where Π_h is linear interpolation $C(\bar{\Omega}) \to V_h$, $Q : \ V_h \to V_0$ is e.g. $a-$ or L_2 projection. The standard analysis then uses elementwise investigation and Friedrichs-type estimate.

For our model problem and strip-like decomposition, it is easy to construct θ_k such that $\theta_k \equiv 1$ in $\Omega_k^\delta \setminus \bigcup_{i \neq k} \Omega_i$, $\theta_k \equiv 0$ outside Ω_k^δ and θ_k linearly varying across the overlaping part of Ω_k^δ.

4 Conclusions

In this paper, we consider a model problem and strip-like domain decomposition which allows to clarify the influence of heterogeneity on efficiency of Schwarz type methods. For one-level Schwarz, the analysis with the above mentioned partition of unity suggests that heterogeneity outside overlapping regions $\Omega_k^\delta \cap \Omega_i^\delta$ does not influence efficiency of Schwarz methods (with exact subdomain solvers). On the other hand, if the heterogeneity in the overlapping regions can not be avoided, then the overlap as big as possible (preserving condition of at most two neighbours subdomains) could be advantageous. These properties are now verified by numerical experiments. For two level methods, there is more possibilities for adaptation to heterogeneity and getting robust solvers, see e.g. [10], [1], [3]. Note that the defined strip decomposition is also robust with respect to orthotropy if strips are orthogonal to the ”weaker”direction [9].

Acknowledgement: This work is supported by the grants GACR105/09/1830 of the Grant Agency CR and the research plan AV0Z30860518 of the Academy of Sciences of the Czech Republic.

References

[1] O. Axelsson, R. Blaheta, M. Neytcheva: *Preconditioning of boundary value problems using elementwise Schur complements.* SIAM J. Matrix Analysis & Appl. 31, 767–789, 2009.

[2] O. Axelsson, R. Blaheta: *Preconditioning of matrices partitioned in 2x2 block form: Eigenvalue estimates and Schwarz DD for mixed FEM.* Submitted.

[3] O. Axelsson, R. Blaheta, V. Sokol: *Schwarz iterative methods for problems with heterogeneous structure.* In progress.

[4] R. Blaheta: *Space decomposition preconditioners and parallel solvers.* In: Numerical Methods and Advanced Applications, M. Feistauer et al. (eds), Springer-Verlag, Berlin, 20–38, 2004.

[5] R. Blaheta, P. Byczanski, P. Harasim: *Multiscale modelling of geomaterials and iterative solvers.* Proceedings of SNA’09, Institute of Geonics AS CR, Ostrava 2009.

[6] R. Blaheta, O. Jakl, J. Starý, K. Krečmer: *Schwarz DD method for analysis of geocomposites.* In: Proceedings of the Twelfth International Conference on Civil, Structural and Environmental Engineering Computing B.H.V. Topping, L.F. Costa Neves and R.C. Barros, (eds.). Civil-Comp Press, Stirlingshire, UK, 2009.

[7] R. Blaheta, O. Jakl, R. Kohut, J. Starý: *GEM – a Platform for Advanced Mathematical Geosimulations.* In: Proceedings of the conference Parallel Processing and Applied Mathematics, PPAM 2009, to appear.

[8] G.A.A. Kahou, L. Grigori, M. Sosokina: *A partitioning algorithm for block-diagonal matrices with overlap.* Parallel Computing 34, 332–344, 2008.

[9] T.P.A. Mathew: *Domain decomposition methods for the numerical solution of partial differential equations.* Lecture Notes in Computational Science and Engineering, Springer, Berlin 2008.

[10] R. Scheichl, E. Vainikko: *Additive Schwarz and aggregation-based coarsening for elliptic problems with highly variable coefficients.* Computing 80, 319–343, 2007.

[11] A. Toselli, O. Widlund: *Domain Decomposition Methods - Algorithms and Theory.* Springer-Verlag, Berlin 2005.

Flow over a rough surface

P. Bauer

Czech Technical University, Prague
Institute of Thermomechanics, AS CR, Prague

Abstrakt

We attempt to model a 2D rough surface by computing non-stationary Navier-Stokes flow over a periodic pattern. The solution is obtained by means of finite element method (FEM). We use non-conforming Crouzeix Raviart elements for velocity and piecewise constant elements for pressure. The resulting linear system is solved by multigrid method. We present computational studies of the problem.

1 Introduction

We consider a polygonal domain $\Omega \subset \mathrm{R}^2$ composed of multiple canyons as an approximation of a rough surface, and solve the incompressible Navier-Stokes equations for velocity $\mathbf{u}$ and pressure p on $[0,T] \times \Omega$:

$$\frac{\partial \mathbf{u}(t,x)}{\partial t} + \mathbf{u}(t,x) \cdot \nabla \mathbf{u}(t,x) - \nu \triangle \mathbf{u}(t,x) + \nabla p(t,x) = 0$$

$$\nabla \cdot \mathbf{u}(t,x) = 0$$

$$\mathbf{u}(0,x) = \mathbf{u}_0(x) \quad x \in \Omega$$

We set no-slip boundary condition for velocity on the terrain, Poiseuille profile on the inlet, Neumann condition on the outlet, and slip condition on the upper boundary.

2 Weak formulation of Navier-Stokes equations

Let $X = (H^{(1)}(\Omega))^2$, $V(\mathbf{u}_{\mathrm{in}}) = \{\mathbf{u} \in X : \mathbf{u}\,|_{\mathrm{terrain}} = \mathbf{0}, \mathbf{u}\,|_{\mathrm{inlet}} = \mathbf{u}_{\mathrm{in}}, \mathbf{u}\,|_{\mathrm{upper}} \cdot \mathbf{n} = 0\}$, $Q = L^2(\Omega)$. We set the following forms:

$(\nabla \mathbf{u}, \nabla \mathbf{v}) = \int\limits_{\Omega} \sum\limits_{i,j=1}^{2} \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j}$, $\;b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \frac{1}{2} \int\limits_{\Omega} \sum\limits_{i,j=1}^{2} (u_j \frac{\partial v_i}{\partial x_j} w_i - u_j v_i \frac{\partial w_i}{\partial x_j})$.

We use the backward Euler difference for the time derivative $\frac{\partial \mathbf{u}(t^n,x)}{\partial t} \approx \frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\tau}$ where $t^n = n\tau$. For each timestep t^n, we seek $\mathbf{u}^n \in V(\mathbf{u}_{\mathrm{in}})$ and $p^n \in Q$, such that $\forall \mathbf{v} \in V(\mathbf{0})$, $\forall q \in Q$:

$$(\mathbf{u}^n, \mathbf{v}) + \tau b(\mathbf{u}^{n-1}, \mathbf{u}^n, \mathbf{v}) + \tau(\nabla \mathbf{u}^n, \nabla \mathbf{v}) - \tau(p^n, \nabla \cdot \mathbf{v}) = (\mathbf{u}^{n-1}, \mathbf{v})$$

$$(q, \nabla \cdot \mathbf{u}^n) = 0$$

Let index h denote the respective finite-dimensional spaces $V^h(\mathbf{u}_{\mathrm{in}})$, Q^h, and the corresponding functions $\mathbf{u}_h^n$, p_h^n. We use the upwinding technique proposed by [2], based on dual elements R_l given by the barycentric nodes of the original mesh (Fig. 1).

We introduce $\mathbf{w}_h \in V^h(\mathbf{u}_{\text{in}})$ to respresent inhomogeneous Dirichlet data. Taking $\mathbf{v}_h = \mathbf{u}_h - \mathbf{w}_h \in V^h(\mathbf{0})$, the discrete problem for each timestep t^n rewritten in the matrix form stands:

$$\begin{aligned} \mathbf{M}\mathbf{v}_h^n + \tau\mathbf{N}(\mathbf{u}_h^{n-1})\mathbf{v}_h^n + \tau\mathbf{A}\mathbf{v}_h^n + \tau\mathbf{B}^{\mathbf{T}}p_h^n &= \tilde{\mathbf{f}}, \\ \mathbf{B}\mathbf{v}_h^n &= \tilde{\mathbf{g}}, \end{aligned} \tag{1}$$

where

$$\begin{aligned} \tilde{\mathbf{f}} &= \mathbf{M}(\mathbf{v}_h^{n-1} + \mathbf{w}_h^{n-1} - \mathbf{w}_h^n) - \tau\mathbf{N}(\mathbf{u}_h^{n-1})\mathbf{w}_h^n - \tau\mathbf{A}\mathbf{w}_h^n, \\ \tilde{\mathbf{g}} &= -\mathbf{B}\mathbf{w}_h^n. \end{aligned}$$

3 Numerical solution using FEM

We choose non-conforming Crouzeix-Raviart elements (Fig. 1) to approximate the components of velocity and piecewise constant elements for pressure.

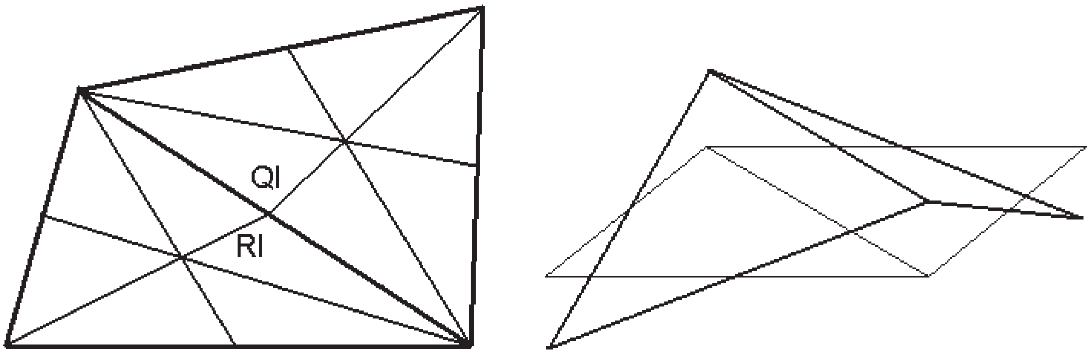


Figure 1: a) Lumped regions, b) Crouzeix-Raviart element.

We use multigrid solver based on Vanka-type smoother to solve the linear system (1). An extension for higher order elements can be found in [3].

4 Numerical results

We consider a periodic pattern composed by seven square canyons. We show the development of the flow over the pattern for $Re = 10^4$.

5 Conclusion

We investigate the flow over periodic structures as the means for parametrization of rough surfaces in models of larger scale in cooperation with the Institute of Thermomechanics of the Academy of Sciences of the Czech Republic.

Acknowledgement: This work has been supported by the research direction project *Applied Mathematics in Technical and Physical Sciences* of the Ministry of Education of the Czech Republic No. MSM6840770010.

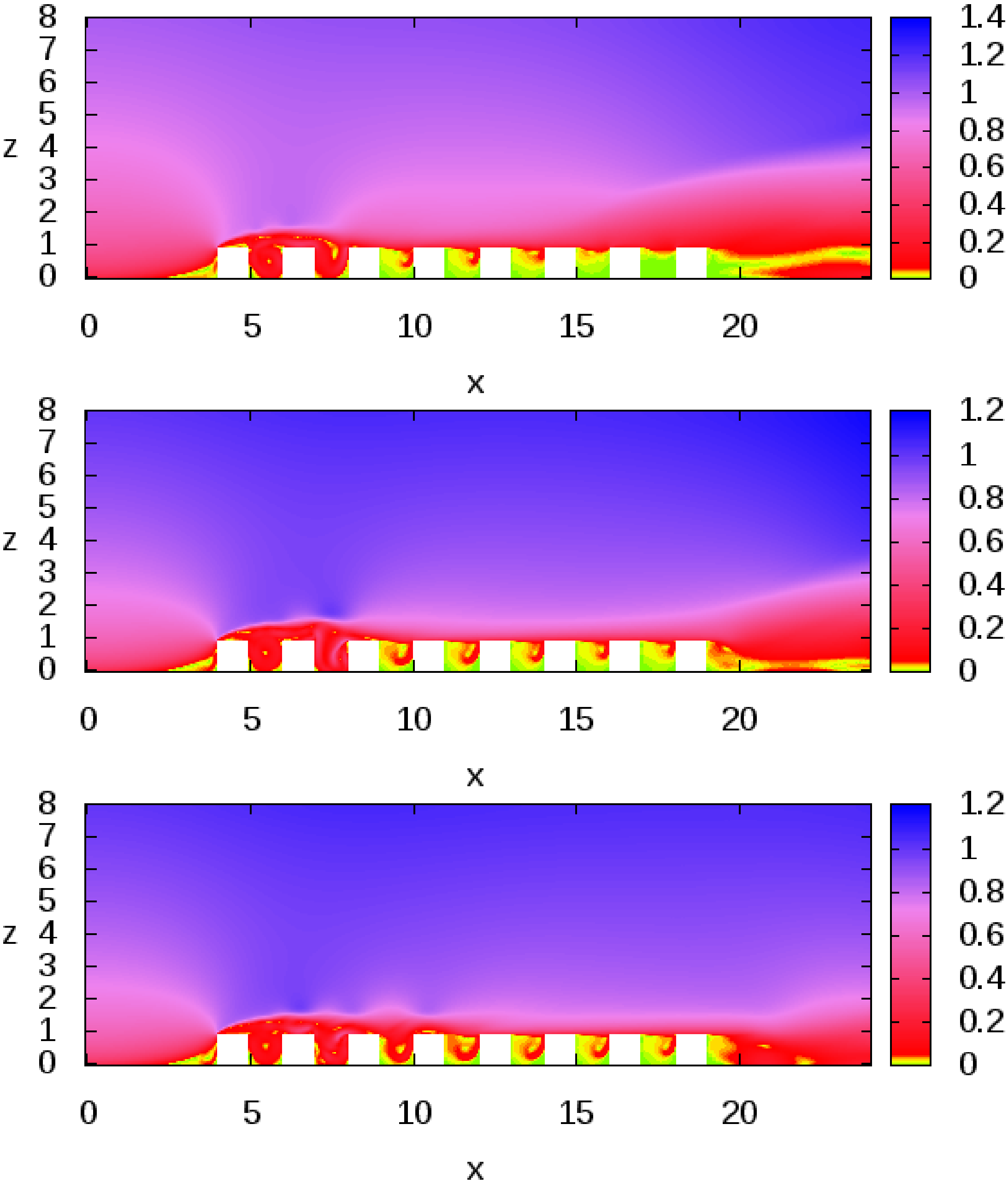


Figure 2: $|u(t)|$ at time $t = 24, 32, 40$.

References

[1] F. Brezzi, M. Fortin, *Mixed and hybrid finite-element methods,* Springer Verlag, New York (1991)

[2] F. Schieweck, L. Tobiska, *An optimal order error estimate for upwind discretization of the Navier-Stokes equation,* Numerical methods in partial differential equations y.12 n.4 (1996), 407–421

[3] V. John, P. Knobloch, G. Matthies, and L. Tobiska, *Non-Nested Multi-Level Solvers for Finite Element Discretisations of Mixed Problems,* Computing 68 (2002), 313–341

On iterative QR pre–processing in the parallel block–Jacobi SVD algorithm

M. Bečka, G. Okša, M. Vajteršic

Institute of Mathematics, Slovak Academy of Sciences, Bratislava

1 Introduction

Recently, an efficient version of the parallel two-sided block Jacobi SVD algorithm with pre-processing was proposed in [4]. When computing all singular values together with all right and left singular vectors of a rectangular matrix A, the pre-processing step consists of the parallel computation of the QR factorization (QRF) with column pivoting (CP) followed Â by the optional LQ factorization (LQF) of R-factor (this is called the QRLQ step). The parallel Â two-sided block Jacobi method with dynamic ordering (cf. [1]) is then applied to the R-factor (or L-factor). The purpose of pre-processing is to concentrate the Frobenius norm near the matrix diagonal so that the Jacobi algorithm may need substantially less parallel steps for convergence than in the case without pre-processing.

However, to perform optimally, Â the parallel Â QRF (or LQF) and the parallel two-sided block Jacobi method need different data layouts. Having p processors, Â the Jacobi algorithm performs very well when the matrix is distributed using the one-dimensional block column distribution. In this case, most of the computations can be performed locally. But this data distribution is not well suited for the serialized block column-oriented parallel QRF. Instead, a block cyclic matrix distribution on a process grid $r \times c$ Â with $p = rc, \;\; r, c \geq 1,$ is needed so that all processors remain busy during the whole parallel QR (or LQ) factorization.

Optimal parameters for the pre-processing step need to be found experimentally for a given parallel architecture. For a cluster of modern computational nodes, it is shown that their values are about $n_b = 100$ and $r \leq c$, $r = \max$, i.e., r is maximized so that both r and c are as close to $\sqrt{p}$ as possible. Numerical experiments suggest that the efficiency of a pre-processing step depends on the distribution of singular values (SVs). In contrast, the dependence on the condition number κ is only mild. The optimal parameters were then used in the pre-processed parallel two-sided block-Jacobi SVD algorithm; its performance was tested for six various distributions of SVs and for well-conditioned ($\kappa = 10^1$) as well as ill-conditioned ($\kappa = 10^8$) random, square, real matrices of order $n = 4000$ and 8000 using $p = 8$ and 16 processors, respectively, with the constant ratio $n/p = 500$. The largest savings in the number of parallel iteration steps needed for the convergence of the whole algorithm were obtained for matrices with a multiple maximal/minimal SV regardless to κ. In these two cases, our algorithm performs better or equally well as the ScaLAPACK routine `PDGESVD`. However, it is shown experimentally that for other four distributions of SVs, the parallel two-sided block-Jacobi SVD algorithm with the optimal pre-processing and dynamic ordering is about 1.2–2.3 times slower than the ScaLAPACK routine.

The un-pivoted QRLQ pre-processing step can be re-formulated and extended to the *QR iteration* (QRI). Using a limited number of QRI steps in the pre-processing can lead to even larger reduction of the off-diagonal Frobenius norm and to the faster convergence of the subsequent Jacobi algorithm with dynamic ordering for certain distributions of SVs. In the serial case, the use of the QRI for the estimates of SVs and singular vectors has been analyzed, e.g., in [2, 3]. To our knowledge, up to now nobody extended its use as the pre-processing method for computing the SVD in parallel. We have implemented the QRI in front of the parallel two-sided block-Jacobi algorithm with dynamic ordering, and we report results for a set of matrices mentioned above. In general, the use of about 6 QRI steps can be recommended before switching to the Jacobi algorithm. Such a strategy can significantly decrease the total parallel execution time of the whole algorithm.

2 Parallel algorithm with dynamic ordering

When using p processors and the blocking factor $\ell = 2p$, a given matrix A is cut column-wise and row-wise into an $\ell \times \ell$ block structure. Each processor contains exactly two block columns of dimensions $m \times n/\ell$ so that $\ell/2$ SVD subproblems of block size 2×2 are solved in parallel in each iteration step.

At the beginning of each parallel iteration step, it is necessary to map one 2×2 block SVD subproblem to each of p processors. This can be achieved by some type of ordering. The so-called *dynamic* ordering is based on the maximum-weight perfect matching that operates on the $\ell \times \ell$ updated weight matrix W using the elements of $W + W^T$, where $(W + W^T)_{ij} = \|A_{ij}\|_\mathrm{F}^2 + \|A_{ji}\|_\mathrm{F}^2$. As shown in [1], this approach leads to a maximum decrease of the off-diagonal Frobenius norm in each parallel iteration step. Moreover, the optimal algorithm for finding the ordering has complexity $O(\ell^3)$.

The convergence of the whole process can be enhanced by a suitable pre-processing of matrix A. As discussed in [4], the QR factorization with column pivoting (QRFCP) can be applied to A at the beginning of computation. Then, the SVD of the matrix R is computed by the PTBJA with dynamic ordering. In the final step, some post-processing in the form of matrix-matrix multiplication is required to obtain the SVD of A.

Alternatively, after the QRFCP of A, one can apply the LQF of R-factor (without column pivoting). Next, the SVD of L is computed by our parallel PTBJA with dynamic ordering. To obtain the SVD of A, two matrix-matrix multiplications are needed in the final post-processing step.

The purpose of the pre-processing step is twofold. First, for rectangular matrices of order $m \times n$, $m \geq n$, the R-factor (L-factor) is a square matrix of order n, which means that for $m \gg n$ huge savings in storage requirements, matrix multiplications and computation of 2×2 block SVDs can be achieved. Second, after the QRFCP (followed by the optional QLF of the R-factor), the Frobenius matrix norm is usually well concentrated near the matrix diagonal, so that only few iteration steps in the parallel two-sided block-Jacobi algorithm are needed for convergence.

3 Optimal data layout for pre-processing

When using p processors with the matrix block cyclic distribution of type $1 \times p$ (i.e., the whole block columns are stored in processors), it is immediately clear that the computation of the block QRF is serialized and synchronized. When the blocking factor in the iterative part of the Jacobi algorithm is $\ell = 2p$ (i.e., each processor stores two block columns), and the block size for the QRF is n_b, then processor i, $0 \leq i \leq p-1$, starts the QRF of its submatrix at step $2\lceil n/\ell \rceil i$ and finishes it at step $2\lceil n/\ell \rceil (i+1)$ (one step here means the processing of one matrix column). During this computation, processor i sends $2\lceil n/\ell\rceil/n_b$-times data to processors to its right for computing updates. At any given time, only one processor computes the QRF; all other processors are computing only the updates and they have to wait for data. As the computation is column-oriented and proceeds from left to right, more and more processors become idle during the computation of the block QRF. This is a highly inefficient use of the computational power.

To increase the efficiency in the pre-processing step, one should use a matrix cyclic block distribution with block size n_b on a process grid $r \times c$, $r, c \geq 1$, with $p = rc$, where r and c is the number of processors in a process row and column, respectively. This type of data distribution is required by all ScaLAPACK matrix routines. Such a data distribution eliminates the synchronization in computing the block QRF and can lead to a better use of computational resources and thus to a faster computation. On the other side, the broadcast of updating data becomes more complex because it needs to be done both across process rows and columns. Moreover, in the triangularization of a given block column A_1 of width n_b, all processors in the appropriate process column are involved and they have to communicate. Despite these differences as compared with the process grid $1 \times p$, the parallel block QRF (with or without column pivoting) on a process grid $r \times c,\ r, c \geq 1,\ p = rc$, can take substantially less time.

We will report our results for all possible combinations of r and c for a given number of processors p together with a variable block column width n_b. Regardless to the number of processors and distribution of singular values, the minimum total parallel execution time was achieved for $n_b = 100$ and the process grid $r \times c$ with $r \leq c$, r closest to $\sqrt{p}$.

4 Numerical experiments with the optimal data layout

We have implemented the parallel two-sided block-Jacobi SVD algorithm (PTBJA) with pre-processing on the Woodcrest Cluster at Regionales Rechenzentrum Erlangen (RRZE), Erlangen-Nuernberg University, Germany. The Woodcrest Cluster consists of 217 computational nodes, each with two Xenon 5160 Woodcrest chips (4 cores organized in 2 dual cores) running at 3.0 GHz. Each dual core contains 4 MB shared Level 2 cache, 8 GB of RAM and 160 GB of local scratch disk. The Infiniband interconnection network has the bandwidth of 10 GBit/s per link and direction.

Each test matrix was generated by the ScaLAPACK routine `DLATMS` as a matrix product using a given distribution of SVs chosen from a group of six possible types, and two random orthogonal matrices with elements from the normal distribution $N(0, 1)$. Two condition numbers were used: $\kappa = 10^1$ and $\kappa = 10^8$.

We will report and comment detailed results from our numerical experiments. These experiments can be divided in two groups. In the first group, four variants of the PTBJA with dynamic ordering were tested: without any pre-processing, with the pre-processing consisting of the QRF with CP, and with the pre-processing where the QRF with/without CP was followed by the LQF of R-factor. For matrices with a multiple maximal/minimal SV, the use of optimal parameters in the pre-processing step (the QRF without CP following by the LQF of R-factor) enabled to achieve the performance better or comparable to that of the ScaLAPACK routine `PDGESVD`. For other distributions of SVs, even the QRFCP + the LQF of R-factor did not lead to a significant decrease of a number of iterations so that the pre-processed PTBJA with dynamic ordering was about 1.2–2.3 times slower than the ScaLAPACK routine `PDGESVD`.

In the second group of experiments, the un-pivoted QRLQ pre-processing was re-formulated and extended to the QR iteration (QRI). There is a well-known connection between the QRI and the QR algorithm applied to a specific sequence of symmetric, positive definite matrices, which enables to use the convergence theory of the QR algorithm for the explanation of experimental results. Specifically, the convergence of the QRI is largely enhanced in the case of multiple SVs (or cluster of close SVs) and in presence of large gap(s) between any consecutive SVs, which is rather typical for ill-conditioned matrices. In such situation it can happen that the SVD is computed only by using, say, 4 QRI steps, without invoking the Jacobi algorithm at all. In general, the use of about 6 QRI steps can be recommended in the pre-processing, followed by a (quite limited) number of parallel iterations in the Jacobi algorithm with dynamic ordering. Such a strategy usually leads to a significant reduction of the total parallel execution time of the whole algorithm for almost all six tested distributions of SVs.

Acknowledgments: Our special thanks go to Prof. Dr. Ulrich Ruede from Erlangen-Nuernberg University for his kind permission and help in using the Woodcrest Cluster. First two authors were supported by the grant no. APVV – 0532 – 07 from the Agency for Science and Research, Slovakia.

References

[1] M. Bečka, G. Okša, M. Vajteršic: *Dynamic ordering for a parallel block-Jacobi SVD algorithm.* Parallel Computing, 28 243–262, (2002).

[2] S. Chandrasekaran, I.C.F. Ipsen: *Analysis of a QR algorithm for computing singular values.* SIAM J. Matrix Anal. Appl., 16(2), 20–535, (1995).

[3] D. A. Huckaby, T. F. Chan: *On the convergence of Stewart's QLP algorithm for approximating the SVD.* Num. Alg., 32, 287–316, (2003).

[4] G. Okša, M. Vajteršic: *Efficient pre-processing in the parallel block-Jacobi SVD algorithm.* Parallel Computing, 32, 166–176, (2006).

Moving boundaries in material science

M. Beneš

Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University in Prague, Trojanova 13, 120 00 Praha 2, Czech Republic
benesmic@kmlinux.fjfi.cvut.cz

The presentation discusses mathematical modelling and numerical simulation of two classes of free-boundary problems arising in material science, which are related to the microstructure formation in solidification of crystalline materials, and to the dislocation dynamics in the crystalline lattice.

The discussed problems occurring in the context of material science have a similar evolution law. This law generally written in the form

$$v_\Gamma = -\kappa_\Gamma + F,$$

is known to describe the motion of curves or surfaces by mean curvature (here denoted by κ_Γ, whereas v_Γ denotes the normal velocity and F a forcing term). The law together with its variants including anisotropy is being extensively studied from the mathematical as well as application viewpoint. We present formulation of mathematical models as well as their numerical solution demonstrating agreement with the experimental understanding of the studied phenomena.

Tvarová optimalizace pro 3D kontaktní problém diskretizovaný metodou hraničních prvků

P. Beremlijski, M. Sadowská

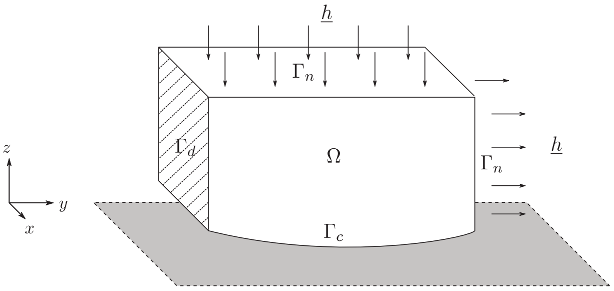
Katedra aplikované matematiky, VŠB - Technická univerzita Ostrava

1 Úvod

V příspěvku se zabýváme úlohou diskrétní tvarové optimalizace trojrozměrného pružného tělesa v jednostranném kontaktu s tuhou překážkou. Pro diskretizaci kontaktní úlohy jsme použili metodu hraničních prvků. Aplikace metody hraničních prvků na kontaktní úlohy můžeme najít například v [2, 5]. Pro malý koeficient tření má diskretizovaná kontaktní úloha s Coulombovým třením jediné řešení, které je navíc závislé lokálně lipschitovsky na řídící proměnné popisující tvar pružného tělesa. Díky jedinému řešení diskrétní úlohy pro fixovanou řídící proměnnou, můžeme použít tzv. přístup implicitního programování, který je založen na minimalizaci nehladké funkce složené z cenové funkce a jednoznačného zobrazení, které řídící proměnné přiřazuje řešení diskrétní úlohy, tzn. stavové proměnné. Pro minimalizaci nehladké funkce jsme použili bundle trust metodu. K získání subgradientní informace, kterou metoda vyžaduje, je nutno použít Morduchovičova a Clarkeova kalkulu.

2 Kontaktní úloha s Coulombovým třením

Buď $\mathcal{O}$ zvolená třída omezených oblastí $\Omega \subset \mathbb{R}^3$ s lipschitzovskou hranicí Γ složenou ze tří navzájem disjunktních částí Γ_d, Γ_n a Γ_c (viz obr. 1). Oblast $\Omega \in \mathcal{O}$ je vyplněna homogenním izotropním materiálem a má tvar „kvádru“ se spodní volnou částí $\Gamma_c = \Gamma_c(\Omega)$, jejíž tvar bude navrhován pomocí zvolených přípustných funkcí $\alpha : \mathcal{R} \mapsto \mathbb{R}$, kde $\mathcal{R}$ označuje pravoúhlý průmět Ω do roviny xy. Část hranice $\Gamma_d \cup \Gamma_n$ je pevná.



Obrázek 1: Geometrie oblasti $\Omega \in \mathcal{O}$.

Na Γ_d budeme uvažovat homogenní Dirichletovu podmínku ve všech souřadných směrech, na Γ_n působí povrchové síly $\underline{h} \in [L^2(\Gamma_n)]^3$ a podél Γ_c je těleso „podepřeno“ tuhou překážkou $\mathbb{R}^2 \times \mathbb{R}_-$ (viz obr. 1), přičemž mezi tělesem a překážkou uvažujeme Coulombovo tření dané koeficientem $\mathcal{F}$. Funkce $\underline{u}(\Omega)$ tedy vyhovuje systému homogenních rovnic rovnováhy $\mathcal{L}\underline{u} = \underline{0}$ lineární homogenní izotropní elastostatiky, předepsané Dirichletově resp. Neumannově podmínce na Γ_d resp. Γ_n, unilaterálním kontaktním podmínkám

$$u_3(x) \geq -x_3, \quad T_3(x) \geq 0, \quad T_3(x)(u_3(x) + x_3) = 0 \quad \text{pro každé } x = (x_1, x_2, x_3) \in \Gamma_c(\Omega),$$

a Coulombovu zákonu tření

$$\begin{cases} \text{pokud} \quad \underline{u}_t(x) := (u_1(x), u_2(x), 0) = \underline{0}, & \text{pak} \quad \|\underline{T}_t(x) := (T_1(x), T_2(x), 0)\| \leq \mathcal{F}T_3(x) \\ \text{pokud} \quad \underline{u}_t(x) \neq \underline{0}, & \text{pak} \quad \underline{T}_t(x) = -\mathcal{F}T_3(x)u_t(x)/\|u_t(x)\| \end{cases}$$

pro každé $x \in \Gamma_c(\Omega)$. Vektor $\underline{T}(x) = (T_1(x), T_2(x), T_3(x))$ značí povrchové napětí v $x \in \Gamma_c(\Omega)$.

3 Slabá hraniční formulace stavového problému

Slabé řešení $\underline{u} \in [H^1(\Omega)]^3$ systému homogenních rovnic rovnováhy lineární homogenní izotropní elastostatiky splňuje Greenův reprezentační vztah

$$u_l(x) = \int\limits_{\Gamma} (\gamma_1 \underline{u}(y),\, \underline{U}_l(x,y))\, \mathrm{d}s_y - \int\limits_{\Gamma} (\gamma_0 \underline{u}(y),\, \gamma_{1,y} \underline{U}_l(x,y))\, \mathrm{d}s_y, \quad x \in \Omega,\ l = 1, 2, 3, \tag{1}$$

kde U je fundamentální řešení lineární elastostatiky známé jako Kelvinův tensor:

$$U_{kl}(x,\, y) := \frac{1+\nu}{8\pi E(1-\nu)} \left((3-4\nu) \frac{\delta_{kl}}{\|x-y\|} + \frac{(x_k - y_k)(x_l - y_l)}{\|x-y\|^3} \right), \quad k, l = 1, 2, 3,$$

$\gamma_0 :\ [H^1(\Omega)]^3 \mapsto [H^{1/2}(\Gamma)]^3$ je operátor stopy a

$$\gamma_1 :\ \{\underline{v} \in [H^1(\Omega)]^3 :\ \mathcal{L}\underline{v} \in [L^2(\Omega)]^3\} \mapsto [H^{-1/2}(\Gamma)]^3$$

je operátor příslušného povrchového napětí splňující pro $\underline{v} \in [C^\infty(\overline{\Omega})]^3$

$$(\gamma_1 \underline{v})_i\, (x) = \sum_{j=1}^{3} \sigma_{ij}(\underline{v}, x) n_j(x), \qquad x \in \Gamma,\ i = 1, 2, 3;$$

$n_j(x)$ je složka vnějšího jednotkového normálového vektoru a σ_{ij} je složka tensoru napětí.

Aplikací operátorů γ_0 a γ_1 na (1) získáme (dle [6]) hraniční vztah

$$\gamma_1 \underline{u} = S(\gamma_0 \underline{u}) \quad \text{na } \Gamma,$$

kde $S :\ [H^{1/2}(\Gamma)]^3 \mapsto [H^{-1/2}(\Gamma)]^3$ je Steklovův-Poincarého operátor, který lze reprezentovat jako

$$S = D + (\frac{1}{2}I + K')V^{-1}(\frac{1}{2}I + K);$$

V je operátor jednoduché vrstvy, K je operátor dvojvrstvy, K' je operátor adjungovaný ke K a D je tzv. hypersingulární operátor. Definice a vlastnosti těchto operátorů jsou k nalezení v [6].

Definujme nyní $W(\Omega) := [H_0^{1/2}(\Gamma(\Omega), \Gamma_d)]^3$ a $X(\Omega) := \{\varphi \in L^2(\mathcal{R}) :$ existuje $\underline{v} \in W(\Omega)$ takové, že na $\Gamma_c(\Omega)$ platí $\varphi = v_3\}$. Buď dále $X'_+(\Omega)$ kužel kladných funkcionálů z duálu $X'(\Omega)$. Slabým hraničním řešením kontaktního stavového problému z 2. kapitoly rozumíme libovolnou dvojici $(\underline{u}, \lambda) \in W(\Omega) \times X'_+(\Omega)$ vyhovující systému

$$\begin{cases} \displaystyle\int\limits_{\Gamma(\Omega)} (S\underline{u}, \underline{v} - \underline{u})\, \mathrm{d}s + \langle \mathcal{F}\lambda, \|\hat{\underline{v}}_t\| - \|\hat{\underline{u}}_t\| \rangle \geq \int\limits_{\Gamma_n} (\underline{h}, \underline{v} - \underline{u})\, \mathrm{d}s + \langle \lambda, \hat{v}_3 - \hat{u}_3 \rangle \quad \forall \underline{v} \in W(\Omega) \\ \langle \mu - \lambda, \hat{u}_3 + \alpha \rangle \geq 0 \quad \forall \mu \in X'_+(\Omega), \end{cases}$$

kde $\langle \cdot, \cdot \rangle$ je dualitní párování mezi prostory $X(\Omega)$ a $X'(\Omega)$, $\hat{v}_3(x') := v_3(x', \alpha(x'))$ a $\hat{\underline{v}}_t(x') := (v_1(x', \alpha(x')), v_2(x', \alpha(x')), 0)$, $x' \in \mathcal{R}$.

4 Diskrétní tvarová optimalizace pro kontaktní úlohu s Coulombovým třením

Než si zformulujeme naši úlohu tvarové optimalizace, zapišme náš stavový problém formou zobecněné rovnosti. K tomu diskretizujeme stavovou úlohu a poté zavedeme rozdělení vektoru posunutí $\boldsymbol{u}$ na $(\boldsymbol{u}_t, \boldsymbol{u}_\nu)$, kde $\boldsymbol{u}_t$ přísluší tečnému posunutí a $\boldsymbol{u}_\nu$ odpovídá normálovému posunutí. Následně vyeliminujeme volné uzly, tj. budeme se zabývat pouze kontaktními uzly (jejich počet je p). Diskretizovanou stavovou úlohu můžeme popsat zobrazením $\mathcal{S} : \boldsymbol{\alpha} \in \mathbb{R}^d \to (\boldsymbol{u}_t, \boldsymbol{u}_\nu, \boldsymbol{\lambda}) \in \mathbb{R}^{4p}$ (tvaru oblasti Ω, který je určen řídícím vektorem $\boldsymbol{\alpha} \in U_{ad}$, je přiřazeno řešení kontaktní úlohy s Coulombovým třením $(\boldsymbol{u}_t, \boldsymbol{u}_\nu, \boldsymbol{\lambda})$ (stavové proměnné)). Zobrazení $\mathcal{S}$ je pro malé koeficienty tření lokálně lipschitzovské. Diskretizovanou stavovou úlohu můžeme ekvivalentně popsat zobecněnou rovností:

$$\begin{aligned} \boldsymbol{0} &\in \boldsymbol{A}_{tt}(\boldsymbol{\alpha})\boldsymbol{u}_t + \boldsymbol{A}_{t\nu}(\boldsymbol{\alpha})\boldsymbol{u}_\nu - \boldsymbol{L}_t(\boldsymbol{\alpha}) + \tilde{\boldsymbol{Q}}(\boldsymbol{u}_t, \boldsymbol{\lambda}) \\ \boldsymbol{0} &= \boldsymbol{A}_{\nu t}(\boldsymbol{\alpha})\boldsymbol{u}_t + \boldsymbol{A}_{\nu\nu}(\boldsymbol{\alpha})\boldsymbol{u}_\nu - \boldsymbol{L}_\nu(\boldsymbol{\alpha}) - \boldsymbol{\lambda} \\ \boldsymbol{0} &\in \boldsymbol{u}_\nu + \boldsymbol{\alpha} + N_{\boldsymbol{R}^p_+}(\boldsymbol{\lambda}), \end{aligned} \tag{2}$$

kde $\boldsymbol{A}(\boldsymbol{\alpha}) \in \mathbb{R}^{3p\times 3p}$ a $\boldsymbol{L}(\boldsymbol{\alpha}) \in \mathbb{R}^{3p}$ jsou matice tuhosti a vektoru pravé strany, které jsme získali po diskretizaci stavové úlohy metodou hraničních prvků,

$$\tilde{\boldsymbol{Q}}(\boldsymbol{u}_{t1}, \boldsymbol{u}_{t2}, \boldsymbol{\lambda}_\nu) = \partial_{(\boldsymbol{u}_{t1}, \boldsymbol{u}_{t2})} j(\boldsymbol{u}_{t1}, \boldsymbol{u}_{t2}, \boldsymbol{\lambda}_\nu), \quad j(\boldsymbol{u}_{t1}, \boldsymbol{u}_{t2}, \boldsymbol{\lambda}_\nu) = \mathcal{F}\sum_{i=1}^{p} \lambda^i ||(\boldsymbol{u}^i_{t1}, \boldsymbol{u}^i_{t2})||$$

a $N_{\boldsymbol{R}^p_+}$ je standardní normálový kužel.

Nyní si popišme úlohu tvarové optimalizace. Hledáme návrhovou proměnnou $\boldsymbol{\alpha}$ řídící tvar Beziérovy plochy, kterou je určena kontaktní hranice Γ_c, tzn. i tvar tělesa Ω, pro kterou nabývá cenový funkcionál $\mathcal{J}(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha}))$ svého minima. Úlohu diskrétní tvarové optimalizace pro kontaktní úlohu s Coulombovým třením pak popíšeme takto:

$$\begin{aligned} &\min \Theta(\boldsymbol{\alpha}) \\ &\text{s omezením} \\ &\quad \boldsymbol{\alpha} \in U_{ad}, \end{aligned}$$

kde $\Theta(\boldsymbol{\alpha}) := \mathcal{J}(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha}))$. Nechť funkcionál $\mathcal{J}$ je spojitě diferencovatelný. K řešení této nehladké úlohy byla použita bundle trust metoda. Tato iterační metoda potřebuje rutinu, která v každém kroce vypočte hodnotu cenového funkcionálu (k tomu potřebujeme vyřešit diskretizovanou stavovou úlohu) a jeden (libovolný) Clarkeův subgradient z Clarkeova zobecněného gradientu $\partial\Theta(\boldsymbol{\alpha})$. Pro jeho konstrukci použijeme tvrzení

$$\partial\Theta(\boldsymbol{\alpha}) = \nabla_1 \mathcal{J}(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})) + \{\boldsymbol{C}^T \nabla_2 \mathcal{J}(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})) | \boldsymbol{C} \in \partial\mathcal{S}(\boldsymbol{\alpha})\}$$

(viz [3]). Dále využijeme nehladkého kalkulu B. Morduchoviče (viz [4]).

Protože platí $\emptyset \neq D^*\mathcal{S}(\boldsymbol{\alpha})(\boldsymbol{y}^*)$ pro všechna $\boldsymbol{y}^*$ a $\operatorname{conv}(D^*\mathcal{S}(\boldsymbol{\alpha}))(\boldsymbol{y}^*) = \{\boldsymbol{C}^T\boldsymbol{y}^* | \boldsymbol{C} \in \partial\mathcal{S}(\boldsymbol{\alpha})\}$, stačí nalézt jeden prvek z množiny $D^*\mathcal{S}(\boldsymbol{\alpha})(\nabla_2 \mathcal{J}(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})))$. Hledání prvků limitní koderivace

$$D^*\mathcal{S}(\boldsymbol{\alpha})(\boldsymbol{y}^*) := \{\boldsymbol{x}^* \in \mathbb{R}^d \,|\, (\boldsymbol{x}^*, -\boldsymbol{y}^*) \in N_{\mathrm{Gr}\,\mathcal{S}}(\boldsymbol{\alpha})\},$$

kde Gr $\mathcal{S}$ je graf $\mathcal{S}$ a $N_{\mathrm{Gr}\,\mathcal{S}}$ je limitní normálový kužel, je značně komplikované a využívá se při něm zápisu zobrazení $\mathcal{S}$ pomocí zobecněné rovnosti (2) (podrobně viz [1]).

5 Závěr

Ve 3D úloze tvarové optimalizace není možné pro citlivostní analýzu využít po částech spojitou diferencovatelnost zobrazení $\mathcal{S}$, které řídícímu vektoru přiřazuje stavové proměnné, jako ve 2D verzi této úlohy. Proto jsme pro citlivostní analýzu optimalizační úlohy, kterou se zabýváme v této práci, museli použít Morduchovičova kalkulu. Pro citlivostní analýzu je nutné vypočíst parciální derivace matice tuhosti a vektoru pravé strany, které získáme při diskretizaci stavové úlohy metodou hraničních prvků, podle jednotlivých řídících proměnných. V této práci jsme tyto derivace získali numericky. V budoucnu bychom chtěli odvodit vztahy pro analytický výpočet těchto derivací.

Poděkování: Tato práce byla podpořena GA ČR 201/07/0294 a MŠMT MSM6198910027.

Reference

[1] P. Beremlijski, J. Haslinger, M. Kočvara, R. Kučera, J. Outrata: *Shape Optimization in Three-Dimensional Contact Problems with Coulomb Friction.* SIAM Journal on Optimization 20(1), 416–444, 2009.

[2] J. Bouchala, Z. Dostál, M. Sadowská: *Scalable Total BETI Based Algorithm for 3D Coercive Contact Problems of Linear Elastostatics.* Computing 85, 189–217, 2009.

[3] F.H. Clarke: *Optimization and Nonsmooth Analysis.* J. Wiley & Sons, 1983.

[4] B.S. Mordukhovich: *Variational Analysis and Generalized Differentiation, Volumes I and II.* Springer-Verlag, 2006.

[5] C. Eck, O. Steinbach, W.L. Wendland: *A Symmetric Boundary Element Method for Contact Problems with Friction.* Math Comput Sim 50, 43–61, 1999.

[6] O. Steinbach: *Numerical Approximation Methods for Elliptic Boundary Value Problems, Finite and Boundary Elements.* Springer-New York, 2008.

A parametric study of the dimensionless closed gas-liquid system

M. Biák, D. Janovská

Department of Mathematics, Institute of Chemical Technology, Prague

1 Introduction

In the original model introduced in [1], the solution depends on ten parameters. The dimensionless formulation significantly reduces the number of parameters only to four. Another obvious advantage is the formulation independency on the model scale.

We study the dependence of the solution of the dimensionless formulation on a given parameter set. All simulations are performed in Matlab software package, namely we use a modified version of the program developed by Petri T. Piiroinen and Yuri A. Kuznetsov, see [3].

2 Transformation of the model equations into a dimensionless form

Let us start with the original dimensional Filippov system that describes the closed gas-liquid system. For the detailed derivation of the model equations, see [1], [2].

$$\mathcal{F}: \quad \frac{\mathrm{d}}{\mathrm{d}\,t}\begin{pmatrix} M_G \\ M_L \end{pmatrix} = \begin{cases} \mathbf{f}^{(1)}(M_G, M_L), & \varphi(M_G, M_L) < 0, \\ \mathbf{f}^{(2)}(M_G, M_L), & \varphi(M_G, M_L) > 0, \end{cases} \tag{1}$$

where

$$\mathbf{f}^{(1)} = \begin{pmatrix} F_G - k_G x \left(\dfrac{M_G \mathrm{R} T}{V - M_L/\rho_L} - P_{out} \right) \\ F_L \end{pmatrix}, \tag{2}$$

$$\mathbf{f}^{(2)} = \begin{pmatrix} F_G \\ F_L - k_L x \left(\dfrac{M_G \mathrm{R} T}{V - M_L/\rho_L} - P_{out} \right) \end{pmatrix}, \tag{3}$$

$$\varphi(M_G, M_L) = M_L - \rho_L V_d. \tag{4}$$

Let $p = (F_G, F_L, \rho_L, V, V_d, T, P_{out}, x, k_L, k_G)^{\mathrm{T}},\ p \in \mathbb{R}^{10}$, be a row vector of the parameters. We consider the Filippov system $\mathcal{F}$ dependent on p:

$$\mathcal{F}(p): \quad \frac{\mathrm{d}}{\mathrm{d}\,t}\begin{pmatrix} M_G \\ M_L \end{pmatrix} = \begin{cases} \mathbf{f}^{(1)}(M_G, M_L, p), & \varphi(M_G, M_L, p) < 0, \\ \mathbf{f}^{(2)}(M_G, M_L, p), & \varphi(M_G, M_L, p) > 0, \end{cases} \tag{5}$$

where

$$\mathbf{f}^{(1)}(M_G, M_L, p) = \begin{pmatrix} F_G - k_G x \left(\dfrac{M_G \mathrm{R} T}{V - M_L/\rho_L} - P_{out} \right) \\ F_L \end{pmatrix}, \tag{6}$$

$$\mathbf{f}^{(2)}(M_G, M_L, p) = \left(\begin{array}{l} F_G \\ F_L - k_L x \left(\dfrac{M_G \mathrm{R} T}{V - M_L/\rho_L} - P_{out} \right) \end{array} \right), \tag{7}$$

$$\varphi(M_G, M_L, p) = M_L - \rho_L V_d. \tag{8}$$

Now, we can define the dimensionless state variables $\tilde{M}_G,\ \tilde{M}_L,\ \tilde{t}$,

$$M_G := M_G^\circ \tilde{M}_G, \quad M_L := M_L^\circ \tilde{M}_L, \quad t := t^\circ \tilde{t}. \tag{9}$$

If we choose the scaling

$$F_G \frac{t^\circ}{M_G^\circ} = 1, \quad F_L \frac{t^\circ}{M_L^\circ} = 1, \quad \frac{\rho_L V}{M_L^\circ} = 1, \tag{10}$$

we obtain a row vector of the dimensionless parameters $\tilde{p} = (\alpha_G, \alpha_L, \beta_L, \beta_G, \gamma)^{\mathrm{T}},\ \tilde{p} \in \mathbb{R}^5$, where

$$\alpha_G = k_G x \frac{t^\circ M_G^\circ \mathrm{R} T \rho_L}{(M_L^\circ)^2} = k_G x \frac{F_G \mathrm{R} T \rho_L}{F_L^2}, \tag{11}$$

$$\alpha_L = k_L x \frac{t^\circ M_G^\circ \mathrm{R} T \rho_L}{(M_L^\circ)^2} = k_L x \frac{F_G \mathrm{R} T \rho_L}{F_L^2}, \tag{12}$$

$$\beta_L = k_L x P_{out} \frac{t^\circ}{M_L^\circ} = \frac{k_L x P_{out}}{F_L}, \tag{13}$$

$$\beta_G = k_G x P_{out} \frac{t^\circ}{M_L^\circ} = \frac{k_G x P_{out}}{F_L}, \tag{14}$$

$$\gamma = \frac{V_d}{V}. \tag{15}$$

The dimensionless Filippov system $\tilde{\mathcal{F}}$ dependent on $\tilde{p} \in \mathbb{R}^5$ has the form:

$$\tilde{\mathcal{F}}(\tilde{p}): \quad \frac{\mathrm{d}}{\mathrm{d}\tilde{t}} \begin{pmatrix} \tilde{M}_G \\ \tilde{M}_L \end{pmatrix} = \begin{cases} \tilde{\mathbf{f}}^{(1)}(\tilde{M}_G, \tilde{M}_L, \tilde{p}), & \varphi(\tilde{M}_G, \tilde{M}_L, \tilde{p}) < 0, \\ \tilde{\mathbf{f}}^{(2)}(\tilde{M}_G, \tilde{M}_L, \tilde{p}), & \varphi(\tilde{M}_G, \tilde{M}_L, \tilde{p}) > 0, \end{cases} \tag{16}$$

where

$$\tilde{\mathbf{f}}^{(1)}(\tilde{M}_G, \tilde{M}_L, \tilde{p}) = \left(\begin{array}{l} 1 - \alpha_G \dfrac{\tilde{M}_G}{1 - \tilde{M}_L} + \beta_G \\ 1 \end{array} \right), \tag{17}$$

$$\tilde{\mathbf{f}}^{(2)}(\tilde{M}_G, \tilde{M}_L, \tilde{p}) = \left(\begin{array}{l} 1 \\ 1 - \alpha_L \dfrac{\tilde{M}_G}{1 - \tilde{M}_L} + \beta_L \end{array} \right), \tag{18}$$

$$\varphi(\tilde{M}_G, \tilde{M}_L, \tilde{p}) = \tilde{M}_L - \gamma. \tag{19}$$

It is useful to separate the parameters F_G and F_L in (11) and (12) in such a way that they may vary independently. Let us set

$$K := \frac{k_G}{k_L}, \tag{20}$$

$$M := \frac{\alpha_L}{\beta_L^2} = \frac{F_G \mathrm{R} T \rho_L}{k_L x P_{out}^2}. \tag{21}$$

We can express α_L and β_G from these two equations,

$$\alpha_L = M\beta_L^2\,, \tag{22}$$

$$\beta_G = K\beta_L. \tag{23}$$

We obtain four dimensionless parametrs, i. e. the row vector $\tilde{q} = (M, K, \beta_L, \gamma)^{\mathrm{T}},\ \tilde{q} \in \mathbb{R}^4$. The resulting dimensionless Filippov system $\tilde{\mathcal{F}}$ then depends only on $\tilde{q}$. It has the form

$$\tilde{\mathcal{F}}(\tilde{q}): \quad \frac{\mathrm{d}}{\mathrm{d}\,\tilde{t}}\begin{pmatrix} \tilde{M}_G \\ \tilde{M}_L \end{pmatrix} = \begin{cases} \tilde{\mathbf{f}}^{(1)}(\tilde{M}_G, \tilde{M}_L, \tilde{q}),\ \varphi(\tilde{M}_G, \tilde{M}_L, \tilde{q}) < 0, \\ \\ \tilde{\mathbf{f}}^{(2)}(\tilde{M}_G, \tilde{M}_L, \tilde{q}),\ \varphi(\tilde{M}_G, \tilde{M}_L, \tilde{q}) > 0, \end{cases} \tag{24}$$

where

$$\tilde{\mathbf{f}}^{(1)}(\tilde{M}_G, \tilde{M}_L, \tilde{q}) = \begin{pmatrix} 1 - KM\beta_L^2 \dfrac{\tilde{M}_G}{1 - \tilde{M}_L} + K\beta_L \\ 1 \end{pmatrix}, \tag{25}$$

$$\tilde{\mathbf{f}}^{(2)}(\tilde{M}_G, \tilde{M}_L, \tilde{q}) = \begin{pmatrix} 1 \\ 1 - M\beta_L^2 \dfrac{\tilde{M}_G}{1 - \tilde{M}_L} + \beta_L \end{pmatrix}, \tag{26}$$

$$\varphi(\tilde{M}_G, \tilde{M}_L, \tilde{q}) = \tilde{M}_L - \gamma. \tag{27}$$

3 Conclusions

We manage to reduce the number of parameters from ten to only four. The system $\tilde{\mathcal{F}}(\tilde{q})$ exhibits a certain slow-fast character. In simulations, it turned out, that two of these parameters, namely M and β_L, substantially affect the behaviour of the system.

Acknowledgement: The work is a part of the research project MSM 6046137306 financed by MSMT, Ministry of Education, Youth and Sports, Czech Republic.

References

[1] K.M. Moudgalya, V. Ryali: *A class of discontinuous dynamical systems I. An ideal gas-liquid system.* Chemical Engineering Science 56, 3595–3609, 2001.

[2] M. Biák, D. Janovská: *Filippov dynamical systems.* In: R. Blaheta, J. Starý (ed.): Seminar on Numerical Analysis & Winter School/Proceedings of the Conference SNA'09, Ostrava, February 2-6, 2009 Appendix, 1–4.

[3] P.T. Piiroinen, Yu.A. Kuznetsov: *An event-driven method to simulate Filippov systems with accurate computing of sliding motions*, ACM Trans. Math. Software 34(13), 1–24, 2008.

Parameter identification in heat flow with a geo-application

R. Blaheta, R. Kohut

Institute of Geonics AS CR, v.v.i., Ostrava, Czech Republic

1 Introduction

Problems of identification of material parameters (mostly parameters appearing in constitutive relations) have application in many fields of engineering including investigation of processes in a rock mass. This paper outlines the structure of parameter identification problems, methods for their solution and describes an identification problem from geotechnics, which will serve as a realistic model example for the showing behaviour of a selected parameter identification method.

