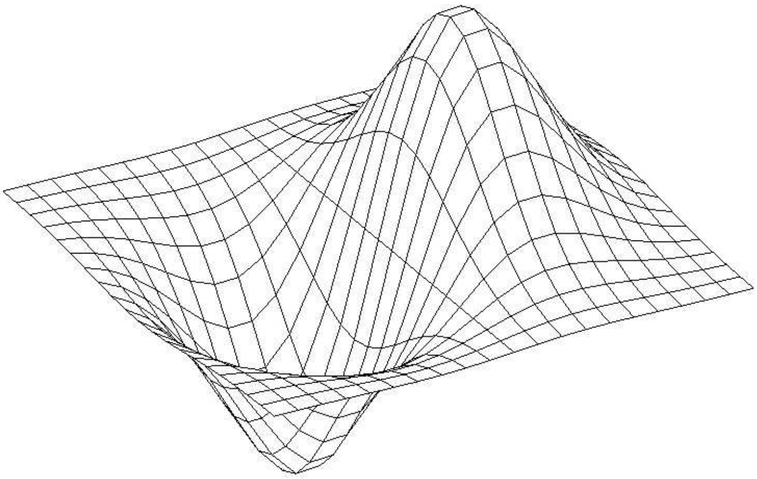
Institute of Computer Science AS CR, Prague

SNA'14

Seminar on Numerical Analysis

Modelling and Simulation
of Challenging Engineering Problems



Winter School

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

Nymburk, January 27 – 31, 2014

Programme committee:

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

Organizing committee:

Hana Bílková	Institute of Computer Science AS CR, Prague
Jurjen Duintjer Tebbens	Institute of Computer Science AS CR, Prague
Miroslav Rozložník	Institute of Computer Science AS CR, Prague
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, Prague
ISBN 978-80-87136-16-4

Preface

Seminar on Numerical Analysis (SNA) is a scientific meeting devoted mainly to mathematical modeling, numerical methods for partial differential equations, numerical linear algebra and parallel computing. Its history goes back to 2003. Since 2005 it has been coupled with the winter school offering tutorials or extended lectures of various selected topics. The venue and organization effort of these meetings is traditionally distributed between Bohemian and Moravian-Silesian organizers from the Academy of Sciences (Institute of Geonics, Ostrava and Institute of Computer Science, Prague) as well as from the universities (Technical University of Ostrava, Czech Technical University and Charles University in Prague). The seminar and the winter school SNA 2014 will be held for the first time in the training centre of the Czech Association of Physical Education and Sports in Nymburk. This year the winter school offers the lectures delivered by the following distinguished researchers:

- Discontinous Galerkin method (*V. Kučera*),
- Operator preconditioning (*Z. Strakoš*),
- FFT-based Galerkin method for homogenization of periodic media (*J. Vondřejc, J. Zeman and I. Marek*)
- Mathematics in image processing (*M. Šorel*).

We believe that the participants will enjoy also the complementary program of contributed presentations and posters. We wish you a pleasant stay in Nymburk.

On behalf of the Programme and Organizing Committee of SNA 2014

Mirek Tůma, Miro Rozložník

Contents

Winter school lectures

V. Kučera
Discontinous Galerkin method

Z. Strakoš
Operator preconditioning

M. Šorel
Mathematics in image processing

J. Vondřejc, J. Zeman, I. Marek
FFT-based Galerkin method for homogenization of periodic media

Isogeometric analysis for Navier-Stokes equations

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová

NTIS – New Technologies for Information Society, University of West Bohemia, Plzeň

1 Introduction

This article is devoted to the simulation of viscous incompressible fluid flow. The numerical model is based on the isogeometrical approach. This is a part of the project devoted to the shape optimization of water turbines. Typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data. Each design change requires generation of new meshes which takes a lot of time. Primary goal of using isogeometric analysis is to be geometrically exact, independently of the discretization. Then we do not need to create any other mesh – the mesh of the so-called "NURBS elements" is acquired directly from CAD representation. Further refinement of the mesh or increasing the order of basis functions are very simple, efficient and robust.

2 NURBS Surfaces

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

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

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

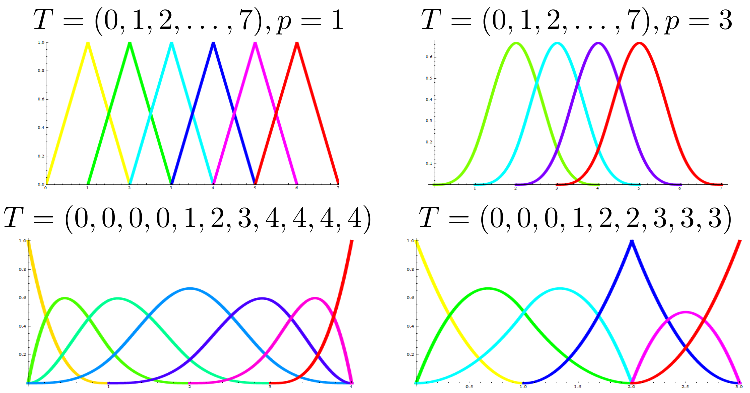
$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & u_i \le t < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p}(u). \end{aligned} \tag{2}$$

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

3 Stationary Navier-Stokes equations

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

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



$T = (0, 1, 2, \dots, 7), p = 1$

$T = (0, 1, 2, \dots, 7), p = 3$

$T = (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$

$T = (0, 0, 0, 1, 2, 2, 3, 3, 3)$

Figure 1: B-spline basis functions

where $\mathbf{u} = \mathbf{u}(\mathbf{x})$ is the vector function describing flow velocity, $p = p(\mathbf{x})$ is the pressure function, ν describes dynamic viscosity and f additional body forces acting on the fluid. We do not assume only very small Reynolds numbers, but there are still some "limits" for which this model gives reasonable solution. The boundary value problem is considered as the system (3) together with boundary conditions

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

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

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

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

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

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

3.1 Approximation using isogeometric analysis

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

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

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

This approach is based on the so-called Picard's method. For isogeometric analysis, basis functions of V_0^h and W^h are NURBS basis functions obtained from the NURBS description of the computational domain (for velocity and pressure). We can express $\mathbf{u}_h^k$ and p_h^k as a linear combination of the basis functions (2) and substitute them to (6) and (7). Linearization is done with help of Picard's iteration and we obtain a sequence of solutions $(\mathbf{u}_h^k, p_h^k) \in V^h \times W^h$, which converges to the weak solution. We obtain a matrix formulation of the problem in the form

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

where

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

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

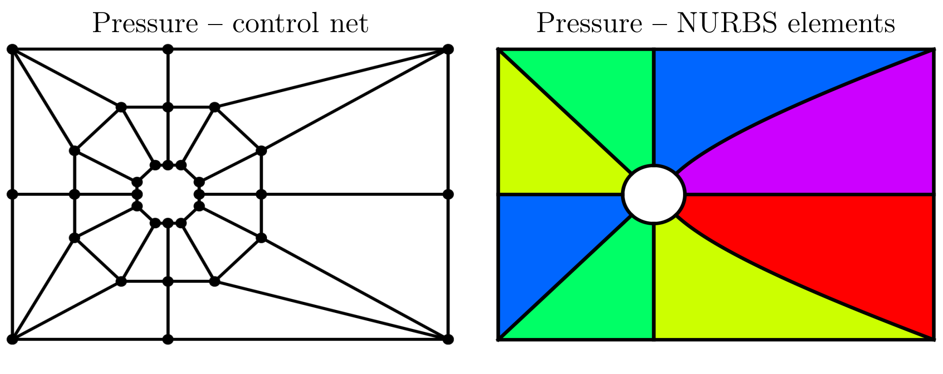
The initial Navier-Stokes problem was transformed to the sequential solving of linear systems.

3.2 LBB (Ladyženskaja-Babuška-Brezzi) condition

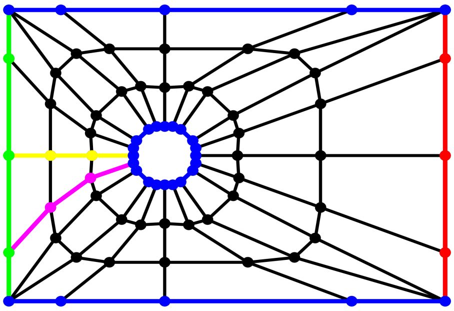
In general, it is not possible to use an arbitrary combination of discretizations for pressure and velocity for solving Stokes problem in order for given discretizations to be stable, it needs to satisfy the so-called LBB condition (or inf-sup condition). It can be shown that such a suitable choice of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

4 Examples

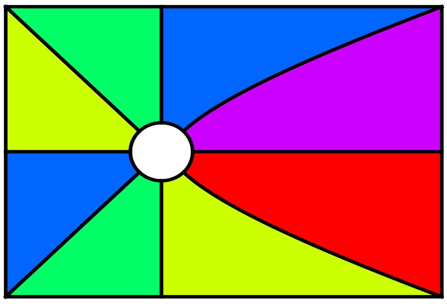
We will present a well-known test example, the so-called fluid flow past a cylinder with parabolic inflow boundary condition (left boundary), no-slip boundary condition on walls ($\mathbf{u} = \mathbf{0}$, upper and bottom boundary) and homogeneous Neumann condition at the outflow (right boundary).



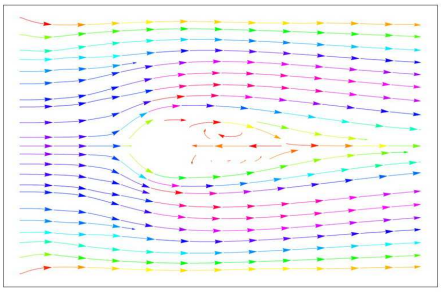
Velocity – control net



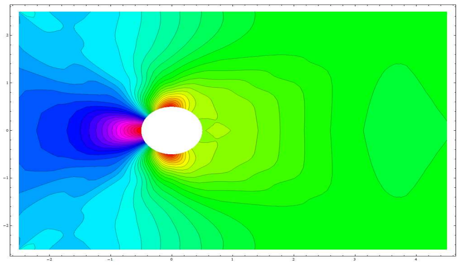
Velocity – NURBS elements



Navier-Stokes problem – velocity



Navier-Stokes problem – pressure



5 Conclusion

Iterative solution of stationary Navier-Stokes equations converges only for relatively low Reynolds numbers. Therefore, it is necessary to use stabilization methods (e.g. SUPG, PSPG, see [2]). The problems with oscillations can be solved by the SOLD methods [3]. The following steps of this project will be focused on turbulence modelling and transient case described by non-stationary Navier-Stokes equations.

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

References

[1] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs: *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement.* Computer Methods in Applied Mechanics and Engineering, 194, 4135–4195, 2005.

[2] A. N. Brooks, T. J. R. Hughes:*Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations.* Computer Methods in Applied Mechanics and Engineering 32, 199–259, 1982.

[3] V. John, P. Knobloch: *On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations: Part I – A review.* Computer Methods in Applied Mechanics and Engineering, 196, 2197–2215, 2007.

GPU implementation of the finite element method

P. Bauer, V. Klement, T. Oberhuber, V. Žabka

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

1 Introduction

Numerical approximation of partial differential equations by means of the finite element method leads to a system of linear equations, generally

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

In case of some nonlinear evolution problems and implicit time-stepping schemes, the system matrix $\mathbf{A}$ depends on time. During the numerical solution of such problems, the system matrix has to be updated after each time step. When implementing the solver on the GPU, special attention has to be paid to the matrix update because it can significantly affect performance of the implementation.

The following three basic approaches to deal with the matrix update on the GPU should be considered. First, the matrix can be assembled from the local element matrices on the CPU and then copied to GPU memory. However, memory transfers between CPU memory and GPU memory are slow and this approach is efficient only if the memory transfers overlap with some computations. Second, the system matrix does not have to be assembled at all. Instead, specific methods operating only on local element matrices can be employed for the solution of (1). The system matrix assembly is then replaced with several less expensive vector disassembly and assembly operations [4, 5]. The third approach is to assemble the system matrix entirely on the GPU [1].

This paper investigates a way of assembling the system matrix by the finite element method entirely on the GPU as described in [1]. We present a general CUDA implementation of the finite element matrix assembly relying on a coloring of the computational mesh. There are no particular limitations on the mesh, so it can be unstructured. We validate the CUDA implementation by comparison with a corresponding CPU implementation.

2 Implementation of the finite element method

Our implementation of the finite element method follows the same pattern as that in DUNE-FEM [2] and DUNE-PDELab [3]. The system matrix is assembled from the local element matrices. Each local element matrix consists of entries computed by integration of the basis functions and their derivatives over a single element. These integrals involving the basis functions are transformed by a geometry transformation to integrals over the reference element and evaluated using quadratures. Hence, the values of the basis functions and their derivatives in the quadrature points on the reference element are only needed. More details can be found in [6].

We implement the algorithm in CUDA. On the GPU, the assembly of the matrix should be done in parallel. For that purpose the mesh is colored (on the CPU) in such a way that elements of the same color do not share any degrees of freedom. Thus, at least elements of the same color can be processed in parallel.

Each CUDA thread is assigned to one element. It computes one entry of the local element matrix, immediately adds it to the corresponding entry of the system matrix and continues with the next entry. When computing the entries of the local matrix, the thread reads the quadrature weights, the basis function values in the quadrature points, the derivatives of the basis functions in the quadrature points, the Jacobian matrices of the geometry transformation in the quadrature points and the indices of the local matrix entries in the system matrix. In our implementation, the quadrature weights, the basis function values and the derivatives are stored in the constant memory space on the GPU because they are the same for all the elements. The Jacobian matrices of the geometry transformations and the indices of the local matrix entries in the system matrix are different for each element and, therefore, they are stored in the global memory space. To achieve maximum GPU memory bandwidth, the data in global memory are structured so that the memory accesses are coalesced.