Most generally, the identification problems appear in investigation of physical processes in material environment. The processes are described by the *state variables* u and driven by the *control variables* f. The material is characterized by parameters κ. Direct problems focus on computation of $u = u_h(\kappa) = u_h(\kappa, x, t)$, where $(x,\, t)$ gives space and time localization, if f and κ are known. On the opposite, identification problems use the knowledge of f and some partial apriori knowledge on the state variable u for (partial or full) determination of κ.

If the apriori information about the state variable u is given by the vector $d = (d_i)$ of measured values $d_i \sim u(x_i, t_i)$, then the search for the unknown material parameters can be formulated as the following minimization problem

$$f(\kappa) = \| \mathcal{M} u_h(\kappa) - d \| \longrightarrow \min_{\kappa \in \mathcal{K}} . \tag{1}$$

Above, $\mathcal{M}$ is an observation operator, which select from u_h values corresponding to d.

In contrary to direct problems, it is known that some identification problems are not well posed, which means that some of the following properties can be violated:

- there exists solution of the problem,
- the solution is unique,
- the solution is stable under small changes of input data.

Although the properties of the minimization problems can be difficult to analyse, a lot of different iterative techniques can be used for the minimization (1) (mostly without theoretical proof of convergence). The range of applicable methods includes

- gradient methods, e.g. Gauss-Newton, Levenberg-Marquardt, conjugate gradients, see [3], [4], [6], [7],
- gradient-free direct method, e.g. Nelder-Mead simplex method [3],
- stochastic methods e.g. [5], genetic algorithms e.g. [6].

In this paper, we shall show the solution of the identification problem, described in the next section, by means of least-square formulation (1) and application of the Nelder-Mead algorithm.

2 A model identification problem

The in-situ Äspö Pillar Stability Experiment (APSE) has been performed at SKBs Äspö Hard Rock Laboratory in south eastern Sweden with the aid of investigation of granite mass damage due to mechanical and thermal loading. The measured data are now used for validation of mathematical models within the DECOVALEX 2011 international project. APSE used electrical heaters to increase temperatures and induce stresses in a rock pillar between holes (Fig. 1) until its partial failure. To determine accurately the temperature changes, a heat flow model is formulated and monitored temperatures are used for identification of heat flow parameters (heat capacity, heat conduction coefficient, heat convection into the holes). The identification should provide parameters taking into account water bearing fractures and water flow and calibrate the model. More details and another approach to the model calibration can be found in [1].

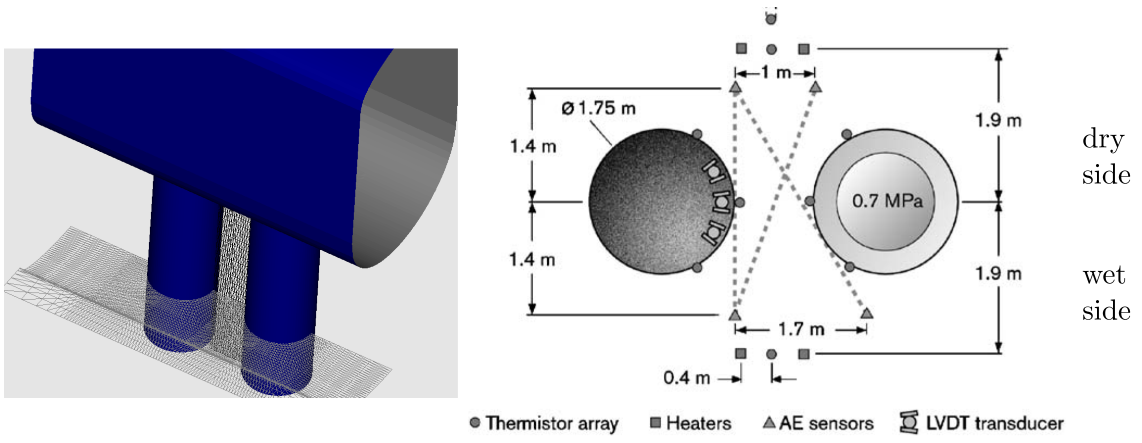


Figure 1: The APSE model - detail of the FE grid around the pillar (GEM software [2]) and plan view on the pillar, holes, location of heaters and points of temperature measurement.

The exploited APSE model, realized by GEM software [2], considers domain of 105 × 125 × 118 m and 99 × 105 × 59 nodes. The grid is refined around the pillar, see Fig. 1. The heaters are producing heat which varies in time. The model assumes original temperature $14.5°C$ on the outer boundaries, zero flux onto the tunnel and nonzero flux given the convection onto the holes. The initial condition is given again by the temperature $14.5°C$.

Monitoring of the temperatures during two month heating phase of APSE is essential for calibration of the thermal model. There are 14 temperature monitoring positions and temperatures are measured in 12 time moments. Altogether 168 values of temperature measurement (vector d) are used for parameter identification, which according to (1) can be written as follows

$$f = f(\lambda_1, c_1, \lambda_2, c_2, \lambda_3, c_3, H_1, H_2, H_3) = \left(\sum_i [u_h(x_i, t_i) - d_i]^2 \right)^{0.5} \longrightarrow \min . \qquad (2)$$

The material parameters represent different conductivity λ and heat capacity c for dry and wet side of model (according to Fig.1.). The rock in the right hole had yielded from a depth of approximately 0.5 m down to 3 m which motivates to introduce third type of material with different λ and c for the damaged part of the pillar. We supposed heat conduction between rock and air in excavated holes determined by different values of the heat conduction coefficient H for individual holes with third coefficient corresponding to surface for the above mentioned damaged part of the pillar. It gives 9 material parameters of the cost functional f in (2).

3 The optimization method and numerical results

For finding the minimum (or at least realizing sufficient decrease) of the cost functional (1), (2) representing agreement between the measured and computed values, we use Nelder-Mead simplex method, see e.g. [3]. To guarantee the positivity of the parameters, we use exponential transformation, i.e. finding x such that $p = e^x$ is the required parameter. As the parameters have quite different orders, we scale the capacity c for having all parameters in order of units.

The Nelder-Mead iterations are stopped when both decrease of the cost functional f is small (below ε_f) and changes of parameters are small (below ε_p). To find very accurate approximation of the parameters, we stop iterations with $\varepsilon_f = 0.001$ and $\varepsilon_p = 0.01$. With a physical initial guess, it requires 764 iterations. The reached minimum value was $f = 33.599$. The obtained material parameters can be seen in Table 1, the convergence behaviour is illustrated in Fig. 2.

We also tested the sensitivity of the cost functional F to change of individual parameters in the vicinity of the computed optimum, i.e. we fixed 8 values from Table 1 and show dependence of f on the remaining one. In Fig. 3, we can see that with respect to λ_1 and c_1, we get stable minimum (a similar observation is for λ_2 and c_2). For H_1 and similarly for λ_3 and c_3, H_2 and H_3 the minimum is unstable.

λ_1	c_1	λ_2	c_2	λ_3	c_3	H_1	H_2	H_3
2.988	2.518e06	4.697	1.167e06	6.556	4.292e06	5.364	5.696	24.901

Table 1: Optimal parameters.

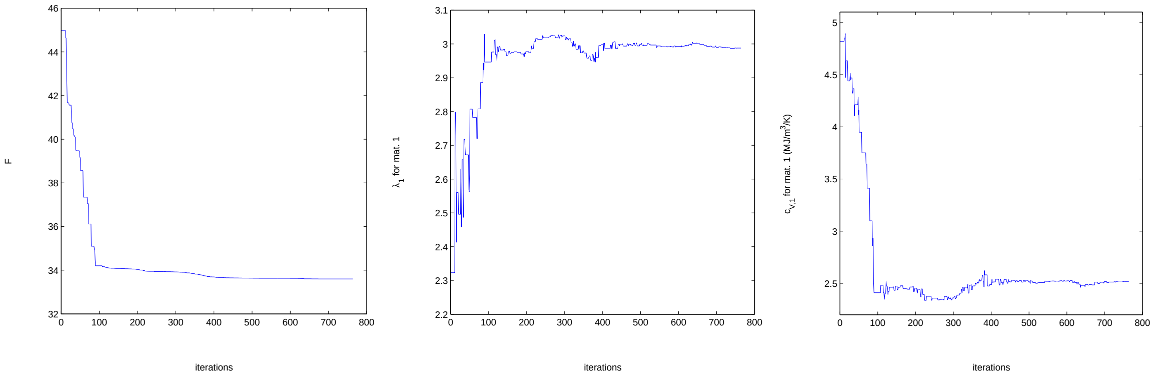


Figure 2: The convergence of - the cost functional F (left), parameter λ_1 (center) and c_1 (right).

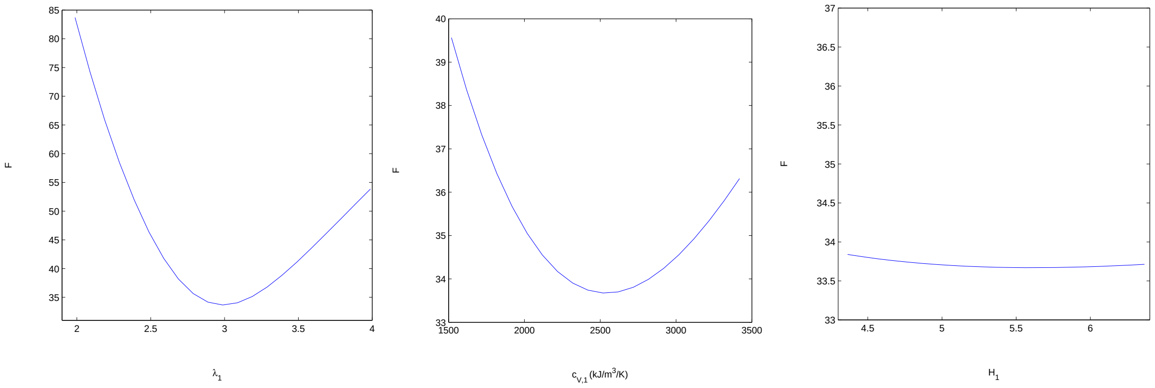


Figure 3: The dependence of the cost functional F on λ_1 (left), c_1 (center) and H_1 (right).

4 Conclusions

The paper describes (1) philosophy of the solution of the identification problems, (2) an application of parameter identification for computation of temperatures in geotechnical problem, where the heat flow take place in complex geologic mass, but where some monitoring data are at disposal (3) behaviour of Nelder-Mead optimization algorithm and question of proper stopping criteria, (4) importance of a suitable choice of parameters to be identified with respect to stability of the minimum of the least-square cost functional. Note that our geotechnical problem can be successfully optimized with only four parameters (λ_1, c_1, λ_2, c_2). The Nelder-Mead method was observed to be able to converge with both physical and non-physical initial guess.

For future, similar identification problems will be applied to another geotechnical problems. As the computational cost is relative high (about 100 iterations requiring solution of mostly one direct problem), we would like to test also the other optimization techniques, especially those involving higher level of parallelism. So far, our method is implemented in GEM software with parallelism exploited in solving the linear systems.

Acknowledgement: The work was conducted within the international DECOVALEX-2011 Project (DEvelopment of COdes and their VALidation against EXperiments). The authors are grateful to the Funding Organisations: CAS, JAEA, KAERI, POSIVA, SKB, TUL and WHU. The views expressed in the paper are however, those of the authors and are not necessarily those of the Funding Organisations. This work is also supported by the research plan AV0Z30860518 of the Academy of Sciences of the Czech Republic.

References

[1] J Ch. Andersson, B. Fälth, O. Kristensson: *Äspö pillar stability experiment TM back calculation.* Advances on Coupled Thermo-Hydro-Mechanical-Chemical Processes in Geosystems and Engineering, HoHai University, Nanjing, China, 675–680, 2006.

[2] R. Blaheta, O. Jakl, R. Kohut, J. Starý: *GEM – a Platform for Advanced Mathematical Geosimulations.* In: Proceedings of the Conference Parallel Processing and Applied Mathematics, PPAM 2009, to appear.

[3] C.T. Kelley: *Iterative Methods for Optimization.* SIAM, Philadelphia 1999.

[4] M.N. Özisik, H.R.B. Orlande: *Inverse Heat Transfer: Fundamentals and Applications.* Taylor and Francis, NY 2000.

[5] R. Mahnken: *Identification of Material Parameters for Constitutive Equations. Encyclopaedia of Computational Mechanics.* Edited by Erwin Stein, Rene de Borst and Thomas J.R. Hughes. Volume 2: Solids and Structures. John Wiley, Chichester 2004.

[6] C. Rechea, S. Levasseur, R. Finno: *Inverse analysis techniques for parameter identification in simulation of excavation support systems.* Computers in Geotechnics 35, 331–345, 2008.

[7] G. Rus, R. Gallego: *Optimization algorithms for identification inverse problems with the boundary element method.* Eng. Analysis with Boundary Elements 26, 315–327, 2002.

Numerical schemes for river flood modelling

M. Brandner, J. Egermaier, H. Kopincová

Departments of Mathematics
University of West Bohemia, Plzeň

1 Introduction

The river flow models are often formulated as one-dimensional problems. In the case of the river flood simulations, it is more convenient to use two-dimensional approach. There are a lot of efficient numerical schemes with different properties. In addition to important properties like conservation, consistency and stability these numerical schemes should satisfy some other ones - positive semidefinitness and computational efficiency especially for wet/dry problems which occur on the whole shoreline.

2 Mathematical model

For the river flood modelling we use two dimensional Saint-Venant equations with the frictional terms

$$\begin{aligned} h_t + (hu)_x + (hv)_y &= 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x + (huv)_y &= -ghB_x - gM^2 \frac{hu\sqrt{(hu)^2+(hv)^2}}{h^{7/3}} \\ (hv)_t + (huv)_x + \left(hv^2 + \frac{1}{2}gh^2\right)_y &= -ghB_y - gM^2 \frac{hv\sqrt{(hu)^2+(hv)^2}}{h^{7/3}}, \end{aligned} \tag{1}$$

where $h = h(x, y, t)$ is the unknown water level, $u = u(x, y, t)$ and $v = v(x, y, t)$ are the orthogonal velocities of the water flow in the x and y directions, $g = 9.81$, $B = B(x, y)$ represents the bottom topography and M is the Mannings coefficient depending on the substrate.

The system can be simply written in the matrix form

$$\mathbf{u}_t + [\mathbf{f}(\mathbf{u})]_x + [\mathbf{g}(\mathbf{u})]_y = \boldsymbol{\psi}(\mathbf{u}, x, y), \tag{2}$$

In the following we suppose the Saint-Venant equations without the frictional terms (the terms containing Mannings coefficient). This terms can be included by fractional stepping.

3 Numerical schemes

We use finite volume methods with the integral averages of the unknown functions on the cells $D_{ij} = [x_{j-1/2,k}, x_{j+1/2,k}] \times [y_{j,k-1/2}, y_{j,k+1/2}]$ of the rectangular grid with the steps Δx and Δy.

$$\mathbf{U}^n_{j,k} \approx \frac{1}{\Delta x \Delta y} \int\limits_{D_{ij}} \mathbf{u}(x,y,t_n)dxdy, \quad \mathbf{F}^n_{j+1/2,k} \approx \tfrac{1}{\Delta t} \int\limits_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{u}(x_{j+1/2},y_k,t))dt,$$

$$\mathbf{G}^n_{j,k+1/2} \approx \frac{1}{\Delta t} \int\limits_{t_n}^{t_{n+1}} \mathbf{g}(\mathbf{u}(x_j,y_{k+1/2},t))dt, \quad \mathbf{\Psi}^n_{j,k} \approx \tfrac{1}{\Delta x \Delta y \Delta t} \int\limits_{t_n}^{t_{n+1}} \int\limits_{D_{ij}} \boldsymbol{\psi}(\mathbf{u},x,y)dxdydt. \tag{3}$$

3.1 Central-upwind

For updating the unknown functions we use the scheme in the form

$$\frac{d}{dt}\mathbf{U}_{j,k} + \frac{1}{\Delta x}[\mathbf{F}_{j+1/2,k} - \mathbf{F}_{j-1/2,k}] + \frac{1}{\Delta y}[\mathbf{G}_{j,k+1/2} - \mathbf{G}_{j,k-1/2}] = \mathbf{\Psi}_{j,k}, \tag{4}$$

with the consistent numerical fluxes. It is also important choose a suitable reconstruction of the unknown functions. In this case we use the following one (for water level)

$$H^-_{j+1/2,k} = \max(0, H_{j,k} + B_{j,k} - B_{j+1/2,k}), H^+_{j+1/2,k} = \max(0, H_{j+1,k} + B_{j+1,k} - B_{j+1/2,k}), \tag{5}$$

$$H^-_{j,k+1/2} = \max(0, H_{j,k} + B_{j,k} - B_{j,k+1/2}), H^+_{j,k+1/2} = \max(0, H_{j,k+1} + B_{j,k+1} - B_{j,k+1/2}), \tag{6}$$

where

$$B_{j+1/2,k} = \max(B_{j,k}, B_{j+1,k}), \qquad B_{j,k+1/2} = \max(B_{j,k}, B_{j,k+1}). \tag{7}$$

This reconstruction ensures positive semidefinitness of the method and allows us to solve problem of dry states (solution between wet and dry cells) by the same procedure as problem between two wet cells. This has the positive influence on the computing time.

In order to preserve special steady state "rest at lake"($u = v = 0$ a $h + B =$ const.) it is used special discretization of the source term (see [2]). In this steady state we have

$$\left(\frac{1}{2}gh^2\right)_x = -ghB_x, \qquad \left(\frac{1}{2}gh^2\right)_y = -ghB_y. \tag{8}$$

By the integrating (8) we obtain

$$-\int\limits_{x_{j-1/2}}^{x_{j+1/2}} ghB_x \mathrm{d}x \approx \frac{1}{2}g(H^-_{j+1/2,k})^2 - \frac{1}{2}g(H^+_{j-1/2,k})^2, \tag{9}$$

$$-\int\limits_{y_{k-1/2}}^{y_{k+1/2}} ghB_y \mathrm{d}x \approx \frac{1}{2}g(H^-_{j,k+1/2})^2 - \frac{1}{2}g(H^+_{j,k-1/2})^2. \tag{10}$$

Therefore the approximation of the source term has the form

$$\mathbf{\Psi}_{j,k} = \begin{bmatrix} 0 \\ \frac{g}{2\Delta x}\left((H^-_{j+1/2,k})^2 - (H^+_{j-1/2,k})^2\right) \\ \frac{g}{2\Delta y}\left((H^-_{j,k+1/2})^2 - (H^+_{j,k-1/2})^2\right) \end{bmatrix} \tag{11}$$

and it is consistent in the following sense

$$\Delta x \mathbf{\Psi}_j = -gh_j \Delta B_j + O(\Delta B_j). \tag{12}$$

3.2 Augmented system

This method is in detail described in [1]. It is based on augmented formulation (we add the fluxes and function $B(x, y)$ as the unknown functions). Then we solve the system

$$\mathbf{w}_t + \mathbf{C}(\mathbf{w})\mathbf{w}_x + \mathbf{D}(\mathbf{w})\mathbf{w}_y = \mathbf{0}, \tag{13}$$

where the vector of unknown functions is

$$\mathbf{w} = \left[h, hu, hv, huv, hu^2 + \frac{1}{2}gh^2, hv^2 + \frac{1}{2}gh^2, B\right]^T. \tag{14}$$

The method is based on the approximate Riemann solver which decomposes the jumps of unknown function and then we construct the fluctuations

$$\mathbf{C}^-\mathbf{W}^{\pm}_{j+1/2,k} = \sum_{p=1}^{7} \min\{s^p_C, 0\}\alpha^p_C \mathbf{r}^p_C, \qquad \mathbf{C}^+\mathbf{W}^{\pm}_{j+1/2,k} = \sum_{p=1}^{7} \max\{s^p_C, 0\}\alpha^p_C \mathbf{r}^p_C, \tag{15}$$

$$\mathbf{D}^-\mathbf{W}^{\pm}_{j,k+1/2} = \sum_{p=1}^{7} \min\{s^p_D, 0\}\alpha^p_D \mathbf{r}^p_D, \qquad \mathbf{D}^+\mathbf{W}^{\pm}_{j,k+1/2} = \sum_{p=1}^{7} \max\{s^p_D, 0\}\alpha^p_D \mathbf{r}^p_D, \tag{16}$$

where s^p_C and s^p_D are approximations of wave speeds, $\mathbf{r}^p_C$ and $\mathbf{r}^p_D$ are approximations of the eigenvectors od Jacobian matrixes and α^p_C and α^p_D are coefficients based on jumps decompositions. To update the solution we use the scheme

$$\mathbf{W}^{n+1}_{j,k} = \mathbf{W}^n_{j,k} - \frac{\Delta t}{\Delta x}(\mathbf{C}^+\mathbf{W}^{\pm}_{j-1/2,k} + \mathbf{C}^-\mathbf{W}^{\pm}_{j+1/2,k}) - \frac{\Delta t}{\Delta y}(\mathbf{D}^+\mathbf{W}^{\pm}_{j,k-1/2} + \mathbf{D}^-\mathbf{W}^{\pm}_{j,k+1/2}). \tag{17}$$

Special approximations of the eigenvectors of the approximate Jacobian matrix of the augmented system ensures preserving all steady states, if one of velocities u or v is identically zero.

One of the most important problems in river flood modelling is correct solution of dry cells problem. Suppose $H_L > 0$ and $H_R = 0$ in one direction. If we use the method on the wet/dry front by the standard way (i.e. like for solution between two wet cells), it can produce spurious results. Especially in the cases where $H_L + B_L < B_R$ can be incorrectly inundate some dry cells. That we can determine the correctly inundate cells, in [1] there is described additional Riemann problem to obtain the middle state h^*. This problem is defined

$$B_R = B_L = 0, \qquad H_R = H_L, \qquad U_R = -U_L. \tag{18}$$

Then the middle state h^* is

$$h^* = \frac{(HU)_L - (HU)_R + s^2 H_R - s^1 H_L}{s^2 - s^1} = H_L + \frac{H_L U_L}{\sqrt{gH_L}}, \tag{19}$$

because the consistent speeds in this problem are $s^1 = -\sqrt{gH_L}$ and $s^2 = \sqrt{gH_L}$. If $h^* > B_R$ the the right cell will be inundate and we can solve the Riemann problem by the method of augmented system with the original values. However, if $h^* \leq B_R$ then the right cell remains dry and we use only left going waves from the additional Riemann problem to update the left cell.

If we solve the additional Riemann problem the middle state h^* represents the maximum of the water elevation. The solution of the additional Riemann problem by the method of augmented system is

$$H^{n+1}_L = H_L + \frac{\Delta t}{\Delta x} H_L U_L. \tag{20}$$

It is easy to see, that the $H_L^{n+1} = h^*$ only if $\frac{\Delta t}{\Delta x} = \frac{1}{\sqrt{gH_L}}$. But the time step Δt has to satisfy the CFL stability condition

$$\max_p \{s_j^p\} \frac{\Delta t}{\Delta x} \le 1, \qquad \forall j \tag{21}$$

so the value $H_L^{n+1} \le h^*$ and give more accurate information if water level is so high to inundate the dry cell in the time $t_n + \Delta t$.

4 Conclusion

We use two numerical schemes for river flood modelling. The central-upwind method is very robust and due to special reconstruction of unknown functions is positive semidefinite and solves the problems on the wet/dry front without any additional conditions. But it preserves only special steady state "rest at lake". The method of augmented system preserves general steady states in one-dimensional problems and some steady states in the two-dimensional ones. But it is necessary to solve additional problem to correct inundation of dry cells. The complete algorithm is more sophisticated but also complicated and it needs longer computing time.

Acknowledgement: This work has been supported by the Research Plan MSM 4977751301 and by Moravian-Silesian region.

References

[1] D.L. George: *Finite Volume Methods and Adaptive Refinement for Tsunami Propagation and Inundation.* University of Washington, Ph.D. Thesis, 2006.

[2] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, B. Perthame: *A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow eater flows.* SIAM Journal on Scientific Computing, Vol. 25 (6), 2050–2065, 2004.

Selection strategy for fixing nodes in FETI-DP method

J. Brož, J. Kruis

Department of Mechanics, Faculty of Civil Engineering
Czech Technical University in Prague

1 Introduction

Nowadays, large scale numerical analyses are popular in the engineering community. These analyses have large demands on the computer capacity. It brings necessity of using of parallel computers. Parallel computers offer large computer memory capacity and computer power. Domain decomposition methods are the most popular numerical methods for solution of wide spectrum of engineering problems on parallel computers. The FETI-DP method is one of non-overlapping domain decomposition methods. This contribution deals with selection strategy for fixing nodes in FETI-DP method. Fixing nodes in the FETI-DP method are needed for non-singular subdomain matrices.

2 FETI-DP method

The FETI-DP (Dual-Primal Finite Element Tearing and Interconnecting) method is one of non-overlapping domain decomposition methods. The method decomposes the original domain into smaller subdomains. This method was introduced by Farhat and coworkers in the article [1]. Development of the method was motivated by difficulties with singular matrices in the original FETI method and complicated modifications due to time-dependent problems with mass or capacity matrices. The FETI-DP method is based on combination of the FETI method and the Schur complement method. The unknowns in the problem are split into two parts. Namely, the fixing and remaining unknowns. The remaining unknowns are further split into the internal and interface unknowns. The continuity condition among subdomain boundaries is enforced by Lagrange multipliers, which are defined between interface remaining unknowns. In the case of fixing nodes, the continuity is enforced by a special ordering of unknowns. Internal unknowns are eliminated and a coarse problem is obtained. The coarse problem is solved by the conjugate gradient method. More information about the FETI-DP method can be found in the article [1] or in the book [3].

3 Fixing nodes

Selection of the fixing unknowns deserves a special attention. The fixing unknowns have to be defined in such a way that the subdomain matrix obtained after removing of the rows and columns belonging to the fixing unknowns is nonsingular. It is clear that there are many possibilities of the fixing unknown definition.

In the case of regular rectangular domains and subdomains, the definition of the fixing unknowns is simple. The unknowns are defined in the corners of the subdomains. In all other cases, the situation is more complicated. Recently, strong influence of the definition of the fixing unknowns on the condition number of the subdomain matrix has been observed [2]. The large condition numbers of subdomain matrices significantly deteriorate the convergence of the iterative methods used for the solution of the coarse problem.

4 Algorithm for selection of fixing nodes

4.1 Algorithm for 2D problems

The proposed algorithm for fixing node selection in 2D has three steps. It is based purely on the knowledge of finite element mesh. In the first step, nodes belonging to more than two subdomains are selected. When the fixing nodes are selected, the number of fixing nodes on each subdomain is checked. Plane strain and plane stress problems require two different interface nodes, three nodes are better for plate problems. Therefore, the minimum number of nodes is three. If there are enough nodes, their mutual distances are computed and compared with estimates of subdomain lengths. If the selected nodes are too close each other, the subdomain matrix has usually very large condition number. If there are subdomains with less than the minimum number of fixing nodes or if the selected fixing nodes do not satisfy geometric conditions, the second step of the algorithm is performed. Interface nodes with only one adjacent interface node are selected as additional fixing nodes. This step selects nodes, where some interface curve starts. The number of selected nodes after two steps of the algorithm can be assumed as the minimum number of nodes. From the mechanical point of view, selected nodes can be assumed as fixed nodes. Additional fixing nodes can be obtained by the third step of the selection algorithm. In order to select additional nodes, nodes belonging to interface curves have to be found. These nodes are denoted as the interface curve nodes (IC nodes). The first and last interface curve nodes are selected yet. Additional nodes can be selected as

- the node closest to the center of the interface curve,
- every n-th node,
- randomly selected node.

4.2 Algorithm for 3D problems

The algorithm for selection of fixing nodes in 3D is based on the nodal multiplicity. The nodal multiplicity of the node is the number of subdomains which share the node. Maximum nodal multiplicity is established before selection of fixing nodes. Afterwards nodes with maximum nodal multiplicity on each subdomain are selected. If there is the minimum number of fixing nodes on each subdomain, the minimum number of nodes in 3D is three, then the selection process finishes. Selection process continues in all other cases until there is the minimum number of fixing nodes. This process starts from maximum nodal multiplicity minus one and continuous to nodal multiplicity which is equal to three until there is the number of fixing nodes on each subdomain greater than three.

If fixing nodes are not chosen by steps with nodal multiplicity then the choice of fixing nodes is based on their geometrical properties. One interface node is selected as the fixing node on the first subdomain. Two different nodes with maximum distance from the first fixing node are selected. These nodes are denoted as the fixing nodes on all neighbor subdomains. A subdomain with at least one fixing node from the previous step is taken into account now. Additional fixing nodes are selected in such a way that their distance from the existing fixing nodes is maximized. This approach is used recursively and at the end of it, there are at least three fixing nodes on each subdomain. Furthermore, the fixing nodes on each subdomain are spread over the subdomain and such positions lead to relatively small condition number of subdomain matrices.

5 Numerical examples in 2D

An irregular domain (called Storey) was chosen in order to check whether the algorithm selects enough fixing nodes which satisfy geometrical conditions. The plane stress linear elasticity problem is assumed. The shape of the domain is depicted in Figure 1. Several densities of finite element mesh were used.

The test results are shown in Figures 3 and 4. The number of iterations of the conjugate gradient method solving the coarse problem with respect to the number of the fixing nodes are plotted in Figure 3. All graphs show that the increasing number of fixing nodes decreases the number of iterations of the conjugate gradient method. The time of solution of the coarse problem therefore also decreases. The total time is decreasing at the beginning but later, it starts to grow due to factorization of the submatrix which contains unknowns defined on fixing nodes. The total time is depicted in Figure 4.

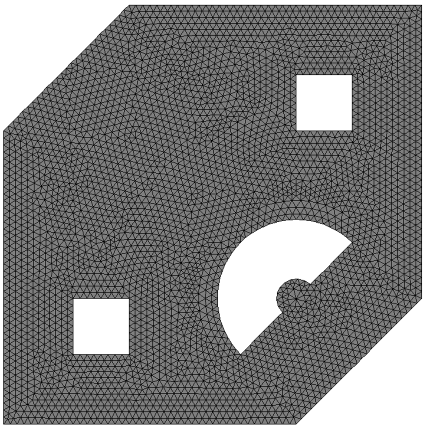


Figure 1: Storey: Original domain.

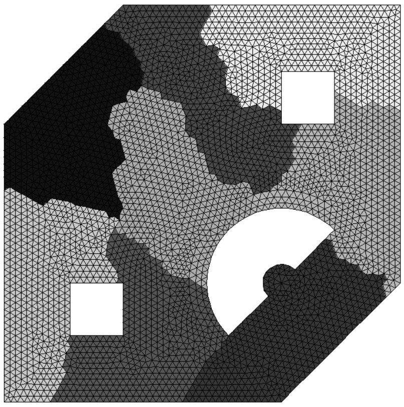


Figure 2: Storey: Mesh decomposed into 8 subdomains.

6 Conclusion

The algorithm for selection of the fixing nodes, which are used in the FETI-DP method, was developed and tested for two dimensional problems. The algorithm was implemented into open source code SIFEL providing parallel computations. The selection algorithm was tested on several regular and irregular domains decomposed into regular and irregular subdomains. It was observed that the minimum as well as maximum number of fixing unknowns is not optimal with respect to elapsed time. The higher number of fixing nodes decreases the number of iterations and reduces time of the factorization of the subdomain matrices. Numerical experiments show that some additional nodes in 2D, e.g. in the center of each interface curve, lead to optimal elapsed times.

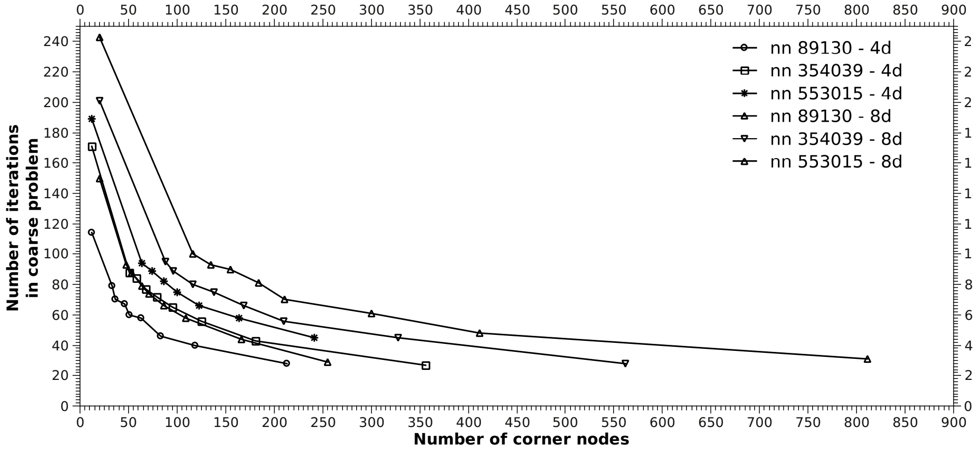


Figure 3: Storey: The number of iterations in the coarse problem.

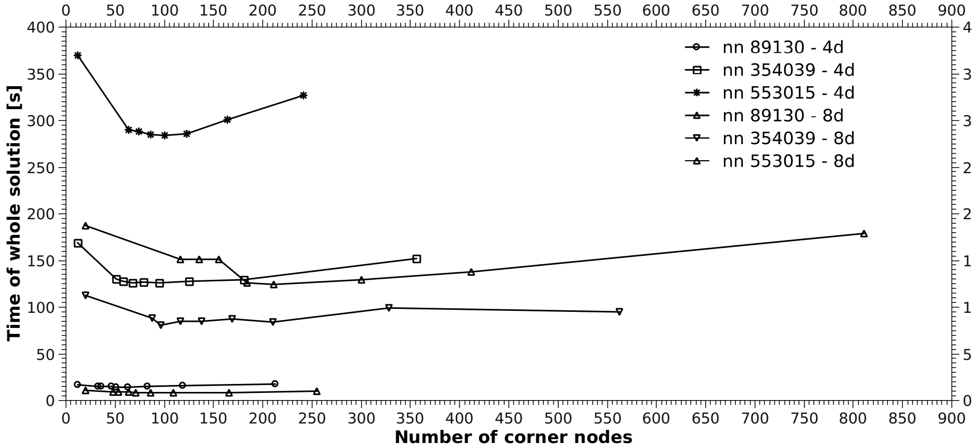


Figure 4: Storey: Time of whole solution.

The algorithm for selection of fixing nodes in 3D is still under development. Recent tests show that there is the same behavior in the case of the number of iterations as in the 2D.

Acknowledgement: This work has been supported by the grant GA ČR number 103/09/H078.

References

[1] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, D. Rixen: *FETI-DP: a Dual-Primal Unified FETI Method—Part I: A Faster Alternative to the Two-Level FETI Method.* In: International Journal for Numerical Methods in Engineering, Vol. 50, 1523–1544, 2001.

[2] P. Kabelíková, Z. Dostál, T. Kozubek, A. Markopoulos: *Generalized Inverse Matrix Evaluation Using Graph Theory.* In: Blaheta R., Starý J. (ed.): Proceedings of the Modelling 2009, Institute of Geonics AS CR, Czech Republic, 2009.

[3] J Kruis: *Domain Decomposition Methods for Distributed Computing*, 1st ed., Saxe-Coburg Publications, Kippen, Stirling, Scotland, 2006.

Canard traveling waves of the spruce budworm population model

L. Buřič, A. Klíč

Department of Mathematics, Institute of Chemical Technology
Technická 5, Prague 6, 166 28, Czech Republic

1 Introduction

We study a mathematical model of the spruce budworm (choristoneura fumiferana) population distribution in the nonconstant environment. See e.g. [1, 2] for introduction to the modeling of the spruce budworm population dynamics. The problem is formulated as the system of two nonlinear parabolic partial differential equations (PDE's) on two-dimensional infinite domain,

$$\varepsilon \frac{\partial b}{\partial t} = \varepsilon^2 \left(\frac{\partial^2 b}{\partial x^2} + \frac{\partial^2 b}{\partial y^2} \right) + f(b, s, \alpha, \gamma)\,, \tag{1a}$$

$$\frac{\partial s}{\partial t} = g(b, s, \delta)\,, \tag{1b}$$

where the functions f and g describing the population dynamics are defined as follows,

$$f(b, s, \alpha, \gamma) = b \left(1 - \frac{b}{\alpha s} \right) - \frac{1}{\gamma} \frac{b^2}{s^2 + b^2}\,, \quad g(b, s, \delta) = s\,(1 - s) - \frac{1}{\delta} b\,.$$

The system (1) is in dimensionless form which was proposed in [3, Section 5.2]. The functions $b(x, y, t)$ and $s(x, y, t)$ define the spruce budworm population density and the foliage population density in the position (x, y) at the time t, respectively. The problem (1) has four positive parameters α, γ, δ and ε. It is important that $0 < \varepsilon \ll 1$.

We are interested in the traveling wave solutions of the system (1). The traveling wave solutions are located as special solutions of a system of ordinary differential equations (ODE's) obtained by the moving coordinate transformation, see e.g. [4, Section 1.5 and Chapter 3].

Let us set $b(x, y, t) = u(\xi)$, $s(x, y, t) = v(\xi)$. The moving coordinate ξ is defined by the relation $\xi = \langle \boldsymbol{n}, \boldsymbol{x} \rangle - ct$, where $\boldsymbol{x} = (x, y)$, $c > 0$ is the unknown wave velocity, and $\boldsymbol{n}$ is the unit vector specifying the direction of the traveling wave propagation. Using new coordinates, we can rewrite the system (1) as the system of the second order ODE's

$$\varepsilon^2 \frac{\mathrm{d}^2 u}{\mathrm{d}\xi^2} + \varepsilon c \frac{\mathrm{d}u}{\mathrm{d}\xi} + f(u, v, \alpha, \gamma) = 0\,, \tag{2a}$$

$$c \frac{\mathrm{d}v}{\mathrm{d}\xi} + g(u, v, \delta) = 0\,. \tag{2b}$$

2 Slow–fast system formulation

In this section we formulate and analyze the system (2) as a *slow–fast system* of ODE's. The slow–fast systems of ODE's are also called the singularly perturbed systems. See e.g. [5] for introduction to the geometric theory of the singularly perturbed systems.

Let us set $p_1 = u$, $p_2 = \varepsilon \frac{\mathrm{d}u}{\mathrm{d}\xi}$, $q = v$. The system (2) is then equivalent to the system of the first order ODE's

$$\varepsilon \frac{\mathrm{d}p_1}{\mathrm{d}\xi} = p_2 \,, \tag{3a}$$

$$\varepsilon \frac{\mathrm{d}p_2}{\mathrm{d}\xi} = -cp_2 - f(p_1, q, \alpha, \gamma) \,, \tag{3b}$$

$$\frac{\mathrm{d}q}{\mathrm{d}\xi} = -\frac{1}{c} g(p_1, q, \delta) \,. \tag{3c}$$

The system (3) is a three-dimensional slow–fast system (in the slow "time" scale) with two fast variables p_1, p_2 and one slow variable q. The term "time" is not meant literally because ξ is in fact the spatial coordinate. After rescaling the independent variable $\vartheta = \xi/\varepsilon$ one can obtain the corresponding system in the fast "time" scale,

$$\frac{\mathrm{d}p_1}{\mathrm{d}\vartheta} = p_2 \,, \tag{4a}$$

$$\frac{\mathrm{d}p_2}{\mathrm{d}\vartheta} = -cp_2 - f(p_1, q, \alpha, \gamma) \,, \tag{4b}$$

$$\frac{\mathrm{d}q}{\mathrm{d}\vartheta} = -\frac{\varepsilon}{c} g(p_1, q, \delta) \,. \tag{4c}$$

The so called *reduced problem* is obtained by taking the system (3) in the limit $\varepsilon \to 0$,

$$0 = p_2 \,, \tag{5a}$$

$$0 = -cp_2 - f(p_1, q, \alpha, \gamma) \,, \tag{5b}$$

$$\frac{\mathrm{d}q}{\mathrm{d}\xi} = -\frac{1}{c} g(p_1, q, \delta) \,. \tag{5c}$$

The reduced problem is actually a dynamical system on the set

$$S_0(\alpha, \gamma) = \big\{ (p_1, p_2, q) \in \mathbb{R}^3 \mid p_2 = 0 \,, f(p_1, q, \alpha, \gamma) = 0 \big\}$$

called the *critical manifold* of the system (3). Assume additionally that $p_1, q > 0$. Since

$$\frac{\partial f}{\partial q}(p_1, q, \alpha, \gamma) = \frac{p_1^2}{\alpha q^2} + \frac{2qp_1^2}{\gamma(q^2 + p_1^2)^2} > 0 \,,$$

then by the Implicit Function Theorem, the critical manifold $S_0(\alpha, \gamma)$ is a smooth curve lying in the plane $p_2 = 0$.

The critical manifold $S_0(\alpha, \gamma)$ is also the set of the equilibria of the *layer problem*

$$\frac{\mathrm{d}p_1}{\mathrm{d}\vartheta} = p_2 \,, \tag{6a}$$

$$\frac{\mathrm{d}p_2}{\mathrm{d}\vartheta} = -cp_2 - f(p_1, q, \alpha, \gamma) \,, \tag{6b}$$

$$\frac{\mathrm{d}q}{\mathrm{d}\vartheta} = 0 \,, \tag{6c}$$

obtained by taking the system (4) in the limit $\varepsilon \to 0$. Let J_ε denote the Jacobian matrix of the right-hand sides of the system (4). Furthermore, let us denote $\lambda_i(\varepsilon)$, $i = 1, 2, 3$, the eigenvalues of the matrix J_ε. We compute the eigenvalues of the matrix J_0,

$$\lambda_1(0) = 0\,, \quad \lambda_{2,3}(0) = \frac{-c \pm \sqrt{c^2 - 4\frac{\partial f}{\partial p_1}}}{2}\,,$$

where J_0 is the Jacobian matrix of the right-hand sides in (6).

The critical manifold $S_0(\alpha, \gamma)$ is divided into the *normally hyperbolic segments*, see e.g. [5], by points at which $\frac{\partial f}{\partial p_1}(p_1, q, \alpha, \gamma) = 0$. The normally hyperbolic segment of $S_0(\alpha, \gamma)$ is stable if $Re\lambda_{2,3}(0) < 0$ i.e., if $\frac{\partial f}{\partial p_1}(p_1, q, \alpha, \gamma) > 0$, and it is unstable if $\frac{\partial f}{\partial p_1}(p_1, q, \alpha, \gamma) < 0$. Therefore, we define the stable part of the critical manifold

$$S_{0,s}(\alpha, \gamma) = \left\{ (p_1, p_2, q) \in S_0(\alpha, \gamma) \mid \frac{\partial f}{\partial p_1}(p_1, q, \alpha, \gamma) > 0 \right\}\,,$$

and the unstable part of the critical manifold

$$S_{0,u}(\alpha, \gamma) = \left\{ (p_1, p_2, q) \in S_0(\alpha, \gamma) \mid \frac{\partial f}{\partial p_1}(p_1, q, \alpha, \gamma) < 0 \right\}\,,$$

respectively. It follows from the continuous dependence of the eigenvalues of the matrix J_ε on ε that the sign of $Re\lambda_{2,3}(\varepsilon)$ is preserved on normally hyperbolic segments of the critical manifold for sufficiently small $\varepsilon > 0$. The sign of the eigenvalue $\lambda_1(\varepsilon)$ for sufficiently small $\varepsilon > 0$ can be determined by the following proposition.

Proposition 1 *Let $\frac{\partial f}{\partial p_1} \neq 0$. Then, the eigenvalue $\lambda_1(\varepsilon)$ has the following asymptotic expansion,*

$$\lambda_1(\varepsilon) = -\left(c\frac{\partial f}{\partial p_1}\right)^{-1} \det \begin{bmatrix} \frac{\partial f}{\partial p_1} & \frac{\partial f}{\partial q} \\ \frac{\partial g}{\partial p_1} & \frac{\partial g}{\partial q} \end{bmatrix} \varepsilon + O(\varepsilon^2)\,.$$

See [3, Section 5.2.2] for the proof.

3 Canard traveling pulse

In this section, we present results obtained by the numerical bifurcation analysis of the system (3) for $\gamma = 0.64$, $\delta = 10$, $\varepsilon = 0.01$. We consider the wave velocity $c \gg \varepsilon$. Namely, we set $c = 1$. We declare that all numerical computations were performed with AUTO-07p software, see [6].

A Hopf bifurcation was detected in the course of the numerical continuation of the equilibria of the system (3) in dependence on the parameter α. The detected bifurcation value is $\alpha_{HB} = 9.27003$. At Hopf bifurcation point a branch of the periodic trajectories emerges. We observed that the periodic trajectory appeared via the Hopf bifurcation undergoes the so called *canard explosion*, see e.g. [7]. The canard periodic trajectories are characterized by that they follow both the stable and the unstable part of the critical manifold.

The canard explosion in the system (3) occurs in the parametric region where three equilibria exist. One of them is a saddle point lying on the stable part of the critical manifold $S_{0,s}(\alpha, \gamma)$. Due to Proposition 1, this saddle has one-dimensional unstable invariant manifold and two-dimensional stable invariant manifold. We observed that the branch of the canard periodic trajectories tends to a canard homoclinic trajectory depicted in Figure 1 (on the left) for $\alpha = 9.23169$. Numerically, the period $T \to 1.43547 \cdot 10^{10}$ and the L_2-norm of the periodic solution tends to the L_2-norm of the saddle point lying on the stable part of the critical manifold. The

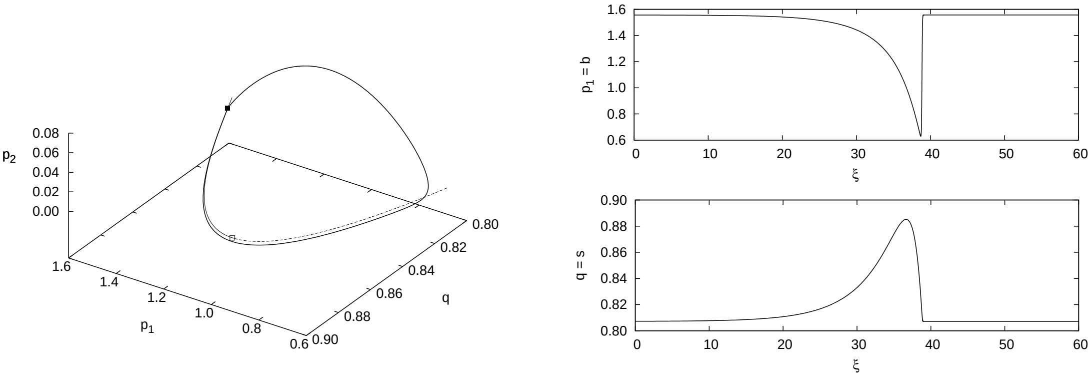


Figure 1: On the left: Canard homoclinic trajectory of the system (3); thick solid line – homoclinic trajectory, thin solid line – stable part of the critical manifold, dashed line – unstable part of the critical manifold, full square – saddle equilibrium, empty square – fold point of the critical manifold. On the right: Canard traveling pulse of the system (1); $c = 1$, $\alpha = 9.23169$, $\gamma = 0.64$, $\delta = 10$, $\varepsilon = 0.01$

traveling pulse of the system (1) corresponding to the homoclinic trajectory of the system (3) is plotted in Figure 1 (on the right). The traveling pulse belongs to a new class of the traveling waves, the so called *canard traveling waves*. This term was introduced in [8]. Note that the traveling pulse mediates the decrease of the spruce budworm population.

Acknowledgement: The research was supported by the grant MSM 6046137306 of the Ministry of Education, Youth and Sports, Czech Republic.

References

[1] D. Ludwig, D. D. Jones, C. S. Holling: *Qualitative Analysis of Insect Outbreak Systems: The Spruce Budworm and Forest.* Journal of Animal Ecology **47**, 315–332 (1978).

[2] D. Ludwig, D. G. Aronson, H. F. Weinberger: *Spatial Patterning of the Spruce Budworm.* Journal of Mathematical Biology **8**, 217–258 (1979).

[3] L. Buřič: *Canard Solutions and Traveling Waves in Reaction-Diffusion Systems.* Ph.D. Thesis, Institute of Chemical Technology, Prague, 2009.
`http://web.vscht.cz/buricl/pub/thesis.pdf`

[4] P. Grindrod: *The Theory and Applications of Reaction-Diffusion Equations.* Oxford University Press, New York, 1996.

[5] C. K. R. T. Jones: *Geometric Singular Perturbation Theory.* In: R. Johnson (Ed.): Dynamical Systems, Lecture Notes in Mathematics 1609, Springer-Verlag, Berlin, 1995, pp. 44–118.

[6] E. J. Doedel, B. E. Oldeman: *AUTO-07P: Continuation and Bifurcation Software for Ordinary Differential Equations.*
`http://indy.cs.concordia.ca/auto/`

[7] M. Krupa, P. Szmolyan: *Relaxation Oscillation and Canard Explosion.* Journal of Differential Equations **174**, 312–368 (2001).

[8] K. R. Schneider, E. A. Shchepakina, V. A. Sobolev: *New type of travelling wave solutions*, Mathematical Methods in the Applied Sciences **26**, 1349–1361 (2003).

Solving an elasto-plastic problem by Newton-like methods

P. Byczanski, S. Sysala

Institute of Geonics AS CR, v.v.i.

1 Introduction

In this contribution, we will apply the semismooth Newton methods with and without damping to solving a one-time step problem in elesto-plasticity. First, we briefly describe the elasto-plastic model and formulate the corresponding one-time step problem in the form of the non-linear variational equation. Then we characterize the used numerical methods. Finally, we illustrate the methods on a 2D numerical example.

2 Elasto-plasticity with hardening

Elasto-plastic problems are the so-called quasi-static problem where the history of loading is taken into account. We consider the von Mises plasticity with linear isotropic strain hardening and with the associative plastic flow rule, see [1, 4]. We use the implicit return mapping scheme to the time discretization and the finite element method with linear simplex elements, see [1, 7].

Let us denote the space of continuous and piecewise linear functions by V_h which approximates the space of all admissible displacements. Let $0 = t_0 < t_1 < \ldots < t_k < \ldots < t_N = T$ be a partition of the time interval $[0, T]$. Then the problem after time and space discretization has the form for $k = 0, 1, \ldots$:

Given the stress σ_h^k, the hardening parameter κ_h^k and the displacement u_h^k at t^k, compute their increments $\triangle\sigma_h^k$, $\triangle\kappa_h^k$, $\triangle u_h^k$:

$$\int_\Omega \Big\langle D\varepsilon(\triangle u_h^k) - a_h^k(\varepsilon(\triangle u_h^k)), \varepsilon(v_h)\Big\rangle \, dx = \triangle f_h^k(v_h) \quad \forall v_h \in V_h, \tag{1}$$

$$\begin{aligned} \triangle\sigma_h^k &= D\varepsilon(\triangle u_h^k) - a_h^k(\varepsilon(\triangle u_h^k)), \\ \triangle\kappa_h^k &= (2\mu\sqrt{3/2})^{-1}\|a_h^k(\varepsilon(\triangle u_h^k))\|. \end{aligned}$$

Put $\sigma_h^{k+1} = \sigma_h^k + \triangle\sigma_h^k$, $\kappa_h^{k+1} = \kappa_h^k + \triangle\kappa_h^k$, $u_h^{k+1} = u_h^k + \triangle u_h^k$.

Here, the matrix D denotes the Hook's matrix, μ, λ are Lamé coefficients, $\triangle f_h^k$ represents the load increment. The function a_h^k is given in the form

$$a_h^k(\varepsilon) := \frac{3\mu}{3\mu + H_m}\left(\|dev(\sigma)\| - \sqrt{\frac{2}{3}}(Y + H_m\kappa)\right)^+ \frac{dev(\sigma_h^k + D\varepsilon)}{\|dev(\sigma_h^k + D\varepsilon)\|}.$$

The function a_h^k is semismooth and has a potential, see [7]. Let us denote its generalized Jacobian and potential by $a_h^{o,k}$ and b_h^k, respectively.

Notice that the main problem in each time step is to solve the non-linear equation (1). If we represent a function $v_h \in V_h$ by the vector $\mathbf{v} \in \mathbb{R}^n$ and miss the index k then (1) can be rewritten as the system of non-linear equations

$$F(\triangle\mathbf{u}) = \triangle\mathbf{f},$$

where

$$\begin{aligned}\langle F(\mathbf{v}),\mathbf{w}\rangle &:= \int_\Omega \langle D\varepsilon(v_h)-a_h(\varepsilon(v_h)),\varepsilon(w_h)\rangle\,dx \quad \forall\mathbf{v},\mathbf{w}\in R^n,\\ \langle\triangle\mathbf{f},\mathbf{w}\rangle &:= \triangle f_h(w_h) \quad \forall\mathbf{w}\in R^n.\end{aligned}$$

We also introduce the following notation

$$\begin{aligned}\langle A(u)\mathbf{v},\mathbf{w}\rangle &:= \int_\Omega \langle D\varepsilon(v_h)-a^o(\varepsilon(u_h))\varepsilon(v_h),\varepsilon(w_h)\rangle\,dx \quad \forall\mathbf{u},\mathbf{v},\mathbf{w}\in R^n,\\ \langle A_e\mathbf{v},\mathbf{w}\rangle &:= \int_\Omega \langle D\varepsilon(v_h),\varepsilon(w_h)\rangle\,dx \quad \forall\mathbf{v},\mathbf{w}\in R^n,\\ J(\mathbf{v}) &:= \frac{1}{2}\|\mathbf{v}\|_E^2-\int_\Omega b(\varepsilon(v_h))dx-\langle\triangle\mathbf{f},\mathbf{v}\rangle \quad \forall\mathbf{v}\in R^n,\end{aligned}$$

where $\|.\|_E=\langle A_e.,.\rangle^{1/2}$. The properties of the function a_h ensure that the problem has a unique solution and can also be formulated as a minimization problem. Notice that $A(\mathbf{v})$ is a symmetric, positive definite matrix and

$$(1-\nu_0)\|\mathbf{w}\|_E^2\le\langle A(\mathbf{v})\mathbf{w},\mathbf{w}\rangle\le\|\mathbf{w}\|_E^2 \quad \forall\mathbf{v},\mathbf{w}\in\mathbb{R}^n,\ \ \nu_0=\frac{3\mu}{3\mu+H_m}. \tag{2}$$

Moreover,

$$\lim_{\mathbf{w}\to 0}\frac{\|F(\mathbf{v}+\mathbf{w})-F(\mathbf{v})-A(\mathbf{v}+\mathbf{w})\mathbf{w}\|_{-E}}{\|\mathbf{w}\|_E}=0 \quad \forall\mathbf{v},\mathbf{w}\in\mathbb{R}^n. \tag{3}$$

3 Semismooth Newton method and its modification

The iterates of the semismooth Newton method (SNM), see [3], have the form

$$\triangle\mathbf{u^{j+1}}=\triangle\mathbf{u^j}+\mathbf{s^j},\quad A(\triangle\mathbf{u^j})\mathbf{s^j}=\triangle\mathbf{f}-F(\triangle\mathbf{u^j}),\ \ j=0,1,\ldots$$

The above properties (2) and (3) ensures that SNM converges locally superlinearly. Notice that the superlinear convergence depends on the finite element discretization. The global convergence of SNM can be proved for sufficiently large H_m.