The system matrix is sparse. We assume that its nonzero pattern does not change in time. Therefore, the matrix can be allocated and the column indices of its entries precomputed on the CPU which is more suitable for this task than the GPU. Similarly, the indices of the local matrix entries in the system matrix and the Jacobian matrices of the geometry transformations can also be precomputed on the CPU prior to the matrix assembly and transferred to GPU memory.

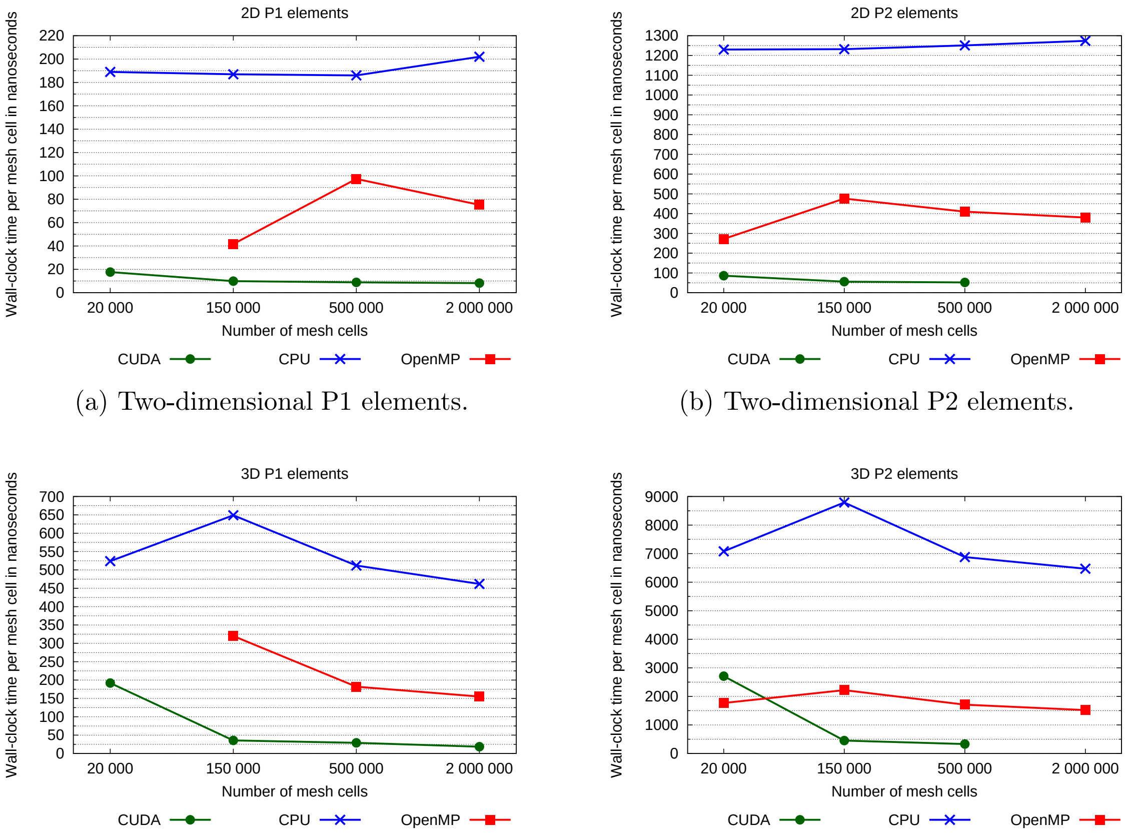
3 Results and conclusion

In order to test the presented GPU implementation of the FEM matrix assembly, we also created corresponding sequential and OpenMP CPU implementations. We applied all the implementations to the numerical solution of the heat equation employing the backward Euler method for the time discretization. We tested them on four different two-dimensional triangular meshes and four different three-dimensional tetrahedral meshes consisting of approximately 20 000, 150 000, 500 000 and 2 000 000 cells. The P1 and P2 finite elements were considered. The resulting matrices assembled by all the three implementations were identical.

We also compared the running times of each implementation. The GPU implementation ran on NVIDIA GeForce GTX 590 (only 1 GPU used), and the CPU implementations ran on AMD Phenom II X6 1090T (6 cores, 3.2 GHz). All the computations were performed using double-precision floating point arithmetic. The results are shown in Figure 1. The times measured do not include computations that can be done in advance and only once for the given mesh, i.e., mesh coloring, initialization of the Jacobian matrices of the geometry transformations and determination of the local matrix entries indices in the system matrix, and the transfer of these data to GPU memory.

It can be seen that the GPU implementation performed 3–10 times better than the OpenMP CPU implementation and up to 25 times better than the sequential CPU implementation. However, the CPU implementation might be more efficient if the mesh contains only a small number of cells (typically less than several thousand cells). In addition, the GPU might be limited by the amount of memory available when processing large amounts of data.

Acknowledgement: This work has been supported by the grant No. SGS11/161/OHK4/3T/14 of the Student Grant Agency of the Czech Technical University in Prague and the project No. TA01020871 of the Technological Agency of the Czech Republic.



(a) Two-dimensional P1 elements.

(b) Two-dimensional P2 elements.

(c) Three-dimensional P1 elements.

(d) Three-dimensional P2 elements.

Figure 1: Comparison of the three FEM matrix assembly implementations: implementation in CUDA, sequential CPU implementation and OpenMP CPU implementation using 6 cores. Results of the GPU computation using the P2 elements on the largest mesh are not available because the GPU ran out of memory. The OpenMP CPU implementation using the P1 elements on the smallest mesh performed much worse than the sequential CPU implementation. These results are thus omitted from the graphs.

References

[1] C. Cecka, A. J. Lew, E. Darve: *Assembly of finite element methods on graphics processors.* International Journal for Numerical Methods in Engineering, 85, (5), 640–669, 2011.

[2] A. Dedner, R. Klöfkorn, M. Nolte, M. Ohlberger: *A generic interface for parallel and adaptive scientific computing: abstraction principles and the DUNE-FEM module.* Computing, 90, (3–4), 165–196, 2010.

[3] DUNE team: *DUNE-PDELab howto.* March 2013. Downloaded from `http://www.dune-project.org/pdelab/`.

[4] R. Mafi, S. Sirouspour: *GPU-based acceleration of computations in nonlinear finite element deformation analysis.* Inter. J. for Numerical Methods in Biomedical Engineering, 2013.

[5] G. Markall, A. Slemmer, D. A. Ham, P. H. J. Kelly, C. D. Cantwell and S. J. Sherwin: *Finite element assembly strategies on multi-and many-core architectures.* International Journal for Numerical Methods in Fluids, 2011.

[6] V. Žabka, T. Oberhuber: *Implementation of the finite element method for the heat equation.* In Proceedings of Doktorandské dny 2013, CTU Publishing House, Prague, 321–330, 2013.

Tvarová optimalizace pro 2D kontaktní problém s Coulombovým třením s koeficientem tření závislým na řešení

P. Beremlijski, J. Haslinger, J. Outrata, R. Pathó*

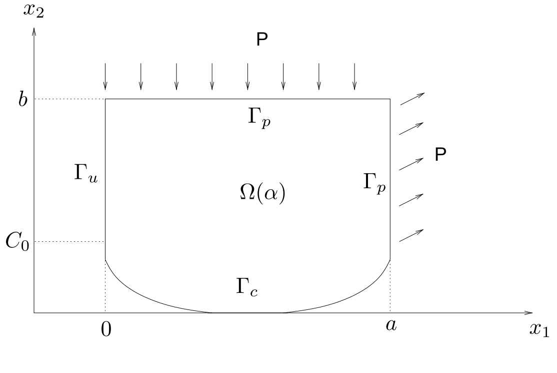
*Centrum excelence IT4Innovations a Katedra aplikované matematiky
Vysoká škola báňská - Technická univerzita Ostrava

1 Úvod

Článek se věnuje diskretizované úloze tvarové optimalizace pro dvojrozměrné pružné těleso v jednostranném kontaktu s tuhou překážkou. Stavová úloha je v tomto případě dána jako Signoriniho problém s Coulombovým třením, kde koeficient tření je závislý na řešení. Tento koeficient navíc nemusí být popsán diferencovatelnou funkcí, ale pouze lokálně lipschitzovskou funkcí. Při splnění jistých podmínek pro koeficient tření má diskrétní kontaktní úloha jediné řešení a toto řešení je závislé lokálně lipschitovsky na návrhové proměnné popisující tvar pružného tělesa. Díky jedinému řešení diskrétní úlohy pro fixovanou řídící proměnnou, můžeme použít tzv. přístup implicitního programování. Ten je založen na minimalizaci nehladké funkce složené z cenové funkce a jednoznačného zobrazení, které návrhové proměnné přiřazuje řešení diskrétní úlohy. Pro minimalizaci nehladké funkce lze použít některou z verzí bundle trust metody. K citlivostní analýze je nutné použít Morduchovičův kalkul. Na závěr příspěvku je ilustrováno použití našeho přístupu. Podrobně se lze s uvedeným přístupem seznámit v [3].

2 Stavová úloha

Nechť $\Omega \subset \mathbb{R}^2$ je pružné těleso s lipschitzovskou hranicí $\partial\Omega$. Hranice $\partial\Omega$ je složena ze tří nepřekrývajících se částí Γ_u, Γ_p a Γ_c (viz obrázek 1).



Obrázek 1: 2D pružné těleso.

Γ_u je hranice s Dirichletovskou podmínkou, na hranici Γ_p působí povrchové síly $P = (P_1, P_2)$, kde $P \in L^2(\Gamma_p)$. Těleso je zdola „podepřeno“ podél hranice Γ_c tuhou překážkou. Tvar Γ_c je určen *návrhovou proměnnou* $\boldsymbol{\alpha} \in \mathbb{R}^d$, přičemž množinu přípustných návrhových proměnných označíme $\mathcal{U}_{ad}$, tzn. $\boldsymbol{\alpha} \in \mathcal{U}_{ad}$. Na této hranici je předepsáno Coulombovo tření s koeficientem tření závislým na řešení $\mathcal{F} : \mathbb{R}_+ \to \mathbb{R}_+$. Navíc platí, že $\mathcal{F}$ je lipschitzovské v $\mathbb{R}_+$.

Algebraická formulace diskrétního Signoriniho problému s Coulombovým třením s koeficientem tření závislým na řešení je následující

$$\left.\begin{array}{l} \text{Najděte } (\boldsymbol{u},\boldsymbol{\lambda}) := (\boldsymbol{u}(\boldsymbol{\alpha}),\boldsymbol{\lambda}(\boldsymbol{\alpha})) \in \mathbb{R}^m \times \mathbb{R}^p_+ : \\ (\boldsymbol{A}(\boldsymbol{\alpha})\boldsymbol{u}, \boldsymbol{v}-\boldsymbol{u})_m + (\mathcal{F}(|\boldsymbol{u}_t|) \bullet \boldsymbol{\lambda}(\boldsymbol{\alpha}), |\boldsymbol{v}_t| - |\boldsymbol{u}_t|)_p \\ \qquad \geq (\boldsymbol{L}(\boldsymbol{\alpha}), \boldsymbol{v}-\boldsymbol{u})_m + (\boldsymbol{\lambda}, \boldsymbol{v}_n - \boldsymbol{u}_n)_p \quad \forall \boldsymbol{v} \in \mathbb{R}^m, \\ (\boldsymbol{\mu}-\boldsymbol{\lambda}, \boldsymbol{u}_n + \boldsymbol{\alpha})_p \leq 0 \quad \forall \boldsymbol{\mu} \in \mathbb{R}^p_+, \end{array}\right\} \tag{1}$$

kde $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ a $\boldsymbol{L} \in \mathbb{R}^m$ jsou matice tuhosti a vektor sil závislé na řídící proměnné $\boldsymbol{\alpha}$, $\boldsymbol{a} \bullet \boldsymbol{b} := (a_1b_1, \ldots, a_pb_p) \in \mathbb{R}^p$, $\boldsymbol{a} = (a_1, \ldots, a_p)$, $\boldsymbol{b} = (b_1, \ldots, b_p)$, $\boldsymbol{u}$ označuje vektor posunutí, kde $\boldsymbol{u}_t$ označuje tečné a $\boldsymbol{u}_n$ normálové posunutí, a $\boldsymbol{\lambda} \in \mathbb{R}^p_+$ (p je počet kontaktních uzlů) je vektor Lagrangeových multiplikátorů. Vektor $(\boldsymbol{u}, \boldsymbol{\lambda})$ nazveme *stavovou proměnnou.*

Dále zredukujeme naši úlohu a budeme se zabývat pouze kontaktními uzly. Stavová úloha realizuje zobrazení $\mathcal{S} : \boldsymbol{\alpha} \in \mathbb{R}^d \to (\boldsymbol{u}_t, \boldsymbol{u}_n, \boldsymbol{\lambda}) \in \mathbb{R}^{3p}$ (řídícímu vektoru $\boldsymbol{\alpha} \in U_{ad}$ je přiřazeno řešení kontaktní úlohy $(\boldsymbol{u}_t, \boldsymbol{u}_n, \boldsymbol{\lambda})$). Diskretizovanou stavovou úlohu lze ekvivalentně popsat zobecněnou rovností (podrobně v [1] a [2]).

$$\left.\begin{array}{l} \boldsymbol{0} \in \boldsymbol{A}_{tt}(\boldsymbol{\alpha})\boldsymbol{u}_t + \boldsymbol{A}_{tn}(\boldsymbol{\alpha})\boldsymbol{u}_n - \boldsymbol{L}_t(\boldsymbol{\alpha}) + Q_t(\boldsymbol{u}_t, \boldsymbol{\lambda}) \\ \boldsymbol{0} = \boldsymbol{A}_{nt}(\boldsymbol{\alpha})\boldsymbol{u}_t + \boldsymbol{A}_{nn}(\boldsymbol{\alpha})\boldsymbol{u}_n - \boldsymbol{A}_{nn}(\boldsymbol{\alpha})\boldsymbol{u}_n - \boldsymbol{L}_n(\boldsymbol{\alpha}) - \boldsymbol{\lambda} \\ \boldsymbol{0} \in \boldsymbol{u}_n + \boldsymbol{\alpha} + N_{\mathbb{R}^p_+}(\boldsymbol{\lambda}), \end{array}\right\} \tag{2}$$

kde multifunkce $Q_t : \mathbb{R}^p \times \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ je definována jako $\big(Q_t(\boldsymbol{x}, \boldsymbol{z})\big)_i := \mathcal{F}(|x_i|)z_i\partial|x_i|$, $\forall i = 1, \ldots, p$ $\forall \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^p$ a $N_{\mathbb{R}^p_+}(\cdot)$ je standardní normálový kužel.

3 Tvarová optimalizace pro kontaktní úlohu s Coulombovým třením s koeficientem tření závislým na řešení

Cílem úlohy tvarové optimalizace je nalezení takové návrhové proměnné $\boldsymbol{\alpha}$ (určující Beziérovu funkci, kterou je modelována kontaktní hranice Γ_c), pro kterou nabývá cenový funkcionál $J(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha}))$ svého minima. Úlohu diskrétní tvarové optimalizace zavedeme jako úlohu

$$\min_{\boldsymbol{\alpha} \in \mathcal{U}_{ad}} \mathcal{J}(\boldsymbol{\alpha}), \quad \mathcal{J}(\boldsymbol{\alpha}) := J(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})), \tag{3}$$

kde funkcionál J je spojitě diferencovatelný. K řešení této obecně nehladké úlohy byla použita bundle trust metoda (podrobně viz [7]). Tato iterační metoda potřebuje rutinu, která v každé iteraci vypočte hodnotu cenového funkcionálu (k tomu potřebujeme vyřešit stavovou úlohu) a jeden (libovolný) Clarkeův subgradient z Clarkeova zobecněného gradientu $\partial\mathcal{J}(\boldsymbol{\alpha})$. Pro jeho nalezení použijeme tvrzení

$$\partial\mathcal{J}(\boldsymbol{\alpha}) = \nabla_1 J(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})) + \{\boldsymbol{C}^T \nabla_2 J(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})),\ \boldsymbol{C} \in \partial\mathcal{S}(\boldsymbol{\alpha})\} \tag{4}$$

(viz [4]). Protože platí $\{\boldsymbol{C}^T \boldsymbol{y}^* | \boldsymbol{C} \in \partial\mathcal{S}(\boldsymbol{\alpha})\} \supset D^*\mathcal{S}(\boldsymbol{\alpha})(\boldsymbol{y}^*)$ pro všechna $\boldsymbol{y}^*$, stačí nalézt jeden prvek z množiny $D^*\mathcal{S}(\boldsymbol{\alpha})(\nabla_2 J(\boldsymbol{\alpha}, \mathcal{S}(\boldsymbol{\alpha})))$. Prvky limitní koderivace $D^*\mathcal{S}(\boldsymbol{\alpha})$ najdeme použitím nehladkého kalkulu B. Morduchoviče (viz [6]). Podrobně v [3].

4 Numerický příklad

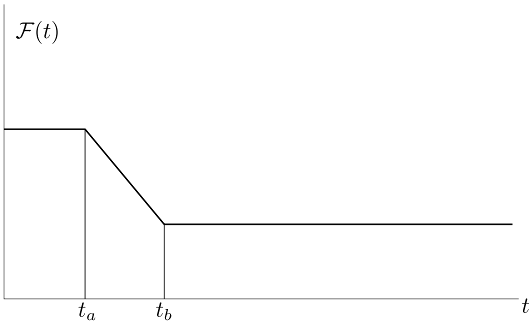
Pro numerické řešení stavové úlohy byla použita metoda postupných aproximací. Numerické řešení stavové i tvarově-optimalizační úlohy bylo implementováno v knihovně MatSol (viz [5]). Tato knihovna byla vyvinuta v prostředí Matlab.

Dříve navržený postup byl použit pro řešení úlohy tvarové optimalizace mající za cíl nalézt tvar kontaktní hranice tak, aby se vektor $\boldsymbol{\lambda}(\boldsymbol{\alpha})$ blížil co nejvíce předepsanému vektoru $\overline{\boldsymbol{\lambda}}$:

$$\min_{\boldsymbol{\alpha} \in \mathcal{U}_{ad},} \quad \|\boldsymbol{\lambda}(\boldsymbol{\alpha}) - \overline{\boldsymbol{\lambda}}\|_6^6 \tag{5}$$

Koeficient tření $\mathcal{F}$ (viz obr. 2) je pro naší úlohu definován takto

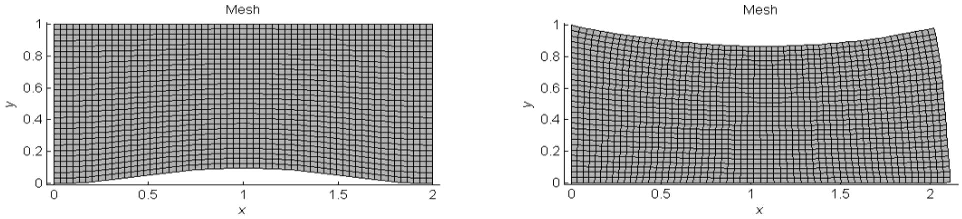
$$\mathcal{F}(t) := \begin{cases} 0.25 & \text{pro } t \in \langle 0, 0.01\rangle, \\ 0.25 \cdot (-60t + 1.6) & \text{pro } t \in (0.01, 0.02), \\ 0.1 & \text{pro } t \in \langle 0.02, \infty). \end{cases} \tag{6}$$



Obrázek 2: Definice koeficientu tření.

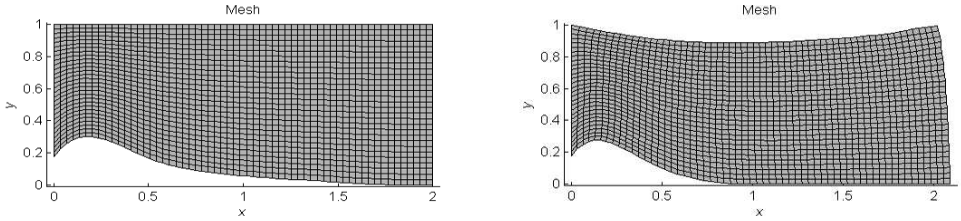
Naši oblast jsme diskretizovali sítí s 1800 uzly, její velikost je 2x1. Povrchové tlaky na hranici Γ_p jsou předepsány takto $\boldsymbol{P}^1 = (0; -60 \text{ MPa})$ na $(0, 1.8) \times \{1\}$ a $\boldsymbol{P}^1 = (0; 0)$ na $(1.8, 2) \times \{1\}$, zatímco $\boldsymbol{P}^2 = (30 \text{ MPa}; 10 \text{ MPa})$ na $\{2\} \times (0, 1)$. Fyzikální parametry oblasti mají tyto hodnoty – Youngův modul $E = 1$ GPa a Poissonova konstanta $\nu = 0.3$. Dimenze návrhové proměnné $\boldsymbol{\alpha}$ řídící Beziérovu funkci, kterou je dána hranice Γ_c, je 20.

Počáteční návrh a jeho deformace je na obrázku 3.

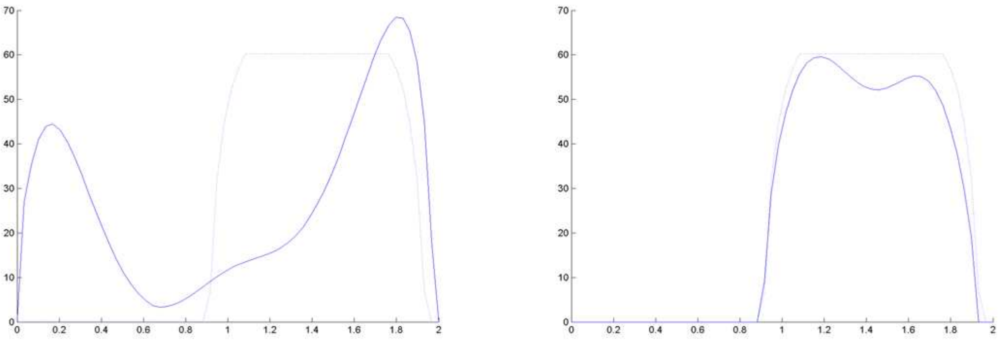


Obrázek 3: Počáteční návrh.

Obrázek 4 zobrazuje optimalizovaný návrh a jeho deformaci.



Obrázek 4: Optimalizovaný návrh.



Obrázek 5: Rozložení normálového napětí na kontaktní hranici pro počáteční návrh (vlevo) a optimalizovaný návrh (vpravo).

Rozložení normálového napětí $\boldsymbol{\lambda}(\boldsymbol{\alpha})$ na kontaktní hranici (plná čára) i předepsaný vektor $\overline{\boldsymbol{\lambda}}$ (tečkovaná čára) pro počáteční i optimalizovaný tvar tělesa jsou zobrazeny na obrázku 5.

Hodnota cenového funkcionálu pro počáteční návrh je $1.0746 \cdot 10^{14}$, zatímco hodnota cenového funkcionálu pro výsledný návrh je $4.7879 \cdot 10^{9}$.

Acknowledgement: Tato práce byla podpořena Evropským fondem regionálního rozvoje (ERDF) v rámci projektu Centra excelence IT4Innovations (CZ.1.05/1.1.00/02.0070) a projektem SPOMECH - Vytvoření multidisciplinárního vědeckovýzkumného týmu pro spolehlivé řešení úloh mechaniky, reg. č. [4] CZ.1.07/2.3.00/20.0070 v rámci Operačního programu Vzdělávání pro konkurenceschopnost a financovaného ze strukturálních fondů EU a státního rozpočtu ČR.

Reference

[1] P. Beremlijski, J. Haslinger, M. Kočvara, J. Outrata: *Shape optimization in contact problems with Coulomb friction.* In: SIAM Journal on Optimization 12, (3), 561–587, 2002.

[2] P. Beremlijski, J. Haslinger, M. Kočvara, R. Kučera, J. Outrata: *Shape optimization in three-dimensional contact problems with Coulomb friction.* In: SIAM Journal on Optimization 20, (1), 416–444, 2009.

[3] P. Beremlijski, J. Haslinger, J. Outrata, R. Pathó: *Shape optimization in contact problems with Coulomb friction and a solution-dependent friction coefficient.* In: SIAM Journal on Control and Optimization (submitted).

[4] F. H. Clarke: *Optimization and nonsmooth analysis.* J. Wiley & Sons, 1983.

[5] T. Kozubek, A. Markopoulos, T. Brzobohatý, R. Kučera, V. Vondrák, Z. Dostál: *MatSol – MATLAB efficient solvers for problems in engineering.* `http://industry.it4i.cz/en/products/matsol/`.

[6] B. S. Mordukhovich, *Variational analysis and generalized differentiation.* Volumes I and II. Springer-Verlag, 2006.

[7] J. Outrata, M. Kočvara, J. Zowe: *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results.* Kluwer Acad. Publ., 1998.

High performance computing in micromechanics

R. Blaheta, R. Hrtus, O. Jakl, J. Starý

Institute of Geonics AS CR, Ostrava

1 Introduction

By micromechanics we understand analysis of the macroscale response of materials through investigation of processes in their microstructure. Here by the macroscale, we mean the scale of applications, where we solve engineering problems involving materials like different metals and composites in aircraft design or rocks and concrete in a dam construction. Different applications are characterized by different characteristic size. At macroscale the materials mostly look as homogeneous or they are idealized as homogeneous or piecewise homogeneous. A substantial heterogeneity is hidden and appears only after more detailed zooming view into the material. This hidden heterogeneity can be called a microstructure. In metals it is created by crystals and grains, in composite materials by matrix and inclusions, in concrete by gravel and mortar or iron reinforcement etc. When the ratio between the characteristic dimensions on macro and microstructure subjects is sufficiently large, then we say that the scales are well separated. In this case, it is not possible to perform the macroscale analysis going into the microstructure details, but it is possible to analyse the macroscopic problems with the use of effective (homogenized) material properties, which are obtained by testing smaller samples of materials. In computational micromechanics, the testing of such samples means solution of boundary value problems on test domains involving the microstructure with loading provided by suitable boundary conditions.

We focus on X-ray CT image based micromechanics of geomaterials with the use of continuum mechanics and the finite element computation of the microscale strains and stresses, see [2]. This means that basic information about the microstructure is provided by analysing (segmentation) of 3D images of real samples. This information should be completed by information on local material properties, i.e. material properties of the individual material constituents.

There is a strong need for high performance parallel computing at several stages of the computational micromechanics, namely at

- analysis of CT scans,
- high resolution finite element solution of boundary value problems,
- solution of inverse problems for determination or calibration of local material properties.