In contrast to SNM, the iterates of the semismooth Newton method with damping (SNMD) have the form

$$\triangle\mathbf{u^{j+1}}=\triangle\mathbf{u^j}+\alpha_j\mathbf{s^j},\quad A(\triangle\mathbf{u^j})\mathbf{s^j}=\triangle\mathbf{f}-F(\triangle\mathbf{u^j}),\ \ \alpha_j=arg\min_{\alpha\in(0,1]}J(\triangle\mathbf{u^j}+\alpha\mathbf{s^j}),\ \ j=0,1,\ldots$$

This method is also locally superlinearly convergent since $\alpha_j\to 1$. Moreover the method converges globally. The corresponding global convergence estimate is independent of the finite element discretization.

We choose $\triangle\mathbf{u^0}=A_e^{-1}\triangle\mathbf{f}$ as a suitable initial approximation of $\triangle\mathbf{u}$. We can also consider the influences of the inexact inner solvers for computing $\mathbf{s^j}$ and α_j. In the below example, the vectors $\mathbf{s^j}$ are founded by a direct solver and the coefficients α_j by the regula-falsi method with respect to the stopping criterion

$$-\delta(\|\tilde{\mathbf{s}}^{\mathbf{j}}\|_E)\le\frac{\langle\nabla J(\mathbf{u^j}+\tilde\alpha_j\tilde{\mathbf{s}}^{\mathbf{j}}),\tilde{\mathbf{s}}^{\mathbf{j}}\rangle}{(1-\nu_0)\tilde\alpha_j\|\tilde{\mathbf{s}}^{\mathbf{j}}\|_E^2}\le\frac{1}{2}q,\quad q<1, \tag{4}$$

where $\delta(\|\mathbf{s^j}\|_E) = \|\mathbf{s^j}\|_E/(\|\mathbf{u^j}+\mathbf{s^j}\|_E+\|\mathbf{u^j}\|_E)$. The first inequality in (4) ensures that the damping is not too strong and the second one ensures the global convergence of the method. Notice that the accuracy of the stopping criterion also depends on the constant $1-\nu_0$. If the constant is small, i.e. if we tend to perfect plasticity, then more exact computing of α_j is required. The SNM and SNMD algorithm are stopped if $\delta(\|\mathbf{s^j}\|_E) < \epsilon$.

4 Numerical example in 2D

We consider a plain strain problem with a thin plate which is represented by the domain Ω, see Figure 1. Homogeneous Dirichlet boundary conditions in the normal direction are prescribed on two sides of Ω. The surface load $g(t) = 450\sin(2\pi t)$, $t \in [0, 1/4]$, is applied to the upper side of Ω. The material parameters are set to $E = 206900$, $\nu = 0.29$, $Y = 450$, $H_m = 1, 100, 10000$ and the time interval is divided into 50 equidistant steps. We consider three different meshes with 2028, 7600 and 29400 elements. The worsest of which is depicted in Figure 1. The tolerance is $\epsilon = 10^{-10}$.

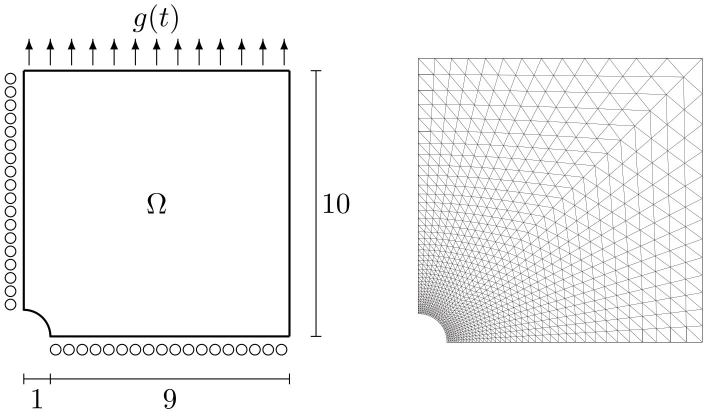


Figure 1: Geometry of the example (left), and the worsest applied mesh (right).

The calculation was performed using a MATLAB 7.0 code, see [2]. Both methods are convergent for this example with respect to the chosen initial iteration. The numbers of SNM and SNMD iterations are practically the same and the superlinear convergence slightly depends on mesh as we can see in Table 1 for SNMD, chosen time steps, three different meshes and $H_m = 100$.

NoTS		25			37			46	
NoFE	2028	7600	29400	2028	7600	29400	2028	7600	29400
j									
0	1.0e-01	1.1e-01	1.1e-01	3.7e-01	4.1e-01	4.3e-01	7.5e-01	7.9e-01	8.0e-01
1	1.6e-02	2.7e-02	2.9e-02	2.7e-02	2.9e-02	4.0e-02	3.1e-02	3.2e-02	3.3e-02
2	1.8e-03	4.3e-04	5.9e-04	1.5e-03	2.1e-03	4.7e-03	1.0e-04	4.4e-03	2.5e-03
3	4.0e-06	3.2e-07	4.0e-05	2.7e-07	4.0e-04	8.7e-04	7.0e-10	3.2e-04	5.2e-04
4	3.2e-11	7.3e-13	8.4e-09	2.0e-14	4.8e-08	1.4e-04	3.8e-15	1.1e-08	2.5e-04
5			3.2e-15		5.7e-15	4.7e-05		4.0e-15	1.5e-07
6						7.0e-09			4.2e-14
7						6.1e-15			

Table 1: Convergence of MSNM (the values of $\delta(\|\mathbf{s^j}\|_E)$) at chosen time steps (NoTS) for 3 meshes characterized by different numbers of elements (NoFE).

For the finest mesh, we test the numbers of regula-falsi iterations in dependence on H_m. We need maximally 0, 1, 2 regula-falsi iteration for $H_m = 10000, 100, 1$, respectively. Thus the additional computing of α_j is not too costly in this example and SNMD gives the same results as SNM for $H_m = 10000$ since $\alpha_j = 1$.

5 Conclusion

The semismooth Newton methods with and without damping have been used to solve the elasto-plastic problem. The main advantage of SNMD is a global convergence which does not hold for SNM in general. On the other hand, computing of the damping coefficients could not be very costly. The proposed stopping criterion (4) ensures all the theoretical convergence results of SNMD and yields good numerical results in combination with the regula-falsi method.

SNMD can also be used for some semicoercive problems where the first inequality in (2) holds only for $\nu_0 = 1$. It can happened for example for perfect plasticity or some simplified elasto-damage-plastic models, see [4]. SNMD has also been used to solving a semicoercive problem with a beam on a non-linear subsoil, see [6]. In such cases, we must replace $1 - \nu_0$ in the stopping criterion (4) by a suitable tolerance parameter.

Acknowledgement: This work has been supported by the Grant Agency of the Czech Republic, contract No. 105/09/1830.

References

[1] R. Blaheta: *Numerical methods in elasto-plasticity.* Documenta Geonica 1998, PERES Publishers, Prague, 1999.

[2] P. Byczanski, S. Sysala: *Modified semismooth Newton method: Numerical example.* In: Proceedings of seminar "SiMoNA 2009", Liberec, 24–30, 2009.

[3] X. Chen, Z. Nashed, L. Qi: *Smoothing methods and semismooth methods for nondifferentiable operator equations.* SIAM J. Numer. Anal. 38, 1200–1216, 2000.

[4] E.A.S. Neto, D. Perić, D.R.J. Owen: *Computational methods for plasticity: Theory and applications.* Wiley, 2008.

[5] L. Qi, J. Sun: *A nonsmooth version of Newton's method.* Mathematical Programming 58, 353–367, 1993.

[6] S. Sysala: *Numerical modelling of semi-coercive beam problem with unilateral elastic subsoil of Winkler's type.* Appl. Math., 55, 2010, in press.

[7] S. Sysala: *Application of the modified semismooth Newton method to some elasto-plastic problems.* Mathematics and Computer in Simulation, Modelling 2009, Submitted.

Study of using corners for BDDC in 3D

M. Čertíková, P. Burda, J. Novotný, J. Šístek

Department of Mathematics, Faculty of Mechanical Engineering and
Department of Mathematics, Faculty of Civil Engineering
Czech Technical University in Prague
Institute of Mathematics and Institute of Thermomechanics
Academy of Sciences of the Czech Republic, Prague

1 Introduction

Numerical solution of linear problems arising from isotropic elasticity discretized by finite elements is important in many areas of engineering. The matrix of such systems is typically large, sparse, and ill-conditioned. For large problems, iterative methods such as the preconditioned conjugate gradients (PCG) are usually less expensive than direct solvers in terms of memory and computational time. However, their convergence rate deteriorates with growing condition number of the solved linear system and good preconditioning becomes essential. The need of efficient preconditioners tailored to the solved problem that can be implemented in parallel gave rise to the field of domain decomposition methods [1].

The Balancing Domain Decomposition by Constraints (BDDC) method [2, 3] is an iterative substructuring primal domain decomposition method. The BDDC method is closesly related to the earlier FETI-DP method. It has been recently proved by Mandel, Dohrmann, and Tezaur [3], that the two methods are spectrally equivalent, which allows for application of numerical results computed for one method to the other.

In both methods, a fundamental role is played by a coarse space defined by a choice of constraints on continuity. Optimal choice of these constraints has strong influence on convergence of the method. However, this choice in practice is not a satisfactorily solved problem yet. In this paper, we study the influence of adding more corners (coarse node constraints), on time effectivity of the computation. Our tests include both scholastic and industrial 3D linear elasticity problems.

2 The BDDC method

After discretization by the finite element method (FEM), the linear system $Ku = f$ is to be solved for a vector u of unknown values of displacements at nodes of a given domain.

The domain is split into nonoverlapping subdomains with the *interface* formed by unknowns common to at least two subdomains. Then the problem is reduced to the *Schur complement* problem with respect to the interface and this reduced problem is solved by PCG method. The BDDC method is used as a preconditioner, that splits the computation of the preconditioned residual needed in every iteration of PCG to solution of independent *subdomain problems* and the global *coarse problem.* The preconditioned residual is obtained as a combination of their solutions (for details see [1, 2, 3]).

The coarse problem is solved on the *coarse space*, which consists of functions that are continuous across the interface at selected degrees of freedom only. Functions from the coarse space are fully determined by their values at these *coarse degrees of freedom* and by the requirement of having minimal energy elsewhere.

Choice of the coarse degrees of freedom has great impact on the performance of the preconditioner. The simplest choice of coarse degrees of freedom is a function value at a selected node on the interface. Such node is then called *corner*. It was shown that while for 2D elasticity problems the BDDC (or FETI-DP) preconditioner is scalable for coarse space defined by corners only, in 3D elasticity problems more general coarse degrees of freedom, such as (weighted) average values over edges and faces, need to be used in order to achieve the scalability, see e.g. Toselli and Widlund [1]. In what follows we deal with 3D problems only.

3 The implementation

In every PCG iteration, the following types of problems are to be solved:

- On every subdomain, a local problem with zero Dirichlet boundary condition on the interface.
- On every subdomain, a local problem with zero Dirichlet boundary condition on the coarse degrees of freedom on interface and zero Neumann boundary condition on the rest of the interface.
- Global coarse problem for coarse degrees of freedom only.

Only values of the boundary conditions, not their type, change from iteration to iteration, so all the factorizations can be prepared in advance. During the PCG cycle only back-substitutions are performed.

We implemented BDDC on top of common components of existing finite element codes – the frontal solver and the element stiffness matrix generation. The coarse problem is solved using standard FEM approach with subdomains playing the role of elements. Thus the coarse matrix in not assembled as a whole but stored distributed among processors as local coarse matrices. Lagrange multipliers are used for implementation of the coarse averages. Detailed description of the implementation can be found in [5], in less detail it is described also in [4].

4 Numerical results

Presented calculations were performed on 12-36 processors of SGI Altix 4700 computer of Supercomputing Centre of Czech Technical University in Prague. The METIS graph partitioner is used as an automatic tool for other than rectangular decompositions.

First we investigated a typical test problem for 3D elasticity: a cuboid with fixed base loaded by pressure of a weight put on the upper face. The geometry is discretized using 24 000 quadratic elements, which leads to 311 943 unknowns. We tested two typical decompositions of the domain: 36 cuboid subdomains obtained by plane sections or 36 subdomains obtained by the graph tool, reffered to as Case A or Case C, respectively. Then we tested three different industrial problems of different sizes. The first one is a problem of elasticity analysis of a turbine nozzle, through which the steam enters the turbine blades. The geometry is discretized using 2 696 quadratic elements, which leads to 40 254 unknowns. The second one is a problem of elasticity analysis of a hip joint replacement which is loaded by pressure from body weight. This mesh consists of 33 186 quadratic elements resulting in 544 734 unknowns. Both meshes were divided into 36 subdomains by the graph tool. The third problem is a problem of stress analysis of a mine reel loaded by its own weight and the weight of of the steel rope. The mesh consisting of

problem	subdomains	vertices	edges	faces	interf. nodes	all nodes
Case A	36	12	52	75	17 303	103 981
Case C	36	82	181	144	20 321	103 981
Turbine nozzle	36	6	60	101	2 714	13 418
Hip replacement	36	1	19	78	9 222	181 578
Mine reel	1 024	2 451	1 209	4 164	117 113	579 737

Table 1: Decomposition characteristics of the tested problems.

140 816 quadratic elements and 1 739 211 unknowns was divided into 1 024 subdomains by the graph tool. Decomposition characteristics of the problems are summarized in Table 1.

We experimented with adding corners to an initial set of corners selected by the algorithm published in [6], with or without using also averages over faces and edges.

When only corners were used for the coarse problem, adding more corners to the initial set did not improved efficiency for small problem of the turbine nozzle, but proved to be beneficial for medium-size problems. Figure 1 left shows results for the turbine nozzle, right for the hip joint replacement problem (very similar results as for hip replacement were obtained also for both cuboid problems). There are three time series depicted on every graph: the overall computational time, the time consumed by factorization of all subproblems, and the time consumed by PCG iterations. For the large problem of mine reel, corners only were not sufficient for convergence.

When also averages were used, adding more corners to the initial set did not improved efficiency for small and medium problems, but proved to be beneficial for the large problem of mine reel. Figure 1 left shows results for the cuboid problem A (very similar results were obtained also for the cuboid problem C), center the hip joint replacement problem, right the mine reel problem.

5 Conclusion

We investigated time efficiency for different choices of the coarse space in the BDDC method. Our tests indicate that the approach of adding more corners to the "minimal“ set of corners can be beneficial for medium problems (either scholar or industrial problems), when the coarse space is defined by corners only. If also averages on edges and faces are used as coarse degrees of freedom, the improvement is small or none. For small industrial problem of the turbine nozzle we found no benefit in adding more corners. On the contrary for the large problem of the mine reel, adding more corners proved to be beneficial for convergence, when averages on edges and faces are used as coarse degrees of freedom (without averages convergence was not achieved). However, for really large problems it seems that some more sofisticated techniques for costruction of the coarse space should be used.

Acknowledgement: We are grateful to Jan Leština (Vamet Ltd.) for providing the problems of turbine nozzle and mine reel for testing. We would also like to thank to Jan Mandel and Bedřich Sousedík for fruitful discussions about BDDC method. This research has been supported by the Czech Science Foundation under grant GA CR 106/08/0403, by the Grant Agency of the Academy of Sciences of the CR under grant IAA200600801, and by projects MSM 6840770001 and MSM 6840770010. It has also been supported by Institutional Research Plans AV0Z 10190503 and AV0Z 20760514.

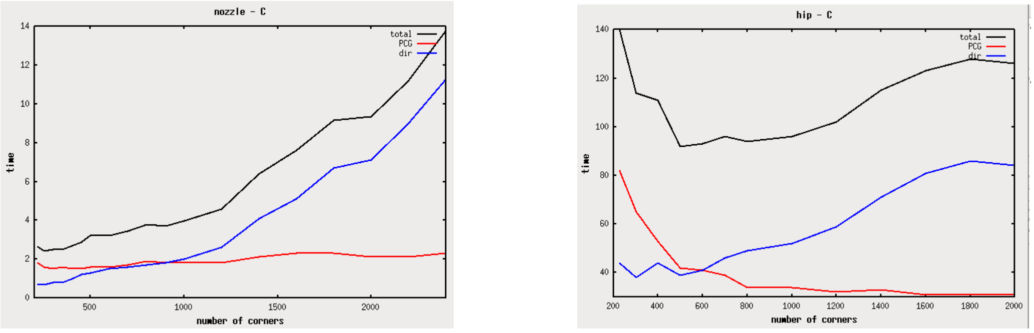


Figure 1: Computational time when using corners only: turbine nozzle problem (left) and hip joint replacement problem (right).

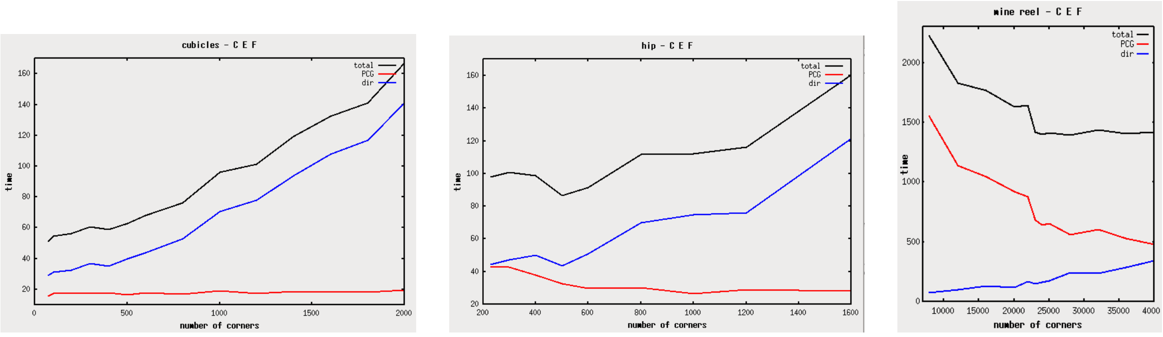


Figure 2: Computational time for coarse space with averages: cuboid problem A (left), hip joint replacement problem (center) and mine reel problem (right).

References

[1] A. Toselli, O. Widlund, O.: *Domain decomposition methods—algorithms and theory*. vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005.

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

[3] J. Mandel, C.R. Dohrmann, R. Tezaur: *An algebraic theory for primal and dual substructuring methods by constraints*. Appl. Numer. Math. 54(2), 167–193, 2005.

[4] J. Šístek, J. Novotný, P. Burda, M. Čertíková: *Implementation of the BDDC method based on the frontal and multifrontal algorithm*. In: Blaheta, R. and Starý, J. (ed.), Proceedings of Seminar on Numerical Analysis, SNA'09, Ostrava, Czech Republic, February 2–6, Institute of Geonics AS CR, Ostrava, 105–108, 2009.

[5] J. Šístek, J. Novotný, J. Mandel, M. Čertíková, P. Burda: , P.: *BDDC by a frontal solver and stress computation in a hip joint replacement*. Math. Comp. Simul. Available online at `http://dx.doi.org/10.1016/j.matcom.2009.01.002`, 2009.

[6] P. Burda, M. Čertíková, J. Novotný, J. Šístek: *On Coarse Space for the BDDC Method* In: Blaheta, R. and Starý, J. (ed.), Modelling 2009, Book of Abstracts, Rožnov pod Radhoštěm, Czech Republic, June 22–26, Institute of Geonics AS CR, Ostrava, 78, 2009.

An algorithm for 3D contact problems with orthotropic friction

Z. Dostál [1], *J. Haslinger* [2], *T. Kozubek* [1], *R. Kučera* [3]

[1]Department of Applied Mathematics, VŠB-TU, Ostrava
[2]Department of Numerical Mathematics, Charles University, Prague
[3]Department of Mathematics and Descriptive Geometry, VŠB-TU, Ostrava

1 Introduction

Contact problems represent a special branch of mechanics of solids whose goal is to find an equilibrium state of deformable bodies being in a mutual contact. Due to non-penetration and friction conditions, problems we have to solve are highly non-linear. For linearly elastic materials obeying a Hook law for small deformations, a linearization of the non-penetration conditions leads to a convex set of kinematically admissible displacements (geometrical nonlinearity). Another non-linearity originates from the presence of friction. In the simplest case with an à-priori given slip bound (Tresca model), the mathematical model is represented by a variational inequality of the second kind. This model is however too simple since the non-penetration and friction phenomena are decoupled. For this reason more realistic models of friction have to be used and the Coulomb friction law is the classical one. The slip bound prescribed in Tresca model is now replaced by the product of a coefficient of friction $\mathcal{F}$ and the norm of the normal contact force. The coupling of unilateral and friction conditions leads to the so-called implicit variational inequality (in terms of displacements) or to a quasivariational inequality (in terms of contact stresses). Due to material or contact surface properties it may happen that the effect of friction is directionally dependent. A discretization of 3D contact problems with orthotropic Coulomb friction characterized by two coefficients of friction $\mathcal{F}_1$ and $\mathcal{F}_2$ in two mutually orthogonal directions was presented in [4]. The scalable algorithm for this problem was developed in [3] while the main ideas may be found in [1].

2 Formulation and algorithm

Let us consider two elastic bodies represented by two non-overlapping domains $\Omega^k \subset \mathbb{R}^3$ with the boundaries $\partial\Omega^k$, $k = 1, 2$. Each boundary consists of three non-empty disjoint parts Γ_u^k, Γ_p^k, and Γ_c^k open in $\partial\Omega^k$, so that $\partial\Omega^k = \overline{\Gamma}_u^k \cup \overline{\Gamma}_p^k \cup \overline{\Gamma}_c^k$. The zero displacements are prescribed on Γ_u^k while surface tractions act on Γ_p^k. On the *contact interface* given by Γ_c^1 and Γ_c^2 we consider contact conditions: the non-penetration of the bodies, the transmission of the contact stresses, and the effect of orthotropic Coulomb friction. Finally we suppose that each body Ω^k is subject to volume forces.

Our algorithm is based on the fixed-point approach in which the solution to the original problem is defined as a fixed-point of an auxiliary mapping acting on the contact interface. To find fixed-point we use the method of successive approximations whose individual iterative steps are given by contact problems with orthotropic Tresca model of friction.

The finite element approximation of the auxiliary problems combined with the TFETI domain decomposition method [2] leads to the following algebraic minimization problem:

$$\begin{aligned} &\text{minimize} \quad && \frac{1}{2}u^\top K u - u^\top f + \sum_{i=1}^{m_c} g_i \|\mathcal{F}_i (T_{1,i}u, T_{2,i}u)^\top\|_2, && (1) \\ &\text{subject to} \quad && B_E u = 0,\ Nu \leq d, && (2) \end{aligned}$$

where $K = \operatorname{diag}(K_1, \ldots, K_s)$ is a symmetric positive semidefinite block-diagonal stiffness matrix of order n, $f \in \mathbb{R}^n$ is the load vector, B_E is an $m \times n$ full rank "gluing"matrix, N denotes an $m_c \times n$ full rank matrix describing together with $d \in \mathbb{R}^{m_c}$ the non-penetration condition, $T_{1,i}, T_{2,i}$ are rows of $m_c \times n$ full rank matrices T_1, T_2, respectively, $\mathcal{F}_i \in \mathbb{R}^{2\times 2}$ are the value of the coefficient of friction, and g_i denote discrete slip bound values at contact nodes.

Even though (1)-(2) is the minimization problem with the unique solution, it is not suitable for direct numerical solution. The reasons are that K is typically singular, the summation term in (1) is non-differentiable, and the feasible set in (2) is in general so complex that the projection into it can hardly be effectively computed. In order to overcome these difficulties, one can apply the duality theory of convex programming [1].

To regularize the non-differentiability we use the following idea based on the Cauchy-Schwarz inequality in $\mathbb{R}^2$:

$$\max_{\|\mathcal{F}_i^{-1}\lambda_{T,i}\|_2 \leq g_i} (T_{1,i}u, T_{2,i}u)\lambda_{T,i} = g_i \|\mathcal{F}_i (T_{1,i}u, T_{2,i}u)^\top\|_2, \tag{3}$$

where $\lambda_{T,i} \in \mathbb{R}^2$ plays the role of Lagrange multipliers. We will denote $\lambda_{T,i} = (\lambda_{T_1,i}, \lambda_{T_2,i})^\top$. It is easily seen that the constraints on $\lambda_{T,i}$ in (3) are the ellipsoidal inequalities.

In the dual formulation of (1)-(2) we use three types of Lagrange multipliers: $\lambda_E \in \mathbb{R}^m$ and $\lambda_N \in \mathbb{R}^{m_c}$ are associated with the equality and the inequality constraints in (2), while $\lambda_{T_1}, \lambda_{T_2} \in \mathbb{R}^{m_c}$ regularize the non-differentiability via (3). To simplify the notation we denote

$$\lambda = \begin{pmatrix} \lambda_E \\ \lambda_N \\ \lambda_{T_1} \\ \lambda_{T_2} \end{pmatrix}, \quad B = \begin{pmatrix} B_E \\ N \\ T_1 \\ T_2 \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ d \\ 0 \\ 0 \end{pmatrix}.$$

The Lagrangian associated with the problem (1)-(2) reads as

$$L(u, \lambda) = \frac{1}{2}u^\top K u - u^\top f + \lambda^\top (Bu - c), \quad (u, \lambda) \in \mathbb{R}^n \times \Lambda(g),$$

and the set of the Lagrange multipliers is given by

$$\Lambda(g) = \{\lambda \in \mathbb{R}^{m+3m_c} : \lambda_{N,i} \geq 0, \|\mathcal{F}_i^{-1}\lambda_{T,i}\|_2^2 \leq g_i^2, i = 1, \ldots, m_c\}.$$

It is well known [1] that (1)-(2) is equivalent to the *saddle-point problem* that is the problem of finding $(\bar{u}, \bar{\lambda}) \in \mathbb{R}^n \times \Lambda(g)$ such that

$$L(\bar{u}, \bar{\lambda}) = \min_{u \in \mathbb{R}^n} \max_{\lambda \in \Lambda(g)} L(u, \lambda).$$

As L is convex in the first variable, $\bar{u}$ can be eliminated by

$$\bar{u} = K^\dagger (f - B\bar{\lambda}) + R\bar{\alpha},$$

where $K^\dagger \in \mathbb{R}^{n\times n}$ is a generalized inverse to K, $R \in \mathbb{R}^{n\times l}$ is a matrix whose columns span the null-space $Ker K$, l denotes the defect of K, and $\bar{\alpha} \in \mathbb{R}^l$ is an appropriate vector. In advance,

$$f - B^\top \bar{\lambda} \in Im K.$$

Therefore, (1)-(2) leads to the dual problem:

$$\text{minimize } \frac{1}{2}\lambda^\top F\lambda - \lambda^\top \widetilde{h}, \quad \text{subject to } \lambda \in \Lambda(g),\ G\lambda = e,$$

where

$$F = BK^\dagger B^\top, \quad \widetilde{h} = BK^\dagger f - c, \quad G = R^\top B, \quad e = R^\top f.$$

After homogenization, using orthogonal projectors, and penalization, we arrive at the following problem:

$$\text{minimize } \frac{1}{2}\lambda^\top (PFP + \rho Q)\lambda - \lambda^\top Ph, \quad \text{subject to } \lambda \in \Lambda(g),\ G\lambda = 0, \tag{4}$$

where $\rho > 0$ is arbitrary and $Q = G^\top(GG^\top)^{-1}G$, $P = I - Q$ denote the orthogonal projectors on $Im G^\top$ and $Ker G$, respectively.

As (4) consists of the minimization of the quadratic objective function subject to separable convex inequalities and linear equality constraints, we use the recently proposed optimization algorithm based on the augmented Lagrangian method [3]. The important property of this algorithm is that the number of iterations needed to get a solution with a given accuracy is uniformly bounded (with respect to the scale of the problem) provided that the spectrum of the Hessian is confined in a given interval. The assumption on the spectrum is satisfied due to TFETI domain decomposition method.

3 Numerical experiments

We use the algorithm for solving contact problem with Coulomb friction with the geometry as in Figure 1. The upper body is made of steel while the lower one is made of aluminium. The applied surface tractions are seen in the figure, the volume forces are neglected. The coefficient of friction is given by $\mathcal{F}_i = \text{diag}(0.3, 0.3)$ (isotropic case).

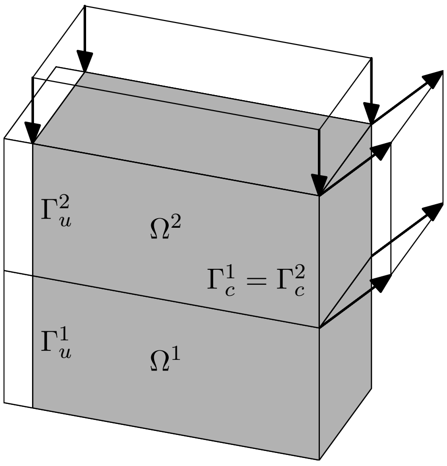


Figure 1: Geometry of the model problem

Each body Ω^k, $k = 1, 2$, is divided into the same number of sub-domains represented by bricks of the same size that are decomposed then into cubes (trilinear finite elements). By H and h we denote the decomposition parameter (diameter of bricks) and the discretization parameter (diameter of cubes), respectively. We apply the inexact implementation of the algorithm so that *iter* are connected iterates of the augmented Lagrangian algorithm and the method of successive approximations. By n_{PFP} we denote the number of matrix vector multiplications by the Hessian matrix. Finally, n, $n_d = m + 3m_c$, and l is the number of primal unknowns, dual unknowns, and rigid body modes. The results of our experiments are summarized in Table 1, where $rel_{\mathit{eff}} := n_{PFP}/n$ is the relative efficiency of the solver.

s	$H/h = 2$	$H/h = 3$	$H/h = 4$	$H/h = 5$
4	(324/153/24) **10/180** *0.5556*	(768/276/24) **10/269** *0.3503*	(1500/435/24) **11/356** *0.2373*	(2592/630/24) **11/470** *0.1813*
32	(2592/1527/192) **11/483** *0.1863*	(6144/2889/192) **11/657** *0.1069*	(12000/4683/192) **11/665** *0.0554*	(20736/6909/192) **12/847** *0.0408*
108	(8748/5493/648) **11/636** *0.0727*	(20736/10506/648) **11/878** *0.0423*	(40500/17139/648) **13/906** *0.0224*	(69984/25392/648) **14/1071** *0.0153*
256	(20736/13419/1536) **12/737** *0.0355*	(49152/25791/1536) **14/939** *0.01910*	(96000/42195/1536) **15/1173** *0.0122*	(165888/62631/1536) **16/1400** *0.0084*
500	(40500/26673/3000) **14/812** *0.0200*	(96000/51408/3000) **15/1039** *0.0108*	(187500/84243/3000) **17/1533** *0.0081*	(324000/125047/3000) **18/1776** *0.0054*

[a] At each position $(n/n_d/l)$, $\boldsymbol{iter/n_{PFP}}$, and rel_{eff} are displayed

Table 1: Scalability and relative efficiency.

Acknowledgement: This work has been supported by the grant 101/08/0574 of the Grant Agency of the Czech Republic and by the Research Project MSM6198910027 of the Czech Ministry of Education.

References

[1] Z. Dostál: *Optimal quadratic programming algorithms: with applications to variational inequalities.* Springer, New York, 2009.

[2] Z. Dostál, D. Horák, R. Kučera: *Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE.* Comm. Num. Meth. Engrg, 22, 1155–1162, 2006.

[3] Z. Dostál, R. Kučera: *An optimal algorithm for minimization of quadratic functions with bounded spectrum subject to separable convex inequality and linear equality constraints.* Submitted to SIAM J. Optimization, (2009).

[4] J. Haslinger, R. Kučera, T. Kozubek: *Discretization and numerical realization of contact problems with orthotropic Coulomb friction.* Submitted to SIAM J. Scientific Computing, 2009.

Preconditioning of FETI-DP using corners on contact interface

Z. Dostál, D. Horák

Department of applied mathematics, FEECS VŠB-Technical University Ostrava, Czech Republic

Our research concerns the preconditioning of FETI-based methods for contact problems. The standard FETI-DP is based on the decomposition into non-overlapping subdomains, where the continuity of the primal solution at crosspoints is implemented directly into the formulation of the primal problem so that one degree of freedom is considered at each crosspoint and the continuity of the solution on auxiliary interfaces is enforced by Lagrange multipliers. The duality transforms the general inequality into the nonnegativity constraints. After eliminating the corners, the problem reduces to a small, relatively well conditioned strictly convex QP (Quadratic Programming) problem with simple bound for Lagrange multipliers that is solved iteratively by efficient algorithms that exploit cheap projections and other tools. For semi-coercive problems the efficiency of the FETI-DP can be further improved by introducing special projectors onto an auxiliary space related to rigid body modes of floating bodies and preconditioners - lumped and Dirichlet's. Let us mention that the preconditioners can be applied only to the linear part and their efficiency is very small. Once the Lagrange multipliers are known, we solve linear problem to find solution for corners. In both phases we need to build the matrix defining so called coarse problem and to factorize it. The scalability of FETI-DPC based on active set strategies with additional planning steps was established by Farhat et al. [6] only experimentally. Dostál et al. proved this scalability theoretically. Numerical scalability for FETI–DP algorithm for coercive problems was proven theoretically and experimentally in [1]. Later, the result was extended to include mortar disctretization [4] and for semicoercive problems [3].

Farhat et al. observed, [5], that the corner nodes on contact interface cause difficulties and recommended to avoid them. These difficulties can be overcome through the additional condition that preserves the nonpenetration in Lagrange multipliers, and moreover in this way it is possible to improve rate of convergence. This richer corner mesh results in better convergence of the method because of better error propagation across the nonlinear interface and in better preconditioning of nonlinear steps using standard FETI-DP preconditioners. We showed experimentally that for unpreconditioned and preconditioned FETI–DP using no corners on contact zone the numbers of CG iterations increase much more rapidly with increasing number of subdomains along the contact interface in comparison to the case we use corners on the contact zone, when the numbers of CG iterations vary very moderately. The results demonstrate that for a given decomposition the use of corners always significantly reduces number of CG iterations for both unpreconditioned and preconditioned systems and this effect is magnified with increasing number of subdomains along the contact interface, i.e. with increasing number of corners on the the contact zone.

Significant modification making from FETI-DP the method of new type is based on definition of all nodes on the contact zone as the corners, i.e. constraint matrix with inequality conditions considers only corner nodes. This approach enable us the splitting of the problem into a very small nonlinear one with corners as unknowns and a linear one with the Lagrange multipliers for equalities. We eliminate the Lagrange multipliers first and the problem reduces to very small, well conditioned strictly convex QP problem with bound for corners that is solved iteratively by efficient algorithms or by application of duality resulting in dual problem with Lagrange multipliers for inequalities. Once the corners are known we solve linear problem to find the solution for Lagrange multipliers for equalities. In both phases we do not need to build any matrix defining the coarse problem and to factorize it. Moreover we can significantly reduce the number of CG interations by applying the standard FETI-DP preconditioners in both cases, i.e., by solving the nonlinear problem with corners and the linear one with Lagrange multipliers as unknowns.

We shall give the results of numerical experiments with parallel implementation using Matlab and PETSc that confirm scalability of the algorithm for contact problems.

Acknowledgement: This research has been supported by the grants: GA CR 101/09/P601.

References

[1] Z. Dostál, D. Horák, D. Stefanica: *A scalable FETI-DP algorithm for a coercive variational inequalities.* IMACS J. Appl. Numer. Math., Vol. 54, 378–390, 2005.

[2] Z. Dostál, D. Horák, D. Stefanica: *An Overview of Scalable FETI-DP Algorithms for Variational Inequalities.* Lecture Notes in Comput. Science and Engineering 55, Proceedings from the 16th Conference on DDM, New York, Springer, 223–230, 2006.

[3] Z. Dostál, D. Horák, D. Stefanica: *A Scalable FETI–DP Algorithm for Semi-coercive Variational Inequality.* Computer Methods in Applied Mechanics and Engineering, Vol. 196(8), ISSN 0045-7825, 1369–1379, 2007.

[4] Z. Dostál, D. Horák, D. Stefanica: *A scalable FETI-DP algorithm with non-penetration mortar conditions on contact interface.* Journal of Comp. and Appl. Mathematics, Vol. 231(2), 577–591, ISSN:0377-0427, 2009.

[5] P. Avery, G. Rebel, M. Lesoinne, C. Farhat: *A numerically scalable dual-primal substructuring method for the solution of contact problems - part I: the frictionless case.* Comput. Methods Appl. Mech. Engrg., 193, 2403–2426, 2004.

[6] G. Rebel, C. Farhat, M. Lesoinne, P. Avery: *A scalable Dual-Primal domain decomposition method for the solution of contact problems with friction.* 7th U.S. National Congress on Computational Mechanics, 2003.

Adaptive *hp*-FEM on dynamical meshes with application to a flame propagation problem

L. Dubcová [1,2], *P. Šolín* [1,3]

[1] Institute of Thermomechanics, Academy of Sciences of the Czech Republic
[2] Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic
[3] Department of Mathematics and Statistics, University of Nevada, Reno

1 Introduction

Flame propagation problems are highly nontrivial from both the modeling and computational points of view. On the computational side, a major difficulty are sharp moving fronts that need to be resolved accurately at all times in order to capture the process dynamics correctly. Nowadays, most practitioners are still tackling these problems with non-adaptive methods such as finite differences (FDM) or non-adaptive low-order finite elements (FEM) [1]. However, extremely fine uniform meshes are usually needed to attain results with sufficient accuracy, which leads to excessive computing times and memory requirements. One of the first space-time adaptive FEM algorithms for flame propagation problems, based on first-order FEM appeared recently [3].

In this paper we present the first algorithm that makes it possible to use adaptive *hp*-FEM [4, 7] for evolutionary PDEs. In Section 2 we presents a low Mach number laminar flame propagation model consisting of two coupled nonlinear parabolic differential equations. Section 3 introduces a novel space-time adaptive algorithm based on a combination of the classical Rothe's method and the novel multi-mesh *hp*-FEM [5, 6]. Numerical results are presented in Section 4.

2 Model problem

We consider a freely propagating laminar flame and its response to a heat-absorbing obstacle represented by a set of cooled parallel rods with a rectangular cross-section. The mathematical model is based on the assumption that the motion of the fluid is independent from the temperature θ and species concentration Y. Fluid motion in the burner is neglected. The model consists of a system of two coupled nonlinear parabolic equations for θ and Y,

$$\begin{aligned} \frac{\partial \theta}{\partial t} - \Delta\theta &= \omega(\theta, Y) \quad \text{in } \Omega \times (0, T_0), & (1)\\ \frac{\partial Y}{\partial t} - \frac{1}{\text{Le}}\,\Delta Y &= -\omega(\theta, Y) \quad \text{in } \Omega \times (0, T_0). & (2)\end{aligned}$$

Here, the reaction rate $\omega(\theta, Y)$ is defined by the *Arrhenius law*

$$\omega(\theta, Y) = \frac{\beta^2}{2\text{Le}} Y e^{\frac{\beta(\theta-1)}{1+\alpha(\theta-1)}}, \qquad (3)$$

where α is the gas expansion coefficient in a flow with nonconstant density, β the non-dimensional activation energy, and Le the Lewis number (ratio of diffusivity of heat and diffusivity of mass). Both θ, $0 \le \theta \le 1$ and Y, $0 \le Y \le 1$ are dimensionless and so is the time t.

3 Adaptive *hp*-FEM on dynamical meshes

The adaptive *hp*-FEM algorithm for time-dependent problems we use is obtained by combining the classical Rothe's method with the novel multimesh *hp*-FEM [5, 6].

The Rothe's method provides a better setting for the application of spatially adaptive algorithms compared to the method of lines (MOL). In every time step, an evolutionary PDE is approximated by means of one or more time-independent ones. The spatial discretization error can be controlled by solving the time-independent equations adaptively, and the size of the time step can be adjusted using standard ODE techniques. In our computations adaptive time integration is carried out using a pair of first-order backward-difference formulas, whose combination yields a second-order scheme [2]. In every time step, the difference between the pair of results provides an estimate of the local error that is used to adapt the time step.

In the $(n+1)$st time step, the approximations $\theta^n(x)$, $Y^n(x)$, that have been obtained in the previous time step, are used as data. Note, however, that they are defined on a locally refined mesh that was created automatically during the nth time step, while the unknowns $\theta^{n+1}(x), Y^{n+1}(x)$ are solved adaptively starting from a coarser mesh. As a result, the meshes obtained on each time level are different, i.e., the mesh changes dynamically in time. In order to keep the algorithms on a reasonable level of complexity, the meshes are not completely unrelated. Each of the meshes $\mathcal{T}_n$ is obtained from a very coarse *master mesh* $\mathcal{T}_m$ using sequence of local mutually independent refinements. In order to evaluate exactly the weak formulation of the coupled problem (1), (2) when the solution pairs $\theta^n(x)$, $Y^n(x)$ and $\theta^{n+1}(x), Y^{n+1}(x)$ are defined on different meshes, we use the *multi-mesh hp*-FEM [5, 6]. In this technology the stiffness matrix is assembled on a *virtual union mesh* $\mathcal{T}_u$ which is the geometrical union of the meshes $\mathcal{T}_n$ and $\mathcal{T}_{n+1}$, as illustrated in Fig. 1. In this way, no additional error arises since no transfer of information between the meshes takes place.

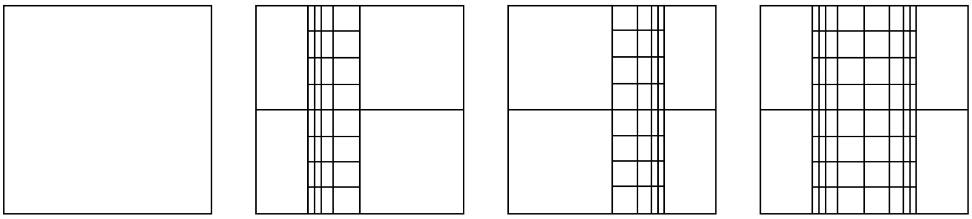


Figure 1: Example of a master mesh $\mathcal{T}_m$, meshes $\mathcal{T}_n, \mathcal{T}_{n+1}$ and the union mesh $\mathcal{T}_u$.

The final space-time adaptive algorithm can be summarized as follows:

1. calculate a temporal error estimate $e_k = ||\omega(\theta_1, Y_1) - \omega(\theta_2, Y_2)||_{L^2}$
2. calculate a spatial error estimate $e_h = ||\omega(\theta_1, Y_1) - \omega(\hat{\theta}, \hat{Y})||_{H_1} / ||\omega(\hat{\theta}, \hat{Y})||_{H_1}$
3. if both $e_k <$ TIMETOL and $e_h <$ SPACETOL, perform mesh coarsening and proceed to the next time level with new time step
4. else
    - if $e_k >$ TIMETOL adjust time step $\tau_k = \tau_k \sqrt{\frac{\text{TIMETOL}}{e_k}}$
    - if $e_h >$ SPACETOL perform mesh adaptation

    and repeat process.

4 Numerical example

We consider the parameters $\alpha = 0.8$, $\beta = 10$, $\mathrm{Le} = 1$ and compare the results of two space-time adaptive computations; *hp*-FEM and *h*-FEM with quadratic elements. In both cases, the full Newton's method was used to resolve the nonlinearity. We do not compare the space-time adaptive *hp*-FEM computations with computations using a fixed mesh and/or timestep because we do not see how these methods could be compared fairly.

Fig. 2 shows the reaction rate $\omega(Y, \theta)$ and the underlying *hp*-FEM and *h*-FEM meshes at time $t = 47.4$. The numbers inside elements indicate their polynomial degrees. Notice that very small elements on the flame front are often adjacent to very large elements. This is possible due to the technique of arbitrary-level hanging nodes [4], and for problems with sharp fronts or curvilinear material interfaces, this saves large amounts of degrees of freedom which otherwise would be needed to keep the mesh regular.

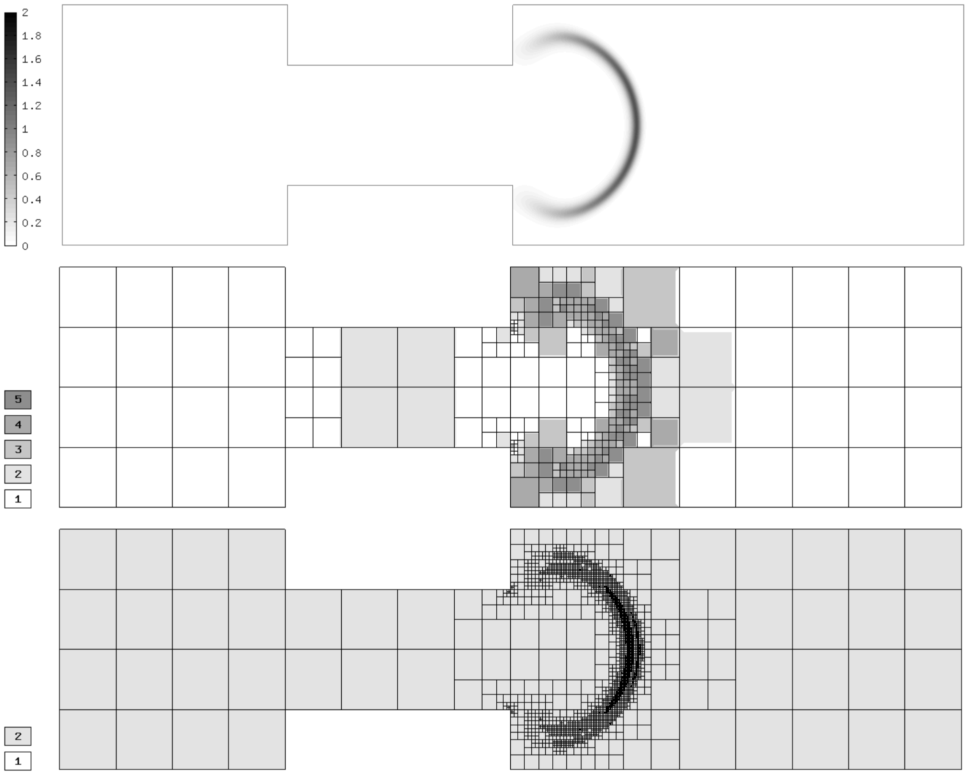


Figure 2: Reaction rate, *hp*-mesh and *h*-mesh at time $t = 47.4$.

Fig. 3 compares the cost of the two computations in terms of the discrete problem size. The reader can see that the adaptive *h*-FEM with quadratic elements required on average 5-6 times more degrees of freedom. It is worth mentioning that also the computational time for low-order FEM computation was more than twice longer. The history of time step size during the computation, for the *hp*-FEM case, is shown in Fig. 3.

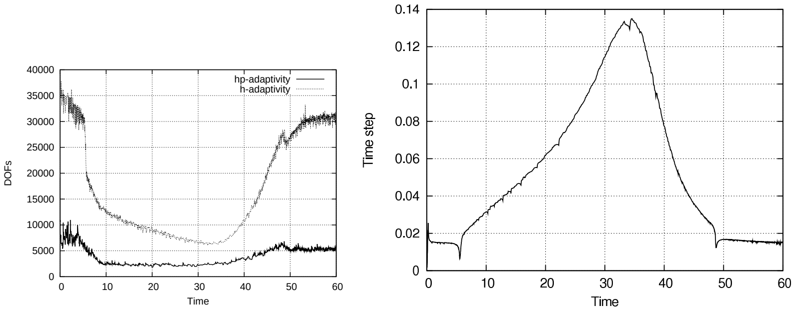


Figure 3: Comparison of discrete problem size as a function of time (left) and the size of time step during the computation (right).

5 Conclusion

We presented a novel PDE-independent space-time adaptive *hp*-FEM algorithm for evolutionary problems based on a combination of the Rothe's method with a novel adaptive multi-mesh *hp*-FEM technique. The method was applied to solve a low Mach number flame propagation model consisting of a pair of coupled nonlinear parabolic PDEs. The method was tested with favorable results against space-time adaptive FEM with low-order (quadratic) elements. The algorithms described in this paper are freely available under the GPL license as part of the modular higher-order finite element C++ library Hermes.

References

[1] R.C. Aldredge: *Turbulent flame propagation with large chemical-heat release.* In: Applied Mathematical Modelling, in press, September 2008.

[2] J. Hoffman, C. Johnson: *Adaptive finite element methods for incompressible fluid flow.* In: T. Barth, H. Deconinck (eds.), 97–157, 2003.

[3] M. Schmich, B. Vexler: *Adaptivity with dynamical meshes for space-time finite element discretizations of parabolic equations.* In: SIAM J. Sci. Comput. Vol. 30 (1), 369–393, 2008.

[4] P. Solin, J. Cerveny, I. Dolezel: *Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM.* In: Math. Comput. Simul 77, 117–132, 2008.

[5] P. Solin, J. Cerveny, L. Dubcova, D. Andrs: *Monolithic discretization of linear thermoelasticity problems via adaptive multimesh hp-FEM.* In: J. Comput. Appl. Math, 2009, doi 10.1016/j.cam.2009.08.092.

[6] P. Solin, L. Dubcova, J. Kruis: *Adaptive hp-FEM with dynamical meshes for transient heat and moisture transfer problems.* In: J. Comput. Appl. Math, 2009, doi 10.1016/j.cam.2009.07.025.

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

On the Ritz values that can be generated by the Arnoldi method

J. Duintjer Tebbens, G. Meurant

Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague

1 Introduction

The Arnoldi method generates approximate eigenvalues of a complex $n \times n$ matrix A by considering a starting unit vector $v \in \mathbb{C}^n$ and a decomposition

$$AV_k = V_k H_k, \qquad V_k e_1 = v,$$

where $V_k^* V_k = I$ and H_k is upper Hessenberg with a positive real lower sub-diagonal. The approximate eigenvalues found in the kth iteration of the Arnoldi method, called Ritz values, are the eigenvalues of H_k. In case A is Hermitian, the method generates a matrix H_k which is tridiagonal and the method is called Lanczos method.

In his 1979 paper, Scott showed that the Lanczos method may converge very slowly in pathologic cases [4]. More precisely, given a Hermitian positive definite matrix A with the eigenvalues

$$\lambda_1 < \lambda_2 < \cdots < \lambda_n,$$

he constructed a perverse starting vector v such that the eigenvalues of H_{n-1} are

$$\frac{\lambda_1 + \lambda_2}{2}, \frac{\lambda_2 + \lambda_3}{2}, \ldots, \frac{\lambda_{n-1} + \lambda_n}{2}.$$

That is, convergence may be postponed until the very last iteration.

This extended abstract deals with generalizations of Scott's result to the Arnoldi algorithm. In the case where A is normal but not Hermitian, we can easily exploit a procedure due to Ericsson to obtain the desired generalization. It turns out that in the next to last iteration one may generate any distribution of Ritz values as long as it satisfies a generalized interlacing property with respect to the spectrum of A. This is done in the next section. The last section presents further generalization for non-normal, diagonalisable matrices and discusses several other issues related to generating prescribed Ritz values in the Arnoldi method.

2 The normal case

The procedure described on page 10 of [1] leads to a method to compute a normal upper Hessenberg matrix $H \in \mathbb{C}^{n \times n}$ with given distinct eigenvalues and given spectrum of its leading principal submatrix. This spectrum, $\mu_1, \ldots, \mu_{n-1}$, must satisfy what is called a generalized interlacing property in [1], namely

$$\Pi^{(r)} \equiv \frac{\prod_{j=1}^{n-1}(\lambda_r - \mu_j)}{\prod_{j=1, j \neq r}^{n}(\lambda_r - \lambda_j)} > 0, \qquad 1 \leq r \leq n, \tag{1}$$

where $\lambda_1, \ldots, \lambda_n$ are the distinct eigenvalues of H.

The following theorem shows how the procedure on page 10 of [1] can be used to construct for a given normal matrix with distinct eigenvalues an initial Arnoldi vector such that the Arnoldi method applied to the normal matrix with the initial Arnoldi vector yields prescribed Ritz values in the next to last step.

Theorem. *Consider a normal matrix A with spectral decomposition $A = W\mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)W^H$ where the eigenvalues $\lambda_1, \ldots, \lambda_n$ are distinct and consider $n-1$ values μ_i such that the generalized interlacing property (1) is satisfied. Let $z \in \mathbb{C}^n$ be any vector satisfying*

$$|z_r|^2 = \Pi^{(r)}, \quad 1 \leq r \leq n,$$

and let

$$\Lambda Z = Z\hat{H}$$

be the Arnoldi decomposition generated by the matrix $\Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)$ and initial vector $Ze_1 = z$, i.e. $Z^H Z = I$ and $\hat{H}$ is upper Hessenberg with a positive real lower subdiagonal. Then the Arnoldi algorithm applied to A with initial vector

$$v \equiv W\bar{Z}e_n$$

generates in the $(n-1)$st iteration the Ritz values $\mu_1, \ldots, \mu_{n-1}$.

Proof: Let H be an upper Hessenberg matrix with eigenvalues $(\lambda_1, \ldots, \lambda_n)$ and leading principal submatrix whose spectrum consists of the values $\mu_1, \ldots, \mu_{n-1}$. Let $HX = X\Lambda$ be the spectral decomposition of H. Paige showed in [2] that

$$|X_{n,r}|^2 = \Pi^{(r)}, \quad 1 \leq r \leq n,$$

see also [3, 1, 5]. Hence there holds $X_{n,r} = e^{i\phi_r} z_r$ for values ϕ_r, $0 \leq \phi_r \leq 2\pi$, and for $1 \leq r \leq n$. If $D = \mathrm{diag}\,(\phi_1, \ldots, \phi_n)$ this means that we have

$$X^T e_n = Dz. \tag{2}$$

Let $P = (e_n, \ldots, e_1)$ be the permutation matrix containing the columns of the identity matrix in reversed order and let $\tilde{H}$ be the upper Hessenberg matrix defined by $\tilde{H} \equiv PH^T P$. Then from $HX = X\Lambda$ we have

$$\Lambda X^T P = X^T P\tilde{H}.$$

This is an Arnoldi decomposition generated by the matrix Λ and initial vector $X^T Pe_1 = X^T e_n = Dz$. On the other hand, from $\Lambda Z = Z\hat{H}$ we obtain

$$D\Lambda Z = \Lambda DZ = DZ\hat{H},$$

i.e. an Arnoldi decomposition generated by the matrix Λ and initial vector $DZe_1 = Dz$. It follows from the uniqueness of the Arnoldi decomposition that the two decompositions are identical and $\hat{H} = \tilde{H}$. Then we obtain from $\Lambda Z = Z\hat{H}$, subsequently,

$$\begin{aligned} \Lambda ZP = ZPP\tilde{H}P &= ZPH^T, \\ PZ^T\Lambda &= HPZ^T, \\ PZ^T W^H W\Lambda W^H &= HPZ^T W^H, \\ AW(PZ^T)^H &= W(PZ^T)^H H. \end{aligned}$$

Because $W(PZ^T)^H$ is unitary, the last equation represents the Arnoldi decomposition generated by the matrix A and initial vector $W\bar{Z}Pe_1 = W\bar{Z}e_n$. Its Hessenberg matrix has the desired Ritz values. □

3 Further generalizations

The previous theorem can be generalized to the case where A is diagonisable but not necessarily normal. The eigenvalues must be distinct and the Ritz values must satisfy a modification of (1).

Theorem. *Let A be diagonalisable with spectral decomposition $A = W\mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)W^{-1}$ where the eigenvalues $\lambda_1, \ldots, \lambda_n$ are distinct and consider $n-1$ values μ_i such that the nonlinear system of equations in the complex variables $z_1, \ldots, z_n$,*

$$diag(z_1, \ldots, z_n)(W^H W)^{-1}\overline{z} = \begin{pmatrix} \Pi^{(1)} \\ \ldots \\ \Pi^{(n)} \end{pmatrix} \tag{3}$$

has a solution. Let $z \in \mathbb{C}^n$ be any vector satisfying (3) and let

$$\overline{\Lambda} Z = Z\hat{H}$$

be the $(W^H W)^{-1}$-orthogonal Arnoldi decomposition generated by the complex conjugate of the matrix $\Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)$ and initial vector $Ze_1 = \overline{z}$, i.e. $Z^H(W^H W)^{-1}Z = I$ and $\hat{H}$ is upper Hessenberg with a positive real lower subdiagonal. Then the Arnoldi algorithm applied to A with initial vector

$$v \equiv WZ^{-H}e_n$$

generates in the $(n-1)$st iteration the Ritz values $\mu_1, \ldots, \mu_{n-1}$.

In our talk we plan to address the proof of this theorem and the geometric meaning of the interlacing properties (1) and (3). We also envisage to discuss generating prescribed Ritz values in arbitrary iteration numbers smaller than $n-1$. Finally, we will mention the problem of generating prescribed Ritz values when A is not given but, as the starting vector v, is constructed. In this case it is possible to prescribe the Ritz values of more than one iteration number.

Acknowledgements: The authors thank Zdeněk Strakoš for initiating work on this topic. The work of the first author was supported by project number KJB100300703 of the Grant Agency of the Academy of Sciences of the Czech Republic.

References

[1] T. Ericsson: *On the eigenvalues and eigenvectors of Hessenberg matrices.* Technical report, 1990.

[2] C.C. Paige: *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices.* Ph.D. Thesis, University of London, 1971.

[3] B.N. Parlett: *The symmetric eigenvalue problem.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1980. Prentice-Hall Series in Computational Mathematics.

[4] D.S. Scott: *How to make the Lanczos algorithm converge slowly.* Math. Comp., 33(145), 239–247, 1979.

[5] J.-P.M. Zemke: *Hessenberg eigenvalue-eigenmatrix relations.* Linear Algebra Appl., 414(2-3), 589–606, 2006.

Potential and Hamiltonian in the Filippov systems

T. Hanus, D. Janovská

Institute of Chemical Technology, Prague

1 Introduction

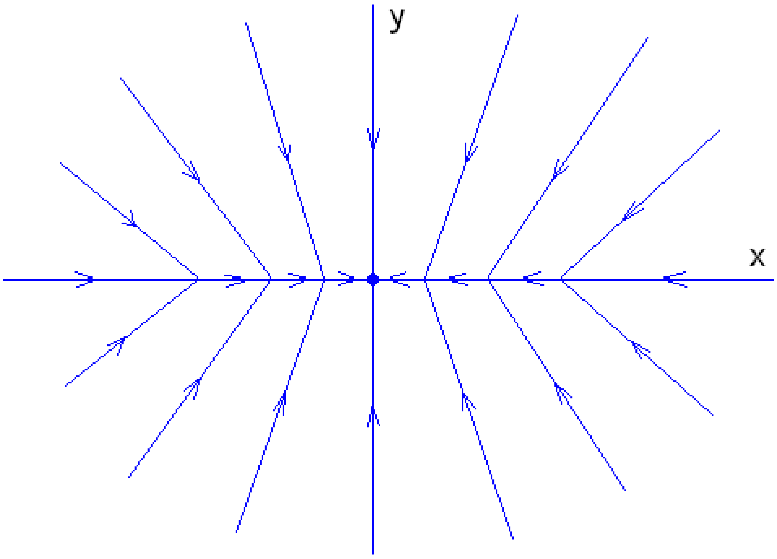
We try to apply the qualitative analysis tools of the continuous dynamical systems onto the discontinuous ones. Functions like a potential, a pseudopotential, a hamiltonian, a pseudohamiltonian can describe the global behaviour of the Filippov system.

2 Potential and pseudopotential

Let us have the Filippov system

$$\mathcal{F}:\begin{cases}\left.\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t}&=-x\\ \frac{\mathrm{d}y}{\mathrm{d}t}&=-y+1\end{aligned}\right\}x\in\mathbb{R},\ y<0,\\ \left.\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t}&=-x\\ \frac{\mathrm{d}y}{\mathrm{d}t}&=-y-1\end{aligned}\right\}x\in\mathbb{R},\ y>0.\end{cases}\tag{1}$$

Its phase portrait is



The function

$$U(x,y)=\begin{cases}-\dfrac{1}{2}x^2-\dfrac{1}{2}(y-1)^2,\, x\in\mathbb{R},\ y<0,\\ -\dfrac{1}{2}x^2-\dfrac{1}{2}(y+1)^2,\, x\in\mathbb{R},\ y\geq 0,\end{cases}\tag{2}$$

is the potential of the system (1). We can see that the gradient of (2) is equal to the vector field of (1) in all points, except for those with $y=0$:

$$\left[\frac{\partial U}{\partial x},\frac{\partial U}{\partial y}\right]=\begin{cases}\left[-x,-y+1\right],\, x\in\mathbb{R},\ y<0,\\ \left[-x,-y-1\right],\, x\in\mathbb{R},\ y>0.\end{cases}$$

The function

$$\widehat{U}(x,y)=(-2)\cdot U(x,y)=\begin{cases}x^2+(y-1)^2,\, x\in\mathbb{R},\ y<0,\\ x^2+(y+1)^2,\, x\in\mathbb{R},\ y\geq 0,\end{cases}\tag{3}$$

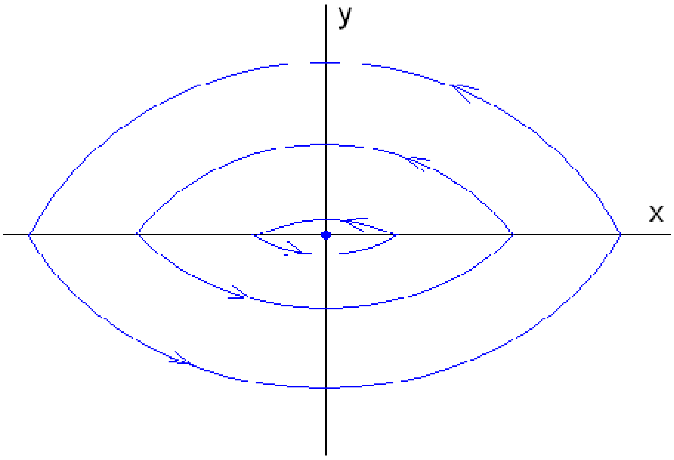
is the pseudopotential of the system (1). The fall lines of (3) projected into the plane xy are the trajectories of (1).

3 Hamiltonian and pseudohamiltonian

Now, let us have the Filippov system

$$\mathcal{F}:\begin{cases}\left.\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t}&=-y+1\\ \frac{\mathrm{d}y}{\mathrm{d}t}&=x\end{aligned}\right\}x\in\mathbb{R},\ y<0,\\ \left.\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t}&=-y-1\\ \frac{\mathrm{d}y}{\mathrm{d}t}&=x\end{aligned}\right\}x\in\mathbb{R},\ y>0,\end{cases}\tag{4}$$

with its phase portrait



The function

$$H(x,y)=\begin{cases}-\dfrac{1}{2}x^2-\dfrac{1}{2}(y-1)^2,\, x\in\mathbb{R},\ y<0,\\ -\dfrac{1}{2}x^2-\dfrac{1}{2}(y+1)^2,\, x\in\mathbb{R},\ y\geq 0.\end{cases} \tag{5}$$

is the hamiltonian of the system (4). We can notice, that the condition

$$\left[\frac{\partial H}{\partial y},-\frac{\partial H}{\partial x}\right]=\begin{cases}\left[-y+1,x\right],\, x\in\mathbb{R},\ y<0,\\ \left[-y-1,x\right],\, x\in\mathbb{R},\ y>0,\end{cases}$$

holds for all points, except for those with $y=0$. So the gradient of (5) is always orthogonal to the vector field of (4).

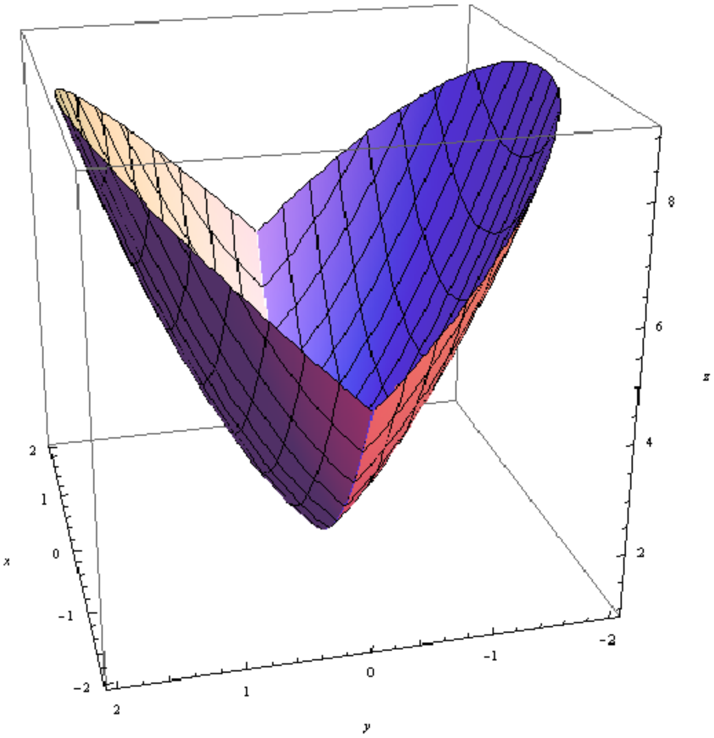
The function

$$\widehat{H}(x,y)=(-2)\cdot H(x,y)=\begin{cases}x^2+(y-1)^2,\, x\in\mathbb{R},\ y<0,\\ x^2+(y+1)^2,\, x\in\mathbb{R},\ y\geq 0.\end{cases} \tag{6}$$

is the pseudohamiltonian of the system (4). The level curves of (6) are the trajectories of (1) free of orientation.

4 Graph

Here we have 3D graph of (3) and (6). They happened to be the same, which is not the rule, of course.



It is a function of two variables, continuous, but not smooth. It is not differentiable at the points with $y=0$.

5 Conclusion

A discontinuous dynamical system is defined piecewise, there is a special formula on each subdomain of the state space. Handling such system requires switching formulas on the boundary between the subdomains. This work is the first step of the long journey. Once we will have one formula of a dynamical system on the entire state space no matter if it is continuous or discontinuous.

Acknowledgement: This work has been supported by the grant MSM 6046137306.

References

[1] A. F. Filippov: *Differencialʹnyje uravnĕnija s razryvnoj pravoj častʹju.* Nauka, Moskva, 1985.

[2] M. W. Hirsch, S. Smale, R. L. Devaney: *Differential equations, dynamical systems, and an introduction to chaos.* Elsevier Academic Press, USA, 2004.

[3] A. Klíč, M. Kubíček: *Matematika III, Diferenciální rovnice.* VŠCHT, Praha, 1992.

[4] Y. A. Kuznetsov, S. Rinaldi, A. Gragnani: *One-parameter Bifurcations in Planar Filippov Systems.* In: Int. J. Bifurcation and Chaos, 2003, 2157–2188.

Redukce diskrétní puklinové sítě a její vliv na řešení úlohy proudění

J. Havlíček, M. Hokr, J. Kopal, P. Rálek

Fakulta mechatroniky, informatiky a mezioborových studií
Technická univerzita v Liberci

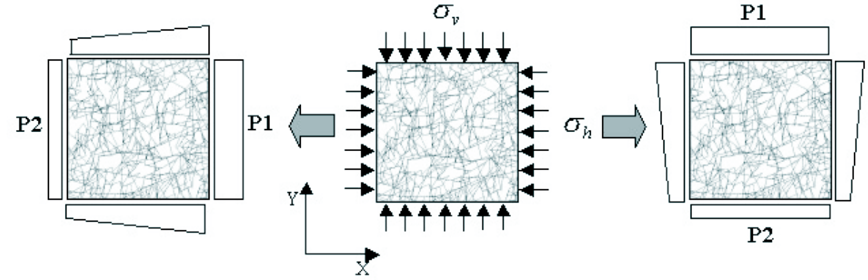
1 Úvod

Příspěvek se věnuje popisu vlastností algoritmu pro redukci dvourozměrné puklinové sítě a vliv této redukce na řešení úlohy proudění v diskrétní puklinové síti s mechanickým zatížením. Problém vznikl jako součást projektu Decovalex-2011, Task C [1].

Potřeba redukovat rozsáhlou puklinovou síť tak, aby zůstaly přibližně zachovány její hydraulické vlastnosti (nebo abychom tuto změnu uměli vysledovat), vznikla při snaze implementovat sdruženou úlohu proudění–mechanika. Vliv mechanického zatížení na rozevření pukliny je v současné době počítán analyticky zvlášť pro každou puklinu, nicméně vzniklý reduktor sítě poskytuje zajímavé poznatky o vlastnostech puklinové sítě, kdy velkou roli z hlediska toku oblastí hraje malý počet velkých puklin.

2 Popis úlohy

Modelovanou oblastí je čtverec o délce strany 20m, na kterém je stochasticky generovaná diskrétní puklinová síť (s parametry podle reálných měření [2], [3]). Rozevření puklin je úměrné jejich délce. Pro úlohu proudění je zadána Dirichletova okrajová podmínka pro tlakovou výšku a gradient tlaku (dvě různé okrajové podmínky jsou znázorněny na obr. 1 vlevo a vpravo). Podrobnému popisu numerického řešení úlohy proudění (bez zahrnutí mechanického zatížení) za použití programu Flow123D se věnuje [5].



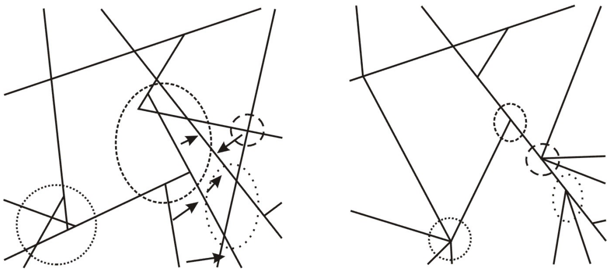
Obrázek 1: Okrajové podmínky pro proudění a mechaniku.

Pro mechaniku jsou zadána horizontální a vertikální napětí (viz obr. 1 uprostřed). Při působícím mechanickém napětí některé pukliny zmenší svoje rozevření a jejich hydraulická vodivost poklesne. U vhodně orientovaných puklin vůči vnějšímu napětí dochází ke zvětšení rozevření a hydraulická vodivost roste. Mechanický model puklin (vzorce pro analytický výpočet napětí, deformace, tuhost puklin, pevnostní kritéria) byl převzat z [3]. Podprobnému popisu vlivu mechaniky na hydraulické vlastnosti puklinové sítě se věnuje [6].

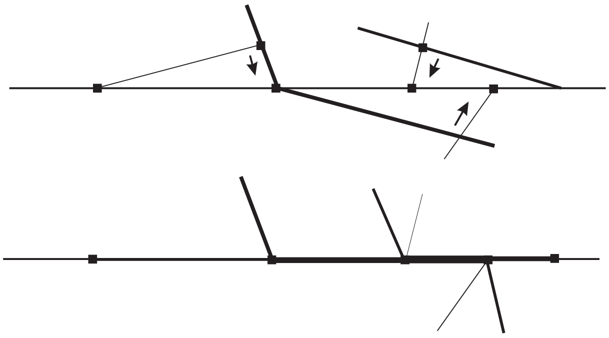
3 Redukce puklinové sítě

Kvůli příliš velké hustotě puklinové sítě vznikl požadavek na její redukci při přibližném zachování hydraulických vlastností tak, abychom na řidší síti již mohli řešit složitější úlohu než prosté proudění (proudění–mechanika, zahrnutí transportu částic). Pro redukci puklinové sítě (tzn. snížení počtu puklin a jejich průsečíků – uzlů) lze použít několik postupů.

- Vymazání (volitelně) malých puklin: u dostatečně malých puklin lze předpokládat, že jejich vliv na celkové proudění bude malý (rozevření puklin je úměrné jejich délce).
- Vymazání slepých úseků puklin: části puklin mezi posledním průsečíkem (uzlem) s jinou puklinou a koncem pukliny nemají vliv na ustálené proudění v puklinové síti.
- Slučování (volitelně) blízkých uzlů (obr. 2): cílem je snížit celkový počet průsečíků puklin. Slučují se vždy dva uzly ležící na téže puklině, zůstane tak zachována sousednost v puklinové síti. Slučováním uzlů lze odstranit miniaturní úseky puklin a zlepšit poměr mezi nejmenším a největším elementem v diskretizaci.
- Slučování souběžných úseků puklin (obr. 3): v rámci slučování uzlů dochází ke sloučení celých úseků puklin. V daném úseku se sečtou rozevření slučovaných puklin a vodivost se přepočte z vodivostí slučovaných puklin tak, aby byl zachován tok puklinou.

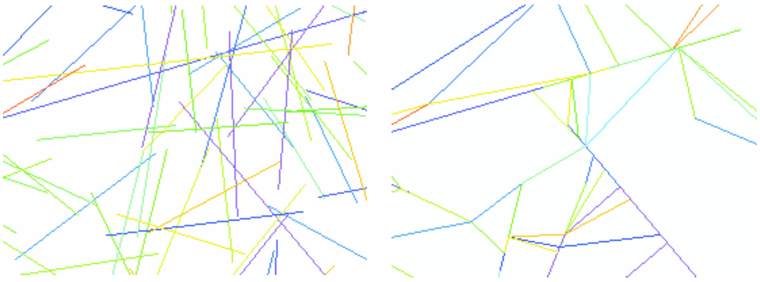


Obrázek 2: Ukázka slučování uzlů.

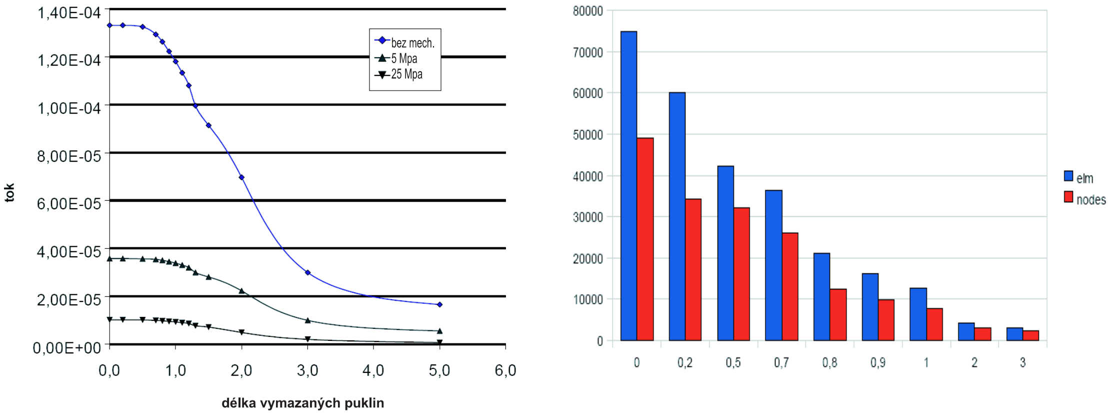


Obrázek 3: Princip slučování úseků puklin.

Uvedené postupy lze různě kombinovat. Geometrické a hydraulické vlastnosti výsledné sítě závisejí na zvolených parametrech redukce (ty mohou být buď robustní nebo jemné). Stejně tak je možné ovlivnit výsledek redukce preferencí dominantních puklin, které by neměly být průběhem redukce ovlivněny. Na obr. 4 je výřez části sítě před a po redukci. Zřetelné jsou dvě dominantní pukliny, které se při redukci nemění.



Obrázek 4: Ukázka části redukované sítě.



Obrázek 5: Závislost výstupního toku (vlevo) a velikosti úlohy (vpravo) na velikosti vymazaných puklin.

4 Ukázky výsledků výpočtů

Byla provedena sada výpočtů s různými parametry redukce, bez zahrnutí vlivu mechanického zatížení i s ním (vertikální napětí je 5 MPa a horizontální napětí 5 Mpa nebo 25 MPa). Výsledky prezentované zde v abstraktu jsou pro okrajové podmínky znázorněné na obr. 1 vpravo, sledován byl výstupní tok dolní částí hranice oblasti (podle předpokladů při zvyšujícím se horizontálním zatížení celkový vertikální tok klesá). Graf na obr. 5 vlevo popisuje změnu výstupního toku hranicemi oblasti v závislosti na množství vymazaných malých puklin. Pukliny do velikosti 1m ovlivňují celkový výstupní tok jen velmi málo. Velikost úlohy však klesne významným způsobem (obr. 5 vpravo). Slučováním uzlů a souběžných puklin lze dosáhnout dalšího snížení velikosti úlohy (např. pro síť s vynechanými puklinami kratšími než 1m s parametrem slučování 0.4 metru o cca polovinu), zde však již hraje velkou roli vliv mechaniky. V úloze proudění bez zahrnutí mechaniky je vliv slučování uzlů a souběžných puklin malý (změna toku je menší než 10%), při zahrnutí mechaniky je tento vliv nárůst hydraulické vodivosti puklin mnohem vyšší, než odpovídá nárůstu vodivosti na neredukované síti. Kromě toho, že dochází ke změně geometrie sítě, to může být způsobeno tím, že při redukci vzniká menší počet mnohem širších puklin, pro které je již použitý mechanický model nepřesný.

5 Závěr

Testovací úlohy ukázaly, že lze výrazným způsobem snížit velikost puklinové sítě, aniž to (pro vhodně zvolené parametry redukce) výrazně ovlivní výslednou bilanci toků hranicí oblasti. Zvolit správně vyvážené parametry redukce není snadné a je třeba pokaždé testovat, nakolik změna sítě ovlivní výsledné toky. Pro úlohy se zahrnutím mechaniky již dochází k velké odchylce ve výsledcích oproti tokům na neredukované síti. Tomu lze částečně předejít spočítáním vlivu mechanického zatížení na dominantní pukliny na neredukované síti a poté provést redukci.

Redukované síti s podobnými hydraulickými vlastnostmi jako síť neredukovaná by měla v budoucnu umožnit složitější výpočty, např. zahrnutí transportu či složitějších mechanických modelů pukliny, než na neredukované síti. Vyvstává úloha nalézt způsob, jak měnit materiálové parametry sítě (např. permeabilitu) tak, aby výsledné toky hranicemi na redukované síti byly shodné (ve větší míře než nyní) s toky na síti neredukované.

Acknowledgement: Tento příspěvek vznikl za podpory MPO ČR, projekt FR-TI1/362, a v rámci účasti na mezinárodním projektu Decovalex.

Reference

[1] J.A. Hudson, I. Neretnieks, L. Jing: *DECOVALEX-2011 project, Technical Definition of the 2-D BMT Problem for Task C.* May 2008.

[2] A. Baghbanan, L. Jing: *Hydraulic properties of fractured rock masses with correlated fracture, length and aperture.* In: International Journal of Rock Mechanics and Mining Sciences, 44(5), 704-719, 2007.

[3] A. Baghbanan, L. Jing: *Stress effects on permeability in fractured rock mass with correlated, fracture length and aperture.* In: International Journal of Rock Mechanics and Mining Sciences, 45(8), 1320-1334, 2008.

[4] B. Malá: *Fracture net.* In: Technical Report TUL, June 2008.

[5] M. Hokr, J. Kopal, J. Havlíček: *Řešení úlohy proudění v rozsáhlé diskrétní síti puklin v kontextu sdružených úloh proudění-mechanika.* In: Sborník SNA 2009.

[6] M. Hokr, J. Havlíček: *Změna hydraulických parametrů v modelu proudění diskrétní puklinovou sítí při zahrnutí vlivu mechaniky.* In: Sborník SIMONA 2009, 45-51.

[7] P. Rálek: *Algoritmus pro redukci puklinové sítě v úloze 2D proudění.* In: Sborník SIMONA 2009, 104-110.

Two-sided quaternionic polynomials

D. Janovská, G. Opfer

Institute of Chemical Technology, Prague
University of Hamburg, Hamburg

1 Introduction

A general, quaternionic polynomial consists of a sum of terms of the type

$$t_j(z) := a_{0j} \cdot z \cdot a_{1j} \cdots a_{j-1,j} \cdot z \cdot a_{jj}, \quad z, a_{0j}, a_{1j}, \ldots, a_{jj} \in \mathbb{H},\ j \geq 0.$$

We call this term a *monomial of degree* j. Since there may be several terms of the same degree we have to enumerate the terms. We do that in the form

$$t_{jk}(z) := a_{0j}^{(k)} \cdot z \cdot a_{1j}^{(k)} \cdots a_{j-1,j}^{(k)} \cdot z \cdot a_{jj}^{(k)}, \quad k = 1, 2, \ldots, k_j,\ k_j \geq 0.$$

The case $k_j = 0$ means that there is no monomial of degree j. *A general, quaternionic polynomial of degree* n takes the form

$$p(z) := \sum_{j=0}^{n} \sum_{k=1}^{k_j} t_{jk}(z). \tag{1}$$

We will treat quaternionic polynomials of the *two-sided type*

$$p(z) := \sum_{j=0}^{n} a_j z^j b_j, \quad z,\ a_j,\ b_j \in \mathbb{H},\ a_0 b_0 \neq 0,\ a_n b_n \neq 0, \tag{2}$$

where $\mathbb{H}$ is the skew field of quaternions. These polynomials include also the *one-sided* polynomials, where all coefficients are located on the left or on the right side of the powers, see [4].

By $\mathbb{R}$, $\mathbb{C}$ we denote the fields of real and complex numbers, respectively. By $\mathbb{H}$ we denote the skew field of quaternions that consists of elements of $\mathbb{R}^4$, equipped with the multiplication rule

$$\begin{aligned} ab \ := \ & (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4, a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3, \\ & a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2, a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1), \end{aligned} \tag{3}$$

where $a := (a_1, a_2, a_3, a_4),\ b := (b_1, b_2, b_3, b_4),\ a_j, b_j \in \mathbb{R},\ j = 1, 2, 3, 4.$

The multiplication rule implies, in particular,

$$\Re(ab) = \Re(ba) \text{ and } ra = ar \text{ for } a, b \in \mathbb{H},\ r \in \mathbb{R}. \tag{4}$$

By 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ we denote the standard units in $\mathbb{H}$. Two quaternions $a, b \in \mathbb{H}$ are called *equivalent*, if there is an $h \in \mathbb{H} \setminus \{0\}$ such that $b = h^{-1}ah$. Equivalent quaternions a, b will be denoted by $a \sim b$. The set

$$[a] := \{u \in \mathbb{H} : u = h^{-1}ah \text{ for all } h \in \mathbb{H} \setminus \{0\}\} \tag{5}$$

will be called an *equivalence class* of a.

Equivalent quaternions a, b can be easily recognized by

$$a \sim b \;\Leftrightarrow\; \Re a = \Re b \text{ and } |a| = |b|, \text{ see [3]}, \tag{6}$$

where $\Re a$ denotes the *real part*, the first component of a, and $|a|$ denotes the *absolute value* of $a = (a_1, a_2, a_3, a_4)$, $|a| := \sqrt{a_1^2 + a_2^2 + a_3^2 + a_4^2}$. We also introduce $\Im a$, the *imaginary part*, the second component of a. Let a be real. Then $[a] = \{a\}$, which means, that in this case, the equivalence class consists only of one element, $\{a\}$. If a is not real, then $[a]$ always contains infinitely many elements,

$$[a] := \{z \in \mathbb{H} : \; \Re z = \Re a, \; |z| = |a|\}, \tag{7}$$

which may be regarded as the surface of a ball in $\mathbb{R}^3$. Let $z \in \mathbb{H}$ be not real and $z = (z_1, z_2, z_3, z_4)$. Then $[z]$ contains exactly two complex numbers, $a \in \mathbb{C}$ and $\overline{a} \in \mathbb{C}$ where a is determined by $\Re a = z_1$ and $\Im a = +\sqrt{z_2^2 + z_3^2 + z_4^2} > 0$. The complex number a will be called the *complex representative* of $[z]$.

All powers $z^j, j \in \mathbb{N}$, of a quaternion z have the form $z^j = \alpha z + \beta$ with real α, β, see [6], [4]. In order to determine the numbers α, β we set up the following iterations, see [4],

$$\begin{aligned} z^j &= \alpha_j z + \beta_j, \quad \alpha_j, \beta_j \in \mathbb{R}, \quad j = 0, 1, \ldots, \text{ where} \\ \alpha_0 &= 0, \quad \beta_0 = 1, \\ \alpha_{j+1} &= 2\Re z \, \alpha_j + \beta_j, \\ \beta_{j+1} &= -|z|^2 \alpha_j, \quad j = 0, 1, \ldots \end{aligned} \tag{8}$$

2 Types of zeros of two-sided polynomials

Let $z \in \mathbb{R}$ be a real zero of p, defined in (1). Since a real z commutes with all quaternions the polynomial can be written in the form

$$p(z) = \sum_{j=0}^{n} A_j z^j \text{ where } A_j := \sum_{k=1}^{k_j} a_{0j}^{(k)} a_{1j}^{(k)} \cdots a_{jj}^{(k)}, \; z \in \mathbb{R}, \tag{9}$$

i.e. as an one-sided quaternionic polynomial, see [4]. We will skip the discussion on real zeros in the sequel.

By means of (8), the polynomial p can be written as

$$p(z) = \sum_{j=0}^{n} a_j z^j b_j = \sum_{j=0}^{n} a_j(\alpha_j z + \beta_j) b_j = \sum_{j=0}^{n} \alpha_j a_j z b_j + \sum_{j=0}^{n} \beta_j a_j b_j = C(z) + B(z), \tag{10}$$

$$\text{where} \qquad C(z) = \sum_{j=0}^{n} \alpha_j a_j z b_j, \quad B(z) = \sum_{j=0}^{n} \beta_j a_j b_j. \tag{11}$$

Let C be defined as in (11). Then, $C : \mathbb{R}^4 \to \mathbb{R}^4$ is a linear mapping over $\mathbb{R}$. Let z_0 be nonreal. Then, $B(z)$, defined in (11), is constant for $z \in [z_0]$. If $p(z) = 0$ for some $z \in \mathbb{H}$, then $C(z) = B(z) = 0$ or $C(z) \neq 0$ and $B(z) \neq 0$. For details, see [5].

We introduce two mappings $\tau_1, \tau_2 : \mathbb{H} \to \mathbb{R}^{4\times 4}$ by

$$\tau_1(a) := \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & -a_4 & a_3 \\ a_3 & a_4 & a_1 & -a_2 \\ a_4 & -a_3 & a_2 & a_1 \end{pmatrix} \in \mathbb{R}^{4\times 4}, \;\; \tau_2(a) := \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix} \in \mathbb{R}^{4\times 4}. \tag{12}$$

The first mapping τ_1 represents the isomorphic image of a quaternion $a = (a_1, a_2, a_3, a_4)$ in the matrix space $\mathbb{R}^{4\times 4}$. Thus, we have $\tau_1(ab) = \tau_1(a)\tau_1(b)$. The two matrices $\tau_1(a), \tau_2(b)$ coincide if and only if $a = b \in \mathbb{R}$, see [2]. The second mapping τ_2, see [1], has the remarkable property that it reverses the multiplication order

$$\tau_2(ab) = \tau_2(b)\tau_2(a).$$

From the definition (12) it follows that

$$\tau_1(a)^{\mathrm{T}} = \tau_1(\overline{a}),\ \tau_2(b)^{\mathrm{T}} = \tau_1(\bar{b}).$$

Both matrices are orthogonal in the sense $\tau_1(a)\tau_1(a)^{\mathrm{T}} = \tau_1(a)\tau_1(\overline{a}) = |a|^2\,\mathbf{I}$, $\tau_2(b)\tau_2(b)^{\mathrm{T}} = |b|^2\,\mathbf{I}$, where $\mathbf{I}$ is the (4×4) identity matrix.

Let $a := (a_1, a_2, a_3, a_4) \in \mathbb{H}$. We introduce an *column* operator $\mathrm{col} : \mathbb{H} \to \mathbb{R}^4$ by

$$\mathrm{col}(a) := \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}.$$

This column operator enables us to regard a quaternion as a matrix with one column and four rows. It is linear over $\mathbb{R}$, i. e.

$$\mathrm{col}(\alpha a + \beta b) = \alpha \mathrm{col}(a) + \beta \mathrm{col}(b), \quad a, b \in \mathbb{H},\ \alpha, \beta \in \mathbb{R}.$$

For arbitrary quaternions a, b, c we have

$$\begin{aligned} \mathrm{col(ab)} &= \tau_1(a)\mathrm{col(b)} = \tau_2(\mathrm{b})\mathrm{col(a)}, \\ \mathrm{col(abc)} &= \tau_1(a)\tau_2(c)\mathrm{col(b)}. \end{aligned}$$

For more properties of these mappings, see [5].

Theorem Let $p(z) := C(z) + B(z)$ be defined as in (10), (11). Then,

$$\begin{aligned} \mathrm{col(p(z))} &= \left(\sum_{j=0}^{n} \alpha_j \tau_1(a_j)\tau_2(b_j)\right) \mathrm{col(z)} + \sum_{\mathrm{j=0}}^{\mathrm{n}} \beta_{\mathrm{j}} \mathrm{col(a_j b_j)} && (13) \\ &=: \mathbf{A}(z)\mathrm{col(z)} + \mathrm{col(B(z))},\ \text{where} && (14) \\ \mathbf{A}(z) &:= \left(\sum_{j=0}^{n} \alpha_j \tau_1(a_j)\tau_2(b_j)\right) \in \mathbb{R}^{4\times 4},\ \mathrm{col(B(z))} := \sum_{\mathrm{j=0}}^{\mathrm{n}} \beta_{\mathrm{j}} \mathrm{col(a_j b_j)}. && (15) \end{aligned}$$

Instead of considering the equation $p(z) = 0$ we consider the equivalent equation

$$P(z) := \mathrm{col(p(z))} = \mathbf{A}(\mathrm{z})\mathrm{col(z)} + \mathrm{col(B(z))} = \mathrm{col}(0) := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} =: 0. \qquad (16)$$

From this formula we obtain: Let z be a nonreal zero of p such that equation (16) is valid. Then, this equation remains valid if in $\mathbf{A}(z)$, $B(z)$ the zero z is replaced with the complex representative z_0 of $[z]$. In order to find the nonreal zeros $z \in \mathbb{H}$ of p, defined in (2), it is sufficient to find the complex representative z_0 of $[z]$, where, in general, z_0 is not a zero of p. The matrix $\mathbf{A}(z)$, occurring in (16) may be singular or non singular.

From these results we obtain the classification of the zeros of p as follows: Let z be a zero of p, defined in (2), and let $z_0 \in [z]$ be the complex representative of $[z]$. We classify the zeros z of p with respect to the rank of $\mathbf{A}(z_0)$. The zero z is called *zero of type k* if $\mathrm{rank}(\mathbf{A}(\mathrm{z_0})) = 4 - \mathrm{k}$, $0 \leq k \leq 4$. A zero of type 4 ($\mathrm{rank}(\mathbf{A}(\mathrm{z_0})) = 0$) is called *spherical zero*. It has the property that all $z \in [z_0]$ are zeros. A zero of type 0 is called *isolated zero*. In this case $z = -(\mathbf{A}(z_0))^{-1}\mathrm{col}\,(\mathrm{B}(\mathrm{z_0}))$ is the only zero in $[z_0]$. We also call a real zero an isolated zero.

Since the polynomial $p(z) := z^2 + 1$ has already infinitely many zeros in $\mathbb{H}$, it makes no sense to count the individual zeros. Let p be any quaternionic polynomial of degree $n \geq 2$. By $\#Z(p)$ we understand the number of equivalence classes in $\mathbb{H}$ which contain zeros of p. We call this number, *essential number of zeros of p*. Let p be a polynomial of degree n of the form described in (2). It can be proofed, see [5], that $\#Z(p)$, the essential number of zeros of p, is not bounded by n, but it will not exceed $2n$.

3 Conclusions

For quaternionic polynomials p of the two-sided type, we have shown that the zeros z may fall into five different classes, where for each zero the class can be determined by looking at the rank of a (4×4) matrix $\mathbf{A}(z)$ defined in (16). If a zero in one class has been found, the described technique allows to find all zeros in the same class.

The representation of a given quaternionic, two-sided polynomial p in the form $P(z) := \mathbf{A}(z)z + B(z)$ can be used not only for the classification of the zeros, but it can be also successfully applied to finding the zeros, by applying Newton's method to $P(z) = 0$. It shows the typical feature, that it may be slow in the beginning, but it will terminate then very quickly with quadratic rate.

Acknowledgement: The research was supported by the research project MSM 6046137306 of The Ministry of Education, Youth and Sports, Czech Republic.

References

[1] L.I. Aramanovitch: *Quaternion Non-linear Filter for Estimation of Rotating body Attitude*. Mathematical Meth. in the Appl. Sciences, 18, 1239–1255, 1995.

[2] K. Gürlebeck, W. Sprössig: *Quaternionic and Clifford Calculus for Physicists and Engineers.* Wiley, Chichester, 371 p., 1997.

[3] D. Janovská, G. Opfer: *Givens' transformation applied to quaternion valued vectors.* BIT, 43, 991–1002, 2003.

[4] D. Janovská, G. Opfer: *A note on the computation of all zeros of simple quaternionic polynomials.* SIAM J. Numer. Anal., 12 p, 2009, accepted.

[5] D. Janovská, G. Opfer: *The classification and the computation of the zeros of quaternionic, two-sided polynomials.* Numer. Math., 20 p., 2009, DOI: 10.1007/s00211-009-0274-y.

[6] A. Pogorui, M. Shapiro: *On the structure of the set of zeros of quaternionic polynomials.* Complex Variables and Elliptic Functions, 49, 379–389, 2004.

Stability of non unique solutions of the Coulomb friction problem

V. Janovský

Charles University, Faculty of Mathematics and Physics, Prague

1 Discrete static contact problems with Coulomb friction

Let $\Omega \subset \mathbb{R}^2$ be a linearly elastic body supported by a rigid foundation along the contact boundary Γ_C, see Figure 1. On Γ_N and Γ_D, Neumann and Dirichlet boundary conditions are prescribed. We consider the static contact problems with Coulomb friction, see e.g. [1]. In particular, we will investigate a discrete version of this problem, see e.g. [2]. This may be understood as a *FEM-approximation* of the continuous mechanical problem:

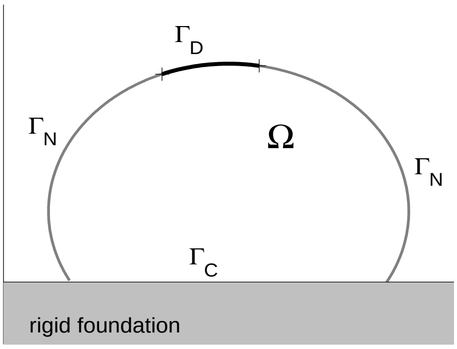


Figure 1: 2D elastic body Ω in frictional contact.

Let integers n and p define the degrees of freedom of the body Ω and the number of contact nodes on Γ_C, $n \geq 2p$. Let $\mathbf{f} \in \mathbb{R}^n$ and $\mathcal{F}$ be the given distributed volume force and the friction coefficient. We seek for

- distributed displacement field $\mathbf{u} \in \mathbb{R}^n$,
- distributed normal and tangential stress components $\boldsymbol{\lambda}_\nu \in \mathbb{R}^p$ and $\boldsymbol{\lambda}_t \in \mathbb{R}^p$.

First, we formulate the static Tresca friction problem: Given a prescribed slip stress $\mathbf{g} = (\mathrm{g}_1, \ldots, \mathrm{g}_p) \in \mathbb{R}^p_+$, find $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(\mathcal{F}, \mathbf{g})$, such that

$$(\mathbb{A}\mathbf{u}, \mathbf{v})_n = (\mathbf{f}, \mathbf{v})_n + (\boldsymbol{\lambda}_\nu, \mathbb{N}\mathbf{v})_p + (\boldsymbol{\lambda}_t, \mathbb{T}\mathbf{v})_p \quad \forall\, \mathbf{v} \in \mathbb{R}^n, \tag{1}$$

$$(\boldsymbol{\mu}_\nu - \boldsymbol{\lambda}_\nu, \mathbb{N}\mathbf{u})_p + (\boldsymbol{\mu}_t - \boldsymbol{\lambda}_t, \mathbb{T}\mathbf{u})_p \geq 0 \quad \forall\, (\boldsymbol{\mu}_\nu, \boldsymbol{\mu}_t) \in \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(\mathcal{F}, \mathbf{g})\,. \tag{2}$$

Here, $\mathbb{A} \in \mathbb{R}^{n\times n}$ is a positive definite stiffness matrix. The full-rank matrices $\mathbb{N} \in \mathbb{R}^{p\times n}$ and $\mathbb{T} \in \mathbb{R}^{p\times n}$ represent the actions of the distributed contact forces along the normal and tangential directions. The sets

$$\boldsymbol{\Lambda}_\nu = \mathbb{R}^p_-\,, \quad \boldsymbol{\Lambda}_t(\mathcal{F}, \mathbf{g}) = \{\boldsymbol{\mu}_t \in \mathbb{R}^p : |\mu_{t,i}| \leq \mathcal{F}\mathrm{g}_i\,, \quad \forall\, i = 1, \ldots, p\} \tag{3}$$

are the sets of Lagrange multipliers. Under generic assumptions, the problem (1)&(2) is uniquely solvable for any data $\mathbf{f}$, $\mathcal{F}$ and $\mathbf{g}$.

To a given $\mathbf{g} \in \mathbb{R}^p_+$, let us assign the solution $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$ of the problem (1)&(2). In particular, we consider the map $\Gamma : \mathbb{R}^p_+ \to \mathbb{R}^p_+$, which is defined as

$$\Gamma(\mathbf{g}) \longmapsto -\boldsymbol{\lambda}_\nu \,. \tag{4}$$

The solution $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$ of (1)&(2), which corresponds to the fixed point $\Gamma(\mathbf{g}) = -\boldsymbol{\lambda}_\nu = \mathbf{g}$ of the map Γ, is defined as a solution of the static Coulomb friction problem. Under generic assumptions, the fixed point exists for any data $\mathbf{f} \in \mathbb{R}^n$ and $\mathcal{F} > 0$. If $\mathcal{F}$ is sufficiently small, the fixed point is unique. In order to find non unique solution of the static Coulomb friction problem, path-following technique was proposed, [4, 5].

2 Stability of the fixed point

Let $r > 0$ be a fixed parameter. The variational inequality (2) is equivalent to the equations

$$\boldsymbol{\lambda}_\nu = P_{\boldsymbol{\Lambda}_\nu}(\boldsymbol{\lambda}_\nu - r\mathbb{N}\mathbf{u})\,, \quad \boldsymbol{\lambda}_t = P_{\boldsymbol{\Lambda}_t(\mathcal{F},\mathbf{g})}(\boldsymbol{\lambda}_t - r\mathbb{T}\mathbf{u})\,, \tag{5}$$

see e.g. [3]. Here, $P_{\boldsymbol{\Lambda}_\nu}$ and $P_{\boldsymbol{\Lambda}_t(\mathcal{F},\mathbf{g})}$ are the orthogonal projections of $\mathbb{R}^p$ onto $\boldsymbol{\Lambda}_\nu$ and $\boldsymbol{\Lambda}_t(\mathcal{F}, \mathbf{g})$, see (3). Hence, solving (1)&(2) is equivalent to finding *roots* of nonlinear equations: Define $\mathcal{G} : \mathbb{R}^p \times \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p$ such that

$$\mathbf{g} \in \mathbb{R}^p\,,\ \mathbf{z} \equiv \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda}_\nu \\ \boldsymbol{\lambda}_t \end{pmatrix} \in \mathbb{R}^{n+2p} \longmapsto \mathcal{G}(\mathbf{g}, \mathbf{z}) \equiv \begin{pmatrix} \mathbb{A}\mathbf{u} - \mathbf{f} - \mathbb{N}^\top \boldsymbol{\lambda}_\nu - \mathbb{T}^\top \boldsymbol{\lambda}_t \\ \boldsymbol{\lambda}_\nu - P_{\boldsymbol{\Lambda}_\nu}(\boldsymbol{\lambda}_\nu - r\mathbb{N}\mathbf{u}) \\ \boldsymbol{\lambda}_t - P_{\boldsymbol{\Lambda}_t(\mathcal{F},\mathbf{g})}(\boldsymbol{\lambda}_t - r\mathbb{T}\mathbf{u}) \end{pmatrix} \in \mathbb{R}^{n+2p}\,. \tag{6}$$

Then, given $\mathbf{g} \in \mathbb{R}^p_+$, we seek for the (unique) root $\mathbf{z} = (\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$ of $\mathcal{G}(\mathbf{g}, \mathbf{z}) = 0$.

The map (6) is *piecewise linear*, see [6]. Hence, the differential $\nabla\mathcal{G}$ is defined for almost all $(\mathbf{g}, \mathbf{z}) \in \mathbb{R}^p_+ \times \mathbb{R}^{n+2p}$. Since $\mathcal{G}$ is obviously related to Γ, see (4), the differential $\nabla\Gamma$ exists for almost all $\mathbf{g} \in \mathbb{R}^p_+$.

We consider a fixed point $\Gamma(\mathbf{g}) = \mathbf{g}$, which is related to the root $\mathcal{G}(\mathbf{g}, \mathbf{z}) = 0$, $\mathbf{z} = (\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$, $\mathbf{g} = -\boldsymbol{\lambda}_\nu$. Hence, $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$ is a solution of the static Coulomb friction problem. We say that the fixed point is *regular*, if the differential $\nabla\Gamma(\mathbf{g}) \in \mathbb{R}^{p\times p}$ exists. Let $\sigma(\nabla\Gamma(\mathbf{g}))$ denote the *spectrum* of the matrix $\nabla\Gamma(\mathbf{g})$:

Definition 1 *We say, that a regular fixed point* $\mathbf{g} \in \mathbb{R}^p_+$ *of* Γ *is* **stable** *if it holds:*

$$\lambda \in \sigma(\nabla\Gamma(\mathbf{g})) \Longrightarrow |\lambda| < 1\,.$$

If there exists $\lambda \in \sigma(\nabla\Gamma(\mathbf{g}))$ *such that* $|\lambda| > 1$*, we say that the fixed point* $\mathbf{g} \in \mathbb{R}^p_+$ *is* **unstable***.*

For the relevance of the above definition, see e.g. [7].

3 Example

Assume $n = 4$, $p = 2$. Let

$$\mathbb{A} = \begin{pmatrix} b & -b & 0 & 0 \\ -b & a & -b & 0 \\ 0 & -b & a & -b \\ 0 & 0 & -b & a \end{pmatrix}\,, \quad \mathbf{f} = \begin{pmatrix} \mathrm{f}_{\nu,1} \\ \mathrm{f}_{t,1} \\ \mathrm{f}_{\nu,2} \\ \mathrm{f}_{t,2} \end{pmatrix}\,.$$

The state variable $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(\mathcal{F}, -\boldsymbol{\lambda}_\nu)$ is structured as follows:

$$(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) = (\mathrm{u}_{\nu,1}, \mathrm{u}_{t,1}, \mathrm{u}_{\nu,2}, \mathrm{u}_{t,2}\,,\quad \lambda_{\nu,1}, \lambda_{t,1}\,,\quad \lambda_{\nu,2}, \lambda_{t,2})^T\,.$$

Let $\mathbf{g} = (\mathrm{g}_1, \mathrm{g}_2)^T \in \mathbb{R}^2_+$ denote the slip stress. Consider a fixed point $\Gamma(\mathbf{g}) = \mathbf{g}$ i.e., $\mathcal{G}(\mathbf{g}, \mathbf{z}) = 0$, $\mathbf{z} = (\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$, $\mathbf{g} = -\boldsymbol{\lambda}_\nu$. Assume that the fixed point is regular. Let us compute $\nabla\Gamma(\mathbf{g}) \in \mathbb{R}^{2\times 2}$:

First, we define $\nabla\mathcal{H} = \mathcal{G}_z(\mathbf{g}, \mathbf{z})$, the partial differential of the function $\mathcal{G} = \mathcal{G}(\mathbf{g}, \mathbf{z})$,

$$\nabla\mathcal{H} = \begin{bmatrix} b & -b & 0 & 0 & -1 & 0 & 0 & 0 \\ -b & a & -b & 0 & 0 & -1 & 0 & 0 \\ 0 & -b & a & -b & 0 & 0 & -1 & 0 \\ 0 & 0 & -b & a & 0 & 0 & 0 & -1 \\ r\chi_1^1 & 0 & 0 & 0 & (1-\chi_1^1) & 0 & 0 & 0 \\ 0 & r(-1+\chi_1^2+\chi_1^3) & 0 & 0 & 0 & 2-\chi_1^2-\chi_1^3 & 0 & 0 \\ 0 & 0 & r\chi_2^1 & 0 & 0 & 0 & (1-\chi_2^1) & 0 \\ 0 & 0 & 0 & r(-1+\chi_2^2+\chi_2^3) & 0 & 0 & 0 & 2-\chi_2^2-\chi_2^3 \end{bmatrix}$$

where

$$\chi_1^1 = (\lambda_{\nu,1} - r\mathrm{u}_{t,1} \le 0) \ , \quad \chi_1^2 = (\mathcal{F}\mathrm{g}_1 + \lambda_{t,1} - r\mathrm{u}_{t,1} \ge 0) \ , \quad \chi_1^3 = (-\mathcal{F}\mathrm{g}_1 + \lambda_{t,1} - r\mathrm{u}_{t,1} \le 0) \ ,$$
$$\chi_2^1 = (\lambda_{\nu,2} - r\mathrm{u}_{t,2} \le 0) \ , \quad \chi_2^2 = (\mathcal{F}\mathrm{g}_2 + \lambda_{t,2} - r\mathrm{u}_{t,2} \ge 0) \ , \quad \chi_2^3 = (-\mathcal{F}\mathrm{g}_2 + \lambda_{t,2} - r\mathrm{u}_{t,2} \le 0) \ ,$$

are characteristic functions (i.e., if $r \le s$ then $(r \le s) \equiv 1$, if $r > s$ then $(r \le s) \equiv 0$.). Then, consider the solutions of two linear systems (7). The relevant right-hand sides are defined as minus the partial differential of $\mathcal{G}(\mathbf{g}, \mathbf{z})$ with respect to g_1 and g_2:

$$\nabla\mathcal{H} \begin{bmatrix} \delta\mathrm{u}^1_{\nu,1} \\ \delta\mathrm{u}^1_{t,1} \\ \delta\mathrm{u}^1_{\nu,2} \\ \delta\mathrm{u}^1_{t,2} \\ \delta\lambda^1_{\nu,1} \\ \delta\lambda^1_{t,1} \\ \delta\lambda^1_{\nu,2} \\ \delta\lambda^1_{t,2} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \chi_1^3 - \chi_1^2 \\ 0 \\ 0 \end{bmatrix} , \quad \nabla\mathcal{H} \begin{bmatrix} \delta\mathrm{u}^2_{\nu,1} \\ \delta\mathrm{u}^2_{t,1} \\ \delta\mathrm{u}^2_{\nu,2} \\ \delta\mathrm{u}^2_{t,2} \\ \delta\lambda^2_{\nu,1} \\ \delta\lambda^2_{t,1} \\ \delta\lambda^2_{\nu,2} \\ \delta\lambda^2_{t,2} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \chi_2^3 - \chi_2^2 \end{bmatrix} . \tag{7}$$

Finally,

$$\nabla\Gamma(\mathbf{g}) = - \begin{bmatrix} \delta\lambda^1_{\nu,1} & \delta\lambda^2_{\nu,1} \\ \delta\lambda^1_{\nu,2} & \delta\lambda^2_{\nu,2} \end{bmatrix} \in \mathbb{R}^{2\times 2} . \tag{8}$$

As an example, we set $a = 2$, $b = 1$ and $\mathcal{F} = 4$. In order to find non-unique solutions of the static Coulomb friction problem, we apply the path-following technique, see [4, 5]: Consider a *loading path*. In particular, let $\mathrm{f}_{\nu,1}(\alpha) = 0.4$, $\mathrm{f}_{\nu,2}(\alpha) = 0.2\alpha + 1.8$, $\mathrm{f}_{t,1}(\alpha) = \mathrm{f}_{t,2}(\alpha) = \alpha$. The *continuous* response of the body is shown in Figure 2 (note that the solution components $\mathbf{u}$ and $\boldsymbol{\lambda}_t$ are uniquely determined by $\boldsymbol{\lambda}_\nu$). For example, for the parameter value $\alpha = \mathrm{f}_{t,1}(\alpha) = -1.5$, we encounter **five** solutions of the Coulomb friction problem. Three of them are stable:

1. **No1 :** $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) = (0; 0; 0; 0; \quad -0.4; 1.5; \quad -1.5; 1.5)^T$
   classification: `contact-stick`, `contact-stick`
   $\sigma(\nabla\Gamma(-\boldsymbol{\lambda}_\nu)) = \{0, 0\} \Longrightarrow$ **stable** fixed point
2. **No2 :** $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) = (-1.4; -1.8; -0.7; -1.1; \quad 0; 0; \quad 0; 0)^T$
   classification: `no contact`, `no contact`
   $\sigma(\nabla\Gamma(-\boldsymbol{\lambda}_\nu)) = \{0, 0\} \Longrightarrow$ **stable** fixed point
3. **No3 :** $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) = (-0.7; -1.1; 0; -0.05; \quad 0; 0; \quad -0.35; 1.4)^T$
   classification: `no contact`, `contact-slip`
   $\sigma(\nabla\Gamma(-\boldsymbol{\lambda}_\nu)) = \{0, 2\} \Longrightarrow$ **unstable** fixed point

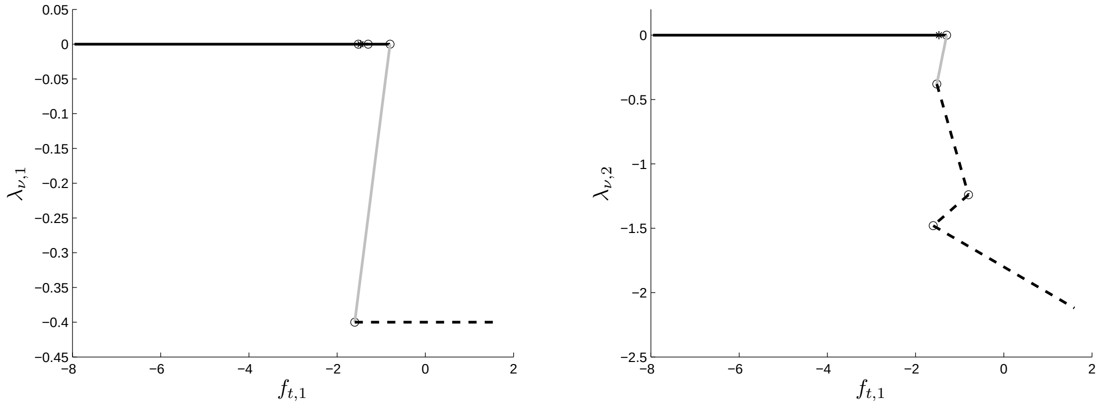


Figure 2: The linear loading path: $\mathrm{f}_{\nu,1}(\alpha) = 0.4$, $\mathrm{f}_{\nu,2}(\alpha) = 0.2\alpha + 1.8$, $\mathrm{f}_{t,1}(\alpha) = \mathrm{f}_{t,2}(\alpha) = \alpha$. The response: solid ... `no contact`, dash ... `contact-stick`, solid-gray ... `contact-slip`.