In this contribution, we focus on the second point, i.e. solving the high resolution finite element systems with tens or hundreds degrees of freedom on available parallel computers at the Institute of Geonics and the IT4Innovations supercomputer centre in Ostrava. Following [3], we describe efficiency of the in-house GEM solvers exploiting the Schwarz domain decomposition method with aggregation by performing computational experiments on the above parallel computers. The solution of these systems is also necessary for building efficient solution methods for inverse material identification problems, see [4] and a further work in progress.

2 High resolution FEM systems and GEM solvers

In analysis of geocomposites (see [2]), the domain Ω is a cube with a relatively complicated microstructure. The FEM mesh is constructed on the basis of CT scans. As benchmarks, we shall use FEM systems arising from CT scanning of a coal-resin geocomposite at CT-lab of the Institute of Geonics. The characteristics of two benchmarks can be seen in Table 1.

Benchmark	Discretization	Size in DOF	Data size
GEOC-2s	257×257× 257	50 923 779	8.5 GB
GEOC-2l	257×257×1025	203 100 675	33.5 GB

Table 1: Benchmarks representing microstructures of two geocomposite samples. Notation, applied discretization meshes and sizes of resulting linear systems.

The elastic response of a representative volume Ω is characterized by homogenized elasticity C or compliance S tensors ($S = C^{-1}$). The elasticity and compliance tensors are determined from the relations

$$C\langle\varepsilon\rangle = C\varepsilon_0 = \langle\sigma\rangle \text{ and } S\langle\sigma\rangle = S\sigma_0 = \langle\varepsilon\rangle, \tag{1}$$

respectively. Here $\langle\sigma\rangle$ and $\langle\varepsilon\rangle$ are volume averaged stresses and strains computed from the solution of elasticity problem

$$-\mathrm{div}(\sigma) = 0, \quad \sigma = C_m\varepsilon, \quad \varepsilon = (\nabla u + (\nabla u)^T)/2 \quad \text{in } \Omega, \tag{2}$$

with boundary conditions

$$u(x) = \varepsilon_0 \cdot x \text{ on } \partial\Omega \text{ and } \sigma \cdot n = \sigma_0 \cdot n \text{ on } \partial\Omega, \tag{3}$$

respectively. Above, σ and ε denote stress and strain in the microstructure, C_m is the variable local elasticity tensor, u and n denote the displacement and the unit normal, respectively. The use of pure Dirichlet and pure Neumann boundary conditions allows us to get a upper and lower bounds for the upscaled elasticity tensor, see e.g. [2].

By using the GEM software [1], the domain is discretized by linear tetrahedral finite elements. The arising systems are then solved by PCG method with a stabilization in the singular case (see [3]). The implementation in the GEM software uses two solvers:

GEM-DD is a solver implemented in the GEM software. It uses one-level additive Schwarz domain decomposition preconditioner with subproblems replaced by displacement decomposition incomplete factorization, see ref. in [3]. The resulting preconditioner is symmetric positive definite even for the singular case.

GEM-DD-CG solver differs in preconditioning, which is now a two-level Schwarz domain decomposition arising from the previous GEM-DD by additive involvement of a coarse problem correction. The coarse problem is created by a regular aggressive aggregation with 3 DOF's per aggregation. In singular case, the coarse problem is also singular with a smaller null space containing only the rigid shifts. The coarse problem is solved only approximately by inner (not stabilized) CG method with a lower solution accuracy - relative residual accuracy $\varepsilon_0 \leq 0.01$.

Note that in the computational experiments described in the next Section, we solve the problems with pure Neumann boundary conditions.

3 Parallel computers and computational experiments

The computational experiments are performed on two computers:

Enna - 64-core NUMA multiprocessor at the Institute of Geonics:

- eight octa-core Intel Xeon E7-8837/2.66 GHz processors
- 256 GB of DDR2 RAM
- CentOS 6.3, Intel Cluster Studio XE 2013, Trilinos 11.4.1

Anselm - multicomputer (cluster) with 207 compute nodes at the Supercomputing Center IT4Innovations. We employed the computing nodes equipped with:

- two octa-core Intel E5-2665/2.4 GHz processors
- 64 GB of memory and 500 GB of local disk capacity
- Infiniband QDR interconnection, fully non-blocking, fat-tree
- Bullx Linux OS (Red Hat family), Intel Parallel Studio XE 2013

Table 2 shows the timings of GEM solvers (without and with coarse grid problem applied) obtained for GEOC2s, i.e. a problem of more than 50 million DOF's, where the performance up to 64 processing elements on Enna and up to 128 processing elements on Anselm could be compared. The stopping criterion was $\|r\|/\|b\| \leq \varepsilon = 10^{-5}$ and the DD-CG solver made use of a coarse problem with aggregation factors 9×9×9 (81 000 DOF's).

	Enna				**Anselm**			
	DD		*DD-CG*		*DD*		*DD-CG*	
# Sd	# It	T_{iter}	# It	T_{iter}	A/E	T_{iter}	A/E	T_{iter}
2	914	8461.2	437	3523.1	0.67	5644.2	0.79	2785.4
4	1129	4973.3	428	1923.6	0.59	3526.2	0.72	1383.4
8	1421	2942.5	416	922.9	0.82	2422.6	0.79	725.7
16	1655	1994.6	376	415.8	0.64	1325.8	0.84	348.7
32	1847	1923.5	329	348.3	0.42	798.3	0.56	194.8
64	2149	3074.9	295	505.9	0.20	620.8	0.23	117.6
128					n/a	515.7	n/a	107.1

Table 2: Timings of the GEOC2s benchmark achieved by the GEM solvers on the multiprocessor Enna and cluster Anselm: Iteration counts (#It), wall-clock time (in seconds) of the solution (T_{iter}) and the corresponding performance ratio Anselm/Enna (A/E) are provided for up to 128 subdomains (# Sd).

For greater number of subdomains, the results confirm the advantage of systems with distributed memory, when the multiprocessors in general suffer from the memory-processor bandwidth contention. Thus, while on Enna the scalability fades out at about 32 cores, the turning point on Anselm is around 128 processing elements, when the small size of subdomains deteriorates the computation/communication ratio.

In absolute figures, we were able to solve the benchmark 3 − 4 times faster on Anselm than on Enna. The advantage of Anselm is to be derived partially from the fact that its newer Intel Sandy Bridge CPU architecture as such outperforms Enna's Westmere one, in our application

	Enna						Anselm	
	DD-9×9×9		*DD-9×9×18*		*DD-9×9×27*		*DD-9×9×27*	
# Sd	# It	T_{iter}	# It	T_{iter}	# It	T_{iter}	# It	T_{iter}
4	751	13719.0	858	15737.6	997	18518.4	997	12671.4
8	690	6237.7	800	6960.8	917	8062.9	917	5803.9
16	585	2717.4	674	4010.6	777	4815.6	777	2576.6
32	585	2483.6	622	2923.8	708	3452.5	708	1157.5
64					627	3637.0	627	558.8
128							652	358.5
256							631	299.6
512							649	333.5

Table 3: Timings of the GEOC2l benchmark achieved by the GEM-DD-CG solver on the multiprocessor Enna and cluster Anselm: Iteration counts(#It) and wall-clock time (in seconds) for the solution time (T_{iter}) are provided now for different sizes of CG problem involved in computations and for various numbers of subdomains (# Sd).

by 20-40%, what can be estimated from the test up to 8 processing elements (one socket) when the processors work in similar conditions.

Table 3 reports computations with the largest benchmark GEOC2l (about 200 million DOF) and demonstrates the impact of the coarse grid size on the time of the solution. We can observe that very aggressive aggregation leads to the best results. We could confirm this observation on Anselm, where the best time in the Table 3 (299.6 s with 256 processing elements and aggregation 9×9×27) was surpassed by an experiment with the coarser aggregation 15×15×31. The overall best GEOC2l solution time of 249.8 s was achieved after 910 iterations on # Sd=512 subdomains (32 compute nodes employed).

A bit surprising decrease of the number of iterations with increasing number of subdomains (processors) as reported in the above Tables, especially for DD-CG, can be explained by the fact that smaller subdomain problems are solved more accurately in our implementation.

Acknowledgement: This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

[1] R. Blaheta, O. Jakl, R. Kohut, J. Starý: *GEM – A platform for advanced mathematical geosimulations.* In: R. Wyrzykowski et al. (eds.), PPAM 2009, Part I, LNCS 6067, 266–275, 2010. and Parallel Computing PARA 2012, LNCS, Springer, accepted.

[2] R. Blaheta, R. Kohut, A. Kolcun, K. Souček, L. Staš: *Micromechanics of geocomposites: CT images and FEM simulations.* In: EUROCK 2013, pp. 399–404, Balkema 2013.

[3] R. Blaheta, O. Jakl, J. Starý, E. Turan: *Iterative solution of singular systems with applications.* Proceedings of the conference SNA 13, Institute of Geonics AS CR, 2013

[4] R. Hrtus, J. Haslinger, R. Blaheta: *Identification problems with a priori given material interfaces.* WOFEX 2013

Elliptic equations on non-compatible meshes of different dimension

J. Březina

Technical University of Liberec, Liberec

1 Introduction

Deep final repositories for nuclear waste are planned in rocks with low permeability, namely in granite, in order to minimize transport of a possible leakage to the surface. A regional model of water flow and transport processes in the granite has to deal with presence of relatively tiny fracture zones of significantly higher hydraulic permeability. One possible approach is to treat these fracture zones as an independent domain of lower dimension and introduce a coupling with a surrounding matrix.

For purpose of this contribution, we shall consider a model problem consisting of a 2d matrix domain Ω_2 and of a fracture domain Ω_1 formed by a network of 1d lines. We also set $\Omega_0 = \emptyset$. We shall consider saturated porous media on both domains described by the Darcy's law

$$\mathbf{v}_d = -\mathbb{K}_d \nabla h_d \quad \text{on } \Omega_d \setminus \Omega_{d-1} \quad \text{for } d = 1, 2; \tag{1}$$

and the continuity equation

$$\mathrm{div} \mathbf{q}_d = F_d \quad \text{on } \Omega_d \setminus \Omega_{d-1} \quad \text{for } d = 1, 2; \tag{2}$$

where $\mathbf{v}_d$ is the velocity $\mathbf{q}_d = \nu_d \mathbf{v}_d$ is the Darcy flux, $\nu_2 = 1$, and ν_1 is the fracture zone cross-section, Other quantities are: the tensor of hydraulic conductivity $\mathbb{K}_d$, the pressure head h_d, and partially integrated density of the water sources F_d. Vectors $\mathbf{q}_d$ and tensors $\mathbb{K}_d$ lives in the corresponding local tangent spaces of domains Ω_d. The principal unknowns of this system are the fluxes $\mathbf{q}_d$ and the pressure heads h_d. We prescribe Dirichlet boundary condition $h_d = H_d$ on the outer boundaries Γ_d of both domains. Furthermore, one boundary condition has to be posed for each of two sides of a line in Ω_1. First condition is the continuity of h_2 from both sides and the second is balance of the fluxes

$$\mathbf{q}_2^+ \cdot \mathbf{n}^+ + \mathbf{q}_2^- \cdot \mathbf{n}^- = Q_2 = \sigma_1 (\mathrm{Tr}\, h_2 - h_1). \tag{3}$$

Here Q_2 is the surface density of the local outer flux from Ω_2 into Ω_1, which is proportional to the difference between the trace of h_2 and h_1 with a given transition coefficient σ_1. The flux Q_2 also appears as a part of the volume source $F_1 = Q_2 + \nu_1 f_1$ on the domain Ω_1.

2 Discrete mixed-hybrid formulation

The coupling used in the model problem, namely continuity of the pressure, is physically relevant only when fracture zone has higher conductivity then matrix (cf. [1]), however it admits approximation of the trace of the pressure head h_2 on Ω_1 even in the case of non-compatible meshes. This is big advantage in real applications, where fracture zone can be complex and generation of compatible meshes becomes problematic.

Let $\mathcal{T}_d = \{T_d^i, i \in \mathcal{I}_d\}$ be a triangulation of the domain Ω_d and $\mathcal{E}_d = \bigcup_{i\in\mathcal{I}_d} \partial T_d^i \setminus \Gamma_d$ set of internal edges. We do not assume any relationship between $\mathcal{T}_2$ and $\mathcal{T}_1$. We shall denote by $V_d \subset H(div, \Omega_d)$ the space of discontinuous lowest order Raviart-Thomas functions (RT_0) and by $P_d \subset L^2(\Omega_d)$ the space of functions piecewise constant on elements of $\mathcal{T}_d$. Further, we introduce $\mathring{P}_d$, the space of functions piecewise constant on edges in $\mathcal{E}_d$. Finally, we denote $V = V_2 \times V_1$ and $P = P_2 \times P_1 \times \mathring{P}_2 \times \mathring{P}_1$.

We say that pair $(\mathbf{q}, \overline{h}) = \big(\mathbf{q}, (h, \mathring{h})\big) \in V \times P$ is mixed-hybrid solution of the problem P_{12} if it satisfies abstract saddle point problem

$$a(\mathbf{q}, \boldsymbol{\psi}) + b(\boldsymbol{\psi}, \overline{h}) = \langle G, \boldsymbol{\psi} \rangle \qquad \forall\, \boldsymbol{\psi} \in V, \tag{4}$$

$$b(\mathbf{q}, \overline{\phi}) - c(\overline{h}, \overline{\phi}) = \langle F, \overline{\phi} \rangle \qquad \forall\, \overline{\phi} = (\phi, \mathring{\phi}) \in P, \tag{5}$$

where the bilinear forms on the left-hand side are

$$\begin{aligned}
a(\mathbf{q}, \boldsymbol{\psi}) &= \sum_{d=1,2} \sum_{i\in\mathcal{I}_d} \int_{T_d^i} \frac{1}{\nu_d} \mathbf{q}_d^i \mathbb{K}_d^{-1} \boldsymbol{\psi}_d^i, \\
b(\mathbf{q}, \overline{\phi}) &= \sum_{d=1,2} \sum_{i\in\mathcal{I}_d} \left(\int_{T_d^i} -\mathrm{div}\mathbf{q}_d \, \phi_d + \int_{\partial T_d^i \setminus \Gamma_d} (\mathbf{q}_d \cdot \mathbf{n}) \mathring{\phi}_d \right), \\
c(\overline{h}, \overline{\phi}) &= \int_{\Omega_1} \sigma \big(R(\overline{h}_1) - T(\overline{h}_2) \big) \big(R(\overline{\phi}_1) - T(\overline{\phi}_2) \big)
\end{aligned}$$

and linear forms on the right-hand side are

$$\begin{aligned}
\langle G, \boldsymbol{\psi} \rangle &= \sum_{d=1,2} \sum_{i\in\mathcal{I}_d} \int_{\partial T_d^i} (\boldsymbol{\psi}_d \cdot \mathbf{n}) H_d, \\
\langle F, \overline{\phi} \rangle &= - \sum_{d=1,2} \int_{\Omega_d} \nu_d f_d \phi_d.
\end{aligned}$$

In the bilinear form c, we have used a reconstruction operator R and operator T approximating the trace of the pressure head h_2 on Ω_1. For more details of the mixed-hybrid formulation and extension to 3d we refer to [2].

3 Numerical experiments

Various choices of operators R and T are possible. In [3], two suitable choices have been proposed. Method $P0$ put simply $R(\overline{h}_1) = h_1$ and $T(\overline{h}_2)$ on element T_1^i is weighted average of h_2 values on intersecting 2d elements with weights proportional to the length of intersection. Method $P1$ use $\mathring{h}_1$ values to reconstruct piecewise linear approximation of the pressure head h_1 and $\mathring{h}_2$ values are interpreted as degrees of freedom of non-conforming $P1$ finite elements, then operator T simply takes trace of this non-conforming approximation on Ω_1. Both methods were tested on very simple geometry in [3] giving convergence rates summarized in Table 1.

Two new topics are covered in this contribution. For first, we better investigate true quality of the velocity field. Table 1 shows rather poor convergence of the velocity, but this is mainly caused by the local error around the fracture since discrete velocity field can not express a jump sitting on the fracture. We shall present numerical tests indicating that velocity field is in fact better out of the fracture. For the second, we shall present comparison of proposed non-compatible methods with a more general model which admits discontinuous pressure on the fracture, but

	pressure head				velocity			
	$\rho = 0.5$		$\rho = 1$		$\rho = 0.5$		$\rho = 1$	
	1d	2d	1d	2d	1d	2d	1d	2d
method P0	2.1	1.60	1.87	1.55	1.82	0.6	1.43	0.56
method P1	2.1	1.68	1.87	1.55	1.82	0.6	1.60	0.56
compatible mesh	1.9	1.9	1.9	1.9	1.9	1	1.9	1

Table 1: Estimated order of convergence of approximated L^2-error for the pressure head and the velocity.

can be applied only for compatible meshes. These comparisons are done on more realistic meshes involving complex fracture domain.

Acknowledgement: This work has been supported by the Technology Agency of the Czech Republic under the project no. TA01021331.

References

[1] V. Martin, J. Jaffrée, J. E. Roberts: *Modeling fractures and barriers as interfaces for flow in porous media.* SIAM Journal on Scientific Computing, 26, (5), 1667, 2005.

[2] J. Březina, M. Hokr: *Mixed-hybrid formulation of multidimensional fracture flow.* In: 7th International Conference, NMA 2010, Borovets, Bulgaria, August 20-24, 2010, pp. 125–132, Springer.

[3] J. Březina: *Mortar-like mixed-hybrid methods for elliptic problems on complex geometries.* In: 19th Conference on Scientific Computing, Algoritmy 2012, Vysoké Tatry – Podbanské, Slovakia, September 9-14, pp. 200–208.

Discontinuous Galerkin method for advection-diffusion equation in domains with fractures

J. Březina, J. Stebel

Technical University of Liberec

1 Introduction

The paper deals with mathematical modeling of transport of dissolved substances in a fractured rock massif. Our goal is to adapt a numerical method that will treat the following aspects of the model:

- *Complex geometry.* The rock matter, such as granite, contains system of thin layers (called fractures) that are difficult to be captured by elements of the same dimension.
- *Heterogeneity.* The real material contains zones with hydraulic conductivity and dispersion tensor differing by orders of magnitude.
- *Advection/diffusion dominance.* At various scales, the problem can have character of first order hyperbolic or second order elliptic PDE.

Throughout the paper, $\Omega^d \subset \mathbb{R}^d$, $d \in \{2,3\}$, will be a Lipschitz domain representing the massif. Our approach is based on explicit treating of the fracture as a $(d-1)$-dimensional manifold Ω^{d-1} inside the massif, i.e. $\Omega^{d-1} \subset \Omega^d$. Accordingly, for $k \in \{d-1, d\}$ we seek for the concentrations $u^k : [0,T] \times \Omega^k \to \mathbb{R}$ satisfying the advection-diffusion equation

$$\partial_t u^k - \operatorname{div}(\mathbb{A}^k \nabla u^k) + \operatorname{div}(\boldsymbol{b}^k u^k) = f^k \text{ in } \Omega^k, \tag{1a}$$

accomplished by the initial and boundary conditions

$$u^k(0,\cdot) = u_0^k \text{ in } \Omega^k, \tag{1b}$$

$$u^k = g^k \text{ on } (0,T) \times \Gamma_D^k, \tag{1c}$$

$$-\mathbb{A}^k \nabla u^k \cdot \boldsymbol{n}^k = h^k \text{ on } (0,T) \times \Gamma_N^k. \tag{1d}$$

Here $\mathbb{A}^k$, $\boldsymbol{b}^k$ is the diffusion tensor and the advection vector field, usually given as the Darcy velocity, $\boldsymbol{n}^k$ stands for the unit outward normal vector to $\partial\Omega^k = \Gamma_D^k \cup \Gamma_N^k$. The mass interchange between the domain and the fracture is realized through the interface condition

$$(-\mathbb{A}^d \nabla u^d + \boldsymbol{b}^d) \cdot \boldsymbol{n}^d = f^{d-1} = q(u^d, u^{d-1}), \tag{1e}$$

where

$$q(u^d, u^{d-1}) := \sigma(u^d - u^{d-1}) + (\boldsymbol{b}^d \cdot \boldsymbol{n}^k)^+ u^d - (\boldsymbol{b}^d \cdot \boldsymbol{n}^d)^- u^{d-1} \text{ on } (0,T) \times \Omega^{d-1},\ \sigma \geq 0, \tag{1f}$$

involves two mechanisms: interchange due to different concentrations (first term) and due to advection (second and third term). Here $f^+ = \max\{0, f\}$ and $f^- = -\min\{0, f\}$ is the positive and the negative part, respectively.

2 Weak formulation and well-posedness

For any $k \in \{d-1, d\}$ we introduce the space

$$V^k := \{v \in W^{1,2}(\Omega^k);\ v_{|\Gamma_D^k} = 0\}.$$

The couple (u^d, u^{d-1}) is said to be a weak solution to (1) if

- $(u^k - g^k) \in L^2(0,T;V^k)$, $\partial_t u^k \in L^2(0,T;(V^k)^*)$, $k \in \{d-1,d\}$;
- $u^k(0,\cdot) = u_0^k$, $k \in \{d-1,d\}$;
- for a.a. $t \in (0,T)$ and $v \in V^d$:

$$\langle \partial_t u^d, v\rangle_{V^d} + \Big(\mathbb{A}^d \nabla u^d - \boldsymbol{b}^d u^d \nabla v\Big)_{\Omega^d} + \Big(\boldsymbol{b}^d \cdot \boldsymbol{n}^d u^d v\Big)_{\Gamma_N^d} + \Big(q(u^d, u^{d-1})v\Big)_{\Omega^{d-1}} = \Big(f^d v\Big)_{\Omega^d} + \Big(h^d v\Big)_{\Gamma_N^d}\ ; \quad (2)$$

- for a.a. $t \in (0,T)$ and $v \in V^{d-1}$:

$$\langle \partial_t u^{d-1}, v\rangle_{V^{d-1}} + \Big(\mathbb{A}^{d-1} \nabla u^{d-1} - \boldsymbol{b}^{d-1} u^{d-1} \nabla v\Big)_{\Omega^{d-1}} + \Big(\boldsymbol{b}^{d-1} \cdot \boldsymbol{n}^{d-1} u^{d-1} v\Big)_{\Gamma_N^{d-1}} = \Big(f^{d-1} v\Big)_{\Omega^{d-1}} + \Big(h^{d-1} v\Big)_{\Gamma_N^{d-1}} + \Big(q(u^d, u^{d-1})v\Big)_{\Omega^{d-1}}. \quad (3)$$

Theorem 1. *Let $\mathbb{A}^k, \boldsymbol{b}^k, \boldsymbol{b}^k \cdot \boldsymbol{n}^k, \sigma$ be bounded, $f^k \in L^2((0,T)\times\Omega^k)$, $h^k \in L^2((0,T)\times\Gamma_N^k)$ and $\mathbb{A}^k$ be uniformly positive definite for $k \in \{d-1,d\}$. Then there exists a unique weak solution (u^d, u^{d-1}), satisfying the estimate*

$$\sup_{t\in(0,T)} \Big(\|u^d(t,\cdot)\|_{2,\Omega^d}^2 + \|u^{d-1}(t,\cdot)\|_{2,\Omega^{d-1}}^2\Big) + \int_0^T \Big(\|\nabla u^d\|_{2,\Omega^d}^2 + \|\nabla u^{d-1}\|_{2,\Omega^{d-1}}^2\Big) \le C,$$

where the constant $C > 0$ depends on the data.

3 Approximation

For the space semi-discretization we have chosen the discontinuous Galerkin (DG) method with weighted averages [1]. The main reason is its capability to treat purely advective as well as diffusive problems and its robustness with respect to space variations of the diffusion tensor. For the time discretization we use the implicit Euler scheme. We shall describe how the method has been extended to the present model.

Let τ, h be the time step and the space discretization parameter, respectively. We assume that $\mathcal{T}_h^k$ is a regular partition of Ω^k into simplices, $k \in \{d-1,d\}$. We define the set $\mathcal{E}_h^k$ of all edges of elements in $\mathcal{T}_h^k$. Further, $\mathcal{E}_{h,I}^k$, $\mathcal{E}_{h,B}^k$ will stand for interior and boundary edges, respectively, and $\mathcal{E}_{h,D}^k$ for edges that coincide with Γ_D^k. For an interior edge E we denote by $T^-(E)$ and $T^+(E)$ the elements sharing E. The unit normal vector $\boldsymbol{n}$ to E is assumed to point from $T^-(E)$ towards $T^+(E)$. We introduce the jump $[f] = f_{|T^-(E)} - f_{|T^+(E)}$, the average $\{f\} = \frac{1}{2}(f_{|T^-(E)} + f_{|T^+(E)})$ and the weighted average $\{f\}_\omega = \omega f_{|T^-(E)} + (1-\omega) f_{|T^+(E)}$. The weight ω is chosen in a specific way taking into account possible inhomogeneity of $\mathbb{A}^k$ (see [1] for details).

At each time instant $t_n = n\tau$ we search for the discrete solution $u^k_{h,n} \in V^k_h$, where

$$V^k_h = \{v : \overline{\Omega^k} \to \mathbb{R};\ v_{|T} \in P_1(T)\ \forall T \in \mathcal{T}^k_h\}$$

is the space of functions piecewise affine on the elements of $\mathcal{T}^k_h$, possibly discontinuous across the element interfaces. For $n = 1, 2, \ldots$, the discrete problem reads:

$$\frac{1}{\tau}\left(u^k_{h,n} - u^k_{h,n-1}v\right)_{\Omega^k} + a^k_{h,n}(u^k_{h,n}, v) + \tilde{a}^k_{h,n}(u^d_{h,n}, u^{d-1}_{h,n}, v) = b^k_{h,n}(v) \quad \forall v \in V^k_h.$$

The forms $a^k_{h,n}$, $\tilde{a}^k_{h,n}$, $b^k_{h,n}$ are defined as follows:

$$\begin{aligned}
a^k_{h,n}(u,v) = {} & \left(\mathbb{A}^k(t_n)\nabla u \nabla v\right)_{\Omega^k} - \left(\boldsymbol{b}^k(t_n) u \nabla v\right)_{\Omega^k} \\
& - \sum_{E \in \mathcal{E}^k_{h,I}} \left(\left(\left\{ \mathbb{A}^k(t_n)\nabla u \right\}_\omega \cdot \boldsymbol{n}^k [v] \right)_E + \theta \left(\left\{ \mathbb{A}^k(t_n)\nabla v \right\}_\omega \cdot \boldsymbol{n}^k [u] \right)_E \right) \\
& + \sum_{E \in \mathcal{E}^k_{h,I}} \left(\boldsymbol{b}^k(t_n) \cdot \boldsymbol{n}^k \{u\}[v] \right)_E + \sum_{E \in \mathcal{E}^k_{h,B}} \left(\boldsymbol{b}^k(t_n) \cdot \boldsymbol{n}^k uv \right)_E \\
& + \sum_{E \in \mathcal{E}^k_{h,I}} \gamma_E \left([u][v]\right)_E + \sum_{E \in \mathcal{E}^k_{h,D}} \gamma_E \left(uv\right)_E ,
\end{aligned}$$

$$\begin{aligned}
\tilde{a}^d_{h,n}(u^d, u^{d-1}, v) &= \left(q(u^d, u^{d-1})v\right)_{\Omega^{d-1}}, \quad v \in V^d_h, \\
\tilde{a}^{d-1}_{h,n}(u^d, u^{d-1}, v) &= -\left(q(u^d, u^{d-1})v\right)_{\Omega^{d-1}}, \quad v \in V^{d-1}_h, \\
b^k_{h,n}(v) &= \left(f^k v\right)_{\Omega^k} + \left(h^k v\right)_{\Gamma^k_N} + \sum_{E \in \mathcal{E}^k_{h,D}} \gamma_E \left(g^k(t_n)v\right)_E .
\end{aligned}$$

The value $\gamma_E > 0$ affects the inter-element jumps of the solution. The constant $\theta \in \{-1, 0, 1\}$ represents the nonsymmetric, incomplete and symmetric variant of the DG method.