4. **No4 :** $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) = (-0.7; -1.1; 0; -0.05; \quad 0; 0; \quad -0.35; 1.4)^T$
   classification: `no contact`, `contact-stick`
   $\sigma(\nabla\Gamma(-\boldsymbol{\lambda}_\nu)) = \{0, 0\} \Longrightarrow$ **stable** fixed point
5. **No5 :** $(\mathbf{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) = (0; -0.05; 0; 0; \quad -0.35; 1.4; \quad -1.45; 1.5)^T$,
   classification: `contact-slip`, `contact-stick`
   $\sigma(\nabla\Gamma(-\boldsymbol{\lambda}_\nu)) = \{2, 0\} \Longrightarrow$ **unstable** fixed point.

Observe that the `contact-slip` at any of the contact points implies that the fixed point becomes unstable.

Acknowledgement: The research was supported by the Grant Agency of the Czech Republic (grant No. 201/07/0294) and also by the research project MSM 0021620839 of The Ministry of Education, Youth and Sports, Czech Republic.

References

[1] C. Eck, J. Jarušek: *Existence results for the static contact problems with Coulomb friction.* Math. Models Methods Appl. Sci. 8, 445–468, 1997.

[2] T. Ligurský: *Discrete Contact Problems with Coulomb Friction.* In: J. Šafránková and J. Pavlů (eds.), WDS'08 Proceedings of Contributed Papers: Part I - Mathematics and Computer Sciences, Matfyzpress, Praha, 49–54, 2008.

[3] P. Hild, Y. Renard: *Local uniqueness and continuation of solutions for the discrete Coulomb friction problem in elastostatics.* Quart. Appl. Math. 63, 553–573, 2005.

[4] V. Janovský: *Solving a discrete contact problem via a path-following technique.* In: T.S. Simos (ed.): International Conference on Numerical Analysis and Applied Mathematics 2009, AIP Conference Proceedings 1168, American Institute of Physics, Melville, New York, 497–500, 2009.

[5] V. Janovský, T. Ligurský: *Computing non unique solutions of the Coulomb friction problem.* Submitted.

[6] S. Scholtes: *Introduction to piecewise differentiable equations.* Institut für Statistik und Mathematische Wirtschaftstheorie, Universität Karlsruhe, 1994.

[7] Yu. A. Kuznetsov: *Elements of Applied Bifurcation Theory.* Springer Verlag, New York, 1995.

FETI-DP averaging for the solution of variational inequalities

M. Jarošová[1], *A. Klawonn*[2], *O. Rheinbach*[2]

[1] VŠB-Technical University of Ostrava, Czech Republic
[2] Universität Duisburg-Essen, Germany

1 Introduction

The application of preconditioning to variational inequalities requires some care, as the preconditioning transforms the variables, turning the bound constraints into more general inequality constraints. In our talk we consider two preconditioning strategies, using edge averages, for FETI-DP (dual-primal finite element tearing and interconnecting) methods for the solution of variational inequalities that describe equilibrium of a system of elastic bodies in unilateral contact. We are interested to solve the partially bound constrained quadratic programming problem to find

$$\min_{\mathbf{u}\in\Omega} \phi(\mathbf{u}), \quad \Omega = \{\mathbf{u} \in \mathbb{R}^n : \ \mathbf{u}_{\mathcal{I}} \geq \ell_{\mathcal{I}}\}, \quad \mathcal{I} = \{1, \ldots, k\}, \tag{1}$$

where $\phi(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathsf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f}$, ℓ and $\mathbf{f}$ are given column n-vectors, $1 \leq k \ll n$, and K is an $n \times n$ symmetric positive definite matrix.

2 Dual-primal FETI method

We consider FETI-DP method, originally introduced by Farhat, Lesoinne, Le Tallec, Pierson, and Rixen [6]. In FETI-DP methods the original domain Ω is decomposed into several nonoverlapping subdomains Ω_i. The continuity of the primal solution at some nodes called vertices (or corners) is implemented directly into the formulation of the primal problem so that one degree of freedom is considered at each vertex. The continuity of the primal variables across the rest of the subdomains interface is enforced by the Lagrange multipliers.

Using the theory of duality we can derivate the dual problem of the problem (1) in the form

$$\min_{\lambda\geq 0} \Theta(\lambda), \quad \Theta(\lambda) = \frac{1}{2}\lambda^T\mathsf{F}\lambda - \lambda^T\mathbf{d}, \tag{2}$$

where

$$\begin{aligned}
\mathsf{F} &= -\mathsf{B}_B\mathsf{K}_{BB}^{-1}\mathsf{B}_B^T - (-\mathsf{B}_B\mathsf{K}_{BB}^{-1}\mathsf{K}_{\Pi B}^T\mathsf{L})^T\tilde{\mathsf{S}}_{\Pi\Pi}^{-1}(-\mathsf{B}_B\mathsf{K}_{BB}^{-1}\mathsf{K}_{\Pi B}^T\mathsf{L}),\\
\mathbf{d} &= -\mathsf{B}_B\mathsf{K}_{BB}^{-1}\mathbf{f}_B - (-\mathsf{B}_B\mathsf{K}_{BB}^{-1}\mathsf{K}_{\Pi B})^T\tilde{\mathsf{S}}_{\Pi\Pi}^{-1}\mathsf{L}^T(\mathbf{f}_\Pi - \mathsf{K}_{\Pi B}\mathsf{K}_{BB}^{-1}\mathbf{f}_B) + \mathbf{c},\\
\tilde{\mathsf{S}}_{\Pi\Pi} &= \mathsf{L}^T(\mathsf{K}_{\Pi\Pi} - \mathsf{K}_{\Pi B}\mathsf{K}_{BB}^{-1}\mathsf{K}_{\Pi B}^T)\mathsf{L}.
\end{aligned}$$

The continuity at the dual displacement variables and inequality constraints are enforced by matrix B. The subscript Π and B denote primal variables and all other variables, respectively. Minimizing $\Theta(\lambda)$ over $\lambda \geq 0$ is equivalent to solving problem (1).

3 Projector preconditioning for FETI-DP

The first preconditioning technique assumed here, preconditioning by a conjugate projector, was proposed for linear systems, e.g., by Dostál [4], and extended for bound constrained problems by Domorádová and Dostál [5].

Let $\mathsf{F} \in \mathbb{R}^{m\times m}$ be a symmetric positive definite matrix. A projector P is an F-*conjugate projector* or briefly a *conjugate projector* if ImP is F-conjugate to Ker$\mathsf{P}$, or equivalently $\mathsf{P}^T\mathsf{F}(\mathsf{I}-\mathsf{P}) = \mathsf{P}^T\mathsf{F}-\mathsf{P}^T\mathsf{F}\mathsf{P} = \mathsf{O}$. If $\mathcal{U}$ is the subspace spanned by the columns of a full column rank matrix $\mathsf{U} \in \mathbb{R}^{m\times p}$, then

$$\mathsf{P} = \mathsf{U}(\mathsf{U}^T\mathsf{F}\mathsf{U})^{-1}\mathsf{U}^T\mathsf{F} \tag{3}$$

is a conjugate projector onto $\mathcal{U}$. Let $\Omega_0 = \{\lambda \in \mathbb{R}^m : \lambda \geq \mathbf{o}\}$. We use the conjugate projectors P and $\mathsf{Q} = \mathsf{I}-\mathsf{P}$ to decompose our dual minimization problem (2) into the minimization on $\mathcal{U}$ and the minimization on $\mathcal{V}\cap\Omega_0$, $\mathcal{V} = \text{Im}\mathsf{Q}$, we can write

$$\min_{\lambda\geq 0}\Theta(\lambda) \;\; = \;\; \Theta(\lambda^0) + \min_{\substack{\mu\in\mathsf{A}\mathcal{V}\\ \mu\geq 0}} \frac{1}{2}\mu^T\mathsf{Q}^T\mathsf{F}\mathsf{Q}\mu + \mu^T\mathbf{g}^0,$$

where $\lambda^0 = \mathsf{P}\mathsf{F}^{-1}\mathbf{d}$ and $\mathbf{g}^0 = -\mathsf{Q}^T\mathbf{d}$. The solution $\widehat{\lambda}$ of the dual problem (2) can then be expressed by $\widehat{\lambda} = \lambda^0 + \mathsf{Q}\widehat{\mu}$.

The matrix U is defined by the elements of the aggregation bases, where the Lagrange multipliers corresponding to the variables of the coinciding edges are aggregated.

4 Transformation of basis

The second closely related preconditioning technique is an explicit transformation of basis introducing edge averages as new primal variables. This turns out to be an efficient method to replace or enhance the coarse problem of the dual-primal FETI method, especially in three space dimensions. We introduce certain edge or face averages or edge first order moments, either additionally or instead of the assembly in a selected number of primal variables; see, e.g., Farhat, Lesoinne, Pierson [7], Klawonn and Widlund [8], Klawonn, Widlund, and Dryja [9], and Klawonn and Rheinbach [10]. This technique was applied to the problems described by variational inequalities by Jarošová, Klawonn and Rheinbach [2].

Let $\hat{\mathbf{u}}_E$ denote the edge unknowns in the new basis, then $\mathbf{u}_E = \mathsf{T}_E\hat{\mathbf{u}}_E$, where T_E is the transformation matrix with the mutually orthogonal columns representing the new basis. This matrix performs the desired change of the basis from the new basis to the original nodal basis. T_E is obtained from $\bar{\mathsf{T}}_E$ using Gram-Schmidt orthogonalization. Ordering averages last, $\bar{\mathsf{T}}_E$ can be written as

$$\bar{\mathsf{T}}_E = \begin{bmatrix} 1 & \dots & 0 & 1 \\ & \ddots & & \vdots \\ 0 & & 1 & 1 \\ -1 & \dots & -1 & 1 \end{bmatrix}. \tag{4}$$

Such transformation matrix can be constructed separately for each edge. The resulting transformation matrix $\mathsf{T}_E^{(i)}$, which operates on all relevant edges of Ω_i, is a direct sum of the relevant transformation matrices T_E associated with the edges of subdomain Ω_i. $\mathsf{T}_E^{(i)}$ is a block diagonal, where each block represents the transformation of variables of one edge.

Since we assume in this case also the vertex constraints, the transformation matrix for all variables of one subdomain Ω_i is of the form

$$\mathsf{T}^{(i)} = \left[\begin{array}{ccc} \mathsf{I}_I^{(i)} & \mathsf{O} & \mathsf{O} \\ \mathsf{O} & \mathsf{I}_V^{(i)} & \mathsf{O} \\ \mathsf{O} & \mathsf{O} & \mathsf{T}_E^{(i)} \end{array} \right], \tag{5}$$

where the subscripts I, V, E denote interior, vertex and edge nodes, respectively. $\mathsf{I}_I^{(i)}$ and $\mathsf{I}_V^{(i)}$ denote identity matrices. Using the same decomposition as in (5), the matrix of the transformed system has the form

$$\mathsf{T}^{(i)T}\mathsf{K}^{(i)}\mathsf{T}^{(i)} \quad = \quad \left[\begin{array}{cc|c} \mathsf{K}_{II}^{(i)} & \mathsf{K}_{VI}^{(i)T} & \mathsf{K}_{EI}^{(i)T}\mathsf{T}_E^{(i)} \\ \mathsf{K}_{VI}^{(i)} & \mathsf{K}_{VV}^{(i)} & \mathsf{K}_{VE}^{(i)T}\mathsf{T}_E^{(i)} \\ \hline \mathsf{T}_E^{(i)T}\mathsf{K}_{EI}^{(i)} & \mathsf{T}_E^{(i)T}\mathsf{K}_{VE}^{(i)} & \mathsf{T}_E^{(i)T}\mathsf{K}_{EE}^{(i)}\mathsf{T}_E^{(i)} \end{array} \right]. \tag{6}$$

The edge variables are now split into two part: the dual variables and averages, so that $\hat{\mathbf{u}}_E = [\hat{\mathbf{u}}_\Delta, \hat{\mathbf{u}}_A]$. Using this notation, we can write

$$\mathsf{T}^{(i)T}\mathsf{K}^{(i)}\mathsf{T}^{(i)} = \left[\begin{array}{cc|cc} \mathsf{K}_{II}^{(i)} & \mathsf{K}_{VI}^{(i)T} & \bar{\mathsf{K}}_{\Delta I}^{(i)T} & \bar{\mathsf{K}}_{AI}^{(i)T} \\ \mathsf{K}_{VI}^{(i)} & \mathsf{K}_{VV}^{(i)} & \bar{\mathsf{K}}_{V\Delta}^{(i)T} & \bar{\mathsf{K}}_{VA}^{(i)T} \\ \hline \bar{\mathsf{K}}_{\Delta I}^{(i)} & \bar{\mathsf{K}}_{V\Delta}^{(i)} & \bar{\mathsf{K}}_{\Delta\Delta}^{(i)} & \bar{\mathsf{K}}_{\Delta A}^{(i)T} \\ \bar{\mathsf{K}}_{AI}^{(i)} & \bar{\mathsf{K}}_{VA}^{(i)} & \bar{\mathsf{K}}_{\Delta A}^{(i)} & \bar{\mathsf{K}}_{AA}^{(i)} \end{array} \right].$$

Ordering the primal variables last, we obtain

$$\mathsf{T}^{(i)T}\mathsf{K}^{(i)}\mathsf{T}^{(i)} = \left[\begin{array}{ccc} \mathsf{K}_{II}^{(i)} & \bar{\mathsf{K}}_{\Delta I}^{(i)T} & \hat{\mathsf{K}}_{\Pi I}^{(i)T} \\ \bar{\mathsf{K}}_{\Delta I}^{(i)} & \bar{\mathsf{K}}_{\Delta\Delta}^{(i)} & \hat{\mathsf{K}}_{\Pi\Delta}^{(i)T} \\ \hat{\mathsf{K}}_{\Pi I}^{(i)} & \hat{\mathsf{K}}_{\Pi\Delta}^{(i)} & \hat{\mathsf{K}}_{\Pi\Pi}^{(i)} \end{array} \right], \quad \text{where } \hat{\mathsf{K}}_{\Pi\Pi}^{(i)} = \left[\begin{array}{cc} \mathsf{K}_{VV}^{(i)} & \bar{\mathsf{K}}_{VA}^{(i)T} \\ \bar{\mathsf{K}}_{VA}^{(i)} & \bar{\mathsf{K}}_{AA}^{(i)} \end{array} \right]. \tag{7}$$

Assembling the primal contributions of each transformed $\mathsf{K}^{(i)}$ to $\tilde{\mathsf{K}}_{\Pi\Pi}$, we obtain the transformed stiffness matrix $\tilde{\mathsf{K}}$. Now we can rewrite problem (2) as

$$\min_{\hat{\mathbf{u}}\in\Omega} \frac{1}{2}\hat{\mathbf{u}}^T\mathsf{T}^T\mathsf{K}\mathsf{T}\hat{\mathbf{u}} - \mathbf{f}^T\mathsf{T}\hat{\mathbf{u}} = \min_{\hat{\mathbf{u}}\in\Omega} \frac{1}{2}\hat{\mathbf{u}}^T\tilde{\mathsf{K}}\hat{\mathbf{u}} - \hat{\mathbf{u}}^T\hat{\mathbf{f}}, \tag{8}$$

where $\hat{\mathbf{u}}$, $\hat{\mathbf{f}}$ denote vector of unknowns and load vector in the new basis, respectively. Using the process described in Section 2 we obtain the solution to this problem. To obtain the primal solution we need to use the backward transformation $\mathbf{u}_E = \mathsf{T}_E\hat{\mathbf{u}}_E$.

The theoretical results show that both described methods iterate in the same subspace and thus have the same rate of convergence [2]. Thus the explicit construction of the dual matrix in the projector can be replaced by the transformation of basis which works locally and can easily be parallelized.

5 Numerical experiments

The theoretical results are confirmed by the results of numerical experiments. For example, in Table 1 and Table 2, we illustrate the improvement on the solution of a model scalar problem, displacement of membrane over a boundary obstacle. For the solution we use MPRGP algorithm with the rate of convergence in terms of the spectral condition number of the Hessian matrix [3, 1].

Acknowledgement: This work has been supported by the grants GA CR 201/07/0294, the Ministry of Education of the Czech Republic No. MSM6198910027 and GA CR 103/09/H078.

	FETI-DP		
H/h	*no preconditioned*	*proj. preconditioning*	*trans. of basis*
4	32	22	22
8	51	30	31
16	82	41	42
32	118	58	61

Table 1: Iteration counts of dual problem for 4 × 4 subdomains and changing H/h.

	FETI-DP		
num. of sub.	*no preconditioned*	*proj. preconditioning*	*trans. of basis*
4 × 4	51	30	31
8 × 8	79	34	35
12 × 12	91	46	45
16 × 16	101	52	51
20 × 20	118	58	57

Table 2: Iteration counts of dual problem for changing number of subdomains and H/h = 8.

References

[1] Z. Dostál: *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities.* 1st edition, SOIA 23, Springer US, New York 2009.

[2] M. Jarošová, A. Klawonn, O. Rheinbach: *Projector preconditioning and transformation of basis in FETI-DP algorithms for contact problems.* Submitted.

[3] Z. Dostál, J. Schöberl: *Minimizing quadratic functions subject to ound constraints with the rate of convergence and finite termination.* Comput. Optim. Appl., 30(1), 23–43, 2005.

[4] Z. Dostál: *Conjugate gradient method with preconditioning by projector.* Intern. J. Computer Math. 23, 315–323, 1988.

[5] M. Domorádová, Z. Dostál: *Projector preconditioning for partially bound-constrained quadratic optimization.* Numerical Linear Algebra with Applications, 14, 791–806, 2007.

[6] Ch. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, D. Rixen: *FETI-DP: A dual-primal unified FETI method - part I: A faster alternative to the two-level FETI method.* International Journal for Numerical Methods in Engineering, 50, 1523–1544, 2001.

[7] Ch. Farhat, M. Lesoinne, K. Pierson: *A scalable dual-primal, domain decomposition method.* Numer. Lin. Alg. Appl., 7, 687–714, 2000.

[8] A. Klawonn, O.B. Widlund: *Dual-primal FETI methods for linear elasticity.* Comm. Pure Appl. Math., 59, 1523–1572, 2006.

[9] A. Klawonn, O.B. Widlund,. M. Dryja: *Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients.* SIAM J.Numer. Anal., 40, 159–179, 2002.

[10] A. Klawonn, O. Rheinbach: *Robust FETI-DP methods for heterogeneous three dimensional elasticity problems.* Comput. Methods Appl. Mech. Engrg., 196(8), 1400–1414, 2007.

Efficient optimization of hybrid neural networks

P. Kordík

Department of Computer Science, Faculty of Information Technology
Czech Technical University in Prague

1 Introduction

Our research focuses on optimization of supervised feed-forward neural networks with hybrid neurons. The search space of possible network topologies is huge and increases even more with growing number of inputs. There are two overlapping problems in neural network optimization, one from the area of discrete optimization, second from continuous optimization domain. The first problem is to find proper topology of the network and the second is to optimize its weights (parameters of transfer functions in neurons).

In traditional (and most popular) algorithms the topology of a neural network is subject of trial and error strategy. One has to estimate number, type and interconnections of neurons in advance and then redefine it, when the optimization of weights fails.

More actual algorithms [1] build network neuron by neuron from a minimal form until a sufficient accuracy is obtained.

These algorithms typically work with a uniform type of neurons (e.g. sigmoid). They can be modified to incorporate more different types of neurons within one network, but it is neither straightforward nor efficient.

In our GAME algorithm, we construct a neural network from neurons of various types. Such hybrid neural network can easier adapt to problems (data sets) of various complexity. For simple problems, a linear transfer neurons can be selected and for complex problems, a multi-layered network with Gaussian and Sine neurons can better reflect relationships in data.

In this contribution, we show how to optimize such networks efficiently.

Topology and Weights Evolving Artificial Neural Networks [4] use evolutionary algorithms to optimize topology together with weights of a network. Disadvantages of such approach are course of dimensionality (in chromosome size) and problematic preserving of bigger networks with not-yet-evolved weights. Our approach is to optimize the topology and weights independently.

2 Topology optimization of hybrid neural networks

We start from a minimal form and add layers of neurons until the accuracy on validation data is increasing. In each layer, we run special niching evolutionary algorithm [3] preserving diversity in the population of neurons. The evolutionary algorithm optimizes inputs of neurons, the type and the structure of their transfer functions. Weights of transfer functions are optimized independently (see next section).

Figure 1 demonstrates the GAME optimization procedure. A hybrid neural network is being evolved and corresponding compatible parts of neurons' chromosomes are crossed over and mutated within the niching evolutionary algorithm running in each layer. For detailed description of the algorithm, see [3].

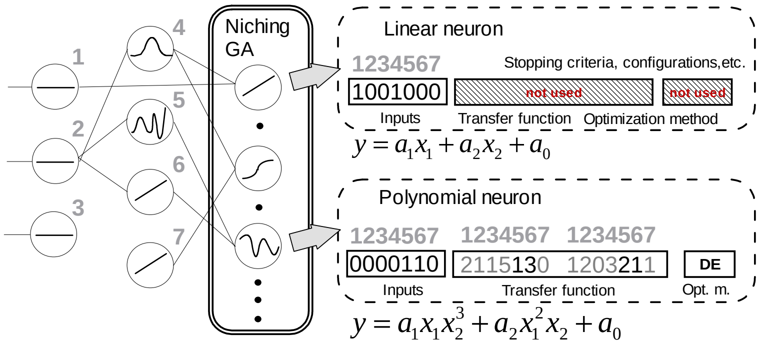


Figure 1: An example of Linear and Polynomial neurons and their chromosomes. The topology of this hybrid neural network is optimized layer by layer by means of a niching genetic algorithm.

3 Weights optimization

In our hybrid neural networks, neurons differ in complexity. The simplest neurons have linear transfer function. Then we have neurons with elementary non-linear functions such as Sigmoid, Gaussian, Exponential and slightly more complex Polynomial, Rational or Sine. The most complex neurons can contain embedded neural network (Cascade correlation network, Multilayered Perceptron, etc.).

The most complex neurons use their own learning algorithms to optimize their weights and these will not be discussed in this section.

Weights of simplest neurons (with a linear transfer function) can be estimated in a single step. We use Least Mean Squared [2] method - weights are computed directly from training data using a matrix inverse. This method works reasonably well except rare cases when a matrix is singular and cannot be inverted.

This simple LMS method can be used also for neurons with polynomial transfer function, but the results are not so good as for linear neurons (the pre-computed matrix is often singular). The reason is that the error surface is more complex and an iterative method is needed.

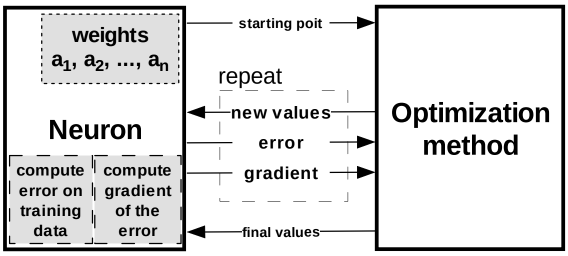


Figure 2: An iterative optimization of single neuron weights. The analytic gradient can be used by the optimization method to fasten convergence.

The weights (coefficients) of non-linear neurons we optimize iteratively. We have implemented several optimization algorithms. Some of them are gradient based, some nature inspired metaheuristics. In this contribution we focus on gradient based techniques (Quasi-Newton method, Conjugate Gradient method, etc.). These algorithms modify weights in order to minimize the training error of a single neuron. For all neurons, we have derived and implemented an analytic gradient of their training error surface [3]. Optimization methods can use this information in order to search the optima more efficiently (see Figure 2).

We have experimented with many different neurons and data sets and we have found some interesting conclusions.

For certain type of neurons, a good starting point is of crucial importance.

3.1 Starting point

The starting point is an initial configuration of weights for the iterative method to start with. For example in case of Sigmoid neuron, it is sufficient to supply a random starting point around zero (weights within $(-0.3, 0.3)$ for normalized data). Large initial weighs prevent optimization method to converge, because the sigmoid function is saturated (it is sensitive just in almost linear part around zero).

$$y_j = e^{-\frac{\sum_{i=1}^{n}\left(x_{ij}-a_i\right)^2}{2*(a_{n+i})^2}} \tag{1}$$

In case or Gaussian neuron (Equation 1), weights a_{n+i} should not be too small. Then, the error surface is flat with a single deep pit and the optimization method is unable to locate this pit with a global minima.

For Gaussian neuron, we use the maximum likelihood estimate of weights (means and standard deviations computed on training data) as the starting point.

3.2 Error surface Inspection

We have implemented an inspection tool allowing us to monitor the the optimization process. The training error is dependent variable and weights are independent variables. To be able to see, how the training error surface looks like, we need to use projections of the multidimensional space. We use scatterplot matrix of training error plots. For each plot, two weights are varied in the interval $(-15, 15)$, all other weights are fixed (values in actual iteration) and the training error is computed in each point of the plot. The darker the background is, the lower the training error. We visualize also the iteration history in each plot and observe, how the process converges.

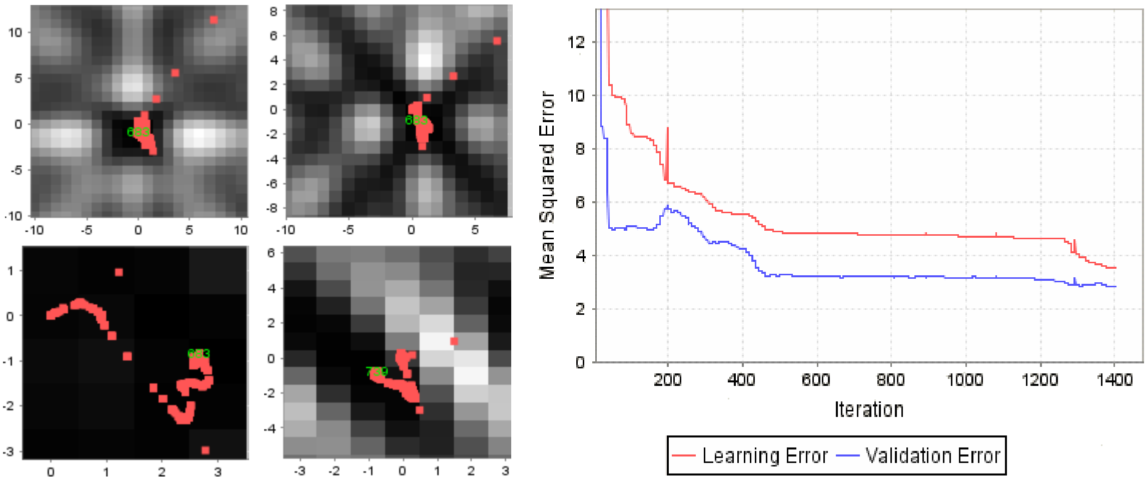


Figure 3: The Sine neuron optimized by Quasi-Newton algorithm with numerical estimates of the analytic gradient. Visualization of the training error surface from different perspectives (left) and the convergence history (right).

This visualization is particulary useful for getting information on training error surface complexity in each dimension (how a change of individual weight influences the error).

In the Figure 3, you can observe the training log of the single Sine neuron on the Bosthouse data set [3]. The Quasi-Newton method (QN) needed almost 1400 iteration to find optimal values of 24 weights (Bosthouse has 12 features, the output of the neuron is $y = \sum_{i=1}^{12} \sin(a_i * x_i + a_{12+i}))$. The algorithm managed to escape from a local minima (iteration 170) and converged to a global optima.

3.3 Analytic gradient

Note that supplying the optimization method the the analytic gradient is not always the best option. In the optimization of the Sine neuron (Figure 3), the gradient was not supplied and QN method estimated it numerically. Therefore much more iterations was needed. On the other hand, when we supply the analytic gradient, QN relies on it much more and get stuck in a local minima after 45 iterations.

In deeper layers of the network, error surface of units becomes increasingly complex and the analytical gradient is important, because it saves the computational time exponentially.

4 Conclusion

We have described the optimization technique for hybrid multilayered neural networks. Our algorithm is able to train networks efficiently, using advanced optimization methods. In this contribution we also present several practical observations, that should help us to improve the algorithm in future.

Acknowledgement: This research is partially supported by the research program "Transdisciplinary Research in the Area of Biomedical Engineering II"(MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] S.E. Fahlman, C. Lebiere: *The cascade-correlation learning architecture.* Technical Report CMU-CS-90-100, Carnegie Mellon University Pittsburgh, USA, 1991.

[2] A.G. Ivakhnenko, J.-A. Müller: *Self-organization of nets of active neurons.* Syst. Anal. Model. Simul., 20(1-2), 93–106, 1995.

[3] P. Kordík: *Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution.* PhD thesis, Czech Technical University in Prague, Praha, 2006.

[4] K.O. Stanley, R. Miikkulainen: *Evolving neural networks through augmenting topologies.* Evolutionary Computation, 10, 99–127, 2002.

Parallel solution of engineering problems in mechanics using MatSol

T. Kozubek

VŠB-Technical University of Ostrava, Department of Applied Mathematics, Czech Republic

Introduction

We first briefly review the TFETI based domain decomposition methodology adapted to the solution of 2D and 3D multibody contact problems of elasticity [2]. Recall that TFETI imposes the prescribed displacements by the Lagrange multipliers, so that all the subdomains are floating and their kernels are a priori known. Then we show that the natural coarse grid of the rigid body motions introduced by Farhat, Mandel, and Roux defines a projector to the subspace of Lagrange multipliers with the solution. Moreover, the preconditioning by the projector reduces the condition number of the dual Schur complement so that it is independent on the discretization parameter h and accelerates also the non-linear steps.

Then we present our in a sense optimal algorithms [1] for the solution of resulting quadratic programming problems. The unique feature of these algorithms is their capability to solve the class of quadratic programming problems with spectrum in a given positive interval in O(1) iterations. The theory yields the error bounds that are independent on conditioning of constraints and the results are valid even for linearly dependent equality constraints.

Finally, we put together the above results to develop scalable algorithms for the solution of both coercive and semi-coercive variational inequalities (see [3] and [4]). Rather surprisingly, the results on the scalability of the TFETI based solution of contact problems are qualitatively the same as the classical results of FETI for linear elliptic problems. The resulting algorithms were implemented in `MatSol` library [5] developed in Matlab environment and tested on solution of 2D and 3D contact problems. For these computations we used an HP Blade system, model BLc7000 with one master node and eight computational nodes, each with two dual core CPUs AMD Opteron 2210 HE. As parallel programming environment we use Matlab Distributed Computing Engine and Matlab Parallel Computing Toolbox. These products allow users to offload work from one Matlab session (the client) to other Matlab sessions (workers), see Figure 1. One can use multiple workers to take advantage of parallel processing or only one worker to take advantage of another computer's speed or to keep the original Matlab client session free. Parallel Computing Toolbox allows to run as many as four Matlab workers on the local machine in addition to the original Matlab client session. On the other hand Matlab Distributed Computing Engine allows us to run as many Matlab workers on a remote cluster of computers as our licensing allows. This parallel environment supports both parallel and distributed programming. An example of parallel computation scenario corresponding to the TFETI based solution of contact problems in mechanics is depicted in Figure 2.

At the end, we give results of numerical experiments with parallel solution of contact problems discretized by up to more than 10 million of nodal variables to demonstrate that the scalability can be observed in computational practice. The power of the results is demonstrated also by the solution of difficult real world problems as analysis of the roller bearing of wind generator (see Figures 3 and 4).

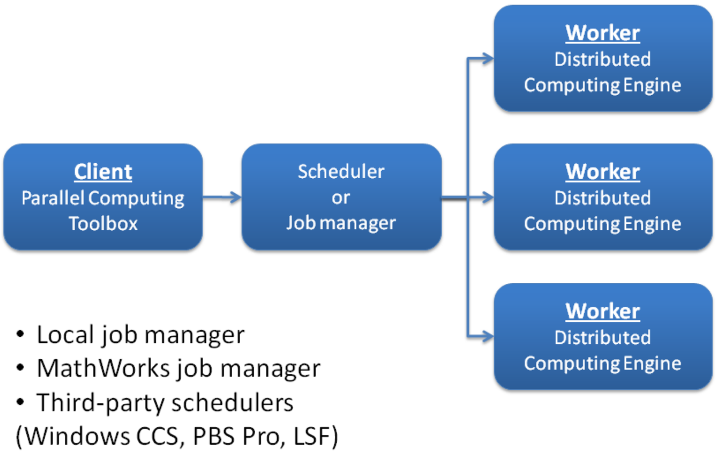


Figure 1: Parallel programming using Matlab Distributed Computing Engine.

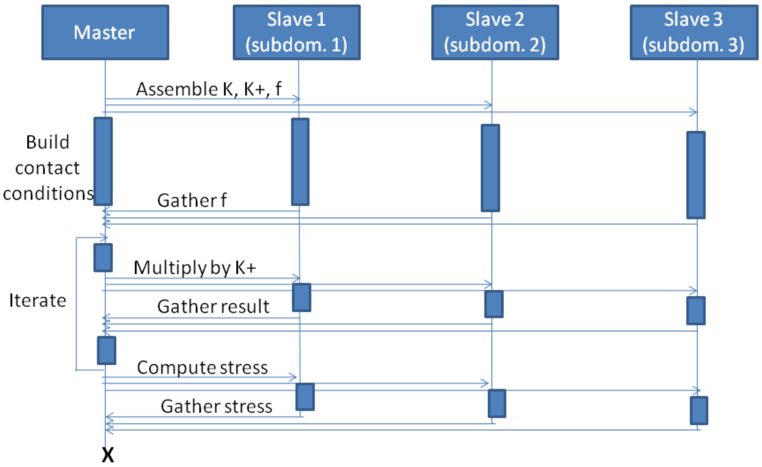


Figure 2: Parallel computation scenario.

Acknowledgement: This research has been supported by the grants GAČR 101/08/0574 and the Ministry of Education of the Czech Republic No. MSM6198910027.

References

[1] Z. Dostál: *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*. 1st edition, Springer US, NY 2009, SOIA 23.

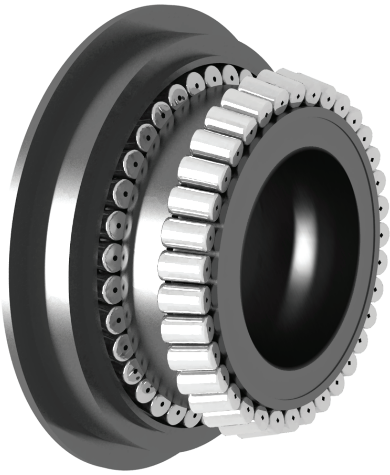


Figure 3: Roller bearing of wind generator.

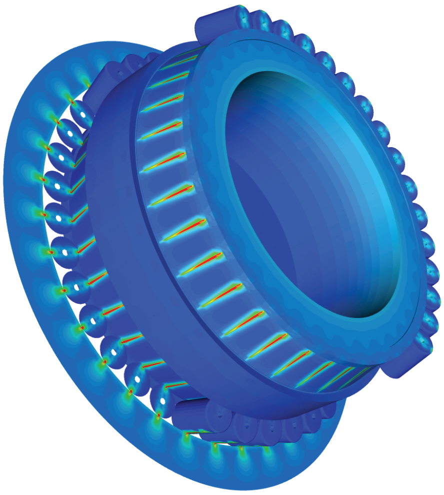


Figure 4: Von Mises stress distribution.

[2] Z. Dostál, D. Horák, R. Kučera: *Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE*, Comm. Num. Meth. Eng. 22, 1155–1162, 2006.

[3] Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, A. Markopoulos: *Scalable TFETI algorithm for the solution of multibody contact problems of elasticity*. Accepted in International Journal for Numerical Methods in Engineering, 2009.

[4] Z. Dostál, T. Kozubek, P. Horyl, T. Brzobohatý, A. Markopoulos: *Scalable TFETI algorithm for two dimensional multibody contact problems with friction*. Submitted to Journal of Computational and Applied Mathematics, 2009.

[5] T. Kozubek, A. Markopoulos, T. Brzobohatý, R. Kučera, V. Vondrák, Z. Dostál: *MatSol - MATLAB efficient solvers for problems in engineering*. `http://www.am.vsb.cz/matsol`, 2009.

Application of BOSS preconditioner to the fourth order problems

J. Kruis, P. Mayer

Department of Mechanics, Faculty of Civil Engineering, Czech Technical University, Prague
Department of Mathematics, Faculty of Civil Engineering, Czech Technical University, Prague

1 Introduction

Systems of linear algebraic equations generated by a numerical method applied to the fourth order problems have usually worse properties than systems obtained from the second order problems. In the case of many degrees of freedom, iterative methods have to be used because the direct methods require unattainable computer memory and unacceptable computer time. The large condition number of the matrix system leads to many iterations needed for prescribed norm of the residual.

The high number of iterations can be reduced by a suitable preconditioner. This contribution deals with the preconditioner based on the BOSS algorithm developed by Brezina in his PhD thesis [1]. The method is based on aggregation of unknowns which is nonoverlapping at the beginning and later the aggregation is changed to overlapping by smoothing. The size of overlap has strong influence on the convergence as well as on the memory requirements.

2 BOSS method

Let system of linear algebraic equations be in the form

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{1}$$

and let the number of unknowns be n. The notation global vector is used for vectors from R^n while local vectors is notation for vectors from R^{n_i}, where n_i denotes the number of unknowns in the i-th aggregate. The global vectors contain all unknowns and they are connected with the whole problem. The local vectors are connected with particular smoothed aggregates and contain only unknowns collected in one aggregate. The localization matrices are defined by the relationship

$$\mathbf{g} = \mathbf{N}_i \mathbf{l}_i \ , \tag{2}$$

where $\mathbf{g}$ denotes the global vector and $\mathbf{l}_i$ denotes the local vector of the i-th aggregate. The reverse relation, i.e. map between global and local vectors, has the form

$$\mathbf{l}_i = \mathbf{N}_i^T \mathbf{g} \ . \tag{3}$$

Local matrices of aggregates are defined by the relationship

$$\tilde{\mathbf{A}}_i = \mathbf{N}_i^T \mathbf{A} \mathbf{N}_i \ . \tag{4}$$

Local correction operator has the form

$$\mathbf{R}_i = \mathbf{N}_i (\tilde{\mathbf{A}}_i)^{-1} \mathbf{N}_i^T \ . \tag{5}$$

The inverse matrix $(\tilde{\mathbf{A}}_i)^{-1}$ is not computed and assembled. The matrices $\tilde{\mathbf{A}}_i$ are factorized into $\mathbf{LL}^T$, $\mathbf{LDL}^T$ or $\mathbf{LU}$ form.

Coarse level correction operator has the form

$$\mathbf{R}_0 = \mathbf{P}(\tilde{\mathbf{A}}_0)^{-1}\mathbf{P}^T \ , \tag{6}$$

where $\mathbf{P}$ is the matrix of smoothed prolongator and the matrix $\tilde{\mathbf{A}}_0$ has the form

$$\tilde{\mathbf{A}}_0 = \mathbf{P}^T\mathbf{A}\mathbf{P} \ . \tag{7}$$

More details about assembling of the matrix $\mathbf{P}$ can be found in [1] or [2]. Similarly to the matrices $\tilde{\mathbf{A}}_i$, the matrix $(\tilde{\mathbf{A}}_0)^{-1}$ is not assembled because $\mathbf{LL}^T$, $\mathbf{LDL}^T$ or $\mathbf{LU}$ factorization is used for the matrix $\tilde{\mathbf{A}}_0$. The BOSS method solves the system (1) iteratively and the flowchart of the method is summarized in Table 1.

```
1. initial vector $\mathbf{z}_0 = \mathbf{x}^{(k)}$

2. local correction on aggregates
   for $i = 1, \ldots, m$: $\mathbf{z}_i = \mathbf{z}_{i-1} + \mathbf{R}_i(\mathbf{b} - \mathbf{A}\mathbf{z}_{i-1})$

3. coarse level correction
   $\mathbf{v}_m = \mathbf{z}_m + \mathbf{R}_0(\mathbf{b} - \mathbf{A}\mathbf{z}_m)$

4. local correction on aggregates - the reverse order
   for $i = m-1, \ldots, 0$: $\mathbf{v}_i = \mathbf{v}_{i+1} + \mathbf{R}_{i+1}(\mathbf{b} - \mathbf{A}\mathbf{v}_{i+1})$

5. vector of results $\mathbf{x}^{(k+1)} = \mathbf{v}_0$
```

Table 1: BOSS method.

The BOSS method is used as a preconditioner of the conjugate gradient method. It means, the step

$$\mathbf{h}^{(k+1)} = \mathbf{C}^{-1}\mathbf{r}^{(k+1)} \ , \tag{8}$$

where the residual $\mathbf{r}^{(k+1)}$ is recalculated into new vector $\mathbf{h}^{(k+1)}$ is solved by the BOSS method.

3 Numerical example

The behaviour of the preconditioner based on the BOSS method is shown on example of plate analysis. Plate deflection is described by the fourth order partial differential equation. Square domain is covered by three different meshes of finite elements which contain 100x100, 200x200 and 300x300 elements and the number of unknowns is 30 300, 120 600 and 270 900, respectively. Two degrees of smoothing are used, namely 2 and 3.

Tables 2 and 3 contain data about aggregation in the case of smoothing degree equal to two respectively to three. The notation used in Tables 2 and 3 is the following: NA is the number of aggregates, MIN N is the minimum number of unknowns in aggregate, MAX N is the maximum number of unknowns in aggregate, MIN NEGM is the minimum number of matrix entries stored in the skyline storage scheme in aggregate, MAX NEGM is the maximum number of matrix entries stored in the skyline storage scheme in aggregate and TOT NEGM is the sum of numbers of matrix entries stored in the skyline storage scheme in aggregates. Table 4 contains the numbers of iterations for degree of smoothing 2 and 3. The number of iterations in the nonpreconditioned conjugate gradient method needed for reduction of residual to the prescribed norm was about 20 200, 85 000 and 150 000 respectively.

NA	MIN N	MAX N	MIN NEGM	MAX NEGM	TOT NEGM
100x100					
10	3,753	5,022	291,246	457,674	3,767,298
40	1,101	1,899	48,273	128,697	3,579,471
70	636	1,266	27,246	69,906	3,659,471
100	459	1,077	15,750	51,603	3,834,897
200x200					
10	13,722	15,060	1,600,305	2,895,477	25,098,081
70	2,181	3,414	158,715	309,684	15,190,809
130	1,191	2,322	66,021	185,634	14,547,495
200	840	1,617	37,851	100,644	14,742,756
300x300					
10	28,131	32,619	5,498,691	9,769,518	70,659,540
100	3,231	4,953	255,795	555,552	36,662,874
300	1,191	2,262	62,376	177,045	33,739,974

Table 2: Plate problem; degree 2.

NA	MIN N	MAX N	MIN NEGM	MAX NEGM	TOT NEGM
100x100					
10	5,535	9,816	555,417	1,273,314	9,636,684
40	2,100	5,571	135,780	585,780	15,615,279
70	1,338	4,125	81,921	414,078	20,233,194
100	1,053	3,723	54,981	338,454	24,394,188
200x200					
10	16,665	23,271	2,606,067	2,895,477	43,294,776
70	3,477	8,133	328,206	309,684	51,998,865
130	2,082	7,518	153,150	185,634	63,772,173
200	1,623	4,935	103,218	100,644	76,664,883
300x300					
10	32,289	44,778	7,257,255	14,741,235	106,364,094
100	4,770	11,508	485,136	1,696,440	110,596,956
300	2,109	6,228	150,369	719,646	152,809,878

Table 3: Plate problem; degree 3.

NA	NI deg 2	NI deg 3
100x100		
10	133	33
40	171	42
70	157	42
100	166	34
200x200		
10	293	69
70	567	111
130	648	122
200	692	134
300x300		
10	530	123
100	1153	216
300	1422	267

Table 4: Plate problem; the number of iterations.

4 Conclusion

The preconditioner based on the BOSS method is very efficient and it reduces significantly the number of iterations. On the other hand, this preconditioner is relatively expensive because it needs additional computer memory and large attention has to be paid to the implementation of the method. Numerical tests show excellent properties for systems obtained from the fourth order problems like plate or shell analyses while for the systems obtained from the second order problems like plane stress the nonpreconditioned iterative method needs shorter time.

Acknowledgement: Financial support for this work was provided by the Grant Agency of the Czech Republic, project n. 201/09/1544 The financial support is gratefully acknowledged.

References

[1] M. Brezina: *Robust Iterative Methods on Unstructured Meshes.* Ph.D. Thesis, University of Colorado at Denver, 1997.

[2] J. Kruis, P. Mayer: *BOSS Preconditioner Applied to Fourth Order Problems.* Submitted to Mathematics and Computers in Simulation.

Computational analysis of stress concentration due to an elliptic hole in a degrading linearized elastic solid

V. Kulvait

Faculty of Mathematics and Physics, Charles University, Prague

1 Introduction

We study boundary-value problems associated with a planar generalized linearized elastic solid body with an elliptic hole in it, when one has a fluid diffusing through the solid. This diffusion of fluid can either enhance or degrade the load carrying capacity of the body, based on how the material moduli of the solid depend on the concentration of the fluid, that is whether the presence of the fluid degrades or strengthens the material. We investigate the nature of the solution when the aspect ratio tends to zero, a problem relevant to the stress singularity at crack tips.

2 The problem setup

2.1 Model geometry

Geometry consists of a plane geometry of a finite square plate with elliptical hole at the centre stretched by constant force on opposite sides. The problem is to find the distribution of stress inside the body. This problem with a circular hole was used as a benchmark problem in [1], see Figure 1.

By appealing to the symmetry of the problem it is possible to solve the problem on a cut-out quarter of the original geometry, see Figure 2.

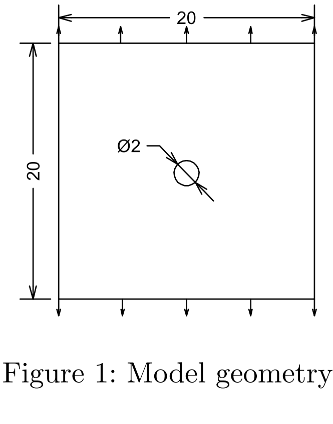


Figure 1: Model geometry.

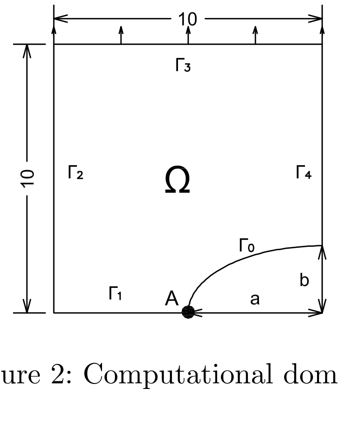


Figure 2: Computational domain.

2.2 Governing equations

For the displacement field $\mathbf{u} : \Omega \to \mathbb{R}^2$, corresponding to that of the linearized elastic solid with elastic moduli dependent on the concentration c of the fluid diffusing through the material, the governing equations take the form:

$$0 = \mathsf{grad}\,(\lambda(c)\mathsf{div}\,\mathbf{u}) + \mathsf{div}\,(\mu(c)(\mathsf{grad}\,\mathbf{u} + (\mathsf{grad}\,\mathbf{u})^T)), \tag{1a}$$

$$\frac{\partial c}{\partial t} = \mathsf{div}\,(k\mathsf{grad}\,c). \tag{1b}$$

Thus, the Cauchy stress tensor τ takes the form

$$\tau = \lambda(c)\mathsf{div}\,\mathbf{u}I + \mu(c)(\mathsf{grad}\,\mathbf{u} + (\mathsf{grad}\,\mathbf{u})^T)\,. \tag{2}$$

2.3 Boundary and initial conditions

The body is loaded in uniform normal stress $T = 4.5$ along the y axis direction, and this provides the boundary condition on Γ_3, see Figure 2. We prescribe symmetry boundary conditions on the Γ_1 and Γ_4. On the remaining parts of boundary Γ_0 and Γ_2 we prescribe homogeneous Neumann condition. Concerning the boundary conditions for c we set $c = 1$ on Γ_0 and homogeneous Neumann condition on the remaining parts of the boundary. The initial conditions at $t = 0$ are for simplicity set to be zero, i.e., $\mathbf{u}(0,\cdot) = \mathbf{0}$ and $c(0,\cdot) = 0$ in Ω.

We carried out computational tests with the values for the model parameters taken partially from the [1] and partially from [3]. For the particular case considered, k does not depend on c. $c \in [0, 1]$. Due to this setting, the Lamé coefficients λ and μ are always positive.

3 Solution for the case of a body whose Lamé coefficients depend on the concentration

We use the system (1) that takes into account the diffusion of the fluid. Particularly the dependence of Lamé coefficients on the concentration c takes the form:

$$\lambda \;=\; \lambda_0(1 - \tfrac{c}{2}), \qquad \mu \;=\; \mu_0(1 - \tfrac{c}{2}). \tag{3}$$

We examine the value of the components of the stress and strain τ_{22} and ε_{22} in the vicinity of the point A and their asymptotic behavior. We solve the model by FEM with approximately 25.000 elements and 100.000 degrees of freedom.

The calculations were carried out for the time interval $t \in [0, 30]$. First, we study the value of τ_{22} and ε_{22} at point A. Then, we examine the distance d, on the line from A to the upper left corner of the geometry, from A to point where the stress drops to half of its value in A. For the strain ε_{22}, this distance is denoted by d_E.

From the Figure 3 we can see that for high values of ratio of $a : b$, the stress is increasing linearly, like in the purely elastic case, with respect to ratio of $a : b$. But its increase is slower. The speed of growth of τ_{22} at the point A is increasing with increasing time. The same is true for ε_{22} at A.

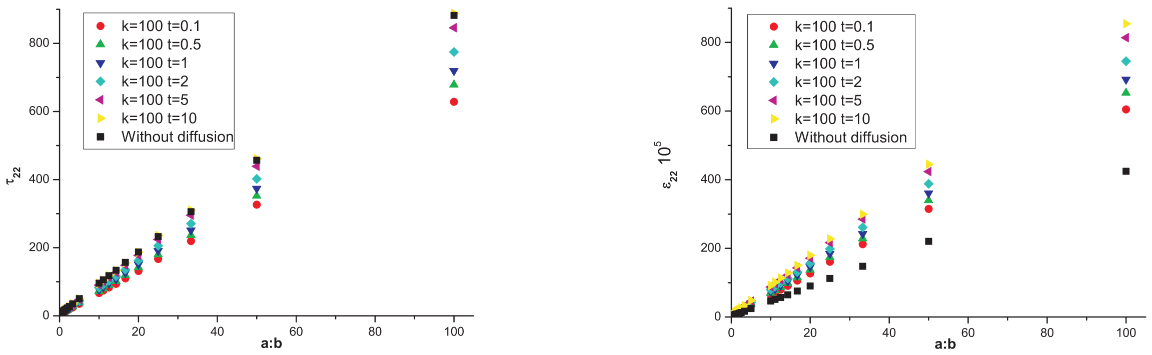


Figure 3: Dependence of τ_{22} and ε_{22} at the point A on the ratio of the axes of the ellipse $a : b$ in coupled model.

In the case of elastostatics the values $(a/b)^2 d$ and $(a/b)^2 d_E$ are approximately constant for sufficiently large ratios of $a : b$. Figure 4 depicts the development of the term $(a/b)^2 d$ for model (1), depending on the value of $a : b$. For low ratios of the axes of the ellipse the term $(a/b)^2 d$ is greater than for the static problem. The asymptotic behavior of the term $(a/b)^2 d$ is the same as in the case of an elastic model. The behavior of the term $(a/b)^2 d_E$ for the strain is much more similar to the case of elastostatics.