In the case when an edge is shared by more than two elements, we consider the so-called ideal mixing, i.e. mass entering the edge through every inlet element ($\boldsymbol{b}^k$ points out of this element) is divided among all outlet elements proportionally to their fluxes.

4 Results

The numerical method described above has been implemented in the software Flow123d [2]. It is demonstrated on the following examples.

4.1 Simple 2D fractured domain

In the first example, the advection is given by a pressure driven flow field (see Figure 1). Diffusion is neglected. The hydraulic conductivity and consequently also the advection field is 10 times larger in the fractures.

The DG method was compared to the finite volume-explicit Euler scheme. The larger advection in fractures leads to a restriction of the time step in the explicit FV method. On the other hand, a significantly larger step used by DG method yields comparable results (see Figure 1).

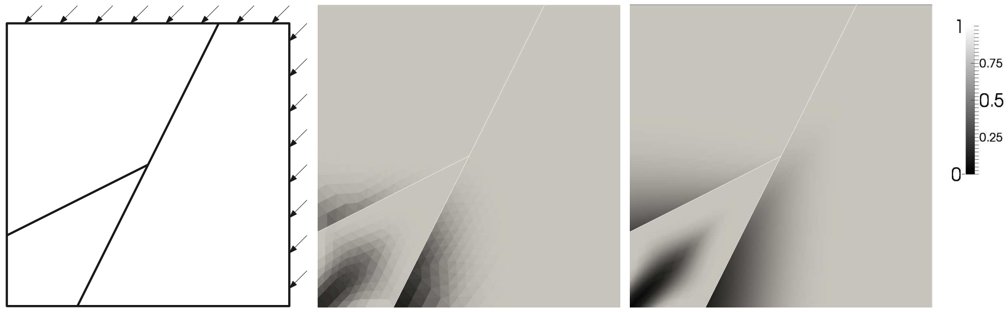


Figure 1: Transport in 2D fractured domain: geometry and direction of advection; FV solution at $t = 0.18$ ($\tau = 4 \cdot 10^{-4}$); DG solution at $t = 0.18$ ($\tau = 10^{-2}$).

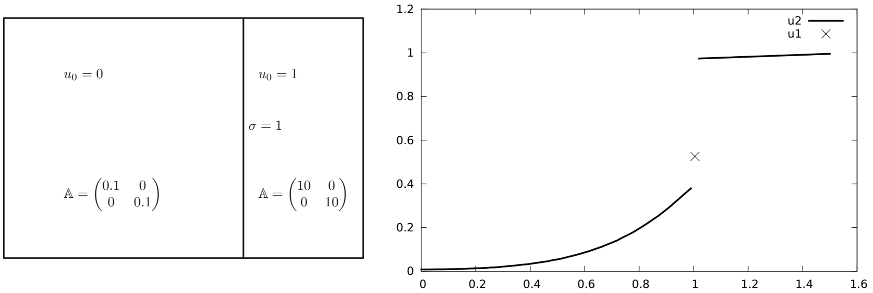


Figure 2: Diffusion through fracture: Geometry; solution at $t = 1$ along horizontal axis.

4.2 Diffusion through fracture

In the second example there is no advection present and the substance is transported by means of diffusion through the fracture. The diffusion tensor has different value in regins divided by the fracture. The performace of the DG method is preserved even for vanishing diffusion.

5 Conclusion

A model for the advection-diffusion equation governing the transport processes in domains with fractures has been presented and its well-posedness was established. The problem has been approximated by the discontinuous Galerkin method which has proven to be robust with respect to the size of advection vector and diffusion tensor as well as to their inhomogeneity.

Acknowledgement: This work has been supported by the Technology Agency of the Czech Republic under the project no. TA01021331.

References

[1] A. Ern, A. F. Stephansen, P. Zunino: *A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity.* IMA Journal of Numerical Analysis, 29, (2), 235–256, 2009.

[2] J. Březina, J. Stebel, J. Hnídek: *Flow123d — simulator of underground water flow and transport in fractured porous media.* Retrieved on 15 December 2013 from `https://dev.nti.tul.cz/trac/flow123d`.

Partition of unity methods for approximation of point sources in porous media

P. Exner

Faculty of Mechatronics, Informatics and Interdisciplinary Studies
Technical University of Liberec

1 Introduction

People often consider in their models of flow in porous media very large areas which can contain various phenomenons of very small scale compared with the size of the areas. These can be some disruptions of the porous media, e.g. cracks and wells, or material inhomogeneities which cause large gradients in pressure head and velocity or even their discontinuities.

Using the standard FEM (Finite Element Method) we are unable to properly approximate the quantities in the vicinity of these disturbances, unless we introduce cells of the same scale in the mesh. This leads to very fine meshes which highly increase computational costs. We use PU (Partition of Unity) methods to overcome this problem and demonstrate it on a steady quasi-three-dimensional model of multi-aquifer system containing hydro-geological wells which cause singularities in solution. We follow the work [5] of R. Gracie and J. R. Craig who have already used the XFEM (eXtended FEM) on a similar model.

We have implemented both the XFEM and SGFEM (Stable Generalized FEM) for our problem, using the Deal II library [2].

2 Model description

We consider steady flow in a system of aquifers (2D layers of given thickness) which are separated by layers with low permeability (aquitards). We suppose the aquitards to be impermeable and so we do about the outer boundary of the aquifers to prescribe homogeneous Neumann boundary condition there.

The distribution of pressure head in m-th aquifer is described by Poisson equation

$$-T^m \Delta h^m = f^m \qquad \text{on } \Theta^m,\ \forall m = 1,\dots,M, \tag{1}$$

where $T^m\,[\mathrm{m}^2\mathrm{s}^{-1}]$ denotes transmisivity, $h^m\,[\mathrm{m}]$ pressure head and $f^m\,[\mathrm{ms}^{-1}]$ source density. Equation (1) is derived from the Darcy law and the continuity equation for incompressible fluid.

The communication between aquifers is possible only through wells which can be seen as 1D problems governed by following equation

$$\int_{\partial B_w^m} \sigma_w^m \left(h^m - H_w^m\right) \mathrm{d}\mathbf{x} = c_w^{m+1}\left(H_w^m - H_w^{m+1}\right) - c_w^m \left(H_w^{m-1} - H_w^m\right), \tag{2}$$
$$\forall m = 1,\dots,M \ \text{ and } \forall w = 1,\dots,W,$$

where $\sigma_w^m\,[\mathrm{ms}^{-1}]$ denotes the permeability coefficient between w-th well and m-th aquifer, H_w^m pressure head in the well w at the level of m-th aquifer, $c_w^m\,[\mathrm{m}^2\mathrm{s}^{-1}]$ permeability of the well w

between aquifers and finally ∂B_w^m is the boundary of the well. The equation (2) puts the flow in and out of the well on right hand side and the flow through the well boundary on the left hand side in balance.

The transfer between wells and aquifers can be treated in two ways. There is a well boundary integral on the left hand side of (2) and it represents flow over the boundary of the well. Second variant uses a surface integral in (2) which represents flow source in the area of aquifer and well cross-section (the units of σ_w^m then changes). The same integral appears also in the weak formulation of (1) – as a boundary integral in the first variant or as a part of the source term in the second variant. Both variants were tested in the work with nearly identical results, only the second approach simplifies the implementation and slightly speeds up the assembly.

3 Numerical methods

We implemented and compared three numerical methods, using the Deal II library (does not provide any XFEM/SGFEM functionality itself):

- h-adaptive FEM with linear finite elements, without any enriching technique
- so called 'corrected' XFEM with local enrichment developed by T. P. Fries in [4], introducing ramp function and shift
- SGFEM introduced by I. Babuška and U. Banerjee in [1]

3.1 Enrichment function

The hydro-geological wells represent sources with very small diameter in the model. If we solve a local problem with circular domain with one well placed in the center, we will see the logarithmic dependence of the pressure head on the distance from the well. If we represented the well only by a point, the pressure head would go to infinity while closing to the point (singularity $|\log 0| \to \infty$).

To capture large gradients of pressure head around the wells, we introduce local enrichment function

$$\phi_w(\mathbf{x}) = \begin{cases} \log(r_w(\mathbf{x})), & r_w > R_w \\ \log(R_w), & r_w \leq R_w, \end{cases} \tag{3}$$

where r_w is the distance from the well center and R_w is the well radius.

3.2 Discretization

According to [5] we used the corrected XFEM method with *ramp function* g_w at first. The pressure approximation we are looking for is in the form

$$h(\mathbf{x}) = \underbrace{\sum_{j\in\mathcal{N}} \alpha_j\, \varphi_j(\mathbf{x})}_{\text{FEM}} + \underbrace{\sum_{w\in\mathcal{W}} \sum_{k\in\mathcal{N}_w} g_w(\mathbf{x})\beta_{w,k}\, \phi_w(\mathbf{x})\varphi_k(\mathbf{x})}_{\text{enrichment}}, \tag{4}$$

where the first part with degrees of freedom α_j corresponds with the linear finite elements φ_j. The second part with degrees of freedom $\beta_{w,k}$ is the enrichment part. The index set $\mathcal{N}$ contains numbers of all mesh nodes, the index set $\mathcal{W}$ contains numbers of all wells and $\mathcal{N}_w$ is the set of all enriched nodes from the well w.

We are enriching nodes of the mesh that are inside a circular neighborhood of given radius r_{enr}. The bigger the enriched area is, the larger amount of additional degrees of freedom are coming from the enrichment. The proper choice of the radius r_{enr} is not trivial and we would like to further investigate this.

To get the optimal convergence rate of the XFEM, it is recommended to use both the *ramp function* and *shift*. We replace the enrichment function ϕ_w in (4) with its shifted version

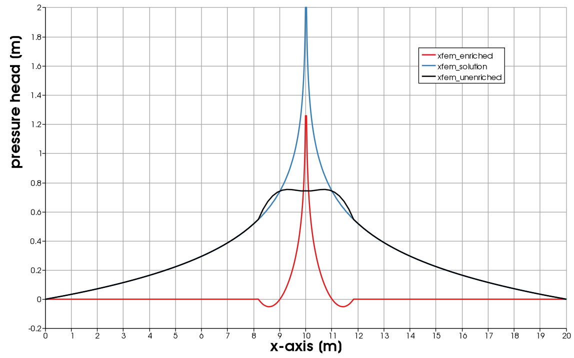
$$\phi_{w,k}(\mathbf{x}) = \phi_w(\mathbf{x}) - \phi_w(\mathbf{x}_k). \tag{5}$$

which guarantees the optimal convergence of the method but possibly brings the problem of ill-conditioning into the linear system. This leads us to the SGFEM which uses only the shift without ramp function in the enriched part of (4). It was proven (for 1D sofar in [1]) that if the finite element part of the trial space is *almost orthogonal* to the enriched part of the trial space then the condition number of the linear system is not worse than the condition number of the FEM part of the linear system.

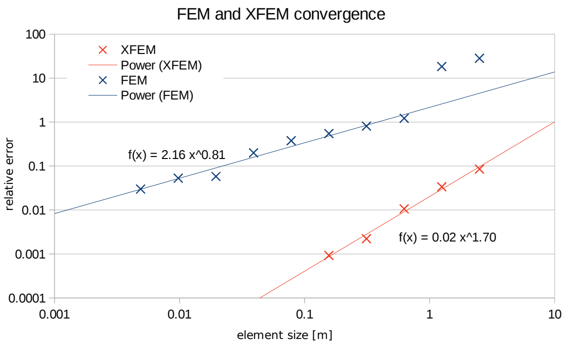
4 Results

In the theoretical part of our work we derived the weak formulation of the problem and we proved the existence and the uniqueness of the weak solution according to Lax-Milgram lemma.

The following figure 1e shows the solution of a simple problem with one well in a single aquifer over x axis. One can see the enriched part (red), the linear finite element part (black) and their sum (blue) corresponding with (4). Analytical solution of this model is known so we could measure convergence rates of the methods – shown in the other figure 1f. We reached higher h-adaptive FEM convergence rate $O(h^{0.8})$ than R. Gracie and J. R. Craig in [5]. XFEM convergence rate $O(h^{1.7})$ is in a good agreement with their results but still suboptimal. The results are in detail described in master's thesis [3].



(e) Decomposed solution.



(f) Convergence rate.

The comparison of corrected XFEM with shifted enrichment and SGFEM is about to be finished.