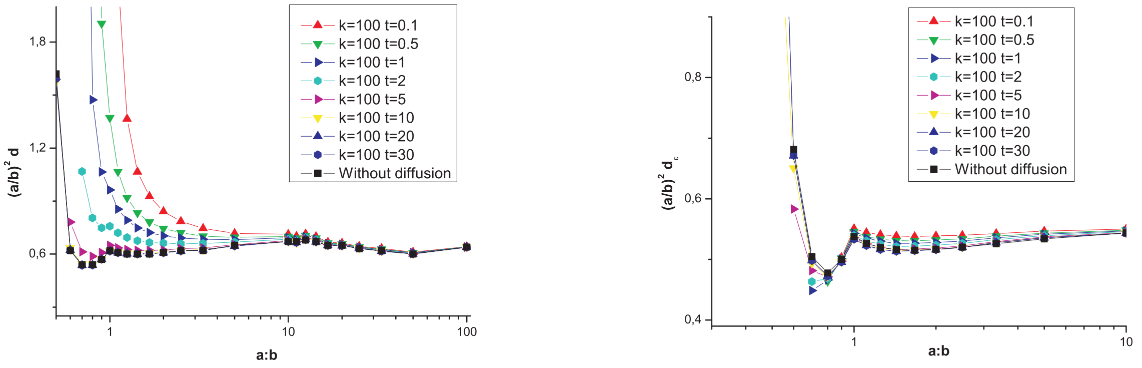


Figure 4: Dependence of term $(a/b)^2 d$ on the ratio of the axes of the ellipse for coefficient of diffusion $k = 100$. On the right is the case with the stress τ_{22} and on the left case with the strain ϵ_{22}.

When the semi-major axis of the ellipse is along the direction of loading, we found that while the classical elastostatic case, the stress component τ_{22} and the strain component ε_{22} change linearly with the aspect ratio $a : b$. However, when diffusion is taking place, we found that the components of stress and strain vary non-linearly for small values of the ratio of $a : b$ but the relationship becomes linear for larger values of the ratio of $a : b$. This result is depicted in Figure 5.

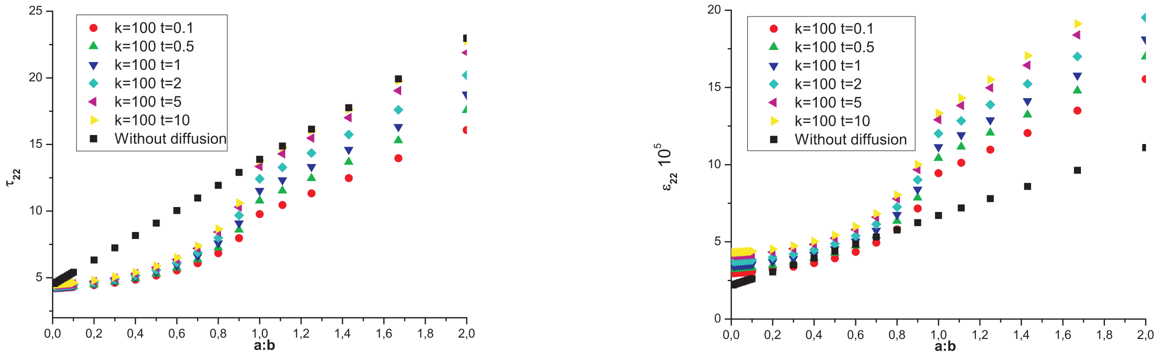


Figure 5: Dependence of τ_{22} and ε_{22} at the point A on the ratio of the axes of the ellipse in coupled model.

4 Conclusion

The system that governs the degradation of the linearized elastic solid due to the diffusion of a fluid leads to stresses and strains that differs significantly from that for the purely linearized elastic model. Numerical results are consistent with the physical expectation that the stress will increase with decreasing ellipse's minor axis b. We find that this increase is linear with respect to the aspect ratio.

Presented results will be published as part of the work [2].

Acknowledgement: V. Kulvait had been partially supported by GACR 201/09/0917.

References

[1] Z. Cai, J. Korsawe, and G. Starke. An adaptive least squares mixed finite element method for the stress-displacement formulation of linear elasticity. *Numerical Methods for Partial Differential Equations*, 21(1):132–148, 2005.

[2] V. Kulvait, J. Málek, and K. R. Rajagopal. Stress concentration due to an elliptic hole in a degrading linearized elastic solid. Submitted for publication in International Journal of Applied Mechanics and Engineering, 2009.

[3] A. Muliana, K. R. Rajagopal, and S. Shankar. Degradation of an Elastic Composite Cylinder due to the Diffusion of a Fluid. *Journal of Composite Materials*, 43(1):1225–1249, 2009.

Theoretical analysis of discrete contact problems with Coulomb friction

T. Ligurský

Charles University in Prague

1 Introduction

Contact problems describe behaviour of loaded deformable bodies in mutual contact. On the contacting parts one often has to take into account non-penetration as well as frictional conditions. The Coulomb law of friction leads to a complicated mathematical problem, in which a lot of issues are still open. In the static case of linear elasticity, existence results have been obtained for a small coefficient of friction $\mathcal{F}$ (see e.g. [7, 1]). More recently, it has been proven in [8] that if a solution possesses a certain property, it is unique provided that $\mathcal{F}$ is small enough. On the other hand, some examples of nonuniqueness are known for large $\mathcal{F}$ ([3, 4]).

In the finite element setting it is known that the discretized problem admits always a solution. There are even results guaranteeing uniqueness of the solution (see e.g. [2]). However, most of them are of global nature and need the assumption on the magnitude of the coefficient of friction $\mathcal{F}$ again. To the author's knowledge the only result concerning local uniqueness of solutions, which admits even large $\mathcal{F}$, has been presented in [5]. Therein, the discrete problem is formulated as a system of non-smooth equations and a suitable version of the implicit function theorem is employed to establish the result.

Having been inspired by this approach, our contribution deals with the local behaviour of discrete solutions. It analyses dependence of solutions not only on the coefficient $\mathcal{F}$ as in [5] but also on loading. In fact, the role of loading seems to be important, as well (see e.g. a discrete model with non-unique solutions in [6]). Besides, qualitative properties of solutions are given.

Throughout the contribution we shall use the following notation: $(.,.)_n$ stands for the scalar product in $\mathbb{R}^n$, $\|.\|_n$ for the corresponding norm, whereas $\|.\|_{n,\infty}$ denotes the max-norm in $\mathbb{R}^n$:

$$\|\boldsymbol{v}\|_{n,\infty} = \max_{i=1,\ldots,n} |v_i|, \quad \boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n.$$

The symbol $\|.\|_n$ is also used for the matrix norm in $\mathbb{R}^{n\times n}$ generated by the vector norm $\|.\|_n$.

2 Problem formulation

Let us consider a two-dimensional Signorini problem with Coulomb friction in which the coefficient of friction $\mathcal{F}$ depends on the spatial variable. A mixed finite element approximation of this model leads to the following variational inequality:

$$\left.\begin{array}{l} \text{Find } (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(\boldsymbol{\mathcal{F}}, -\boldsymbol{\lambda}_\nu) \text{ such that} \\ (\boldsymbol{A}\boldsymbol{u}, \boldsymbol{v})_n = (\boldsymbol{f}, \boldsymbol{v})_n + (\boldsymbol{\lambda}_\nu, \boldsymbol{B}_\nu \boldsymbol{v})_p + (\boldsymbol{\lambda}_t, \boldsymbol{B}_t \boldsymbol{v})_p \quad \forall\, \boldsymbol{v} \in \mathbb{R}^n, \\ (\boldsymbol{\mu}_\nu - \boldsymbol{\lambda}_\nu, \boldsymbol{B}_\nu \boldsymbol{u})_p + (\boldsymbol{\mu}_t - \boldsymbol{\lambda}_t, \boldsymbol{B}_t \boldsymbol{u})_p \ge 0 \quad \forall\, (\boldsymbol{\mu}_\nu, \boldsymbol{\mu}_t) \in \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(\boldsymbol{\mathcal{F}}, -\boldsymbol{\lambda}_\nu), \end{array}\right\} \qquad (\boldsymbol{M})$$

where $\boldsymbol{u}$ represents the displacement vector, $\boldsymbol{\lambda}_\nu$ and $\boldsymbol{\lambda}_t$ are the corresponding normal and tangential Lagrange multipliers and n, p stand for the numbers of degrees of freedom and of the contact nodes, respectively. Further, $\boldsymbol{\mathcal{F}} = (\mathcal{F}_1, \ldots, \mathcal{F}_p) \in \mathbb{R}^p_+$ characterizes the distribution of $\mathcal{F}$ in the contact nodes and the Lagrange-multiplier sets $\boldsymbol{\Lambda}_\nu$, $\boldsymbol{\Lambda}_t(.)$ are defined by

$$\begin{aligned}\boldsymbol{\Lambda}_\nu &= \mathbb{R}^p_-,\\ \boldsymbol{\Lambda}_t(\boldsymbol{\mathcal{F}}, \boldsymbol{g}) &= \{\boldsymbol{\mu}_t = (\mu_{t,1}, \ldots, \mu_{t,p}) \in \mathbb{R}^p \,|\, |\mu_{t,i}| \le \mathcal{F}_i g_i \ \forall\, i = 1, \ldots, p\}, \quad \boldsymbol{g} = (g_1, \ldots, g_p) \in \mathbb{R}^p_+.\end{aligned}$$

By $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ we denote the symmetric stiffness matrix satisfying

$$\exists\, \gamma > 0: \quad (\boldsymbol{A}\boldsymbol{v}, \boldsymbol{v})_n \ge \gamma \|\boldsymbol{v}\|_n^2 \quad \forall\, \boldsymbol{v} \in \mathbb{R}^n$$

and by $\boldsymbol{B}_\nu, \boldsymbol{B}_t \in \mathbb{R}^{p\times n}$ the matrices which represent the linear mappings associating with a displacement vector its normal and tangential component on the contact zone, respectively. Next we suppose that:

$$\left.\begin{array}{l}(j) \text{ the Euclidean norm of each row vector of } \boldsymbol{B}_\nu,\ \boldsymbol{B}_t \text{ is equal to one;}\\ (jj) \text{ each column of } \boldsymbol{B}_\nu,\ \boldsymbol{B}_t \text{ contains at most one nonzero element;}\\ (jjj)\ \exists\, \beta > 0: \quad \sup\limits_{\boldsymbol{0}\ne\boldsymbol{v}\in\mathbb{R}^n} \dfrac{(\boldsymbol{\mu}_\nu, \boldsymbol{B}_\nu\boldsymbol{v})_p + (\boldsymbol{\mu}_t, \boldsymbol{B}_t\boldsymbol{v})_p}{\|\boldsymbol{v}\|_n} \ge \beta \|(\boldsymbol{\mu}_\nu, \boldsymbol{\mu}_t)\|_{2p} \quad \forall\, (\boldsymbol{\mu}_\nu, \boldsymbol{\mu}_t) \in \mathbb{R}^{2p}.\end{array}\right\}$$

Finally, $\boldsymbol{f} \in \mathbb{R}^n$ is the load vector.

3 Theoretical results

In this talk it will be shown that there exists at least one solution to problem $(\boldsymbol{M})$ for any $\boldsymbol{f} \in \mathbb{R}^n$, $\boldsymbol{\mathcal{F}} \in \mathbb{R}^p_+$ and that this solution is unique provided that $\|\boldsymbol{\mathcal{F}}\|_{p,\infty} < \beta\gamma/\|\boldsymbol{A}\|_n$. Furthermore, confining ourselves to $\boldsymbol{\mathcal{F}}$ such that $\|\boldsymbol{\mathcal{F}}\|_{p,\infty} \le \mathcal{F}_{\max}$ for an arbitrary $\mathcal{F}_{\max} \in [0, \beta\gamma/\|\boldsymbol{A}\|_n)$, the unique solution is a Lipschitz-continuous function of $\boldsymbol{\mathcal{F}}$:

$$\exists\, \delta > 0: \quad \|\boldsymbol{\mathcal{S}}_{\boldsymbol{f}}(\boldsymbol{\mathcal{F}}) - \boldsymbol{\mathcal{S}}_{\boldsymbol{f}}(\bar{\boldsymbol{\mathcal{F}}})\|_{n+2p} \le \delta \|\boldsymbol{\mathcal{F}} - \bar{\boldsymbol{\mathcal{F}}}\|_{p,\infty} \quad \forall\, \boldsymbol{\mathcal{F}}, \bar{\boldsymbol{\mathcal{F}}} \in \mathbb{R}^p_+,\ \|\boldsymbol{\mathcal{F}}\|_{p,\infty}, \|\bar{\boldsymbol{\mathcal{F}}}\|_{p,\infty} \le \mathcal{F}_{\max}.$$

Here $\boldsymbol{\mathcal{S}}_{\boldsymbol{f}} : \mathbb{R}^p_+ \to \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p$ denotes the solution map for a *fixed* $\boldsymbol{f} \in \mathbb{R}^n$:

$$\boldsymbol{\mathcal{S}}_{\boldsymbol{f}}(\boldsymbol{\mathcal{F}}) = (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t), \quad \boldsymbol{\mathcal{F}} \in \mathbb{R}^p_+,\ \|\boldsymbol{\mathcal{F}}\|_{p,\infty} \le \mathcal{F}_{\max},$$

where $(\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$ is the solution to $(\boldsymbol{M})$ with the coefficient $\boldsymbol{\mathcal{F}}$ and the load vector $\boldsymbol{f}$.

To present local analysis of the solutions we shall restrict ourselves to coefficients $\boldsymbol{\mathcal{F}}$ from the following set:

$$\boldsymbol{\mathcal{A}} = \{\boldsymbol{\mathcal{F}} \in \mathbb{R}^p \,|\, \mathcal{F}_i > 0 \ \forall\, i = 1, \ldots, p\}.$$

In addition, we shall introduce the Lagrange-multiplier set $\boldsymbol{\Lambda}_t(.)$ which does not depend on $\boldsymbol{\mathcal{F}}$:

$$\boldsymbol{\Lambda}_t(\boldsymbol{g}) := \{\boldsymbol{\mu}_t = (\mu_{t,1}, \ldots, \mu_{t,p}) \in \mathbb{R}^p \,|\, |\mu_{t,i}| \le g_i \ \forall\, i = 1, \ldots, p\}, \quad \boldsymbol{g} = (g_1, \ldots, g_p) \in \mathbb{R}^p_+.$$

The equivalent formulation of $(\boldsymbol{M})$ reads as follows:

$$\left.\begin{array}{l}\text{Find } (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(-\boldsymbol{\lambda}_\nu) \text{ such that}\\ (\boldsymbol{A}\boldsymbol{u}, \boldsymbol{v})_n = (\boldsymbol{f}, \boldsymbol{v})_n + (\boldsymbol{\lambda}_\nu, \boldsymbol{B}_\nu\boldsymbol{v})_p + (\boldsymbol{F}\boldsymbol{\lambda}_t, \boldsymbol{B}_t\boldsymbol{v})_p \quad \forall\, \boldsymbol{v} \in \mathbb{R}^n,\\ (\boldsymbol{\mu}_\nu - \boldsymbol{\lambda}_\nu, \boldsymbol{B}_\nu\boldsymbol{u})_p + (\boldsymbol{F}(\boldsymbol{\mu}_t - \boldsymbol{\lambda}_t), \boldsymbol{B}_t\boldsymbol{u})_p \ge 0 \quad \forall\, (\boldsymbol{\mu}_\nu, \boldsymbol{\mu}_t) \in \boldsymbol{\Lambda}_\nu \times \boldsymbol{\Lambda}_t(-\boldsymbol{\lambda}_\nu),\end{array}\right\} \qquad (\boldsymbol{M}^*)$$

where $\boldsymbol{F} := \boldsymbol{F}(\boldsymbol{\mathcal{F}}) = \mathrm{diag}\{\mathcal{F}_1, \ldots, \mathcal{F}_p\} \in \mathbb{R}^{p\times p}$.

Our first result concerning local uniqueness of solutions to $(\boldsymbol{M}^*)$ says that the analysis of local behaviour of a solution as a function of the coefficient $\boldsymbol{\mathcal{F}}$ can be converted to the one of local behaviour of the solution as a function of the load vector $\boldsymbol{f}$. To state the assertion more precisely we introduce the set-valued mappings $\boldsymbol{S}^*_{\boldsymbol{\mathcal{F}}} : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n+2p}$ and $\boldsymbol{\mathcal{S}}^*_{\boldsymbol{f}} : \boldsymbol{\mathcal{A}} \rightrightarrows \mathbb{R}^{n+2p}$ by

$$\begin{aligned} \boldsymbol{S}^*_{\boldsymbol{\mathcal{F}}}(\boldsymbol{f}) &= \{\boldsymbol{y}\}, \quad \boldsymbol{f} \in \mathbb{R}^n, \\ \boldsymbol{\mathcal{S}}^*_{\boldsymbol{f}}(\boldsymbol{\mathcal{F}}) &= \{\boldsymbol{y}\}, \quad \boldsymbol{\mathcal{F}} \in \boldsymbol{\mathcal{A}}, \end{aligned}$$

where $\{\boldsymbol{y}\}$, $\boldsymbol{y} \equiv (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t)$, denotes the set of all solutions to problem $(\boldsymbol{M}^*)$ with the coefficient $\boldsymbol{\mathcal{F}}$ and the load vector $\boldsymbol{f}$ in both cases. The only difference between these mappings is that $\boldsymbol{S}^*_{\boldsymbol{\mathcal{F}}}$ associates the set of solutions to *varying* $\boldsymbol{f}$ for $\boldsymbol{\mathcal{F}}$ *fixed* whereas $\boldsymbol{\mathcal{S}}^*_{\boldsymbol{f}}$ associates the set of solutions to *varying* $\boldsymbol{\mathcal{F}}$ for $\boldsymbol{f}$ *fixed*.

Theorem 1 *Let $\boldsymbol{\mathcal{F}}^0 \in \boldsymbol{\mathcal{A}}$, $\boldsymbol{f} \in \mathbb{R}^n$ be arbitrary and let $\boldsymbol{y}^0 \equiv (\boldsymbol{u}^0, \boldsymbol{\lambda}^0_\nu, \boldsymbol{\lambda}^0_t) \in \mathbb{R}^{n+2p}$ be a solution to $(\boldsymbol{M}^*)$ with the coefficient $\boldsymbol{\mathcal{F}}^0$ and the load vector $\boldsymbol{f}$. If $\boldsymbol{S}^*_{\boldsymbol{\mathcal{F}}^0}$ has a locally Lipschitz-continuous branch containing $\boldsymbol{y}^0$ in a vicinity of $\boldsymbol{f} \in \mathbb{R}^n$, i.e. there exist a single-valued Lipschitz-continuous function $\boldsymbol{\varphi}_{\boldsymbol{\mathcal{F}}^0}$ from a neighbourhood $\boldsymbol{O}$ of $\boldsymbol{f}$ into $\mathbb{R}^{n+2p}$ and a neighbourhood $\hat{\boldsymbol{V}}$ of $\boldsymbol{y}^0$ such that*

$$\boldsymbol{\varphi}_{\boldsymbol{\mathcal{F}}^0}(\boldsymbol{f}) = \boldsymbol{y}^0 \qquad \textit{and} \qquad \boldsymbol{\varphi}_{\boldsymbol{\mathcal{F}}^0}(\boldsymbol{\xi}_f) = \boldsymbol{S}^*_{\boldsymbol{\mathcal{F}}^0}(\boldsymbol{\xi}_f) \cap \hat{\boldsymbol{V}} \quad \forall\, \boldsymbol{\xi}_f \in \boldsymbol{O},$$

then there are neighbourhoods $\boldsymbol{U}$, $\boldsymbol{V}$ of $\boldsymbol{\mathcal{F}}^0$, $\boldsymbol{y}^0$, respectively, and a single-valued Lipschitz-continuous function $\boldsymbol{\sigma}_{\boldsymbol{f}} : \boldsymbol{U} \to \boldsymbol{V}$ satisfying

$$\boldsymbol{\sigma}_{\boldsymbol{f}}(\boldsymbol{\mathcal{F}}^0) = \boldsymbol{y}^0 \qquad \textit{and} \qquad \boldsymbol{\sigma}_{\boldsymbol{f}}(\boldsymbol{\mathcal{F}}) = \boldsymbol{\mathcal{S}}^*_{\boldsymbol{f}}(\boldsymbol{\mathcal{F}}) \cap \boldsymbol{V} \quad \forall\, \boldsymbol{\mathcal{F}} \in \boldsymbol{U}.$$

To complete this analysis we shall give sufficient conditions guaranteeing the existence of locally Lipschitz-continuous branches of the solution map $\boldsymbol{S}^*_{\boldsymbol{\mathcal{F}}}$ for $\boldsymbol{\mathcal{F}} \in \boldsymbol{\mathcal{A}}$ fixed. We shall also show that these conditions are not satisfied only for $\boldsymbol{f}$ restricted to a union of some subspaces of dimension strictly lower than n.

4 An elementary example

In the end of the talk our theoretical approach will be illustrated on an elementary example corresponding to a single linear triangular finite element depicted in Figure 1. This example is taken from [5] and it is nothing else than a special case of a model studied in [6].

Denoting $\boldsymbol{u} := (u_\nu, u_t)$ and $\boldsymbol{f} := (f_\nu, f_t)$, its formulation can be written as follows:

$$\left.\begin{aligned} &\text{Find } (u_\nu, u_t, \lambda_\nu, \lambda_t) \in \mathbb{R}^4 \text{ such that} \\ &au_\nu - bu_t - \lambda_\nu - f_\nu = 0, \\ &-bu_\nu + au_t - \lambda_t - f_t = 0, \\ &\lambda_\nu - P_{(-\infty,0]}(\lambda_\nu - ru_\nu) = 0, \\ &\lambda_t - P_{[\mathcal{F}\lambda_\nu, -\mathcal{F}\lambda_\nu]}(\lambda_t - ru_t) = 0, \end{aligned}\right\}$$

where the constants $a := (\lambda + 3\mu)/2$ and $b := (\lambda + \mu)/2$ depend on the Lamé coefficients $\lambda \geq 0$, $\mu > 0$ characterizing the considered isotropic and homogeneous material and $r > 0$ is an arbitrarily chosen parameter. Further, $P_{(-\infty,0]}$ and $P_{[\mathcal{F}\lambda_\nu, -\mathcal{F}\lambda_\nu]}$ are projections of $\mathbb{R}^1$ onto the intervals $(-\infty, 0]$ and $[\mathcal{F}\lambda_\nu, -\mathcal{F}\lambda_\nu]$, respectively.

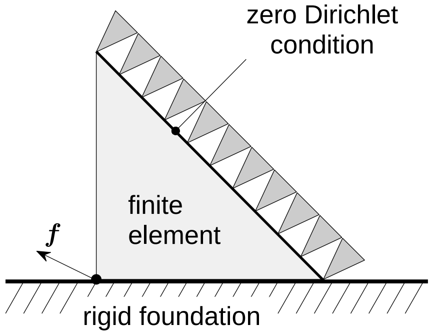


Figure 1: Geometry of the elementary example.

It will be shown that the unicity of the solutions to this example depends not only on the coefficient $\mathcal{F}$ but also on the load vector $\boldsymbol{f}$. Moreover, we shall see that the structure of solutions with respect to $\boldsymbol{f}$ is much simpler than the one with respect to $\mathcal{F}$.

Acknowledgements: This work has been supported by the grant No. 18008 of the Charles University Grant Agency and under grant No. 201/07/0294 of the Grant Agency of the Czech Republic. The support of the Nečas Center for Mathematical Modeling is also acknowledged. Last but not least, the author would like to thank Prof. Jaroslav Haslinger for his help and constructive discussions.

References

[1] C. Eck, J. Jarušek: *Existence results for the static contact problem with Coulomb friction.* Math. Models Methods Appl. Sci. 8(3), 445-468, 1998.

[2] J. Haslinger: *Approximation of the Signorini problem with friction, obeying the Coulomb law.* Math. Methods Appl. Sci. 5, 422–437, 1983.

[3] P. Hild: *An example of nonuniqueness for the continuous static unilateral contact model with Coulomb friction.* C. R., Math., Acad. Sci. Paris 337(10), 685–688, 2003.

[4] P. Hild: *Non-unique slipping in the Coulomb friction model in two-dimensional linear elasticity.* Q. J. Mech. Appl. Math. 57(2), 225–235, 2004.

[5] P. Hild, Y. Renard: *Local uniqueness and continuation of solutions for the discrete Coulomb friction problem in elastostatics.* Quart. Appl. Math., 63, 553–573, 2005.

[6] V. Janovský: *Catastrophic features of Coulomb friction model.* The Mathematics of Finite Elements and Applications IV, MAFELAP 1981, Proc. Conf., Uxbridge/Middlesex 1981, 259–264, 1982.

[7] J. Nečas, J. Jarušek, J. Haslinger: *On the solution of the variational inequality to the Signorini problem with small friction.* Boll. Unione Mat. Ital., V. Ser., B, 17, 796–811, 1980.

[8] Y. Renard: *A uniqueness criterion for the Signorini problem with Coulomb friction.* SIAM J. Math. Anal. 38(2), 452–467, 2006.

Fast boundary elements for 3D Helmholtz equation

D. Lukáš[1], *J. Szweda*[2]

[1]Department of Applied Mathematics
[2]Department of Mechanics, VŠB–Technical University of Ostrava

We are faced with a problem of acoustic noise analysis of a railway wheel while aiming at shape optimization of its profile such that scattering at certain frequencies will be damped in the end. This contribution presents the related forward problem and its efficient numerical solution using a novelty variant of the adaptive cross approximation algorithm when applied to hypersingular or double–layer Helmholtz operators, the compression time of which originally suffers from the multiple-element support of the piecewise linear ansatz functions.

1 Galerkin direct BEM for the exterior Helmholtz problem

We consider an elastic structure of a railway wheel fixed along the axle and loaded with a harmonic force on the perimeter. For a given material density ρ, elastic modulus E, Poisson ratio ν, damping β and a load f of an angular frequency ω, we search for the harmonic displacement field $\mathcal{U}(x,t) := \mathrm{Re}\{\mathbf{u}(\mathbf{x})\,\mathrm{e}^{\mathrm{i}\omega t}\}$ such that

$$\begin{array}{rcll} -\nabla\cdot\boldsymbol{\sigma}(\mathbf{u}) - \rho\,\omega^2\,\mathbf{u} & = & \mathbf{0} & \text{in } \Omega, \\ \mathbf{u} & = & \mathbf{0} & \text{on } \Gamma_{\mathrm{D}}, \\ \boldsymbol{\sigma}\cdot\mathbf{n} & = & f\mathbf{n}\,\delta_{\mathbf{a}} & \text{on } \Gamma_{\mathrm{N}}, \end{array}$$

where $\sigma_{ij}(\mathbf{u}) = \lambda\delta_{ij}\nabla\cdot\mathbf{u} + \mu(\partial_i u_j + \partial_j u_i)$, $\lambda := \frac{E\nu}{(1+\nu)(1-2\nu)}\,(1+2\beta\mathrm{i})$, $\mu := \frac{E}{2(1+\nu)}\,(1+2\beta\mathrm{i})$, and $\delta_{\mathbf{a}}$ is a boundary Dirac distribution at $\mathbf{a}\in\Gamma_{\mathrm{N}}$. For the numerical solution we employ the lowest–order tetrahedral finite elements.

The resulting deformations provide the Neumann boundary data for the exterior Helmholtz problem

$$\begin{array}{rcll} -\triangle p - \kappa^2\,p & = & 0, & \text{in } \Omega^e := \mathbb{R}^3 \setminus \overline{\Omega}, \\ \partial p/\partial\mathbf{n} & = & v_n := \omega^2\rho\mathbf{u}\cdot\mathbf{n}, & \text{on } \Gamma := \partial\Omega, \\ \left|\left(\frac{\mathbf{x}}{|\mathbf{x}|}, \nabla p(\mathbf{x})\right) - i\kappa p(\mathbf{x})\right| & = & O(|\mathbf{x}|^{-2}), & \mathbf{x}\to\infty, \end{array}$$

where $\kappa := \omega/c$ with the speed of sound c. The acoustical pressure is given by $\mathcal{P}(\mathbf{x},t) := \mathrm{Re}\{p(\mathbf{x})\,\mathrm{e}^{\mathrm{i}\omega t}\}$. We consider the Galerkin direct approach to the boundary integral equation of the latter problem, cf. [1], which reads as follows: Find $p\in H^{-1/2}(\Gamma)$ such that

$$\forall v\in H^{-1/2}(\Gamma)\ :\ \langle Dp, v\rangle_\Gamma = \langle(-1/2I + K')v_n, v\rangle_\Gamma,$$

where I denotes the identity, the Helmholtz fundamental solution reads $P(\mathbf{x},\mathbf{y}) := \mathrm{e}^{\mathrm{i}\kappa r}/(4\pi|r|)$ with $r := |\mathbf{x}-\mathbf{y}|$ and where

$$\langle Dp, v\rangle_\Gamma := \int_\Gamma\int_\Gamma P(\mathbf{x},\mathbf{y})\,\left[(\mathbf{n}(\mathbf{x})\times\nabla\widetilde{v}(\mathbf{x}))\cdot(\mathbf{n}(\mathbf{y})\times\nabla\widetilde{p}(\mathbf{y})) - \kappa^2\mathbf{n}(\mathbf{x})v(\mathbf{x})\cdot\mathbf{n}(\mathbf{y})p(\mathbf{y})\right]\,dS(\mathbf{y})\,dS(\mathbf{x}),$$

$$\langle K'v_n, v\rangle_\Gamma := \int_\Gamma\int_\Gamma (\partial P(\mathbf{x},\mathbf{y})/\partial\mathbf{n}(\mathbf{y}))\,v_n(\mathbf{y})v(\mathbf{x})\,dS(\mathbf{y})\,dS(\mathbf{x}).$$

We employ the lowest–order boundary element discretization using a numerical quadrature of regularized kernels [2].

2 Element–based adaptive cross approximation

Typically, the boundary of the railway wheel is discretized into 15112 triangles and 22668 nodes, therefore, we aim at a sparsification of the orginally fully populated BEM matrices. The adaptive cross approximation (ACA) [3] turns out to be a good choice. It relies on a hierarchical decomposition of a matrix into the admissible (far field) and nonadmissible (near field) blocks so that the nonadmissible blocks are assembled completely, while the admissible ones are adaptively approximated by a sum of rank–one matrices. The algorithm is easy to include into an existing code, since it only requires assembling of rows and columns. This works fast for the lowest–order discretization of the single–layer V as well as the double–layer operator K, since either the rows or columns are related to the piecewise constant ansatz functions. However, in case of the hypersingular matrix D the piecewise linear ansatz functions have multi–element supports, thus, the entry evaluation costs more, moreover, the quadrature over many couples of elements is performed several times. In our case, the ACA applied to K compressed to 12% takes 25 minutes, while it takes 3.4 hours in the case of D compressed to 15%. Note that this is not the case for the Laplace problem, since we have the representations $D_\triangle = T_\triangle^T \cdot V_\triangle \cdot T_\triangle$.

As a remedy for the Helmholtz operator D, we propose to hierarchically cluster the matrix element–wise, rather than nodal–wise, and approximate the admissible blocks by piecewise constant ansatz functions, instead of piecewise linear, while using the admissible part of V similarly to the Laplace case. Our element–based ACA approximation is as follows:

$$
\begin{aligned}
D = \sum_i \sum_j D^{(i,j)} &= \sum_i \sum_j T_i^T \, V_{ij} \, T_j - \kappa^2 R_i^T \left(\int_{\Gamma_i} \int_{\Gamma_j} P(\mathbf{x}, \mathbf{y}) \, \phi_k^{(i)}(\mathbf{x}) \, \phi_l^{(j)}(\mathbf{y}) \, dS(\mathbf{y}) \, dS(\mathbf{x}) \right)_{k,l} R_j \\
&\approx T^T \cdot V^{\mathrm{adm}} \cdot T - \frac{\kappa^2}{4} \sum_{(i,j)\in\mathcal{N}^{\mathrm{adm}}} R_i^T \cdot \begin{pmatrix} 1/2 & 1/2 \end{pmatrix} \cdot V_{ij}^{\mathrm{adm}} \cdot \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \cdot R_j + \sum_{(i,j)\in\mathcal{N}^{\mathrm{non}}} D^{(i,j)},
\end{aligned}
$$

where $\mathcal{N}^{\mathrm{adm}}$ and $\mathcal{N}^{\mathrm{non}}$ denotes the admissible and the nonadmissible blocks, respectively. Here, each element couple is entered at most once. In our case, the matrix D was assembled in less than one hour. However, the application of D costs more. Note that our approach can also be applied to K.

Finally, to validate our method, we applied it to a unit ball and cube for the problem with the given solution $p(\mathbf{x}) := \mathrm{e}^{\mathrm{i}\kappa|\mathbf{x}-\mathbf{x}_\mathrm{s}|}/(4\pi|\mathbf{x}-\mathbf{x}_\mathrm{s}|)$, where $\kappa := 2\pi\, 151.6/340$ and with the scatterer placed at $\mathbf{x_s} := (0.05, 0.05, 0.05)$. For example, the ball surface was discretized on 6 levels into 40–40960 elements. the relative solution error measured in $L^2(\Gamma)$ converged from 10^{-1} to 10^{-4}. The ACA assembling time for D at the finest level took 42397 seconds comparing 5515 seconds in our approach, however, the 123 GMRES iterations took 496 seconds vers. our 6662 seconds.

References

[1] O. Steinbach, S. Rjasanow: *The Fast Solution of Boundary Integral Equations.* Springer, 2007.

[2] S. Sauter, C. Schwab: *Randelementmethoden: Analyse, Numerik und Implementierung schneller Algorithmen.* B.G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2004.

[3] M. Bebendorf, R. Grzhibovskis: *Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation.* Math. Meth. Appl. Sci. 29, 1721–1747, 2006.

A recursive formulation of limited memory variable metric methods

L. Lukšan, J. Vlček

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Praha 8, and
Technical University of Liberec, Hálkova 6, 461 17 Liberec

Variable metric methods with limited memory can be efficiently used for large-scale unconstrained optimization in case the sparsity pattern of the Hessian matrix is not known. These methods are usually realized in the line-search framework so that they generate a sequence of points $x_i \in \mathcal{R}^n$, $i \in \mathcal{N}$, by the simple process

$$x_{i+1} = x_i + \alpha_i d_i, \tag{1}$$

where $d_i = -H_i g_i$ is a direction vector, H_i is a positive definite approximation of the inverse Hessian matrix and $\alpha_i > 0$ is a scalar step-size chosen in such a way that

$$F_{i+1} - F_i \le \varepsilon_1 \, \alpha_i \, d_i^T g_i, \quad d_i^T g_{i+1} \ge \varepsilon_2 \, d_i^T g_i \tag{2}$$

(the weak Wolfe conditions), where $F_i = F(x_i)$, $g_i = \nabla F(x_i)$ and $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$. Matrices H_i, $i \in N$, are computed either by using a limited number ($m \ll n$) of variable metric updates applied to the scaled unit matrix or by updating low dimension matrices. The first approach, used in [9], is based on the computation of the direction vector d_i using the Strang recurrences [8]. The second approach, used in [1], is based on the matrix expression described below. To simplifying notation, we omit index i and replace index $i+1$ by $+$.

Variable metric method from the Broyden class use the update

$$\begin{aligned} H_+ &= H + UMU^T = H + [d, Hy] \begin{bmatrix} m_1, & m_2 \\ m_2, & m_3 \end{bmatrix} \begin{bmatrix} d \\ Hy \end{bmatrix} \\ &= H + \frac{1}{b} dd^T - \frac{1}{a} Hy(Hy)^T + \frac{\eta}{a} \left(\frac{a}{b} d - Hy \right) \left(\frac{a}{b} d - Hy \right)^T , \end{aligned} \tag{3}$$

where $d = x_+ - x$, $y = g_+ - g$, $a = y^T Hy$, $b = y^T d$ and η is a free parameter. We need to express m consecutive steps of (3) (with the initial matrix γI) in the form $H_+ = \gamma I + \bar{U} \bar{M} \bar{U}^T$, where $\bar{U} \in R^{n \times 2m}$ and $\bar{M} \in R^{2m \times 2m}$. In [1], the authors propose explicit expressions of the matrix $\bar{M}$ for three classic variable metric updates: DFP ($\eta = 0$), BFGS ($\eta = 1$) and the rank one ($\eta = b/(b-a)$). For other values of the parameter η, such explicit expressions are not known. In this contribution we describe another way, based on recursive construction of the matrix $\bar{M}$, which allows us to realize any member of the Broyden class of the variable metric updates. The following theorem is proved in [7].

Theorem 1 *Let H_+ be a matrix defined by (3) and $H = \gamma I + \bar{U} \bar{M} \bar{U}^T$. Then*

$$H_+ = H_1 + \bar{U}_+ \bar{M}_+ \bar{U}_+^T,$$

where $\bar{U}_+ = [\bar{U},\, d,\, H_1 y]$ and

$$\bar{M}_+ = \begin{bmatrix} \bar{M} + m_3 \, zz^T, & m_2 \, z, & m_3 \, z \\ m_2 \, z^T, & m_1, & m_2 \\ m_3 \, z^T, & m_2, & m_3 \end{bmatrix}. \tag{4}$$

Here $m_1 = (1/b)(\eta a/b + 1)$, $m_2 = -\eta/b$, $m_3 = (\eta - 1)/a$ are elements of matrix M, $z = \bar{M} \bar{r}$ and $\bar{r} = \bar{U}^T y$.

If $i \le m$, the construction of matrix H_{i+1} follows straightforwardly from Theorem 1. Thus we describe the construction of matrix H_{i+1} in case $i > m$. We will assume that $H_{i+1-m} = \gamma_i I$. At the beginning of the i-th iteration, we have available the rectangular matrix $\bar{U}_{i-1} = [d_{i-m}, y_{i-m}, \dots, d_{i-1}, y_{i-1}]$ and the block upper triangular matrix

$$\bar{R}_{i-1} = \left[\begin{array}{lll} d_{i-m}^T y_{i-m}, & \dots & d_{i-m}^T y_{i-1} \\ y_{i-m}^T y_{i-m}, & \dots & y_{i-m}^T y_{i-1} \\ \dots\dots\dots & \dots & \dots\dots\dots \\ 0, & \dots & d_{i-1}^T y_{i-1} \\ 0, & \dots & y_{i-1}^T y_{i-1} \end{array} \right],$$

whose every block contains two rows and one column. First we determine matrix $\bar{U}_i = [d_{i-m+1}, y_{i-m+1}, \dots, d_i, y_i]$ from matrix $\bar{U}_{i-1}$ by deleting the first two columns and adding the last two columns. Similarly easily we obtain matrix $\bar{R}_i$ from matrix $\bar{R}_{i-1}$. Only the last column $\bar{U}_i^T y_i$ of this matrix has to be computed. Furthermore, we compute recursively matrix $\bar{M}_i = \bar{M}_i^i$ in such a way that we set

$$\bar{M}_{i-m+1}^i = \left[\begin{array}{ll} m_{i-m+1}^1, & m_{i-m+1}^2 \\ m_{i-m+1}^2, & m_{i-m+1}^3 \end{array} \right]$$

(indices 1, 2, 3 are now placed up) and for $i-m+1 \le j \le i-1$, compute vector $z_j = \bar{M}_j^i \bar{r}_j$, where $\bar{r}_j$ is $j-i+m$-th column of matrix $\bar{R}_i$, whose every even element is multiplied by number γ_i (since $H_{i+1-m} = \gamma_i I$), and set

$$\bar{M}_{j+1}^i = \left[\begin{array}{lll} \bar{M}_j^i + m_{j+1}^3 \, z_j z_j^T, & m_{j+1}^2 \, z_j, & m_{j+1}^3 \, z_j \\ m_{j+1}^2 \, z_j^T, & m_{j+1}^1, & m_{j+1}^2 \\ m_{j+1}^3 \, z_j^T, & m_{j+1}^2, & m_{j+1}^3 \end{array} \right].$$

Vector $H_{i+1} g_{i+1}$ is computed by the formula

$$\begin{aligned} H_{i+1} g_{i+1} = \gamma_i g_{i+1} \quad + \quad & [d_{i-m+1}, \, \gamma_i y_{i-m+1}, \dots, d_i, \, \gamma_i y_i] \, \bar{M}_i \\ & [d_{i-m+1}, \, \gamma_i y_{i-m+1}, \dots, d_i, \, \gamma_i y_i]^T g_{i+1} \end{aligned}$$

(even columns of matrix $\bar{U}_i$ are multiplied by number γ_i). As we can see, approximately $6mn$ operations (addition and multiplication) are consumed in i-th iteration. However, approximately $2(m-1)n$ operations can be saved, if we compute and store inner products $d_j^T g_{i+1}$, $y_j^T g_{i+1}$ instead of $d_j^T y_i$, $y_j^T y_i$, $i-m+1 \le j \le i$. Then the first $m-1$ inner products $d_j^T y_i$, $y_j^T y_i$, $i-m+1 \le j \le i-1$ can be determined from the previously computed inner products by the formulas $d_j^T y_i = d_j^T g_{i+1} - d_j^T g_i$, $y_j^T y_i = y_j^T g_{i+1} - y_j^T g_i$, $i-m+1 \le j \le i-1$. Thus it is necessary to compute only two inner products $d_i^T y_i$, $y_i^T y_i$. Inner products $d_j^T g_{i+1}$, $y_j^T g_{i+1}$, $i-m+1 \le j \le i$ can be used for the computation of direction vector s_{i+1}, so we save $2mn$ operations.

The method described has been tested by using a set of 60 test problems with 1000 variables. This set (Test25) was obtained by merging the sets Test14, Test15, Test18 described in [6], which can be downloaded from `http://www.cs.cas.cz/luksan/test.html` (together with report [6]). The results of the tests are listed in Table 1, where `NIT` is the total number of iterations, `NFV` is the total number of function and gradient evaluations, `NF` is the number of failures and `TIME` is the total CPU time. We have tested the original `LBFGS` subroutine, described in [2], and our realizations of limited memory variable metric methods implemented in the UFO system [5]. In Table 1, `BFGSSTR` denotes the limited memory BFGS method with the Strang recurrences [9] (an analogy of `LBFGS`), `BFGSBNS` denotes the limited memory BFGS method with compact matrices described in [1], `BFGSNEW` denotes the limited memory BFGS method with recursive construction of matrix $\bar{M}$ described above, LMVMNEW denotes the limited memory variable metric method with recursive construction of matrix $\bar{M}$ that use parameter η proposed in [4], and CG denotes the conjugate gradient method. Note that the first four rows in Table 1 correspond to different implementations of the BFGS method and that our approach gives the best results.

Method	NIT	NFV	F	TIME
LBFGS	110406	117226	2	43.38
BFGSSTR	99125	104085	-	37.56
BFGSBNS	91650	96235	-	36.89
BFGSNEW	85430	89796	-	33.50
LMVMNEW	92877	99033	-	34.61
CG	144990	222460	1	60.77

Table 1: Test results.

Acknowledgement: This work has been supported by the Czech science foundation, project No. 201/09/1957, and the institutional research plan No. AV0Z10300504.

References

[1] R.H.Byrd, J.Nocedal, R.B.Schnabel: *Representation of quasi-Newton matrices and their use in limited memory methods.* Math. Programming 63, 129-156, 1994.

[2] D.C.Liu, J.Nocedal: *On the limited memory BFGS method for large scale optimization.* Mathematical Programming 45, 503-528, 1989.

[3] L.Lukšan: *Numerické optimalizační metody.* Nepodmíněná minimalizace. Výzkumná zpráva V-1058, Ústav informatiky AV ČR, Praha 2009.

[4] L.Lukšan, E.Spedicato: *Variable metric methods for unconstrained optimization and nonlinear least squares.* Journal of Computational and Applied Mathematics 124, 61-93, 2000.

[5] L.Lukšan, M.Tůma, J.Vlček, N.Ramešová, M.Šiška, J.Hartman, C.Matonoha: *Interactive System for Universal Functional Optimization.* Research Report V-1040, Institute of Computer Science Czech Academy of Sciences, Prague 2008.

[6] L.Lukšan, J.Vlček: *Sparse and partially separable test problems for unconstrained and equality constrained optimization.* Research Report V-767, Institute of Computer Science, Czech Academy of Sciences, Prague 1998.

[7] L.Lukšan, J.Vlček: *Limited memory variable metric methods from the Broyden class.* Research Report V-1059, Institute of Computer Science Czech Academy of Sciences, Prague 2009.

[8] H.Matthies, G.Strang: *The solution of nonlinear finite element equations.* Int. J. for Numerical Methods in Engineering 14, 1613-1623, 1979.

[9] J. Nocedal: *Updating quasi-Newton matrices with limited storage.* Math. Comp. 35, 773-782, 1980.

Quantitative analysis of numerical solution for the Gray-Scott model

J. Mach

Czech Technical University, Prague

1 Introduction

Reaction-diffusion systems are a class of systems of partial differential equations of parabolic type. It includes mathematical models describing various phenomena e.g. in the fields of physics, biology and chemistry. Gray-Scott model is one of these models. It was first introduced in 1984 by P. Gray and S. K. Scott [1]. It is a mathematical model of the autocatalytic chemical reaction $U + 2V \longrightarrow 3V$, $V \longrightarrow P$. U, V are reactants and P is final product of the reaction. Chemical substance U is being continuously added into the reactor and the product P is being continuously removed from the reactor during the reaction. Later it has been extensively studied e.g. by Wei [2], Winter [3], Ueyama [5], Dkhil [6], Doelman [7]. This model is well known to exhibit rich dynamics, see e.g. Nishiura [4]. There exist chemical systems exhibiting features similar to those of the Gray-Scott model, see e.g. Mazin [8] and references therein.

2 Problem formulation

We study the Gray-Scott in 2D. Assume that $\Omega \equiv (0, L) \times (0, L)$ is an open square representing the square reactor, where the chemical reaction takes place, $\partial\Omega$ is its boundary andν is its outer normal. Then initial-boundary value problem for the Gray-Scott model is a system of two partial differential equations of parabolic type $u_t = a\Delta u - uv^2 + F(1 - u)$, $v_t = b\Delta v + uv^2 - (F + k)v$ in $\Omega \times (0, T)$ with initial conditions $u(\cdot, 0) = u_{ini}$, $v(\cdot, 0) = v_{ini}$ and zero Neumann boundary conditions $\frac{\partial u}{\partial \nu} \mid_{\partial\Omega}= 0$, $\frac{\partial v}{\partial \nu} \mid_{\partial\Omega}= 0$. Functions u, v are unknowns representing concentrations of chemical substances U, V. Parameter F denotes the rate at which the chemical substance U is being added during the chemical reaction, $F + k$ is the rate of $V \to P$ transformation and a, b are constants characterizing the environment in the reactor. This system may be rewritten in several dimensionless forms. We use the one which is used also e.g. in [3, 8, 9].

3 Numerical schemes

Computational studies of the Gray-Scott model show difficulties in convergence. We compare two numerical schemes for solution of the initial-boundary value problem defined in Sect. 2 in order to disclose details of these problems. Both of them are based on the method of lines. For spatial discretization we used structured numerical grids consisting of squares for the finite difference method and of triangles for the finite elements method To solve resulting systems of ordinary differential equations Runge-Kutta-Merson method (see e.g. [10], [11]) was used.

3.1 FDM based numerical scheme

Let h be mesh size such that $h = L/(N-1)$ for some $N \in \mathbf{N}^+$. We define numerical grid as set $\overline{\omega}_h = \{(ih, jh) \mid i = 0, \dots, N-1, j = 0, \dots, N-1\}$. For function $u : \mathbf{R}^2 \to \mathbf{R}$ we define a projection on $\overline{\omega}_h$ as $u_{ij} = u\,(ih, jh)$. We introduce finite differences $u_{x_1,ij} = (u_{i+1,j} - u_{i,j})/h$, $u_{\overline{x}_1,ij} = (u_{i,j} - u_{i-1,j})/h$ $u_{x_2,ij} = (u_{i,j+1} - u_{i,j})/h$, $u_{\overline{x}_2,ij} = (u_{i,j} - u_{i,j-1})/h$, and define approximation Δ_h of the Laplace operator Δ as $\Delta_h u_{ij} = u_{\overline{x}_1 x_1,ij} + u_{\overline{x}_2 x_2,ij}$. Then semi-discrete scheme has the following form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} u_{ij}(t) &= a\Delta_h u_{ij} + F(1-u_{ij}) - u_{ij}v_{ij}^2, \\ \frac{\mathrm{d}}{\mathrm{d}t} v_{ij}(t) &= b\Delta_h v_{ij} - (F+k)v_{ij} + u_{ij}v_{ij}^2, \end{aligned} \tag{1}$$

plus discrete initial and boundary conditions. This system is solved by Runge-Kutta-Merson method.

3.2 FEM based numerical scheme

To induce the semi-discrete scheme we begin with variational formulation of the problem defined in Sect. 2 and rewrite it in weak form. Then we proceed to discretize this problem in space. Let $\mathcal{T}_h$ be a partition of domain Ω into disjoint triangles τ. At vertices P_j of $\mathcal{T}_h$ we define pyramid functions $\Phi_1, \dots, \Phi_{N_h}$, $\Phi_i(P_j) = \delta_{ij}$ and define finite dimensional space $S_h \subset H^1(\Omega)$ to be spanned by these functions. We search for Galerkin approximation u_h, v_h of weak solution in S_h thus $u_h = \sum_1^{N_h} \alpha_j \Phi_j$ and $v_h = \sum_1^{N_h} \beta_j \Phi_j$. Real-valued functions α_j, β_j are solution of Galerkin approximation problem

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}(u_h, \varphi_1) + a(\nabla u_h, \nabla\varphi) &= (f_1, \varphi),\ \forall\varphi \in S_h \\ \frac{\mathrm{d}}{\mathrm{d}t}(v_h, \varphi_2) + b(\nabla v_h, \nabla\varphi) &= (f_2, \varphi),\ \forall\varphi \in S_h \end{aligned}$$

with $u_h|_{t=0} = u_{ini,h}$, $v_h|_{t=0} = v_{ini,h}$, where $f_1(u,v) = F(1-u) - uv^2$, $f_2(u,v) = -(F+k)v + uv^2$ and $(\cdot,\cdot)$ denote the L_2 inner product. Substituting basis functions $\Phi_1, \dots, \Phi_{N_h}$ instead of arbitrary functions $\varphi \in S_h$ we get system of $2N_h$ ODEs with initial conditions

$$\begin{aligned} A\dot{\alpha(t)} + aB\alpha(t) &= C(\alpha(t), \beta(t)), \\ A\dot{\beta(t)} + bB\beta(t) &= E(\alpha(t), \beta(t)), \\ A\alpha(0) &= D, \\ A\beta(0) &= F. \end{aligned} \tag{2}$$

Lagrange interpolation was used for numerical integration to get approximation of vectors C, E, D, F. Using method of lumped masses and renumbering unknowns we can rewrite the problem for finding functions α_j, β_j in the following form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} u_{ij}(t) &= \frac{2a}{3h^2}[u_{i+1,j} + u_{i+1,j+1} + u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + \\ &\quad + u_{i-1,j+1} - 6u_{ij}] + F(1-u_{ij}) - u_{ij}v_{ij}^2 \\ \frac{\mathrm{d}}{\mathrm{d}t} v_{ij}(t) &= \frac{2b}{3h^2}[v_{i+1,j} + v_{i+1,j+1} + v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + \\ &\quad + v_{i-1,j+1} - 6v_{ij}] - (F+k)v_{ij} + u_{ij}v_{ij}^2 \end{aligned} \tag{3}$$

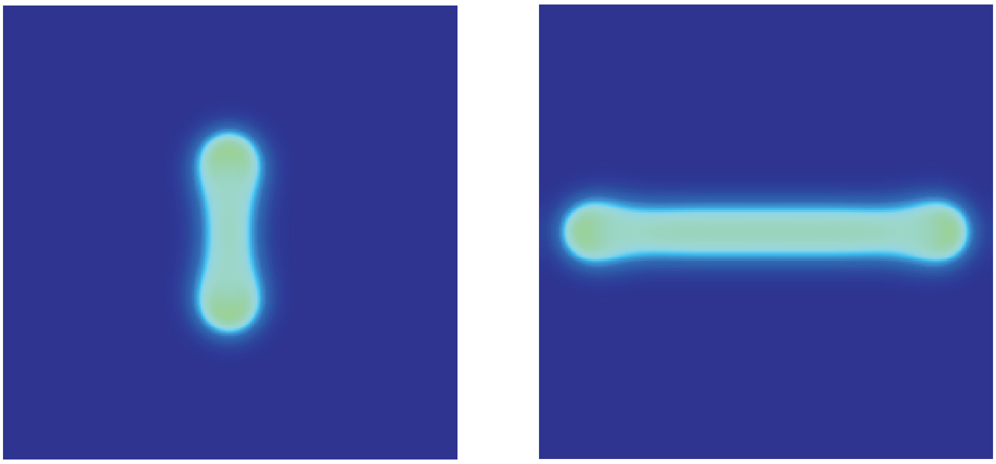
plus corresponding initial and boundary conditions. This system is solved by Runge-Kutta-Merson method.

4 Numerical simulations

We performed a series of computations to perform quantitative comparison of our 2D numerical schemes. According to our results the Gray-Scott model is sensitive on the mesh parameter size, which means, the numerical solution may change notably when refining the computational grid.

In our numerical simulations we use square domain $\Omega \equiv (0.0, 0.5) \times (0.0, 0.5)$. Initial data are considered such that $u_{ini} + v_{ini} = 1$ hold within the computational domain Ω and v_{ini} consists of one or several spots.

We met initial data and model parameter values combinations for which the following situations occurred. First, we have results where FDM based numerical scheme is less dependent on the space stepping, that is, the numerical results are visually more similar in wider range of mesh parameter sizes then in case of the FEM based numerical scheme. We have also results, where the FEM based numerical scheme is less dependent on the space stepping. We were able to see agreement in numerical results obtained in both of these cases from certain mesh parameter size. But we have also met combinations where we were not able to see the solutions becoming visually more and more similar while refining the numerical grid. Example is given below in Fig. 1.



(a), FDM, grid 800×800

(b), FEM, grid 800×924

Figure 1: Dependence of pattern in numerical solution on numerical scheme and grid size for given model parameters ($a = 2 \cdot 10^{-5}$, $b = 1 \cdot 10^{-5}$, $F = 0.0737$, $k = 0.061882$, $L = 0.5$) and initial data (one spot in the middle of the domain) at fixed time $t = 8000$.

In Fig. 1 we demonstrate the case, where we were not able to obtain agreement of numerical results by the numerical schemes from Sect. 3. Using the same model parameters and initial data we could see lines growing in orthogonal directions. Depicted are solutions at time $t = 8000$. Each of numerical schemes provided the same pattern within wide range of grid sizes. We tried to successively refine the numerical grid up to 2000×2000 and corresponding size of triangle grid. For the same model parameters as in Fig. 1 an agreement of numerical results was observed for different initial data.

5 Conclusion

We focused on quantitative comparison of two numerical schemes which solve the Gray-Scott model in 2D. Our numerical simulations show that for certain combinations of initial data and model parameter values we may not get an agreement of numerical results provided by these numerical schemes while refining the numerical grid. Example result is given.

Acknowledgement: The work has been performed under the Project HPC-EUROPA (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action under the FP6 "Structuring the European Research Area" Programme. Partial support of the project of "Necas Center for Mathematical modeling", No. LC06052 and of the project "Applied Mathematics in Physics and Technical Sciences", No. MSN6840770010 of the Ministry of Education, Youth and Sports of the Czech Republic is acknowledged.

References

[1] P. Gray, S.K. Scott: *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and instabilities in the system $A + 2B \rightarrow 3B$, $B \rightarrow C$.* In: Chem. Eng. Sci., 39, 1087–1097, 1984.

[2] J. Wei: *Pattern formation in two-dimensional Gray-Scott model: existence of single-spot solutions and their stability.* In: Physica D, 148, 20–48, 2001.

[3] J. Wei, M. Winter: *Asymmetric spotty patterns for the Gray-Scott model in* $\mathbf{R}^2$. In: Stud. Appl. Math., 110(1), 63–102, 2003.

[4] Y. Nishiura, D. Ueyama: *Self-Replication, Self-Destruction, and Spation-Temporal Chaos in the Gray-Scott model.* In: Forma, 15, 281–289, 2000.

[5] Y. Nishiura, D.Ueyama: *Spatio-temporal chaos for the Gray-Scott model.* In: Physica D, 150, 137–162, 2001.

[6] F. Dkhil, E. Logak, Y. Nishiura: *Some analytical results on the Gray-Scott model.* In: Asymptotic Analysis, 39, IOS Press, 225–261, 2004.

[7] A. Doelman et al.: *Pattern formation in the one-dimensional Gray-Scott model.* In: Nonlinearity, 10, 523–563, 1997.

[8] W. Mazin, K.E. Rasmussen, E. Mosekilde et al.: *Pattern formation in the bistable Gray-Scott model.* In: Mathematics na Computers in Simulations, 40, Elsevier Science, 371–396, 1996.

[9] J.S. McGough, K. Riley: *Pattern Formation in the Gray-Scott model.* In: Nonlinear Analysis: Real World Application, 5, Elsevier Science, 105–121, 2004.

[10] J. Šembera, M. Beneš: *Nonlinear Galerkin Method for Reaction-Diffusion Systems Admitting Invariant Regions.* In: Journal of Computational and Applied Mathematics, 136(1-2), 163–176.

[11] V. Minárik, J. Kratochvíl, K. Mikula, M. Beneš: *Numerical simulation of dislocation dynamics.* In: Numerical Mathematics and Advanced Applications, ENUMATH 2003 (peer reviewed proceedings), Springer Verlag, 631–641, 2004.

Effect of hydrodynamic mixing on the photosynthetic microorganism growth: Revisited

Š. Papáček, C. Matonoha

Institute of Physical Biology, University of South Bohemia, Nové Hrady
Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague

1 Introduction

Biotechnology with microalgae and photo-bioreactor (PBR) design is nowadays regaining attention thanks to emerging projects of CO_2 sequestration and algae biofuels. Nevertheless, there do not exist reliable methods as well as programming software neither for modeling, simulation and control of microbial growth in photo-bioreactors, nor for PBR design [3]. Modeling in a predictive way the photosynthetic response in the three-dimensional flow field seems today unrealistic, because the global response depends on numerous interacting intracellular reactions, with various time-scales. The physiological state of any cellular system and its impact on growth and product formation is the result of a complex interplay between the extracellular environment and the cellular machinery. The design of PBR in which microalgae cells function as factories as well as the prediction of suitable PBR operating conditions is further complicated because of the dynamic variations of the extracellular environment.

Our main goal is to develop and implement the mathematical model of microalgae growth in a general PBR as tool in the design of photo-bioreactors and the optimization of their performance. In our previous works we studied an adequate multi-scale lumped parameter model which well describes the principal physiological mechanisms in microalgae: photosynthetic light-dark reactions and photoinhibition [5], as well as its model parameter estimation [8, 7]. In [6] we presented how to construct a distributed parameter model consisting mainly in determination of hydrodynamic dispersion coefficient as function of space coordinates.

This paper deals with the non-homogeneous steady-state one-dimensional reaction-diffusion system (3) with a special boundary condition. However, equation (3) is rewritten in form of two ordinary differential equations (ODE), which leads after re-scaling to the standard form of the singularly perturbed system [4]. The purpose of such an operation is to infer the asymptotic properties of the reaction-diffusion system (3).

2 Modelling photosynthetic microorganism growth

The photosynthetic microorganism growth description is usually based on the so-called microbial kinetics, i.e. on the lumped parameter models (LPM) describing the photosynthetic response in small cultivation systems with a homogeneous light distribution [9]. However, there is an important phenomenon, the so-called flashing light enhancement, which demands some other model than it residing in the artificial connection between the steady state kinetic model and the empiric one describing the photosynthetic productivity under fluctuating light condition. Nevertheless, even having an adequate dynamical LPM of microorganism growth, see e.g. phenomenological model of so-called photosynthetic factory [5, 8], another serious difficulty resides in the description of the microalgal growth in a PBR, i.e. in a distributed parameter system.

In order to develop the distributed parameter model (DPM) of a microorganism growth, two main approaches for transport and bioreaction processes modelling are usually chosen: (i) Eulerian infinitesimal, and (ii) Eulerian multicompartmental. While the Eulerian infinitesimal approach, leading to the partial differential equations (PDE), is an usual way to describe transport and reaction systems, the multicompartmental modelling framework, resulting in an ODE system, is mostly used in the process engineering area. This second approach, based on balance equation among compartments with finite control volume, has been recently treated by Bezzo *et al.* [2]. The authors presented there a rigorous mathematical framework for constructing *hybrid multicompartment/CFD models*. *Hybrid* there means that the fluid flow description is resolved by a CFD code, and does not make a part of the ODE system of governing equations.

In the sequel, we adopt the first approach aiming to clarify in an analytical manner the role of hydrodynamic mixing, or more precisely, the mechanism of the photosynthetic microorganism growth enhancement due to the microbial cell transport by radial dispersion. Nevertheless, in the future work, our results should serve to develop a numerical scheme for setting up the optimal compartment size in the multicompartment/CFD models.

3 Model development

Transport equation for microbial cells (concentration c) as the function of spatial coordinates and time gets the next form [1]:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{v}c) - \nabla \cdot (D_e \nabla c) = R(c) \ , \tag{1}$$

where $R(c)$ is the source term (representing microbial growth, unit: cell m^{-3}s^{-1}), $\mathbf{v}$ represents the velocity field, and D_e is the dispersion coefficient, which corresponds to diffusion coefficient in microstructure description and becomes mere empirical parameter suitably describing mixing in the system. D_e is influenced by the molecular diffusion and velocity profile. When mixing is mainly caused by the turbulent micro-eddies, the phenomenon is called the turbulent diffusion and a *turbulent diffusion coefficient* is introduced e.g. in [1]. The reaction obviously depends on some variables, usually called as substrates. For our special case of photosynthetic growth in a PBR, the role of only one limiting substrate (the nutrients are supposed to be present in a sufficient amount, i.e. they do not limit the growth) fulfills the irradiance, in other words, an external forcing input u. Moreover we suppose the rectangular PBR geometry illuminated from one side, i.e. the irradiance level is decreasing from the PBR wall to PBR core. Thus, the PBR volume (our computational domain) can be divided into layers with the same irradiance level, transforming the 3D problem into the one-dimensional. Consequently, the description of cell motion in direction of light gradient, i.e. perpendicular to PBR wall and at the same time perpendicular to the direction of convective flow, is of most interest. This motion is caused by the just mentioned turbulent diffusion. Furthermore, we can introduce the dimensionless spatial coordinate x, and the dimensionless dispersion coefficient $p(x)$ by

$$r := xL \ , \ D_e := p(x) \ D_0 \ ,$$

where L and D_0 (unit: m^{2}s^{-1}) are the PBR length in direction of light gradient, and a constant with some characteristic value, respectively.

Furthermore we introduce the dimensionless concentrations c and c_{ss} as

$$y := \frac{c}{c_m} \ , \ y_{ss} := \frac{c_{ss}}{c_m} \ ,$$

where c_m is a characteristic (e.g. maximal) concentration of c.

Based on the photosynthetic factory model [5, 8] we have for the reaction term R the relation

$$R(c) = -k\ (c - c_{ss})\ , \tag{2}$$

where k is the rate (unit: s^{-1}) associated with the dynamic process by which is the concentration c approaching to some value c_{ss} depending only on the external input $u(x)$.

As we are interested on the steady state solution of (1), i.e. $\frac{\partial c}{\partial t} = 0$, we finally obtain

$$-\left[p(x)y'\right]' + q(x)\ y = q(x)\ y_{ss},\ \ y'(0) = 0,\ y'(1) = 0\ , \tag{3}$$

where $q(x) := \frac{k(u(x))\ L^2}{D_0}$.

4 Asymptotic properties of the reaction-diffusion system (3)

In the process engineering literature, there exists a concept of well mixed unit. This construct is further used e.g. in the multicompartmental or multizonal models [2, 6]. The crucial question is: When a compartment with finite volume is well mixed? For a reaction-diffusion system, it has to depend on the so-called *Damköhler number*.

In our previous work, in sake of the benchmark problem, we were looking for an analytical solution of the equation (3). Realizing that it was impossible, we did not search the solution in the usual form of $y = y(x)$, but we wanted to find the mean value of y in the interval $x \in [0.1]$, i.e. to compute the expression $\int_0^1 y(x)\ \mathrm{d}x$. Based on [10], the boundary value problem (3) was transformed into the related initial value problem. It consisted in finding solutions of two homogeneous equations, two differential equations with the right-hand side and computing a solution of a system of two algebraic equations. By this procedure, we could have obtained a function value and its derivative in an arbitrary point. The original differential equation with boundary conditions was thus transformed into a differential equation with an initial condition. As we have needed only a solution in several points, we could apply the above procedure repeatedly. Finally, the value $\int_0^1 y(x)\ \mathrm{d}x$ would be obtained by a suitable numerical method.

Now, we are developing an asymptotic method. Let first define $\frac{\mathrm{d}}{\mathrm{d}x}y := z$, then the resulting first order ODE system is

$$\frac{\mathrm{d}}{\mathrm{d}x}y = z\ ,\ \frac{\mathrm{d}}{\mathrm{d}x}\left[p(x)z\right] = q(x)\ (y - y_{ss})\ ,\ z(0) = 0,\ z(1) = 0\ \ . \tag{4}$$

Consequently, if we define k_0 as follows: $k := k_A(u(x))\ k_0$, then the *Damköhler number* of second type could be defined as $Da_{II} := \frac{k_0 L^2}{D_0}$, and the dependence of the solution of (4) on $Da_{II} := \varepsilon \to 0$ could be studied.

The following ODE (5)

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[p(x)z\right] = \varepsilon k_A(u(x))\ (y - y_{ss})\ ,\ z(0) = 0,\ z(1) = 0\ \ , \tag{5}$$

thanks to the properties of its right hand side clearly satisfies the sufficient condition for applying the averaging method [4]. One can therefore approximate (4) as follows (always when $\varepsilon \to 0$):

$$\frac{\mathrm{d}}{\mathrm{d}x}y = z\ ,\ \frac{\mathrm{d}}{\mathrm{d}x}\left[p(x)z\right] = \varepsilon \int_0^1 \left[k_A(u(x))\ (y - y_{ss})\right] \mathrm{d}x\ ,\ z(0) = 0,\ z(1) = 0\ \ . \tag{6}$$

Acknowledgement: This work was supported by the grant MŠMT MSM 600 766 58 08 of the Ministry of Education, Youth and Sports of the Czech Republic, and by the Grant Agency of Czech Republic through the research grant No. 102/05/0011.

References

[1] W.J. Beek, K.M.K. Muttzall, J.W. van Heuven: *Transport Phenomena.* Wiley & Sons, 2000.

[2] F. Bezzo, S. Macchietto, C.C. Pantelides: *Computational issues in hybrid multizonal/computational fluid dynamics models.* AIChE Journal, 51, 1169–1177, 2005.

[3] M. Janssen, J. Tramper, L.R. Mur, R.H. Wijffels: *Enclosed Outdoor Photobioreactors: Light Regime, Photosynthetic Efficiency, Scale-Up, and Future Prospects.* Biotechnology and Bioengineering, 81, 193–210, 2003.

[4] H.K. Khalil: *Perturbation and averaging.* Nonlinear systems. Prentice Hall, 2002.

[5] Š. Papáček, S. Čelikovský, D. Štys, J. Ruiz-León: *Bilinear system as modelling framework for analysis of microalgal growth.* Kybernetika, 43, 1–20, 2007.

[6] Š. Papáček, D. Štys, P. Dolínek, K. Petera: *Multicompartment/CFD modelling of transport and reaction processes in Couette-Taylor photobioreactor.* Applied and Computational Mechanics, 1, 577–586, 2007.

[7] Š. Papáček, S. Čelikovský, B. Rehák, D. Štys: *Experimental design for parameter estimation of two time-scale model of photosynthesis and photoinhibition in microalgae.* Math. Comput. Simul., doi: 10.1016/j.matcom.2009.06.033, 2009.

[8] B. Rehák, S. Čelikovský, Š. Papáček: *Model for Photosynthesis and Photoinhibition: Parameter Identification Based on the Harmonic Irradiation O_2 Response Measurement.* IEEE Transactions on Circuits and Systems I: Regular Papers, Special Issue: 101–108.

[9] K. Schugerl, K.-H. Bellgardt: *Bioreaction Engineering, Modeling and Control.* Springer-Verlag, Berlin Heidelberg, 2000.

[10] E. Vitásek: *Numerické metody*, Nakladatelství technické literatury, Praha, 1987.

Modelling of the airflow through vocal folds

J. Prokopová, M. Feistauer, V. Kučera, J. Horáček

Charles University, Faculty of Mathematics and Physics, Prague
Institute of Thermomechanics, Academy of Sciences of the Czech Republic, Prague

1 Introduction

The simulation of compressible flow in time dependent domains plays an important role in several areas of human activities, for example development of aircrafts and turbines, civil engineering, car industry or medicine. The presented work introduces implementation of numerical techniques like the ALE (Arbitrary Lagrangian-Eulerian) methods and the discontinous Galerkin finite element methods in these domains and creates the bases of the further work in the direction of fluid-structure interaction. We are specially interested in the medical apllications of this type of problem. For this reason the problem of the airflow through vocal folds is treated.

2 Continuous problem

We deal with compressible flow in a bounded domain $\Omega_t \subset I\!R^2$ depending on time $t \in [0,T]$. We assume that the boundary of Ω_t consists of three disjoint parts $\partial\Omega_t = \Gamma_I \cup \Gamma_O \cup \Gamma_{W_t}$, where Γ_I and Γ_O represent the inlet and outlet and Γ_{W_t} represents moving impermeable walls.

We consider the Navier-Stokes equations in the conservative form [1]:

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{2} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = \sum_{s=1}^{2} \frac{\partial \boldsymbol{R}_s\left(\boldsymbol{w}, \nabla \boldsymbol{w}\right)}{\partial x_s} \text{ in } \Omega_t,\ t \in [0,T]\,, \tag{1}$$

where

$$\begin{aligned}
&\boldsymbol{w} = (\rho, \rho v_1, \rho v_2, E)^T \in I\!R^4,\\
&\boldsymbol{f}_s(\boldsymbol{w}) = (\rho v_s, \rho v_1 v_s + \delta_{1s} p, \rho v_2 v_s + \delta_{2s} p, (E+p) v_s)^T,\ s = 1,2,\\
&\boldsymbol{R}_s\left(\boldsymbol{w}, \nabla \boldsymbol{w}\right) = (0, \tau_{s1}, \tau_{s2}, \tau_{s1} v_1 + \tau_{s2} v_2 + k \frac{\partial \theta}{\partial x_s})^T,\ s = 1,2,\\
&\tau_{ij} = \lambda \delta_{ij} \mathrm{div} \boldsymbol{v} + 2\mu d_{ij}(\boldsymbol{w}),\ d_{ij}(\boldsymbol{w}) = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right),\ i,j = 1,2.
\end{aligned}$$

We use the following notation: ρ – density, p – pressure, E – total energy, (v_1, v_2) – velocity vector, θ – absolute temperature, $c_v > 0$ – specific heat at constant volume, $\gamma > 1$ – Poisson adiabatic constant, $\mu > 0, \lambda$ – viscosity coefficients, $k > 0$ – heat conduction coefficient. We set $\lambda = -2\mu/3$. System (1) is completed by the thermodynamical relations

$$p = (\gamma - 1)(E - \rho\, |\boldsymbol{v}|^2/2), \quad \theta = \left(\frac{E}{\rho} - \frac{1}{2}\,|\boldsymbol{v}|^2\right)/c_v, \tag{2}$$

and equipped with inicial condition $\boldsymbol{w}(\boldsymbol{x},0)=\boldsymbol{w}^0(\boldsymbol{x}),\ \boldsymbol{x}\in\Omega_t$ and boundary conditions:

$$\begin{array}{rl}
\text{Inlet } \Gamma_I: & \rho|_{\Gamma_I\times(0,T)}=\rho_D, \\
& \boldsymbol{v}|_{\Gamma_I\times(0,T)}=\boldsymbol{v}_D=(v_{D1},v_{D2})^T, \\
& \displaystyle\sum_{j=1}^{2}\left(\sum_{i=1}^{2}\tau_{ij}n_i\right)v_j+k\frac{\partial\theta}{\partial\boldsymbol{n}}=0 \quad \text{on } \Gamma_I\times(0,T); \\
\text{Moving wall } \Gamma_W: & \boldsymbol{v}_{\Gamma_W\times(0,T)}=\boldsymbol{z}_D, \quad \dfrac{\partial\theta}{\partial\boldsymbol{n}}=0; \\
\text{Outlet } \Gamma_O: & \displaystyle\sum_{i=1}^{2}\tau_{ij}n_i=0, \quad \frac{\partial\theta}{\partial\boldsymbol{n}}=0\ j=1,2.
\end{array}$$

Here $\boldsymbol{z}_D$ is the velocity of the moving wall.

3 ALE formulation

The time dependence of the domain is taken into account with the aid of a regular one-to-one ALE mapping (cf. [3])

$$\mathcal{A}_t:\bar{\Omega}_0\longrightarrow\bar{\Omega}_t,\ \text{i.e. } \mathcal{A}_t:\boldsymbol{X}\longmapsto\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{X},t)=\mathcal{A}_t(\boldsymbol{X}). \tag{3}$$

We define the ALE velocity:

$$\begin{aligned}
\tilde{z}(\boldsymbol{X},t)&=\frac{\partial}{\partial t}\mathcal{A}_t(\boldsymbol{X}), \quad t\in[0,T],\ \boldsymbol{X}\in\Omega_0, \\
\boldsymbol{z}(\boldsymbol{x},t)&=\tilde{z}(\mathcal{A}_t^{-1}(\boldsymbol{x}),t), \quad t\in[0,T],\ \boldsymbol{x}\in\bar{\Omega}_t
\end{aligned} \tag{4}$$

and the ALE derivative of a function $f=f(\boldsymbol{x},t)$ defined for $\boldsymbol{x}\in\Omega_t$ and $t\in[0,T]$:

$$\frac{D^A}{Dt}f(\boldsymbol{x},t)=\frac{\partial\tilde{f}}{\partial t}(\boldsymbol{X},t), \quad \text{where } \tilde{f}(\boldsymbol{X},t)=f(\mathcal{A}_t(\boldsymbol{X}),t),\ \boldsymbol{X}\in\Omega_0. \tag{5}$$

By the chain rule,

$$\frac{D^Af}{Dt}=\frac{\partial f}{\partial t}+\boldsymbol{z}\cdot\nabla f=\frac{\partial f}{\partial t}+\operatorname{div}(\boldsymbol{z}f)-f\operatorname{div}\boldsymbol{z}. \tag{6}$$

This leads us to the ALE form of the Navier-Stokes equations:

$$\frac{D^{\mathcal{A}}\boldsymbol{w}}{Dt}+\sum_{s=1}^{2}\frac{\partial\boldsymbol{g}_s(\boldsymbol{w})}{\partial x_s}+\boldsymbol{w}\mathrm{div}\boldsymbol{z}=\sum_{s=1}^{2}\frac{\partial\boldsymbol{R}_s(\boldsymbol{w},\nabla\boldsymbol{w})}{\partial x_s}, \tag{7}$$

where $g_s(\boldsymbol{w})=f_s(\boldsymbol{w})-z_s\boldsymbol{w},\ s=1,2.$

4 Discretization

The problem is discretized in space by the discontinuous Galerkin finite element method ([1], [2]) using piecewise polynomial approximations of the components of the state vector $\boldsymbol{w}$, in general discontinuous on interfaces between neighbouring elements from a triangulation of the polygonal approximation Ω_{ht} of the domain Ω_t. In this work we use the simplest variant of the discontinuous

Galerkin finite element method for solving Navier-Stokes equations, the *incomplete interior penalty Galerkin* (IIPG) scheme.

Scheme obtained by space discretization by the discontinuous Galerkin finite element method represents a system of ordinary differential equations, which must be discretized with respect to time. We use the method developed in [2] for inviscid flow. A backward Euler method is used for a discretization of the ALE derivative

$$\frac{D^{\mathcal{A}}\boldsymbol{w}_h}{Dt}(\boldsymbol{x}, t_{k+1}) \approx \frac{\boldsymbol{w}_h^{k+1}(\boldsymbol{x}) - \hat{\boldsymbol{w}}_h^k(\boldsymbol{x})}{\tau_k}, \quad \boldsymbol{x} \in \Omega_{ht_{k+1}}, \tag{8}$$

where $\hat{\boldsymbol{w}}_h^j(\boldsymbol{x}) = \boldsymbol{w}_h^j\left(\mathcal{A}_{t_j}\left(\mathcal{A}_{t_{k+1}}^{-1}\right)(\boldsymbol{x})\right)$, $\boldsymbol{x} \in \Omega_{ht_{k+1}}$ and $\boldsymbol{w}_h^j(\boldsymbol{x}) = \boldsymbol{w}_h(\boldsymbol{x}, t_j)$. The nonlinear terms in the scheme are linearized with the aid of properties of expressions $\boldsymbol{g}_s$ and $\boldsymbol{R}_s$. This treatment leads to the linear system,which is is solved on each time level by the GMRES method with a block diagonal preconditioning.

5 Numerical experiments

We consider compressible flow in the channel, whose shape is inspired by a shape of vocal folds and supraglottal spaces as shown in Figure 1. The lower channel wall between the points A and D is changing the shape according to the given function of time and axial coordinate:

$$\begin{aligned} y(x,t) &= (a_1 + a_t)\left[\sin\left(\frac{3\pi}{2} + \pi\frac{x - x_A}{x_C - x_A}\right) + 1\right] + d, \ x \in [x_A, x_C], \\ y(x,t) &= 2(a_1 + a_t)\cos\left(\frac{\pi}{2}\frac{x - x_C}{x_D - x_C}\right) + d, \ x \in [x_C, x_D], \\ a_t &= a_2 \sin(2\pi f t), \ t \in [0,T]; \ a_1 = 0.18, \ a_2 = 0.015, \end{aligned} \tag{9}$$

where $f = 5.38 \cdot 10^3$. The motion of the upper wall of the channel is treated in a similar way. This movement is interpolated inside the domain resulting in the ALE mapping $\mathcal{A}_t$.

Figures 2 show streamlines at different time instants $t = 504, 558, 612, 666$ during the fourth period of the motion.

6 Conclusion

Using our program code based on the finite element approximation, we solved the viscous compressible flow in time-dependent domains with shape motivated by the simulation of the airflow

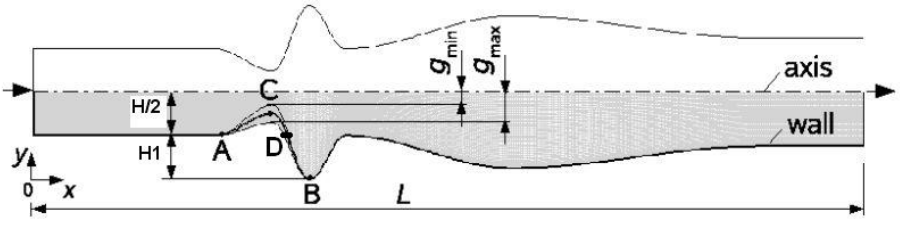


Figure 1: Geometry of computational domain.

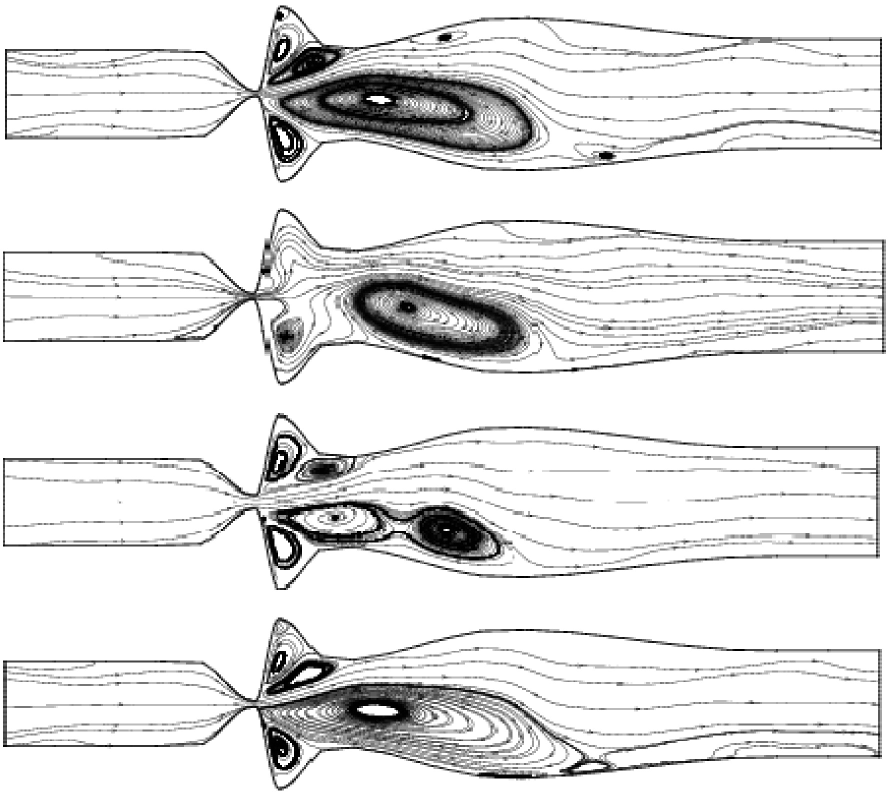


Figure 2: Streamlines at time $t = 504$, 558, 612, 666.

in human vocal folds. The computational results show that it is not possible to simplify the mathematical model supposing the axisymmetry of the solution, because the unsymmetric flow structure is developed in spite the computational domain is axisymmetric.

Future work will be focused on the better approximation of the computational domain to the real geometry of the glottis and the vocal tract and mainly on the application of the fluid-structure interaction consisting in the solution of coupled system describing flow and structure behaviour.

Acknowledgement: This work was supported by the Project OC 09019 "Modelling of voice production based on biomechanics"within the program COST 2103 and by the project No. MSM 0021620839 of the Ministry of Education of the Czech Republic.

References

[1] M. Feistauer, J. Felcman, J. Straškraba: *Mathematical and Computational Methods for Compressible Flow.* Clarendon Press, Oxford, 2003.

[2] M. Feistauer, V. Kučera, J. Prokopová: *Discontinuous Galerkin solution of compressible flow in time dependent domains..* In: Mathematics and Computers in Simulation. In press, doi: 10.1016/j.matcom.2009.01.020.

[3] T. Nomura, T.J.R. Hughes: *An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body.* Comput. Methods Appl. Mech. Engrg. 95, 115–138, 1992.

Algebraic multilevel preconditioning of coarse problems of balanced domain decomposition methods with nodal constraints

I. Pultarová

Department of Mathematics, Faculty of Civil Engineering, CTU in Prague

1 Coarse problems of balanced domain decomposition methods by constraints

A method of balanced domain decomposition by constraints (BDDC) [3] is an iterative algorithm for numerical solution of partial differential equations discretized by the finite element (FE) method. The BDDC method exploits a nonoverlaping partition of a domain, and within each iteration, the main computation consists in solving particular boundary problems on every subdomain and in solving a certain coarse grid problem. There are many ways how to define the coarse base functions. In our considerations, they are defined by nodal values (degrees of freedom, DOFs) in few nodes on interfaces of subdomains and have minimal energy on each subdomain and null normal derivatives on all interfaces. Then in general, the coarse functions are discontinuous on interfaces of subdomains up to the nodes where the coarse DOFs are defined. Since the coarse problem itself can be large, an appropriate preconditioning is desired [2].

2 Algebraic multilevel preconditioning of the coarse problem

We present a new strategy of preconditioning of the coarse problem of BDDC. This is based on an algebraic multilevel (AML) preconditioning technique [1]. In spite of classical application of AML preconditioning directly to finite element bases, we utilize a hierarchical splitting of the coarse space within the BDDC algorithm. A quality of AML preconditioning is measured by the constant γ in the strengthened Cauchy-Buniakowski-Schwarz (CBS) inequality [1].

We provide some numerical estimates of the CBS constants for two- and three-dimensional elliptic problems: an equation of diffusion and an equation of linear elasticity. For discretisation, bilinear or trilinear FEs are used with rectangular or prismatic supports, respectively. The subdomains are of a rectangular or prismatic shape as well. Coarse base functions are defined by all corner nodal values on subdomains. In every problem a hierarchical splitting of the coarse base system is constructed with coefficients that are equal to that for hierarchical transformation of bilinear or trilinear FEs, respectively. In each test we specify only a bilinear form $a(\cdot,\cdot)$ used in a weak formulation of the problem, because neither boundary conditions nor a right hand side of the equation influence the estimates of the CBS constant γ. The estimate of γ can be calculated from exploiting the properties of the coarse function on a single reference subdomain only. A mesh of a reference subdomain may influence the value of γ. Then in each graph, a number of elements in a reference subdomain is indicated. Our main interest is to examine a behavior of γ when varying the coefficients of the bilinear form $a(\cdot,\cdot)$.

Diffusion equations (D2) and (D3). Bilinear form $a(\cdot,\cdot)$ is

$$a(u,v) = \int_\Omega (\nabla u)^T C \nabla v \, \mathrm{d}x,$$

where

$$C = \begin{pmatrix} 1 & c \\ c & d \end{pmatrix} \qquad \text{or} \qquad C = \begin{pmatrix} 1 & c_{12} & c_{13} \\ c_{12} & d_2 & c_{23} \\ c_{13} & c_{23} & d_3 \end{pmatrix}$$

for two- and three-dimensional problems (D2) and (D3), respectively. The values of c, d, d_i and c_{ij} are constant on subdomains and matrix C is positive definite on every subdomain.

The estimates of γ^2 for the case $c = 0$ and $d \in (0, 1\rangle$ are displayed in Figure 1 on the left for five different meshes of a reference subdomain. On the right, values of γ^2 are presented for $d = 1$ and $c \in (-1, 0\rangle$. The estimates of γ^2 for the case $d_2 = 1$, $c_{ij} = 0$ and $d_3 \in (0, 1\rangle$ are shown in the left part of Figure 2 for four different meshes of a reference prismatic subdomain. In the right hand side of Figure 2, values of γ^2 for the case $d_i = 1$ and $c_{12} = c_{13} = c_{23} \in (-0.5, 0\rangle$ are presented.

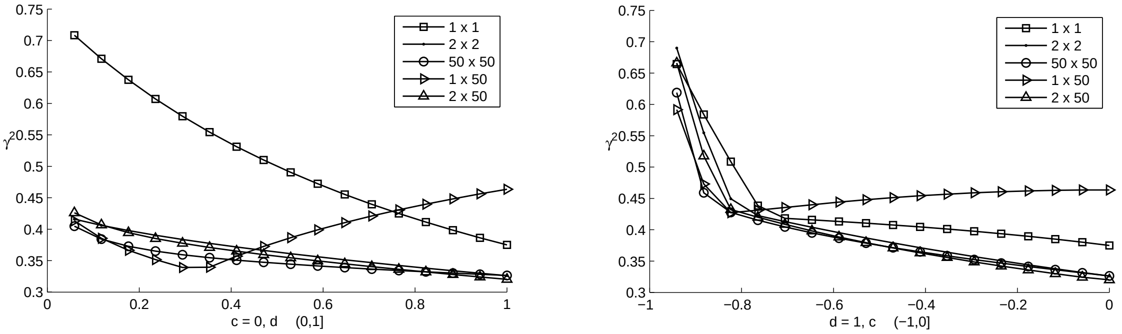


Figure 1: Numerical estimates of γ^2 for hierarchical splitting of coarse problems for equation (D2). Coefficients $c = 0$ and $d \in (0, 1\rangle$ (left), and the case $d = 1$ and $c \in (-1, 0\rangle$ (right).

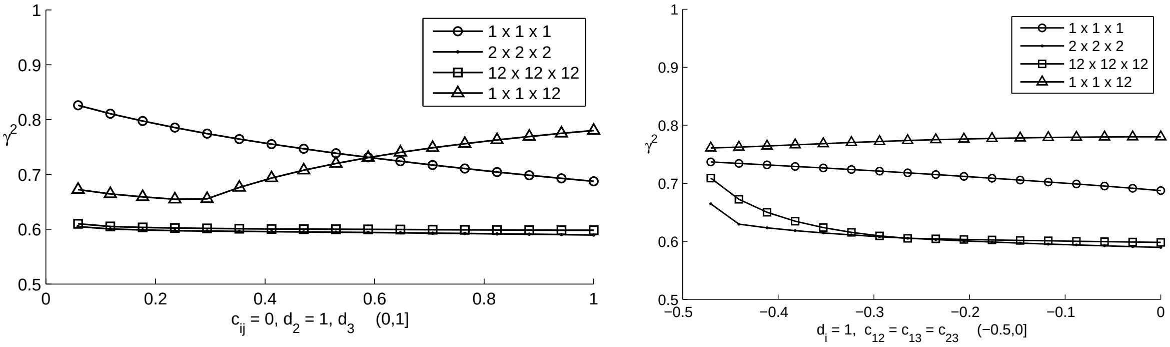


Figure 2: Numerical estimates of γ^2 for hierarchical splitting of coarse problems of BDDC for equation (D3). Coefficient $d_2 = 1$ and $c_{ij} = 0$ and varying $d_3 \in (0, 1\rangle$ (left), and $d_2 = d_3 = 1$ and $c_{12} = c_{13} = c_{23} \in (-0.5, 0\rangle$ (right) for different meshes of a reference subdomain.

Elasticity equations (E2) and (E3). Bilinear form $a(\cdot,\cdot)$ is

$$a(\mathbf{u},\mathbf{v}) = \int_\Omega 2\mu(\nabla^{(s)}\mathbf{u})^T\nabla^{(s)}\mathbf{v} + \lambda \operatorname{div}\mathbf{u}\cdot\operatorname{div}\mathbf{v}\,\mathrm{d}x,$$

where $\nabla^{(s)}\mathbf{u} = \varepsilon(\mathbf{u})$ and

$$\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right).$$

The share modulus of a material μ and the Lamé constant λ can be substituted by the Poisson ratio ν and by the modulus of elasticity E via $\mu = E/2/(1+\nu)$ and $\lambda = E\nu/(1+\nu)/(1-2\nu)$.

The estimates of γ^2 for elasticity equations for $\nu \in \langle 0, 0.5)$ can be found in Figure 3. Two-dimensional cases for partitions of a rectangular subdomain 1×1, 5×5, 50×50 and 1×50, respectively, are shown on the left. A three-dimensional case for five different meshes of a reference prismatic subdomain can be found in the right part of Figure 3.

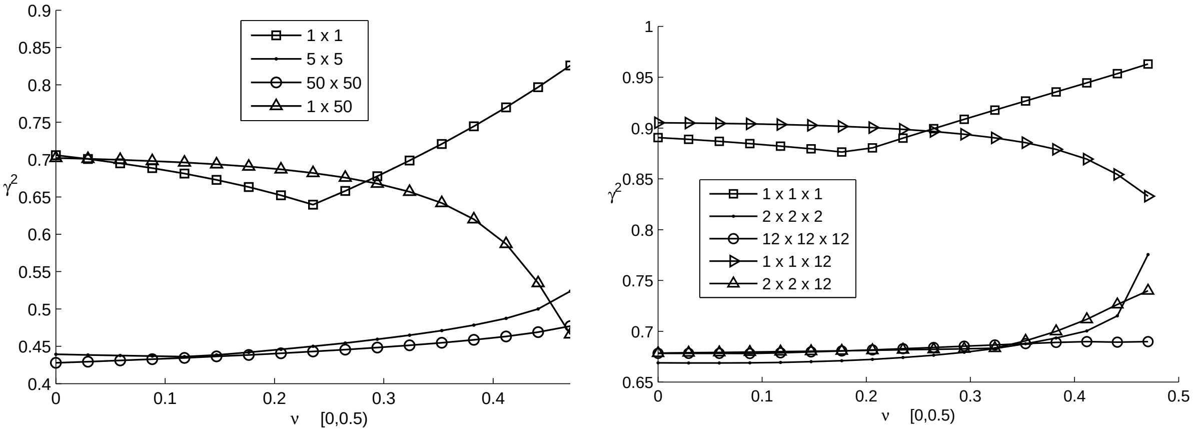


Figure 3: Numerical estimates of γ^2 for hierarchical splitting of the coarse problems for equations (E2) (left) and (E3) (right). Varying Poisson ratio $\nu \in \langle 0, 0.5)$ and different meshes of a reference subdomains.

Acknowledgement: The research was supported by the project CEZ MSM 6840770001 and by the Grant Agency of The Czech Republic, No. 201/09/P500.

References

[1] O. Axelsson: *Iterative Solution Methods.* Cambridge University Press, 1996.

[2] J. Mandel, B. Sousedik, C.R. Dohrmann: *Multispace and Multilevel BDDC.* Computing 83, 55-85, 2008.

[3] Numerical Methods in Computational Mechanics, 2009, Edited by M. Okrouhlík. e-book, `http://www.it.cas.cz/files/u1784/Num_methods_in_CM.pdf`.

Hybrid dynamical systems: verification and error trajectory search

S. Ratschan

Institute of Computer Science
Academy of Sciences of the Czech Republic

Modern complex technical systems usually consist not only of physical components, but also of computer equipment interacting with these physical components. As a consequence, such systems cannot be modeled based on physical laws formulated in the language of continuous mathematics alone. In addition, one needs discrete modeling formalisms. Hybrid (dynamical) systems are a formalism for modeling the resulting combined continuous-discrete behavior. In the talk we will describe this formalism, and discuss algorithms for the automatic analysis of such hybrid systems.

When restricted to their continuous part, such hybrid systems amount to ordinary differential equations, when restricted to their discrete part, to finite state machines. However, the interaction between those two parts introduces significant additional difficulty that obstructs the use of classical numerical error analysis. Moreover, such systems are usually non-deterministic, in the sense that they not only have a single initial state, but a whole set of initial stats, and in the sense that starting from a given unique state, they may allow an uncountable set of trajectories (e.g., resulting from differential inequalities). A further difficulty results from the fact that one is often interested in analyzing the behavior of hybrid systems over an unbounded time horizon.

As a consequence, more or less all problems of analyzing hybrid systems are undecidable [4], although one can get much further with a weaker notion of quasi-decidability [3, 1, 5].

In the talk we will discuss discuss algorithms for proving that a given hybrid system does not reach an element of a set of states considered to be unsafe (in whatever unbounded time) [6, 2]. Moreover, we will discuss the use of optimization techniques to find trajectories the violate this property [7].

Acknowledgement: This work has been supported by GAČR grant 201/08/J020 and by the institutional research plan AV0Z100300504.

References

[1] W. Damm, G. Pinto, and S. Ratschan. Guaranteed termination in the verification of LTL properties of non-linear robust discrete time hybrid systems. *International Journal of Foundations of Computer Science (IJFCS)*, 18(1):63–86, 2007.

[2] T. Dzetkulič and S. Ratschan. How to capture hybrid systems evolution into slices of parallel hyperplanes. In *ADHS'09: 3rd IFAC Conference on Analysis and Design of Hybrid Systems*, pages 274–279, 2009.

[3] M. Fränzle. Analysis of hybrid systems: An ounce of realism can save an infinity of states. In J. Flum and M. Rodriguez-Artalejo, editors, *Computer Science Logic (CSL'99)*, number 1683 in LNCS. Springer, 1999.

[4] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. *Journal of Computer and System Sciences*, 57:94–124, 1998.

[5] S. Ratschan. Safety verification of non-linear hybrid systems is quasi-decidable. 2009. submitted.

[6] S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. *ACM Transactions in Embedded Computing Systems*, 6(1), 2007.

[7] S. Ratschan and J.-G. Smaus. Finding errors of hybrid systems by optimising an abstraction-based quality estimate. In C. Dubois, editor, *Tests and Proofs*, volume 5668 of *LNCS*, pages 153–168. Springer, 2009.

Comparison of the sparse matrix storage formats

I. Šimeček

Faculty of Information Technologies,
Czech Technical University, Prague

1 Introduction

Computations with sparse matrices are widespread in scientific projects. The performance of mathematical operations with sparse matrices depends strongly on the used matrix storage format. In this paper, we compare the performance during the execution of some basic routines from linear solvers and its dependency on the used format.

The paper consists of four parts: a general introduction of sparse matrix storage formats, performance testing, conclusions, and suggestions for future work.

2 Usual sparse matrix formats

In the following text, we assume that A is real sparse matrix of order n. Let n_Z be the total number of nonzero elements in A.

2.1 The coordinate (XY) format

The coordinate (XY) format is the most simplest sparse format. The matrix A is represented by three linear arrays $Elem, X$, and Y. The array $Elem[1, \ldots, n_Z]$ stores the nonzero values of A, arrays $X[1, \ldots, n_Z]$ and $Y[1, \ldots, n_Z]$ contain X- and Y-positions, respectively, of the elements with the nonzero value.

2.2 The compressed sparse row (CSR) format

The most common format (see [4]) for storing sparse matrices is the compressed sparse row (CSR) format. A matrix A stored in the CSR format is represented by three linear arrays $Elem, Addr$, and Ci. The array $Elem[1, \ldots, n_Z]$ stores the nonzero elements of A, the array $Addr[1, \ldots, n]$ contains indexes of initial nonzero elements of rows of A, and the array $Ci[1, \ldots, n_Z]$ contains column indexes of nonzero elements of A. Hence, the first nonzero element of row j is stored at the index $Addr[j]$ in array $Elem$.

2.3 Register blocking formats

Widely-used formats(XY and CSR) are easy to understand, however sparse operations (like matrix-vector or matrix-matrix multiplication) using these formats are slow (mainly due to indirect addressing). Sparse matrices often contain dense submatrices (blocks), so various blocking formats were designed to accelerate matrix operations. Compared to the CSR format, the aim of these formats (like SPARSITY[2] or CARB[3]) is to consume less memory and to allow a better use of registers and the vectorization of the computation. Algorithms using these formats are very fast, because they are tuned for a target architecture.

2.4 Quadtree data format

Quadtree (for details see [6, 5, 1]) is the recursive tree data structure. Such a tree represents a partition of the matrix into submatrices ("nodes" in the graph terminology). There are different types of nodes in the tree. Inner nodes of the quadtree are divided into "Mixed" or "Empty" nodes. Leafs of the quadtree are divided into "Full" or "Empty" nodes.

Great advantages of the quadtree are the following:

- Easy and fast conversion from common sparse matrix storage formats like CSR or XY.
- Modifications (adding or removing nonzero elements) of the quadtree are relatively easy.
- The recursive style of programming and recursive style of storage ("Divide and Conquer" approach) leads to codes with a surprising performance due to the better cache memory utilization.

3 Discussion about formats

3.1 Drawbacks of usual formats

- **XY and CSR formats:** These formats doesn't support fast adding or removing nonzero elements.
- **Register blocking formats:** These formats suffer from a large transformation overhead, are designed only for limited set of operations, doesn't support fast adding or removing nonzero elements.
- **Quadtree data format:** A big drawback of the quadtree structure is a larger control and data overhead compared to standard formats. The standard quadtree implementation leads to a space (and execution) inefficiency. To remove inefficiencies, we use the additional types of leafs: modified versions of the XY and the CSR formats. The modification means that we express all coordinates relatively to the beginning of the submatrix (node). We call "XY" and "CSR" respectively this type of node.

  Our second improvement is the elimination of "Empty" nodes, because they do not contain any useful information. They are simply represented by the NULL pointer.

3.2 Dynamic formats

All mentioned formats (except quadtree) doesn't support fast adding or removing nonzero elements. We must modify previous formats to eliminate this drawback. For example, the implementation of the CSR format is modified in this way: The array $Addr[1, \ldots, n]$ contains pointers to rows of A (separated arrays containing the values of nonzero elements and its column indexes). Since all information are stored in dynamic arrays, we called these modified formats "dynamic".

3.2.1 Combined format

The new "combined" format consists of dynamic CSR and CSC format. So, this format consumes two times more space but it should be fast independently on the memory pattern (row-like or column-like).

4 Evaluation of the results

4.1 Test applications

We have implemented these basic routines from the linear algebra that are often used in linear solvers:

- to get an value at the given location in the sparse matrix (operation GET_XY),
- to set an value at the given location in the sparse matrix (operation SET_XY),
- to find the maximal value in the row in the sparse matrix (operation MAX),
- to add values of given row to other row in the sparse matrix (operation ADD),
- the transposition of the sparse matrix (operation TRANSP),
- the multiplication of a sparse matrix by a dense vector (operation SPMV),

4.2 Test data

We have used 32 real matrices from various technical areas from the MatrixMarket and Harwell sparse matrix test collection.

4.3 HW and SW configuration

All results were measured on Intel Core 2 Quad Q8200 (only one core was used) at 2.33 GHz, 4 GB of the main memory at 400 MHz, running OS Windows XP Professional SP3 with the following cache parameters: L1 cache is 32 KB data cache, L2 cache is 2 MB data.
SW: Microsoft Visual Studio 2003, Intel compiler version 10.1 with switches for maximal performance.

4.4 Measured results

Comments on Table 1:

- The performance of GET_XY and SET_XY operations depend on the ratio $\frac{nz}{n}$, for higher values ("long" rows) can be improved by the binary search method.
- The performance of SET_XY operation is different in the case if the former value of given location is nonzero (the update of the value=hit) or is zero (the new nonzero entry=miss).
- The TRANSP operation in Combined format consists only of exchanging few pointers!

operation/format	CSR	CSC	Combined	Quad
operation GET_XY	$\{0 \ldots ++\}$	$\{0 \ldots ++\}$	$\{0 \ldots ++\}$	$+$
operation SET_XY (hit)	$\{0 \ldots ++\}$	$\{0 \ldots ++\}$	$\{0 \ldots ++\}$	$+$
operation SET_XY (miss)	$\{- \ldots +\}$	$\{- \ldots +\}$	$\{- \ldots 0\}$	$+$
operation MAX	$++$	$-$	$++$	$+$
operation ADD	$++$	$--$	$-$	0
operation TRANSP	0	0	$++$	$+$
operation SPMV	$+$	$-$	$+$	0

Table 1: Performance comparison of the formats. The symbol $--$ denotes very slow, $-$ denotes slow, 0 denotes average, $+$ denotes fast, $++$ denotes very fast.

5 Conclusion

We have tested the performance of some very basic routines used in linear solvers. Measured results satisfy the theoretical assumption that used data format affects strongly the performance. Since every format has some drawbacks and overheads, it is difficult to choose a “winner”, but if the numbers of these operations and the memory pattern are known, we can choose the suitable data storage format.

6 Future works

- We should optimize some routines and deeply measure the performance on various platforms.
- We should develop a new format similar to combined format but with “weak coherency”.

Acknowledgement: This research has been supported by MŠMT under research program MSM6840770014.

References

[1] J.D. Frens, D.S. Wise: *Matrix inversion using quadtrees implemented in gofer.* 1995.

[2] E. Im: *Optimizing the Performance of Sparse Matrix-Vector Multiplication - dissertation thesis.* Dissertation thesis, University of Carolina at Berkeley, 2001.

[3] I. Šimeček: *A new format for sparse matrix-vector multiplication.* In: Seminar on Numerical Analysis, Ostrava, Ústav geonomy AV ČR, 101–104, 2007.

[4] I. Šimeček: *Performance aspects of sparse matrix-vector multiplication.* Acta Polytechnica, 46(3/2006), 3–8, January 2007.

[5] D.S. Wise: *Matrix algorithms using quadtrees* (invited talk). In: ATABLE-92, 11–26, 1992.

[6] D.S. Wise: *Ahnentafel indexing into morton-ordered arrays, or matrix locality for free.* In: Euro-Par 2000 Parallel Processing, volume 1900 of Lecture Notes in Computer Science, 774—783, 2000.

Application of the BDDC method to the Stokes problem

J. Šístek, P. Burda, J. Mandel, J. Novotný, B. Sousedík

Institute of Mathematics, Academy of Sciences of the Czech Republic, Prague
Department of Mathematics, Faculty of Mechanical Engineering,
Czech Technical University in Prague
Department of Mathematical and Statistical Sciences,
University of Colorado Denver, Denver, USA
Department of Mathematics, Faculty of Civil Engineering,
Czech Technical University in Prague

1 Introduction

The Balancing Domain Decomposition based on Constraints (BDDC) is one of the most advanced preconditioners suitable for parallel iterative solution of large systems of linear equations arising from finite element (FE) analysis. The method was introduced by Dohrmann [2] and the theory is due to Mandel and Dohrmann [6]. Li and Widlund reformulated the BDDC method in [5] to a simple global approach. The underlying theory of the BDDC method covers problems with symmetric positive definite matrix. An important application that leads to such kind of systems is structural analysis by linear elasticity theory.

The solution of the incompressible Stokes problem by a mixed finite element method leads to a saddle point system with symmetric indefinite matrix. Thus, the standard theory of BDDC does not cover this important class of problems. In the first attempt to apply BDDC to the incompressible Stokes problem proposed by Li and Widlund [4], the optimal preconditioning properties of BDDC were recovered. The approach is based on the notion of *benign* subspaces, which is restricted to using discontinuous pressure approximations, and the authors present results for piecewise constant approximations. Moreover, the approach in [4] requires quite nonstandard constraints between subdomains, thus making the implementation more problem specific and difficult.

In this paper, we follow a different approach. We have implemented a parallel version of the BDDC method and verified its performance on a number of problems arising from linear elasticity (e.g. [9]). Here, we investigate the applicability of the method and its implementation to the Stokes flow 'as is'. Although it is beyond the standard theory of the BDDC method, contributive results are obtained using only minor changes and minimal amount of custom coding to the implementation for elasticity problems.

It has been known for a long time, that conjugate gradient method is able to reach solution also for many indefinite cases (e.g. [7]), although it may fail in general. Our investigation is also motivated by recent trends of numerical linear algebra to investigate and often prefer the use of PCG method with a suitable indefinite preconditioner over more robust but also more expensive iterative methods for solving saddle point systems such as MINRES, BiCG or GMRES [8].

Results for the Stokes flow in two and three dimensions are presented. All these problems are obtained using mixed discretization by Taylor–Hood finite elements. These elements use piecewise (bi/tri)linear pressure approximation, which does not allow the approach via benign spaces of [4], but are very popular in the computational fluid dynamics community.

2 BDDC domain decomposition method

Let Ω be a bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, let U be a finite element space of piecewise polynomial functions v continuous on Ω and U' its dual space. Let $a(\cdot,\cdot)$ be a bilinear form on $U \times U$ and $f \in U'$, and let $\langle\cdot,\cdot\rangle$ denote the duality pairing of U' and U. Consider an abstract variational problem: *Find $u \in U$ such that*

$$a(u,v) = \langle f,v\rangle \quad \forall\, v \in U\,. \tag{1}$$

For the case of steady Stokes flow we adopt the following slightly unusual notation

$$\begin{aligned} a(u,v) &= \nu \int_\Omega \nabla \mathbf{u}_h : \nabla \mathbf{v}_h \mathrm{d}\Omega - \int_\Omega p_h \nabla\cdot\mathbf{v}_h \mathrm{d}\Omega + \int_\Omega \psi_h \nabla\cdot\mathbf{u}_h \mathrm{d}\Omega, && (2)\\ \langle f,v\rangle &= \int_\Omega \mathbf{f}\cdot\mathbf{v}_h \mathrm{d}\Omega. && (3)\end{aligned}$$

Solution $u = (\mathbf{u}_h, p_h)$ consists of the discretized vector field of velocity and the discretized scalar field of pressure, ν represents the kinematic viscosity of the fluid, $\mathbf{f}$ represents the external load, and $v = (\mathbf{v}_h, \psi_h)$. For the case of Stokes problem, $a(u,v)$ is only symmetric indefinite [3].

Write the matrix problem corresponding to (1) as $Au = f$. The domain Ω is decomposed into N nonoverlapping subdomains Ω_i, $i = 1,...,N$. Each subdomain is a union of several finite elements of the underlying mesh. Unknowns common to at least two subdomains are called *boundary unknowns* and the union of all boundary unknowns is called the *interface* Γ. Let W_i be the space of finite element functions on subdomain Ω_i and put $W = W_1 \times \cdots \times W_N$. It is the space where subdomains are completely disconnected, and functions on them independent of each other. Clearly, $U \subset W$.

The main idea of the BDDC preconditioner in the abstract form is to construct an auxiliary finite dimensional space $\widetilde{W}$ such that

$$U \subset \widetilde{W} \subset W, \tag{4}$$

and extend the bilinear form $a(\cdot,\cdot)$ to a form $\widetilde{a}(\cdot,\cdot)$ defined on $\widetilde{W} \times \widetilde{W}$, such that solving the variational problem (1) with $\widetilde{a}(\cdot,\cdot)$ in place of $a(\cdot,\cdot)$ is cheaper and can be split into independent computations performed in parallel. Then the solution projected to U is used for the preconditioning of (1). Space $\widetilde{W}$ contains functions continuous at selected coarse degrees of freedom such as values at selected nodes called *corners* as well as averages over *edges* or *faces*.