5 Future work

There are several problems we would like to take interest in and do tests on our model:

- the choice of the size of the enrichment area (is it possible automatic?)
- adaptive integration on the enriched elements – there is an idea of precomputed quadrature points which would be then only mapped from the reference element
- improvement in solving of the linear system

After finishing this primal research of XFEM/SGFEM method we will aim our effort in the implementation of XFEM in the mixed hybrid method to compute both pressure head and velocity of a fluid. Further extension of our model on flow in 3D system with cracks is planned. Long term aim is to implement this method in the software Flow123d which specializes in computations on complex meshes consisting of simplicial elements of different dimensions.

Acknowledgement: This work is made under the sincere guidance of Mgr. Jan Březina, Ph.D.

References

[1] I. Babuška, U. Banerjee: *Stable generalized finite element method (SGFEM).* Mathematics Faculty Scholarship. Paper 139, 2011. web: `http://surface.syr.edu/mat/139`.

[2] W. Bangerth, R. Hartmann, G. Kanschat: *deal.II – a general purpose object oriented finite element library.* ACM Trans. Math. Softw., 33, (4), 24/1–24/27, 2007.

[3] P. Exner: *Partition of unity methods for approximation of point water sources in porous media.* Master's thesis. Technical University of Liberec, Faculty of Mechatronics, Informatics and Interdisciplinary Studies, 2013. web: `http://bacula.nti.tul.cz/~pavel.exner/files/dipl_Pavel_Exner.pdf`.

[4] T. P. Fries: *A corrected xfem approximation without problems in blending elements.* International Journal for Numerical Methods in Engineering, 75, 503–532, 2007.

[5] R. Gracie, J. R.Craig: *Modelling well leakage in multilayer aquifer systems using the extended finite element method.* Finite elements in Analysis and Design, Elsevier B.V., 46, 504–513, 2010.

Tracking the trajectory in finite precision CG computations

T. Gergelits, Z. Strakoš

Department of Numerical Mathematics
Faculty of Mathematics and Physics, Charles University in Prague, Prague

1 Introduction

The method of conjugate gradients (CG) [4] is the method of choice for solving linear systems of algebraic equations

$$Ax = b, \qquad A \in \mathbb{F}^{N\times N},\ b \in \mathbb{F}^{N}, \quad \text{where } \mathbb{F} \text{ is } \mathbb{C} \text{ or } \mathbb{R},$$

with a large and sparse matrix A which is Hermitian and positive definite. It is well known that the numerical convergence behaviour of the CG method can be strongly affected by the influence of rounding errors. However, whereas the CG *convergence rate* may be substantially *different* in finite precision and exact arithmetic, we observe that the *trajectories* of approximations as well as the corresponding *Krylov subspaces* are very *similar*.

2 Delay of convergence & rank-deficiency

Computationally, the CG method is based on short recurrences. Assuming exact arithmetic, short CG recurrences ensure the global orthogonality of the residual vectors, which span at the kth step the k-dimensional Krylov subspace $\mathcal{K}_k(A, r_0)$. Here $r_0 = b - Ax_0$ is the initial residual and x is approximated by $x_k \in x_0 + \mathcal{K}_k(A, r_0)$. In practical computations, however, the use of

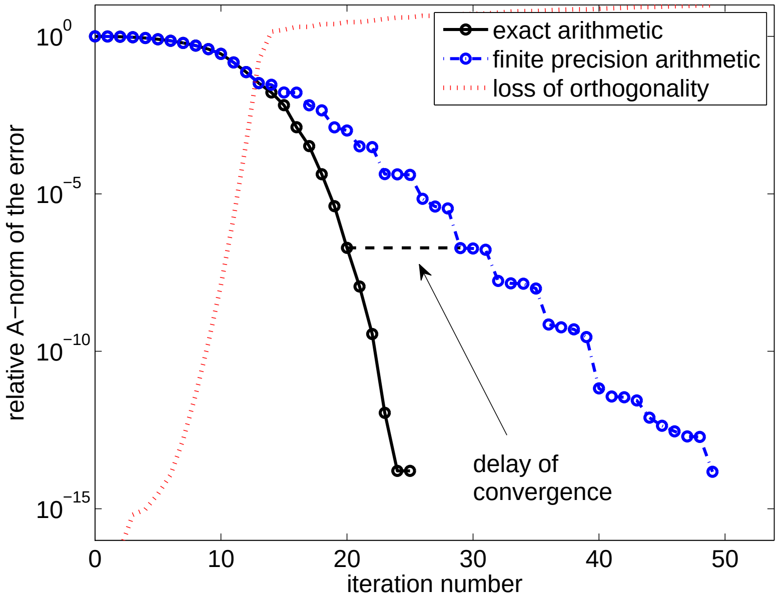


Figure 1: Illustration of the delay of convergence in finite precision CG computations.

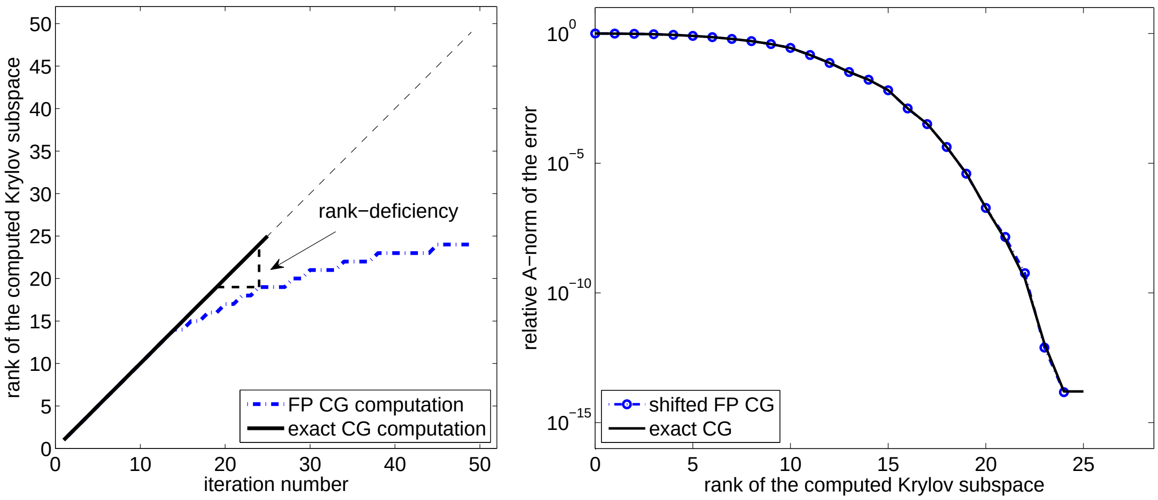


Figure 2: Delay of convergence is determined by the rank-deficiency of the computed Krylov subspace.

short recurrences inevitably leads to the loss of the global orthogonality among the computed residual vectors; they often become even (numerically) linearly dependent. Consequently, the computed residuals may, at the kth step, span a subspace of the dimension $\bar{k}$ smaller than k. This rank-deficiency of the computed Krylov subspaces then causes a delay of convergence of finite precision CG computations; see Fig. 1.

The correspondence between the delay of convergence and the rank-deficiency (see, e.g., [7] and [6, Section 5.9.1]) is illustrated in Fig. 2. In the right, the exact CG convergence curve[1] is compared with the curve of finite precision CG computations which is shifted back by the (numerical) rank-deficiency $k - \bar{k}$ where

$$\bar{k} = \mathrm{rank}(\mathcal{K}_k(A, r_0))$$

is the numerical rank of the computed Krylov subspace. The threshold criterion for computation of the numerical rank is set to 0.1, loss of orthogonality is even for this “weak” threshold very substantial.

We observe that the kth error $\|x - \bar{x}_k\|_A$ of finite precision CG computations corresponds to the $\bar{k}$th error $\|x - x_{\bar{k}}\|_A$ of exact CG computations, i.e., the convergence of finite precision CG computations is delayed, with respect to exact CG computations applied to the same data A, b and x_0, by $k - \bar{k}$ iterations; cf. [3].

3 Inertia of computed approximations and Krylov subspaces

Since approximations $\bar{x}_k$ generated by finite precision CG computations and $x_{\bar{k}}$ generated by exact CG computations both lie in the same space $\mathbb{F}^N$, we can compare the vectors themselves (see Fig. 3). Surprisingly, the distance between the exact precision approximation and the shifted finite precision CG approximation is small in comparison with the actual size of the error (red solid line), i.e.,

$$\frac{\|\bar{x}_k - x_{\bar{k}}\|_\infty}{\|x - x_{\bar{k}}\|_\infty} \ll 1.$$

[1] The exact arithmetic is simulated by full double reorthogonalization of the computed residuals.

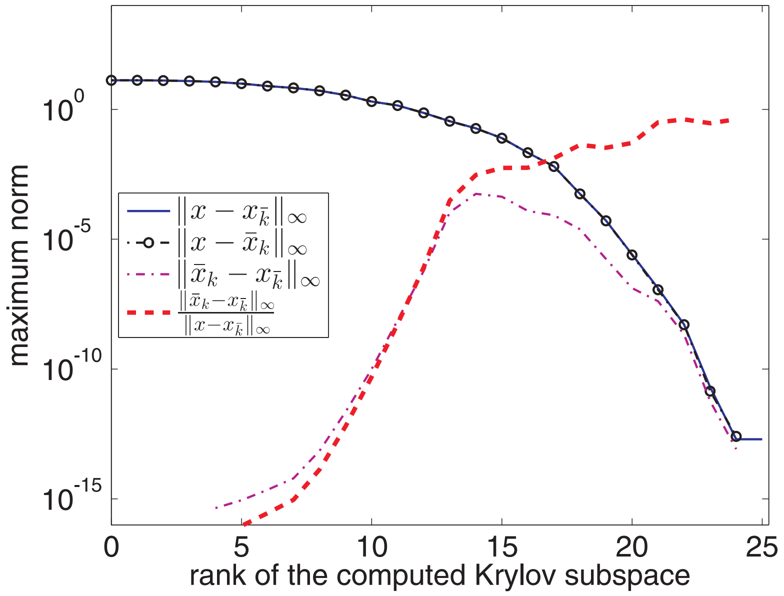


Figure 3: The closeness of exact CG and shifted FP CG approximations in comparison with the actual size of the error is illustrated by the red solid line.

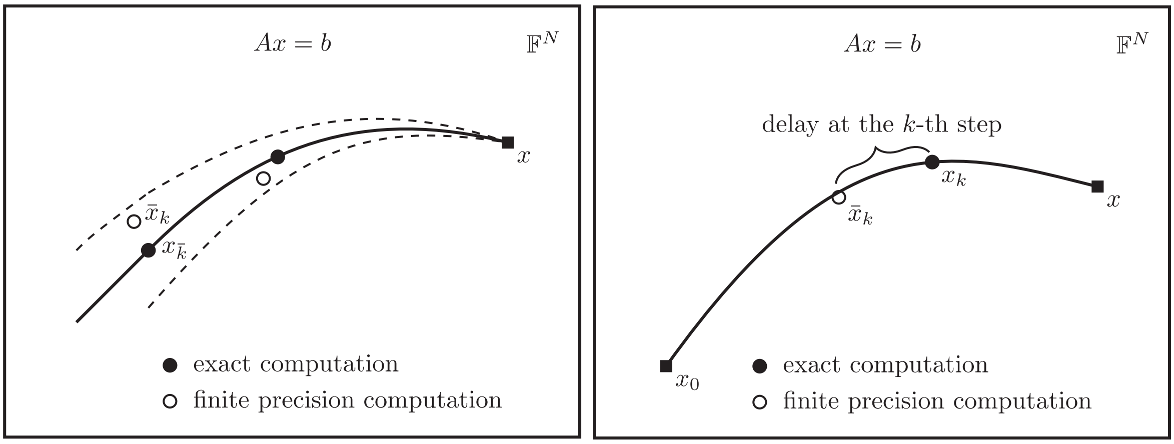


Figure 4: The trajectory of approximations $\bar{x}_k$ generated by finite precision CG computations tightly follows the trajectory of exact CG approximations $x_{\bar{k}}$ with the delay given by the rank-deficiency of the computed Krylov subspace.

Thus we formulate an observation that the trajectory of approximation vectors generated by the CG method in finite precision arithmetic applied to linear system $Ax = b$ closely follows the trajectory of approximation vectors from the CG method in exact arithmetic applied to the same system; see the illustration in Fig. 4.

The observed correspondence among the approximation vectors from the finite precision CG computations and CG in exact arithmetic suggests that also the associated Krylov subspaces are in some sense close to each other. Indeed, we have observed that the computed rank-deficient Krylov subspace span numerically nearly the same subspace as the Krylov subspace of the corresponding dimension generated by the CG method in exact arithmetic.

4 Conclusion

To our best knowledge, the relation between the k-th Krylov subspace computed by the CG process in finite precision arithmetic and the Krylov subspace of the corresponding dimension $\bar{k}$ generated by exact CG process has not been adressed in literature. The somewhat related problem of sensitivity of Krylov subspaces to small perturbations was studied in several papers; see, e.g., [1, 5] or [8]. Krylov subspaces can be in general sensitive to small perturbations of the matrix A. The observed stability (or inertia) of computed Krylov subspace represents a very remarkable phenomenon which needs further investigation.

Acknowledgement: This work has been supported by the ERC-CZ project LL1202, by the GAČR grant 201/13-066845S and by the GAUK grant 695612.

References

[1] J.-F. Carpraux, S. K. Godunov, S. V. Kuznetsov: *Condition number of the Krylov bases and subspaces.* Linear Algebra Appl., 248, 137–160, 1996.

[2] T. Gergelits, Z. Strakoš: *Composite convergence bounds based on Chebyshev polynomials and finite precision conjugate gradient computations.* Numerical Algorithms, DOI: 10.1007/s11075-013-9713-z. June, 2013.