In computation, the corresponding matrix denoted $\widetilde{A}$ is used. It is larger than the original matrix of the problem A, but it possesses a simpler structure suitable for direct solution methods. This is the reason why it can be used as a preconditioner.

The projection $E : \widetilde{W} \to U$ is realized as a weighted average of values from different subdomains at unknowns on the interface Γ followed by the discrete harmonic extension from boundary to interior of each subdomain (see [9]).

Let $r \in U'$ be the residual in an iteration of an iterative method. The BDDC preconditioner $M_{BDDC} : U' \to U$ in the abstract form produces the preconditioned residual $v \in U$ as

$$M_{BDDC} : r \to v = Ew,$$

where $w \in \widetilde{W}$ is obtained as the solution to problem

$$w \in \widetilde{W} : \widetilde{a}(w,z) = (r, Ez) \quad \forall z \in \widetilde{W}, \tag{5}$$

or in terms of matrices as

$$v = E\widetilde{A}^{-1}E^T r. \tag{6}$$

	no	BDDC		ILUT with threshold		
method	preconditioner	corners only	corners+faces	10^{-3}	10^{-4}	10^{-5}
BICGSTAB	n/a	45	22	n/a	331	10
GMRES	759	49	38	472	87	18

Table 1: Number of iterations for BICGSTAB and GMRES without preconditioning, and preconditioned by BDDC and ILU, lid driven cavity.

3 Numerical results

Our parallel implementation of the BDDC preconditioner has been extensively tested on problems with symmetric positive definite matrices arising from linear elasticity (e.g. [9]). The current version is based on the multifrontal massively parallel sparse direct solver MUMPS [1], which is used for factorization of matrix $\widetilde{A}$ in (6). The preconditioned problem is solved by parallel PCG method run on problem (1) reduced to interface Γ by static condensation (see [9] for details). For the Stokes problem, matrix $\widetilde{A}$ is symmetric indefinite and as such is factorized and the problem repeatedly solved by MUMPS representing an indefinite preconditioner.

As the first example, we select a 2D case of lid driven cavity, a popular benchmark problem for methods for viscous flow. The case of uniform mesh of 128 ×128 Taylor–Hood finite elements was chosen. It was divided into 8 subdomains by METIS package. Solution of the problem by our earlier solver based on a serial frontal algorithm took 231 seconds on one 1.5 GHz Intel Itanium 2 processor of SGI Altix 4700 computer in CTU Supercomputing Centre, Prague, compared to 40.5 seconds on 8 processors of the same computer necessary for the solution by the new implementation of BDDC. The stopping criterion of PCG was chosen as $\|r\|_2/\|g\|_2 < 10^{-6}$, resulting in 50 PCG iterations. To investigate the performance of the BDDC preconditioner in combination with standard iterative methods for general matrices, namely BICGSTAB and GMRES, we have also performed several experiments with our serial code written in MATLAB. In Table 1, we compare the resulting number of iterations of these methods preconditioned by BDDC and by the ILU preconditioner for several values of threshold τ for dropping entries in incomplete factorization. Where 'n/a' is present in the table, BICGSTAB failed to converge.

In the second example, a 3D geometry of a channel with a sudden reduction of diameter is considered. Due to the symmetry of the channel, only a quarter of it is considered in the computation. This problem consists of 3 393 Taylor–Hood finite elements with 54 248 unknowns, and it is solved to relative residual $\|r\|_2/\|f\|_2 < 10^{-6}$ by 33 PCG iterations. Division into 4 subdomains obtained by METIS and solution at Reynolds number 100 are presented in Figure 1.

4 Conclusion

We present a parallel implementation of the BDDC preconditioner and explore its applicability to problems with indefinite matrices, namely the Stokes problem. Although the available theory either does not cover this case, or treats it differently [4], the presented experiments suggest promising ways for this effort. We have performed several experiments, for which PCG was successfully used although the system was indefinite. However the reason why a breakdown was not observed deserves further investigation. Our serial experiments also led to promising results for combination of BDDC method with standard iterative methods for solving systems with general matrices, such as BICGSTAB and GMRES.

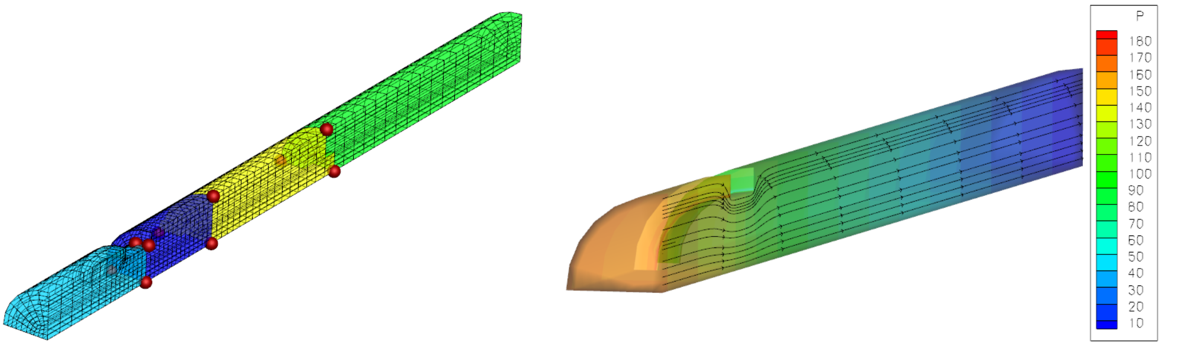


Figure 1: Division of mesh into 4 subdomains (left) and pressure with streamlines (right) for channel with sudden reduction of diameter, only a quarter of the channel is considered for symmetries.

Acknowledgement: This research has been supported by National Science Foundation under grant DMS-0713876, by Czech Science Foundation under grant GA CR 106/08/0403, and by the Grant Agency of the Academy of Sciences of the Czech Republic under grant IAA200600801. A part of this work was done during Jakub Šístek's visit at the University of Colorado Denver.

References

[1] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent: *Multifrontal parallel distributed symmetric and unsymmetric solvers.* Comput. Methods Appl. Mech. Engrg. 184 501–520, 2000.

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

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

[4] J. Li, O. Widlund: *BDDC algorithms for incompressible Stokes equations.* SIAM J. Numer. Anal. 44(6), 2432–2455, 2006.

[5] J. Li, O.B. Widlund: *FETI-DP, BDDC, and block Cholesky methods.* Internat. J. Numer. Methods Engrg. 66(2), 250–271, 2006.

[6] J. Mandel, C.R. Dohrmann: *Convergence of a balancing domain decomposition by constraints and energy minimization.* Numer. Linear Algebra Appl. 10(7) 639–659, 2003.

[7] C.C. Paige, M.A. Saunders: *Solutions of sparse indefinite systems of linear equations.* SIAM J. Numer. Anal. 12(4), 617–629, 1975.

[8] M. Rozložník, V. Simoncini: *Krylov subspace methods for saddle point problems with indefinite preconditioning.* SIAM J. Matrix Anal. Appl. 24(2), 368–391, 2002.

[9] J. Šístek, J. Novotný, J. Mandel, M. Čertíková, Burda, P.: *BDDC by a frontal solver and stress computation in a hip joint replacement.* Mathematics and Computers in Simulation, in print, available online January 21, 2009, DOI `10.1016/j.matcom.2009.01.002` (2009).

Modelování podzemního proudění jako sdružené úlohy v 3D-2D-1D geometrii se složitou diskretizací

I. Škarydová, M. Hokr

Technická univerzita v Liberci

1 Úvod

V příspěvku se zabýváme modelováním režimu proudění vody v okolí vodárenského tunelu v Bedřichově v Jizerských horách, v podobě sdružené úlohy podzemního a povrchového proudění. Přitom využíváme modelu s 3D-2D-1D geometrií a výpočet provádíme v programu Flow123D [1], vyvíjeném na Technické univerzitě v Liberci. Úloha je náročná na diskretizaci kvůli specifickému tvaru modelu (lokality), který je charakterizován velkým poměrem mezi vertikálními a horizontálními rozměry a přítomností tunelu s velmi malým průměrem proti rozměrům úlohy.

Úlohu řešíme v rámci projektu Decovalex, který se zabývá simulací termo-hydro-mechano-chemických (THMC) procesů za účelem analýzy hlubinného úložiště a bezpečného ukládání jaderného odpadu. Spojitost zde nalézáme v podobném charakteru lokality z hlediska přírodních podmínek, s obdobnými tunely v horninovém masivu se počítá i v hlubinném úložišti.

2 Popis modelu

Úloha je definována jako potenciálové proudění (parabolické parciální diferenciální rovnice 2.řádu) v oblasti, která je kombinací oblastí různých dimenzí $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ (tzv. „multidimenzionální model“, podrobněji v [2]), vyjádřené jako systém Darcyho zákona a rovnice kontinuity

$$\mathbf{u_i} = -K_i \nabla p_i \qquad \text{na } \Omega_i\,,\ i = 1, 2, 3 \tag{1}$$

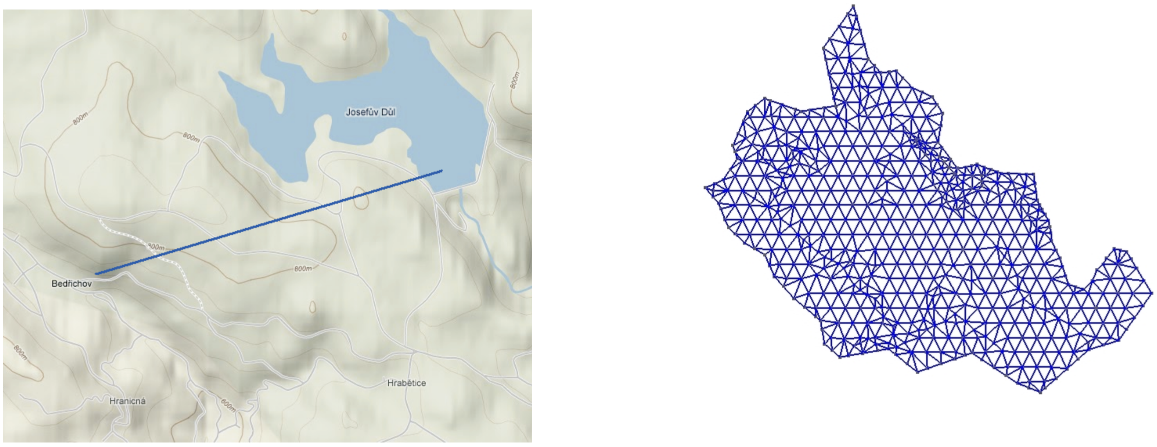
$$\kappa_i \frac{\partial p_i}{\partial t} - \nabla \cdot \mathbf{u_i} = q_i \qquad \text{na } \Omega_i\,,\ i = 1, 2, 3 \tag{2}$$

kde $\mathbf{u_i}$ [m.s^{-1}] je tzv. darcyovská rychlost (odpovídá plošné hustotě toku), K_i [m.s^{-1}] hydraulická vodivost a p_i [m] piezometrická výška, κ [m^{-1}] specifická storativita, q_i [s^{-1}] zdroje t [s] čas [3]. Na částech hranice oblasti předepisujeme standardní Dirichletovy a Neumannovy okrajové podmínky. Úloha je numericky řešena pomocí smíšené-hybridní metody konečných prvků, která umožňuje přirozené zavedení interakce mezi jednotlivými dimenzemi v diskretizované formě [2].

Diskretizační síť modelu je tvořena čtyřstěny (3D oblast podzemní vody v horninovém masivu), 2D trojúhelníkové elementy umístěné na horním povrchu 3D oblasti reprezentují tok po povrchu terénu (srážkové dotace zde vystupují jako zdroje) a 1D liniové elementy potoky a řeky. Použití multidimenzionálního modelu pro sdruženou úlohu povrchového a podzemního proudění zobecňuje původní myšlenku využití 2D prvků ve Flow123D jako reprezentace puklin v hornině, i když model potenciálového proudění je pouze empirickou náhradou fyzikálních řídících rovnic povrchového proudění s omezenou přesností (předpokládáme zde jako dostatečnou pro vyjádření hydrologické bilace bez interpretace dynamiky).

3 Úloha bedřichovského tunelu

Modely lokality bedřichovského tunelu řešíme ve dvou variantách. Kvůli odladění a správné funkčnosti nejprve model lokality bez tunelu a poté i s tunelem. Geometrie modelu (Obr. 1) je již zadána s triangulací o straně 200 metrů, abychom mohli postihnout nerovnosti na povrchu terénu, lokálně je triangulace zjemněná kvůli lepšímu popisu řek a dalších detailů. Odlišnost úlohy je také v náročnosti diskretizace. Horizontální rozměry modelu jsou v řádu kilometrů, naopak vertikální pouze několik stovek metrů a geometrie navíc obsahuje i tunel s malým průměrem. Proto je složitější zvolit zjemnění sítě tak, aby plynule přecházelo od jemné kolem tunelu k hrubší na okraji modelu na malé vzdálenosti, protože se tunel nachází pouze asi 100 metrů pod zemským povrchem, Tab.1. Geometrii s triangulací dále diskretizujeme pomocí elementů o straně 150 metrů nebo menšími. Důležité je ale zjemnění ve svislém směru, kterého je možno dosáhnout například vytvořením předdefinovaných vrstev před generováním sítě.



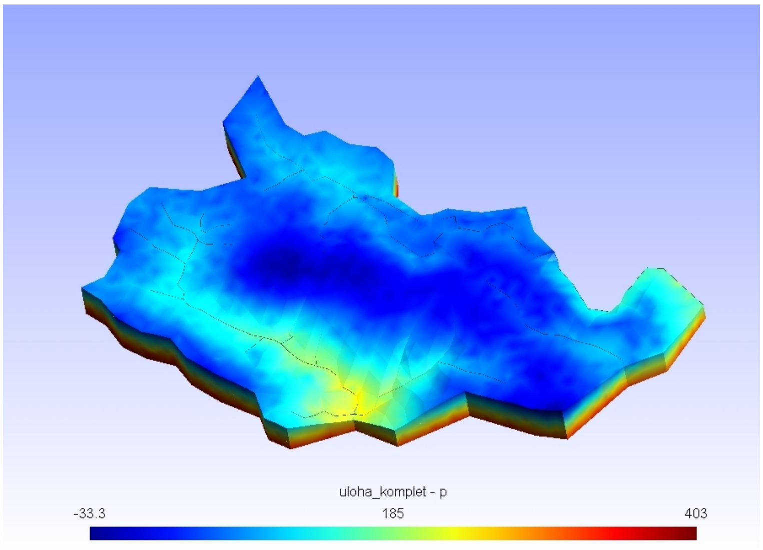
Obrázek 1: Poloha tunelu, podle [5] a pohled shora na geometrii lokality tvořenou triangulací.

přibližné rozměry [m]	5250 × 6007 × 400
nadmořské výšky [m n. m.]	500 - 840
rozměry tunelu [m] (průměr, délka)	3.6 × 2590
průměrná mocnost nadloží tunelu [m]	100
počet uzlů 150m síť	4253
počet elementů 150m síť	21185
strana elementu 150m síť [m]	cca 150
počet uzlů 100m síť	12573
počet elementů 100m síť	62331
strana elementu 100m síť [m]	cca 100

Tabulka 1: Číselné charakteristiky úlohy a dvou možností diskretizace modelu, [4]

4 Závěr

V tomto modelu jsme ukázali možnost využití 3D-2D-1D koncepce na úloze proudění a varianty nastavení parametrů pro diskretizaci. Úloha je ve fázi rozpracovanosti, ale podle výsledků (např.



Obrázek 2: Model bez tunelu - výsledné rozložení tlakové výšky [m].

tlakové výšky na Obr. 2) je vidět, že voda proudí podle fyzikálních představ: částečně stéká po povrchu a do řek, částečně se vsakuje, a na úpatí kopců vyvěrá. Pole rychlostí je na názornou vizualizaci složitější.

Poděkování: Tato práce vznikla za podpory Ministerstva průmyslu a obchodu, projekt č. FR-TI1/362.

Reference

[1] O. Severýn, M. Hokr, J. Královcová, J. Kopal, M. Tauchman: *Flow123D: Numerical simulation software for flow and solute transport problems in combination of fracture network and continuum.* Technical report. TU Liberec, 2008.

[2] J. Šembera, J. Maryška, J. Královcová, O. Severýn: *A novel approach to modelling of flow in fractured porous medium.* Kybernetika 4/2008, 577–588.

[3] J. Valentová: *Flow123D: Hydraulika podzemní vody.* ČVUT Praha, 2007.

[4] SÚRAO: *Geologická a strukturní analýza granitoidů z tunelu v Bedřichově v Jizerských horách. Příloha 1.*. Technická zpráva, Praha, 2003.

[5] D.Šustr: *Numerické simulace přítoku podzemní vody do tunelu, diplomová práce*, TU Liberec, 2009.

Probabilistic system approach to LC-MS analysis

J. Urban, J. Vaněk, D. Štys

Institute of Physical Biology, University of South Bohemia
Nové Hrady CZ

1 Introduction

The cell state may be ultimately characterized by state of metabolomic and signaling pathways and by state of formation of patterns in the cellular structures. These parameters complement each other. The pathways and metabolite transformations define and maintain the cellular structures. However, the current state of knowledge does not allow to devise state of pathways from observation of structure and vice versa. The state of metabolite fluxes is aspect of the far-from equilibrium physico-chemical state of the cell or whole culture. A chemical analysis of huge amount of collected data from measured samples is difficult and it take a long time to be done manually. Automatics tools could be helpful and are able to save important part of the time. This paper is focussed on automatic processing of liquid chromatography in combination with mass spectrometry. A set of tools was developed: e.g. noise reduction, peaks detection or substance comparison.

Today, the intensity y produced by the spectrometer is usually shown as two 2D graphs, $y1(t)$ and $y2(m)$, where t is time and m is the mass to charge ratio. We take a more general approach, looking at peeks in 3D, $y(t,m)$. At each point (t,m), where t is the retention time and m is mass m/z, the output intensity of the spectrometer is composed of several parts:

$$y(t,m) = s(t,m) + q(t,m) + r(t,m), \tag{1}$$

where $y(t,m)$ is the useful signal, $s(t,m)$ is the useful signal, $q(t,m)$ is the systematic noise $ps(t,m)$ = probablility that $y(t,m)$ is useful signal, $s(t,m)$ and $r(t,m)$ is the random noise $pq(t,m)$ = probablility that $y(t,m)$ is not random noise, $r(t,m) = 0$.

Exact properties of each component, both theoretical and practical, are documented in the comment section of the code. Additional discussion of the individual steps was presented in [1].

Our software automatically evaluates the given instrument, detects peaks, and calculates the probability of error for individual peaks. There are no artificial, user-defined parameters. The program not only quantifies the accuracy of the interpretation, but it also detects many peaks which, using the existing methods, are not distinguished from the noise. Exactly the same algorithm is used to evaluate the preliminary blank run, and the peaks detected in this measurement. Then, in the regular runs, if a similar peak is detected for the same (t,m) point, the peak is eliminated — it is a peak associated with the mobile phase and washing of the colon. Many software packages simply subtract $y(t,m)$ of the blank from $y(t,m)$ of the sample output, but that clearly makes no sense.

Typical measurement output data from HPLC/MC is set of points in three dimensional space which is defined by axes: retention time, molecular mass and intensity. Analytes elute in every retention time point from HPLC column, obviously because of delay proportional to some chemical property, and enter the MS ionization chamber. Intensity for each detectable mass is

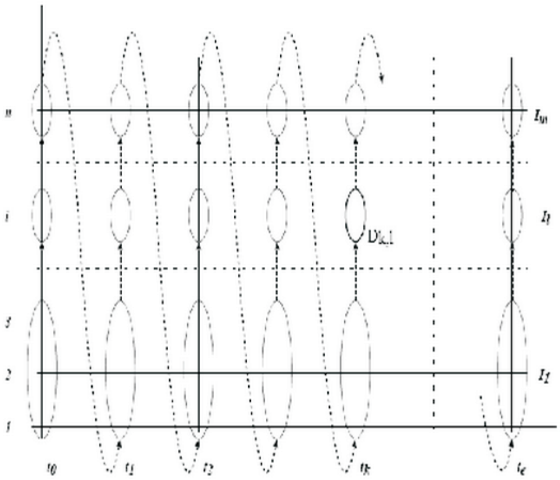


Figure 1: Mathematical space of system atributes.

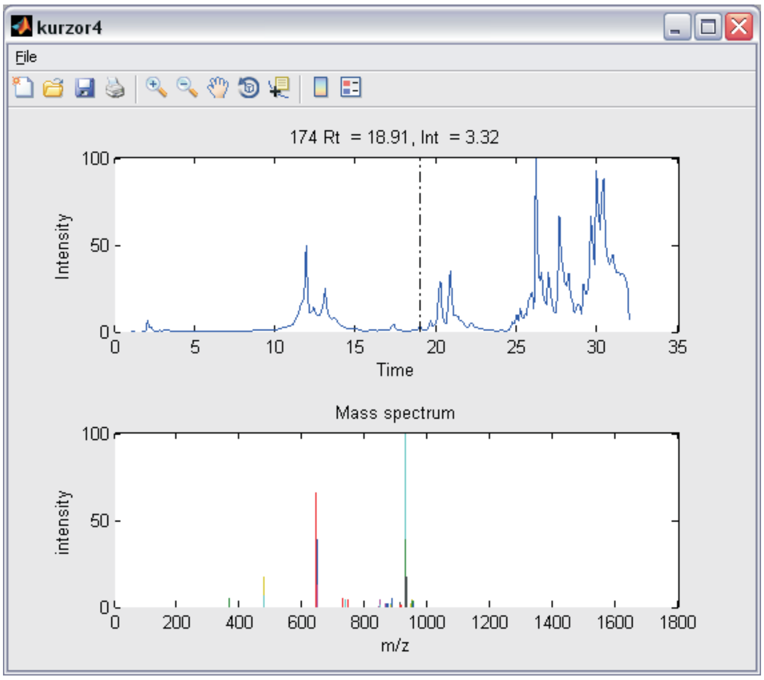


Figure 2: Filtered LC-MS measurement with probability > 75 % in classic graphical representation.

measured inside the MS and this value represented amount of ionized molecules of individual mass in exact retention time point [2, 3, 4].

2 Conclusion

Ours approach is focused on proper characterization of presented noise. Noise produced by mobile phase is characterised separately to random noise contribution. Information about the both of noise characterizations were integrated into probability factor. All algorithms were implemented in MATLAB environment [5].

Acknowledgement: This work has been supported by the grant HCTFOOD A/CZ0046/1/0008 of EEA funds.

References

[1] J. Urban, J. Vaněk, J. Soukup, D. Štys: *Expertomica metabolite profiling: getting more information from LC-MS using the stochastic systems approach.* Bioinformatics, 25(20), 2764–7, 2009.

[2] R.E. Ardrey: *Liquid Chromatography Mass Spectrometry: An Introduction.* Wiley, 2003.

[3] M.C. McMaster: *HPLC, a practical user's guide.* Wiley, 2007.

[4] W. Weckwerth (ed.): *Metabolomics: Methods and Protocols.* Humana Press, Totowa NJ, 2007.

[5] MATLAB software, www.mathworks.com, The Mathworks, Natick, Massachusetts, USA.

Using graphic cards for high-performance computing tasks

J. Vaněk, J. Urban, Dalibor Štys

Institute of Physical Biology, Nové Hrady

1 Introduction

Recently, graphics hardware (GPU) architectures have begun to emphasize versatility, offering rich new ways to programmable reconfigure the graphics engine. In this abstract, we introduce whether current graphics architectures can be applied to problems where general-purpose vector processors might traditionally be used. Comparing the speed of graphics cards to standard CPUs, it illustrate high raw performance power, as well as room for improvement speed in many tasks. The main features and tricks of the GPU programming is described below. Based on our results and current trends in GPU development, we believe that efficient use of graphics hardware will become increasingly important to high-performance computing on commodity hardware.

2 Implementation of algorithms on GPU

Actual generation of graphic cards could be used for very realistic 3D computer gaming. However, their high computation power can be used also for another applications - for high performance computing [1]. Comparison of raw computation power between GPUs and CPUs are illustrated on figure 1.

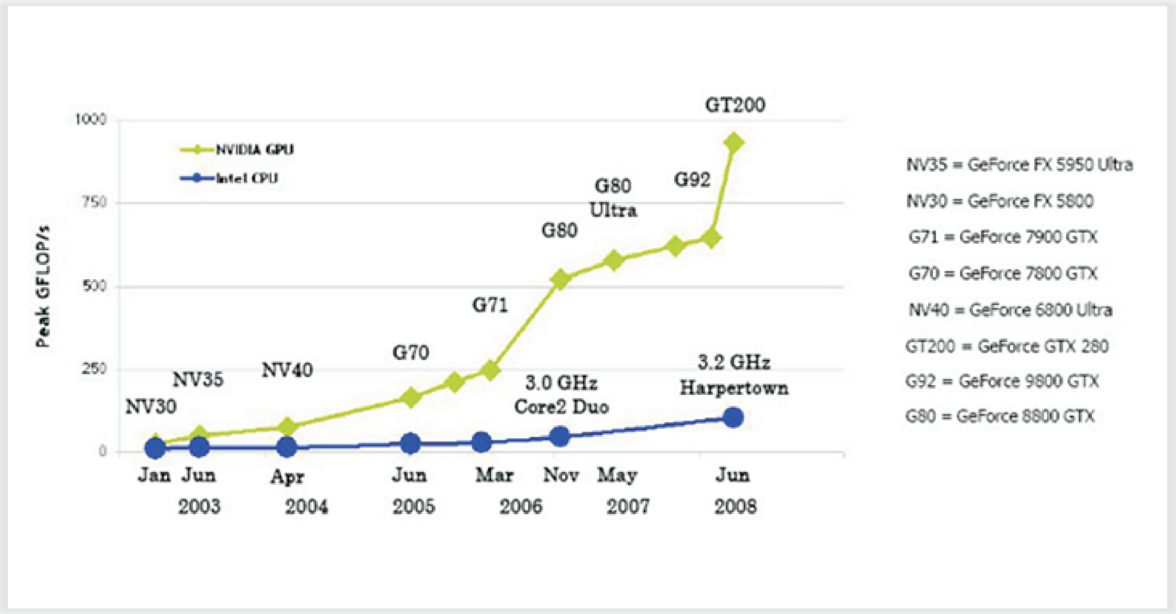


Figure 1: An comparison between GPUs and CPUs raw computation power in recent years.

Currently, two high-level GPU programming technologies from both main manufactures are usable. Brook+ (now called ATI Stream SDK) from AMD/ATI [2] and CUDA from NVIDIA [3]. The algorithm was done in CUDA because of NVIDIA hardware. GPU programming is not so easy especially if maximum speed is necessary. But the final speed of implementation satisfies more sophisticated programming style.

The key to high performance of implementation is to fit the algorithm to GPU highly-parallel architecture. The architecture of NVIDIA GPUs is illustrated on figure 2.

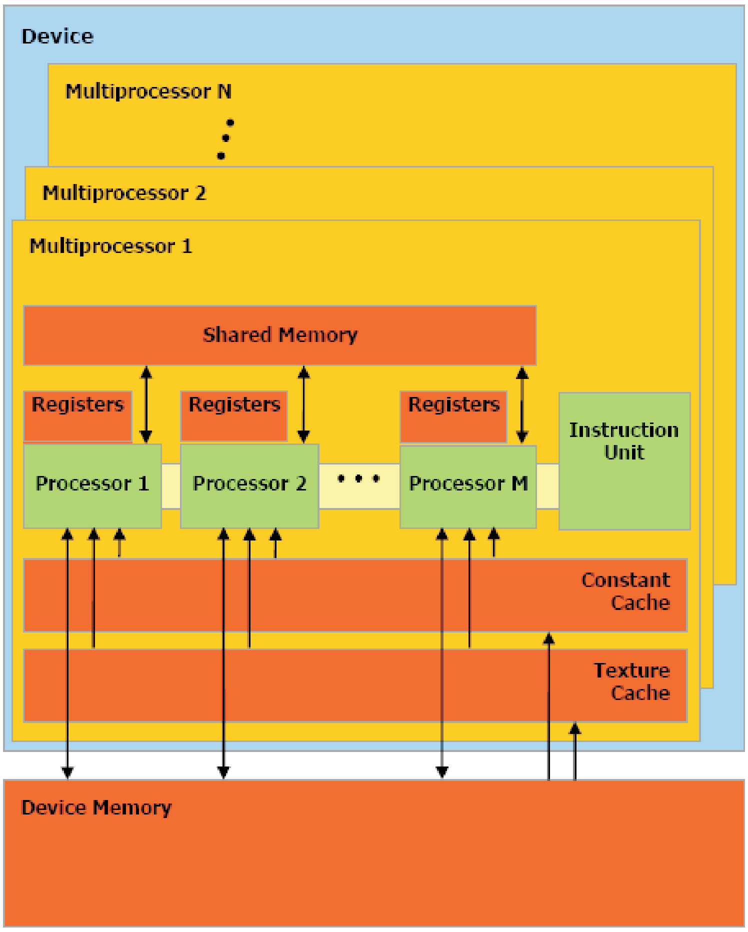


Figure 2: CUDA double-hierarchy data-parallel programming model.

For the architecture the double-hierarchy is typical. Whole GPU is the set of multiprocessors as well as the multiprocessor (MP) is a set of eight scalar processors (SP). All multiprocessors can access data in the device memory. For reading they can employ texture or constant cache. SPs inside one multiprocessor can utilize excepting its registers also joint shared cache. Read and write accesses can be synchronized and the SPs inside one multiprocessors can cooperate this way. The survey of all memory kinds are listed for better understanding:

- **Host (CPU) memory** — “normal” memory where all data have to be prepared before transfer to GPU memory through PCI-Express bus. The results are transferred back from GPU to host memory after computation. It is faster to transfer less amount of larger memory blocks than transfer high number of small blocks. Keep in mind that the bus is relatively very slow and could be a bottle-neck of the computation performance.

- **Device (GPU, global) memory** — main memory installed on graphics card. It has high delay therefore the implementation has to look at this fact. Random read/write of single data-types affects the performance a lot. Using block read/write ("coalesced access") is necessary. The second option is to use constant or texture cache.

- **Constant cache** — limited amount of this kind of memory can be used for reading. It is advisable to use it for example for look-in tables.

- **Texture cache** — read-only data in device memory can be cached via texture cache. It can be allocated either linear memory or in 2D manner. It is advisable to use it for all data which are read-only.

- **Shared memory** — limited amount of on-chip memory which is shader-clocked and it belongs to individual multiprocessor. It is accessible only for SPs of this multiprocessor. The implementation should avoid "bank conflicts" which reduce performance. Appropriate using of this kind of memory can be a key part of high-performance implementation.

- **Registry** — memory which belongs to individual SPs. They are used to store the internal variables.

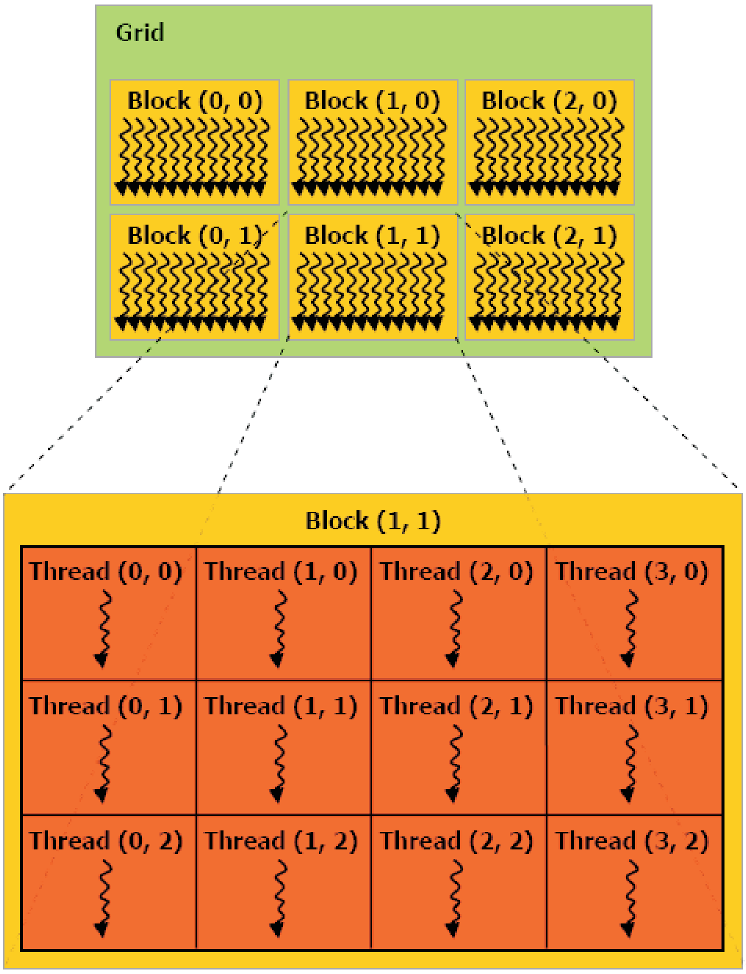


Figure 3: CUDA double-hierarchy data-parallel programming model.

CUDA data-parallel programming model is based on the hardware double-hierarchy. The data have to be split into algorithmically independent parts on two levels. At the first level the data are split into grid of blocks. Each block is processed with the same algorithm which is called “kernel”. During processing of the block several number of thread is running. It is the second level of the hierarchy (it is illustrated in fig. 3). All threads which evaluate one block are running on one multiprocessor and they can utilize its shared memory for data interchange.

To manage this programming model, SIMT (single-instruction multiple thread) architecture is employed. The multiprocessor maps each thread to one SP core, and each scalar thread executes independently with its own instruction address and register state. The multiprocessors threads are executed in parallel groups. The programmer has to implement its algorithm this parallel way too. At least 32 threads should do the same work.

3 Conclusion

This abstract introduces abilities of actual generation of graphic cards and shortly describes main programming features and tricks how to achieve good effectivity of the algorithm. Our practical experiences with GPU programming lead us to conclusion that it is a promising technology. Moreover, it is usable way how to cheaply speed-up many algorithms and applications.

Acknowledgement: This work was supported by the grant HCTFOOD A/CZ0046/1/0008 of EEA funds.

References

[1] General-Purpose Computation Using GPUs, www.gpgpu.org.

[2] AMD/ATI Stream computing SDK, ati.amd.com/technology/streamcomputing.

[3] NVIDIA CUDA gpgpu development tools, www.nvidia.com/cuda.

Behaviour of the augmented GMRES method

J. Zítko, D. Nádhera

Charles University, Faculty of Mathematics and Physics, Prague

1 Introduction

Let us consider the linear system

$$Ax = b, \quad A \in \mathbb{C}^{n\times n}, \quad x, b \in \mathbb{C}^n, \tag{1}$$

where $\mathbb{C}^n$ denotes the complex $n-$dimensional space. It is assumed that A is a nonsingular, non-Hermitian and in praxis usually large and sparse matrix.

The restarted GMRES method with restart m, usually denoted by GMRES(m), is a well known and popular iterative process for solving above mentioned systems and will be a base for the next consideration. However, restarting slows down the convergence and in many cases GMRES(m) causes stagnation. The research for reducing negative effects of restart develops in several ways. Let us mention the principal two: construction of a good preconditioner or augmentation of the current Krylov subspace. In general, both processes are not stationary. The preconditioner and the additional subspace are updated after each restart.

We will consider the GMRES(m, k) method, i.e., the restarted GMRES with restart m where the subspace of dimension k is added in each restart. The additional subspace will be constructed on the base of information gathered in previous restarts. The classical estimate of the norm of the residual vector leads to the following deduction. The following text can be found in the book [3] on the page 55. “Eigenvalues all around the origin are bad because (by the maximum principle) it is impossible to have a polynomial that is 1 at the origin and less than 1 everywhere on some closed curve around the origin.” On the opposite side “eigenvalues tightly clustered about some single point (away from the origin) are good”. Hence the idea to remove the eigenvalues that are small in magnitude from the spectrum of A seems to be natural. The technique how to do it is studied by many authors. Let us mention [5, 1, 11] here. The implementation presented in [5] generates first the Krylov subspace and then adds approximate eigenvectors corresponding to the smallest eigenvalues in magnitude. In this paper, k generalized harmonic Ritz values are calculated and the corresponding eigenvectors are added to the Krylov subspace. The GMRES(m, k) method is introduced in Section 2. The numerical behaviour of spaces, which are added, is studied in the next Section 3. The non-stagnation conditions are mentioned and numerically tested in Section 4. Some comments to the generalization of non-stagnation conditions are in the last section.

If $Y \in \mathbb{C}^{n\times k}$, then Range($Y$) is the space generated by the columns of Y. In the whole paper the following notational conventions will be used: x_0 … an initial approximation, $r_0 = b - Ax_0$ the corresponding residual, it is supposed that $r_0 \neq 0$, $v_1 = r_0/\|r_0\|$, $\|.\|$… the Euclidean norm, $\mathcal{K}_m(A, r_0)$ =Range($[r_0, Ar_0, \ldots, A^{m-1}r_0]$) … the Krylov subspace, S_n … the unit sphere in $\mathbb{C}^n$, $\mathcal{P}_m$ … the set of polynomials of degree m, $\mathcal{P}_m^0$ … all polynomials from $\mathcal{P}_m$ which equal zero in zero. Let $\emptyset \neq \mathcal{Z} \subseteq \mathbb{C}^n$ be a subspace. Define the norm $\|A\|_{\mathcal{Z}} = \sup\limits_{x\in\mathcal{Z}\cap S_n} \|Ax\|$ and $W_{\mathcal{Z}}(A) = \{x^H Ax|\, x \in \mathcal{Z} \cap S_n\}$ is the field of values of A, with respect to $\mathcal{Z}$. Let $s := m + k$. It is assumed that all Krylov and augmented spaces have maximal dimension and that the matrix A is diagonalizable. The following process could be analyzed without the last assumption. However, in this case, the formulas would be more complicated without any new theoretical contribution.

2 Description of GMRES(m, k)

ALGORITHM 2.1. ONE RESTARTED RUN OF GMRES(m, k).

1 Input Let $x_0^{(j)}$ and $r_0^{(j)}$ be the starting vector and corresponding residual vector at the beginning of the jth restart. Let the space Range(Y_j) where $Y_j = [y_1^{(j)}, y_2^{(j)}, \dots, y_k^{(j)}]$ be added to $\mathcal{K}_m(A, r_0^{(j)})$. Put $v_1 := r_0^{(j)}/\|r_0^{(j)}\|$.

2 Construction of the orthogonal basis Calculate for $i = 1, 2, \dots, s$ the vectors $v_{i+1} = P_i^{\perp} Au_i / \|P_i^{\perp} Au_i\|$ where

- for $i \leq m$ we substitute $u_i := v_i$ and $P_i^{\perp}$ denotes the orthogonal projection onto the orthogonal complement of the Krylov subspace $\mathcal{K}_i(A, v_1)$;
- for $m+1 \leq i \leq s$ we put $u_i = y_{i-m}^{(j)}$ and $P_i^{\perp}$ denotes the orthogonal projection onto the orthogonal complement of the space Range($[v_1, v_2, \dots, v_i]$).

The motivation for this more general formulation can be found in [9] and [11]. The above procedure is the shortest formulation of the idea of all known processes for the construction of an orthonormal basis of Range($[v_1, Av_1, \dots, A^m v_1, Ay_1^{(j)}, Ay_2^{(j)}, \dots, Ay_k^{(j)}]$).

3 Output Define the matrices W, $\tilde{W}$ and the $(s+1) \times s$ Hessenberg matrix $\tilde{H}$ by the relations

$$W = [v_1, v_2, \dots, v_m, y_1^{(j)}, \dots, y_k^{(j)}], \quad \tilde{W} = [v_1, v_2, \dots, v_m, v_{m+1}, \dots, v_{s+1}], \quad AW = \tilde{W}\tilde{H}.$$

- The new iteration $x_s^{(j)} = x_0^{(j)} + w_s$, where $r_0^{(j)} - Aw_s \perp$ Range (AW).
- The new matrix $Y_{j+1} = [y_1^{(j+1)}, y_2^{(j+1)}, \dots, y_k^{(j+1)}]$ is calculated by the following way: $Y_{j+1} = WG$, where $G = [g_1, g_2, \dots, g_k]$ and g_i for $i \in \{1, 2, \dots, k\}$ solve the eigenvalue problem

$$W^H A^H W g_i = \theta_i^{-1} W^H A^H A W g_i$$

The k smallest harmonic Ritz values, denoted θ_i, $i = 1, 2, \dots k$, with the correspondig harmonic Ritz vectors g_i are calculated.

For more details see for example [5], [1]. The algorithm of GMRES(m, k) is now obvious. Now we will study the numerical behaviour of the sequence $\mathcal{Y}_1, \mathcal{Y}_2, \dots$, where $\mathcal{Y}_j :=$Range(Y_j). The convergence was studied in special cases for example in [8]. The problem we focus is to find cheaply an integer $p > 0$ such that the subspaces $\mathcal{Y}_j$ and $\mathcal{Y}_{j+1}$ do not differ to much. Various tests could be considered for example whether the gap $\Theta(\mathcal{Y}_j, \mathcal{Y}_{j+1}) \leq tol \ \forall j > p$ where tol is a given tolerance.

3 Numerical experiments

More numerical examples can be found in [6]. Let the upper bidiagonal matrix A has 0.1, 0.2, 0.3, 4..., 1000 on the main diagonal and the superdiagonal elements are equal 0.1. The Figure 1 shows that GMRES(15) stagnates but GMRES(m, k), (where $m + k = 15$), converges if $k \geq 3$. The approximate eigenvectors are calculated according to the part **3** in ALGORITHM 2.1. This experiment leads to the second stage, where the improving of $\mathcal{Y}_j$ terminates after p restarted runs and $\mathcal{Y}_p$ is added in all the next restarts. The Figure 2 leads us to investigate the behaviour of the numbers $\Theta(\mathcal{Z}, A\mathcal{Y}_j)$, $\|Ay^{(j)} - \theta y^{(j)}\|/\|y^{(j)}\|$ (see FIGURE 4) and $\Theta(A\mathcal{Y}_j, A\mathcal{Y}_{j+1})$. The symbol $\mathcal{Z}$ denotes the eigenspace corresponding to the three small eigenvalues. Index j denotes the jth restart.

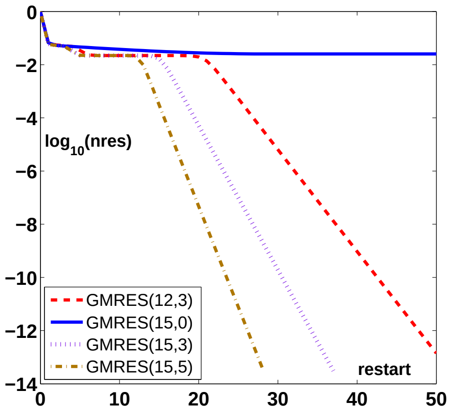


Figure 1:

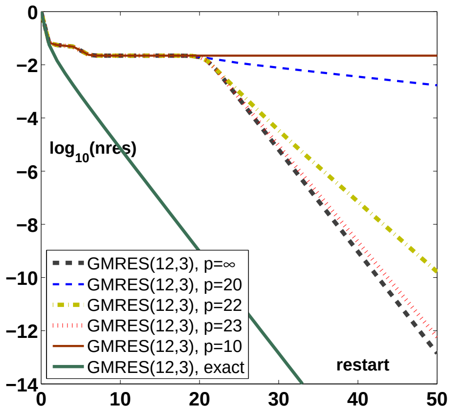


Figure 2:

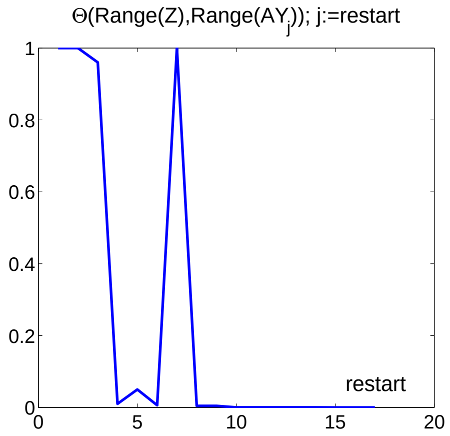


Figure 3:

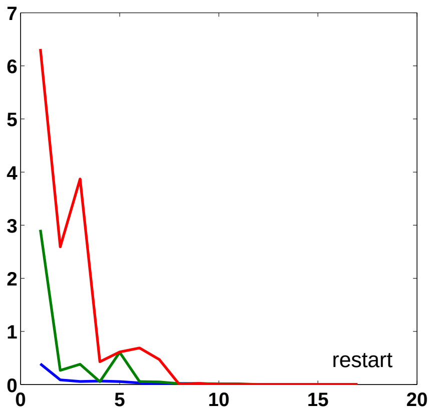


Figure 4:

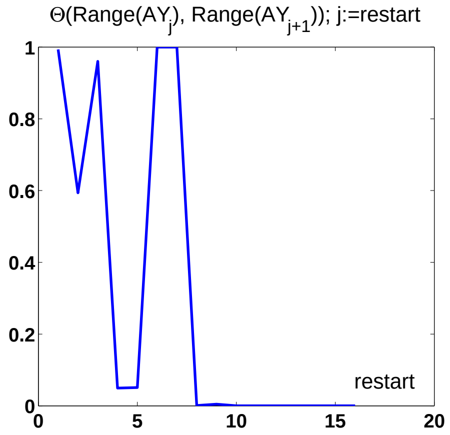


Figure 5:

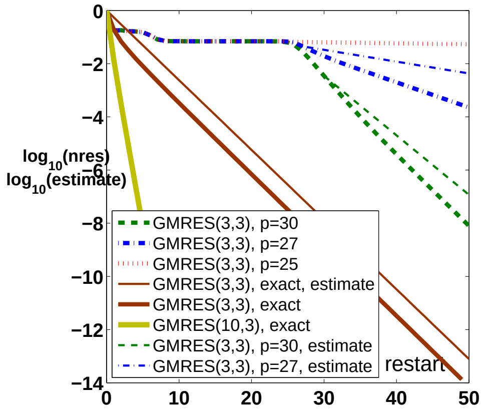


Figure 6:

The matrix A has the same structure as in the previous case, only the dimension will be smaller, in this case $A \in \mathbb{R}^{100\times 100}$. The behaviour of $\Theta(\mathcal{Z}, A\mathcal{Y}_j)$ is drawn in Figure 3. The Figure 4 describes the numbers $\|Ay^{(j)} - \theta y^{(j)}\|/\|y^{(j)}\|$ for three smallest harmonic Ritz values θ_1 (the curve above), θ_2 (the curve in the middle) and θ_3. The gap between the spaces $A\mathcal{Y}_j$ and $A\mathcal{Y}_{j+1}$ illustrates Figure 5.

4 Estimates of residual vector

More detailed calculations show that the convergence of GMRES(m,k) algorithm can be very slow for small k or when a wrong spaces Y_j are added. The behaviour of the algorithm is well described by the estimates of the residual norm that are independent on the number of restart. The following theorem [12] generalize the result of [4] and the presented estimate approximate the course of GMRES(m,k) very well.

Theorem 4.1 Let $m,k,s \in \{1,2,\ldots,n-1\}$, $s = m+k < n$, $j > 1$. Let the subspace $\mathcal{Y}_k$ =Range(Y_k) be added to the corresponding Krylov subspace in the jth restart.
1) Then the following inequalities

$$\|r_s^{(j)}\|^2/\|r_0^{(j)}\|^2 \le \min_{q\in\mathcal{P}_m^0}\left(1-\min_{\substack{w\in S_n\\ w\perp A\mathcal{Y}_k}}\frac{|w^Hq(A)w|^2}{\|q(A)w\|^2}\right) \le \min_{q\in\mathcal{P}_m^0}\left(1-\min_{\substack{w\in S_n\\ w\perp A\mathcal{Y}_k}}\frac{|w^HH_qw|^2}{\|q(A)w\|^2}\right) \tag{2}$$

hold, where H_q is the Hermitian part of $q(A)$. If S_q denotes the skew-Hermitian part of $q(A)$ then we obtain further estimates substituting S_q instead of H_q in (2). Let us remark that $w^TS_qw = 0$ for real vectors and matrices.
2) Let m be an integer, $m+k < n$,. If a polynomial $q \in \mathcal{P}_m^0$ exists such that the system of equations

$$w^Hq(A)w = 0 \quad \text{or} \quad w^HH_qw = 0 \quad \text{or} \quad w^HS_qw = 0 \tag{3}$$

does not have any solution in the set $(A\mathcal{Y}_k)^\perp \cap S_n$, then GMRES$(m,k)$ is convergent.

5 Conclusion

The presented theorem is an example of estimates that guarantee the convergence in the case that H_q or S_q is positive or negative definite. Further generalization yields the paper [7]. The generalization of the basic theorem in the just mentioned paper will be formulated for GMRES(m,k) now. Let $q \in \mathcal{P}_m^0$, $q(A) = H_q+\mathbf{i}S_q$, where H_q and S_q is the Hermitian and skew-Hermitian part of the matrix $q(A)$, respectively. It is easy to see that

$$x^Hq^2(A)x = x^H(H_q+\mathbf{i}S_q)^2x = \|H_qx\|^2 - \|S_qx\|^2 + \mathbf{i}2x^HL_qx. \tag{4}$$

where $\mathbf{i}^2 = -1$ and $L_q = (H_qS_q + S_qH_q)/2 = (H_qS_q + (H_qS_q)^H)/2$ is the Hermitian part of the matrix H_qS_q. Let the subspace $\mathcal{Y}_k$ be added to the considered Krylov subspace in all restarts and define $\mathcal{Z} = (A\mathcal{Y}_k)^\perp$, $H_q(\mathcal{Z}) = \{H_qz|z \in \mathcal{Z}\}$ and $S_q(\mathcal{Z}) = \{S_qz|z \in \mathcal{Z}\}$. It is easy to see that

$$\Big\{\|H_qx\| < \|S_qx\| \;\forall x \in \mathcal{Z}\cap S_n\Big\} \Leftrightarrow \Big\{Re(x^Hq^2(A)x) < 0 \;\forall x \in \mathcal{Z}\cap S_n\Big\}, \tag{5}$$

and analogous relation could be written if $\|S_qx\| < \|H_qx\|$. Because the set $\mathcal{Z}\cap S_n$ is compact, the field of values $W_{\mathcal{Z}}(q^2(A))$ does not contain 0 if $\|S_qx\| < \|H_qx\|$ or $\|S_qx\| < \|H_qx\|$ or if $x^HL_qx \neq 0\,\forall x \in \mathcal{Z}\cap S_n$.

If S_q is nonsingular, then

$$\Big\{\|H_qx\| < \|S_qx\| \;\forall x \in \mathcal{Z}\cap S_n\Big\} \Leftrightarrow \left\{\left\|H_qS_q^{-1}\overbrace{\frac{S_qx}{\|S_qx\|}}^{y}\right\| < 1 \;\forall x \in \mathcal{Z}\cap S_n\right\}$$
$$\Leftrightarrow \left\{\overbrace{\sup_{y\in S_q(\mathcal{Z})\cap S_n}\|H_qS_q^{-1}y\| < 1}^{\Leftrightarrow\|H_qS_q^{-1}\|_{S_q(\mathcal{Z})}<1}\right\}$$

and analogous relation could be written if if the matrix H_q is nonsingular and $\|S_q x\| < \|H_q x\|$. The just formulated thoughts for the matrix $q(A)$ form another proof of the original result of Simoncini and Szyld (see [7]) for the matrix A.

Theorem 5.1 Let $q \in \mathcal{P}_m^0$ be arbitrary. Let the subspace $\mathcal{Y}_k$ of dimension k be added to the Krylov subspace in all restarted runs. Define $\mathcal{Z} = (A\mathcal{Y}_k)^{\perp}$. Let the matrix S_q is nonsingular and $\|H_q S_q^{-1}\|_{S_q(\mathcal{Z})} < 1$ or the matrix H_q is nonsingular and $\|S_q H_q^{-1}\|_{H_q(\mathcal{Z})} < 1$ or $x^H L_q x \neq 0\, \forall x \in \mathcal{Z} \cap S_n$.
Then GMRES$(2m, k)$ is convergent.

Acknowledgement: The work is a part of the research project MSM0021620839 financed by MSMT. The authors thanks for the support.

References

[1] A. Chapman, Y. Saad: *Deflated and augment Krylov subspace techniques.* Numer. Lin. Algebra with Appl., 4, 43–66, 1997.

[2] M. Eiermann, O.G. Ernst: *Geometric aspects in the theory of Krylov subspace methods.* Acta Numerica, 251–312, 2001.

[3] A. Greenbaum: *Iterative methods for solving linear systems.* SIAM, Philadelphia 1997.

[4] J.F. Grcar: *Operator coefficient methods for linear equations. A restarted GMRES method augmented with eigenvectors.* Technical Report SAND89-8691, Sandia National Laboratories, November 1989. Available at `http://seesar.lbl.gov/ccse/Publications/sepp/ocm/SAND89-8691.pdf`.

[5] R.B. Morgan: *A restarted GMRES method augmented with eigenvectors.* SIAM J. Matrix Anal. Appl., 16, 1154–1171, 1995.

[6] D. Nádhera: *Conditions for convergence of the restarted and augmented GMRES method.*(Czech) Diploma thesis, Charles University, Faculty of Mathematic and Physics, Prague 2009.

[7] V. Simoncini, D.B. Szyld: *New conditions for non-stagnation of minimal residual methods.* Report 07-4-17, April 2007. Available at `http://www.math.temple.edu/ szyld`.

[8] D.C. Sorensen: *Implicit application of polynomial filters in k-step Arnoldi method* . SIAM J. Matrix Anal. Appl., 13(1), 357–385, 1992.

[9] H.F. Walker, L. Zhou: *A simpler GMRES.* Numer. Lin. algebra with Appl., 1(6), 571–588, 1994.

[10] J. Zítko: *Generalization of convergence conditions for a restarted GMRES.* Numer. Linear Algebra Appl., 7, 117–131, 2000.

[11] J. Zítko: *Convergence conditions for a restarted GMRES method augmented with eigenspaces.* Numer. Lin. algebra with Appl., 12, 373–390, 2005.

[12] J. Zítko: *Some remarks on the restarted and augmented GMRES method.* ETNA, 31, 221–227, 2008.