[3] A. Greenbaum: *Behaviour of slightly perturbed Lanczos and conjugate-gradient recurrences.* Linear Algebra Appl., 113, 7–63, 1989.

[4] M. R. Hestenes, E. Stiefel: *Methods of conjugate gradients for solving linear systems.* J. Research Nat. Bur. Standards, 49, 409–436, 1952.

[5] S. V. Kuznetsov: *Perturbation bounds of the Krylov bases and associated Hessenberg forms.* Linear Algebra Appl., 265, 1–28, 1997.

[6] J. Liesen, Z. Strakoš: *Krylov subspace methods: principles and analysis.* Numerical Mathematics and Scientific Computation, Oxford University Press, 2012.

[7] C. C. Paige, Z. Strakoš: *Correspondence between exact arithmetic and finite precision behaviour of Krylov space methods.* In XIV. Householder Symposium, University of British Columbia, Whistler, 1999, 250–253.

[8] C. C. Paige, P. Van Dooren: *Sensitivity analysis of the Lanczos reduction.* Numer. Linear Algebra Appl., 1, (6), 29–50, 1999.

FLLOP: A novel massively parallel QP solver

V. Hapla, D. Horák, A. Markopoulos, L. Říha

VSB-Technical University Ostrava,
IT4Innovations National Supercomputing Center,
VSB-Technical University of Ostrava
17. listopadu 15/2172
708 33 Ostrava – Poruba, Czech Republic

Discretization of most engineering problems, described by partial differential equations, leads to large sparse linear systems of equations. However, problems expressible as elliptic variational inequalities, such as those describing the equilibrium of elastic bodies in mutual contact, are more naturally discretized to quadratic programming problems (QP). They can be thought of as a generalization of linear systems of equations being subject to equality and inequality constraints and take the form

$$\text{find} \quad \mathbf{x} = \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} \quad \text{subject to} \quad \mathbf{B}_E\mathbf{x} = \mathbf{c}_E,\ \mathbf{B}_I\mathbf{x} \leq \mathbf{c}_I,\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}.$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the symmetric positive semidefinite *Hessian* matrix, $\mathbf{b} \in \mathbb{R}^n$ is the *right hand side vector*, $\mathbf{B}_E \in \mathbb{R}^{m_E\times n}$ is the *equality constraint matrix*, $\mathbf{c}_E \in \mathbb{R}^{m_E}$ is the *equality constraint right hand side vector*, $\mathbf{B}_I \in \mathbb{R}^{m_I\times n}$ is the *inequality constraint matrix*, $\mathbf{c}_I \in \mathbb{R}^{m_I}$ is the *inequality constraint right hand side vector*, $\mathbf{l} \in \mathbb{R}^n$ is the *lower bound vector*, $\mathbf{u} \in \mathbb{R}^n$ is the *upper bound vector.*

We present here our novel package FLLOP for quadratic programming and FETI domain decomposition, built on top of PETSc (similarly to TAO and SLEPc packages). Currently tested applications include mainly engineering problems of structure mechanics: linear elasticity, contact problems, elasto-plasticity, and shape optimization.

FLLOP API is designed to be easy-to-use but at the same time efficient and HPC-centric. One of the principal design decisions is decoupling of concepts of QP problems, QP transforms and QP solvers. A QP transform is a mapping deriving from the given original QP a new QP which is simpler or has some better properties. It is of course required that the solution of the original QP can be computed from the solution of the derived one. QP transforms often allow use of efficient solvers that are not compatible with the original QP. However, they are themselves solver-neutral. The algebraic part of the FETI DDM method is a special case of QP transform.

The typical workflow when solving a QP with FLLOP is as follows:

1. specification of the QP by the user,
2. an automatic or user-specified series of QP transforms,
3. an automatic or manual choice of a suitable solver,
4. solution of the most derived QPs by the chosen solver,
5. a series of reconstructions to get a solution of the original QP (triggered automatically by the solver).

Each function representing a QP transform creates a new instance `QP`$_2$ of the `QP` class based on the original `QP`$_1$. The data objects being altered by the given QP transform are copied from

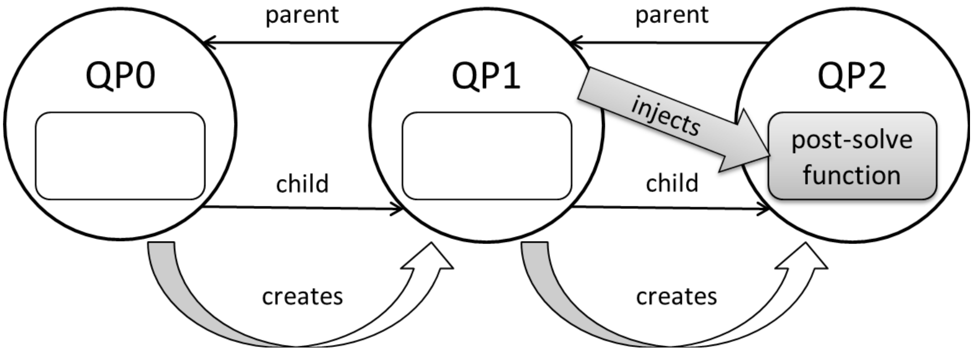


Figure 1: Chain of QP transforms.

QP_1, modified and stored to QP_2. Otherwise, QP_1 and QP_2 only share pointers to the same data object. Furthermore, links between QP_1 and QP_2 are created; QP_1 obtains a *child* link to QP_2, QP_2 gets a *parent* link to QP_1. Thus, a doubly linked list is generated where every node is a QP (Figure 1).

The solution $\mathbf{x}\langle\text{QP}_2\rangle$ is generally not equal to the solution $\mathbf{x}\langle\text{QP}_1\rangle$. The associated reconstruction function $\big(\text{QP}_2{\to}\text{QP}_1\big)$ must be called to carry out $\mathbf{x}\langle\text{QP}_1\rangle = \big(\text{QP}_2{\to}\text{QP}_1\big)\big(\mathbf{x}\langle\text{QP}_2\rangle\big)$. For the above-mentioned case, the reconstruction function is $\big(\text{QP}_2{\to}\text{QP}_1\big)(\mathbf{x}) = \mathbf{x}{+}\mathbf{x}_P$, so it holds that $\mathbf{x}\langle\text{QP}_1\rangle = \mathbf{x}\langle\text{QP}_2\rangle{+}\mathbf{x}_P$. In FLLOP, the reconstruction function is injected to the child QP by the transform function. Once the solution of the last QP is computed, the solver triggers a series of the reconstruction functions in LIFO manner, i.e. the reconstruction function of the last QP is called first.

We also need to store somewhere the auxiliary data created by a transform and needed by the asociated reconstruction function. In our case, it is the vector $\mathbf{x}_P$. For this purpose so called *reconstruction context* is used; it is a `void` pointer, injected to the child QP together with the reconstruction function. The notions mentioned above are illustrated by Figure 2.

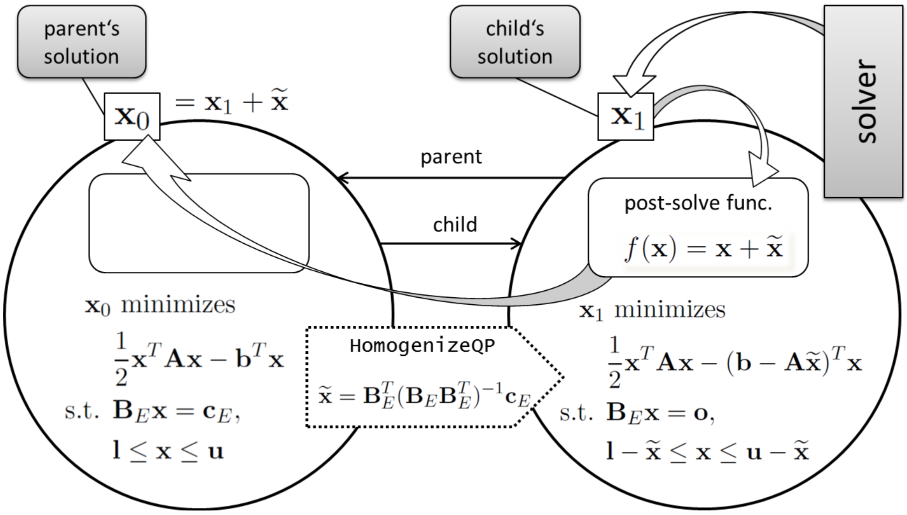

parent's solution
child's solution
$\mathbf{x}_0$
$= \mathbf{x}_1 + \widetilde{\mathbf{x}}$
$\mathbf{x}_1$
solver
parent
child
post-solve func.
$f(\mathbf{x}) = \mathbf{x} + \widetilde{\mathbf{x}}$
$\mathbf{x}_0$ minimizes
$\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$
s.t. $\mathbf{B}_E\mathbf{x} = \mathbf{c}_E,$
$\mathbf{l} \le \mathbf{x} \le \mathbf{u}$
HomogenizeQP
$\widetilde{\mathbf{x}} = \mathbf{B}_E^T(\mathbf{B}_E\mathbf{B}_E^T)^{-1}\mathbf{c}_E$
$\mathbf{x}_1$ minimizes
$\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - (\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}})^T\mathbf{x}$
s.t. $\mathbf{B}_E\mathbf{x} = \mathbf{o},$
$\mathbf{l} - \widetilde{\mathbf{x}} \le \mathbf{x} \le \mathbf{u} - \widetilde{\mathbf{x}}$

Figure 2: Example of QP transform and reconstructions of the solution – homogenization of the equality constraints.

References

[1] V. Hapla, D. Horak: *A comparison of FETI natural coarse space projector implementation strategies.* Proceedings of PARENG2013, Paper 6, 2013. DOI: 10.4203/ccp.101.6

[2] V. Hapla, D. Horak, M. Merta: *Use of direct solvers in TFETI massively parallel implementation.* Proceedings of PARA 2012, Lecture Notes in Computer Science, 7782 LNCS, 192–205, 2013. DOI: 10.1007/978-3-642-36803-5_14

[3] M. Merta, A. Vasatova, V. Hapla, D. Horak: *Massively parallel implementation of Total-FETI method DDM with application to medical image registration.* Proceedings of DD21 2012, Lecture Notes in Computational Science and Engineering, accepted 2013.

[4] D. Horák, V. Hapla: *TFETI coarse problem massively parallel implementation.* ECCOMAS 2012 e-Book Full Papers, 8260–8267, 2012. ISBN: 978-395035370-9

[5] V. Hapla, D. Horák: *TFETI coarse space projectors parallelization strategies.* Proceedings of PPAM 2011, Lecture Notes in Computer Science, 7203 LNCS (PART 1), 152–162, 2012, DOI: 10.1007/978-3-642-31464-3_16

[6] T. Kozubek, V. Vondrák, M. Menšík, D. Horák, Z. Dostál, V. Hapla, P. Kabelíková, M. Čermák: *Total FETI domain decomposition method and its massively parallel implementation.* Advances in Engineering Software 60–61, 14–22, 2013. DOI: 10.1016/j.advengsoft.2013.04.001

[7] PRACE-2IP WP9 Deliverable: *Support for industrial applications year 1.* `https://bscw.zam.kfa-juelich.de/bscw/bscw.cgi/d799956/D9.1.1.pdf`, 2012.

[8] Z. Dostál: *Optimal quadratic programming algorithms, with applications to variational inequalities.* 1st edition. SOIA 23. Springer US, New York, 2009.

[9] Z. Dostál, T. Kozubek: *An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. Mathematical Programming* 135, (1–2), 195–220, 2012. DOI: 10.1007/s10107-011-0454-2

[10] Z. Dostál, D. Horák, R. Kučera: *Total FETI – an easier implementable variant of the FETI method for numerical solution of elliptic PDE.* Communications in Numerical Methods in Engineering, 22, (12), 1155–1162, 2006. DOI: 10.1002/cnm.881

[11] Dostal, Z., Kozubek, T., Vondrak, V., Markopoulos, A., Brzobohaty, T.: *Scalable TFETI algorithm for the solution of multibody contact problems of elasticity.* International Journal for Numerical Methods in Engineering, 82, (11), 1384–1405, 2010. DOI: 10.1002/nme.2807

[12] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang: *PETSc users manual.* Tech. Rep. ANL-95/11 – Revision 3.2, Argonne National Laboratory, 2011.

[13] S. Balay, W. D. Gropp, L. C. McInnes, B.Ḟ. Smith: *Efficient management of parallelism in object oriented numerical software libraries.* Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, (Eds.), Birkhäuser Press, 163–202, 1997.

[14] *PETSc Web page.* `http://www.mcs.anl.gov/petsc/`

[15] *FLLOP Web Page.* `http://industry.it4i.cz/en/products/fllop/`

On parameter dependent static contact problems

J. Haslinger, V. Janovský, R. Kučera

Charles University, Faculty of Mathematics and Physics, Prague
Department of Mathematics and Descriptive Geometry, VŠB-TU, Ostrava

1 Discrete static contact problems with Coulomb friction

Consider deformable bodies in mutual contact. The relevant mathematical description consists in modeling both non-penetration conditions and a friction law. The widely accepted Coulomb friction law represents a serious mathematical and numerical problem.

In particular, we consider the static contact problem with Coulomb friction on a planar domain. The problem is uniquely solvable, provided that the friction coefficient $\mathcal{F} > 0$ is sufficiently small. Note that no essential contribution was made concerning solvability of this problem for general data. In a natural finite element (FEM) approximation, the discrete problem has always a solution, disregarding the size of $\mathcal{F}$, see e.g. [2, 6].

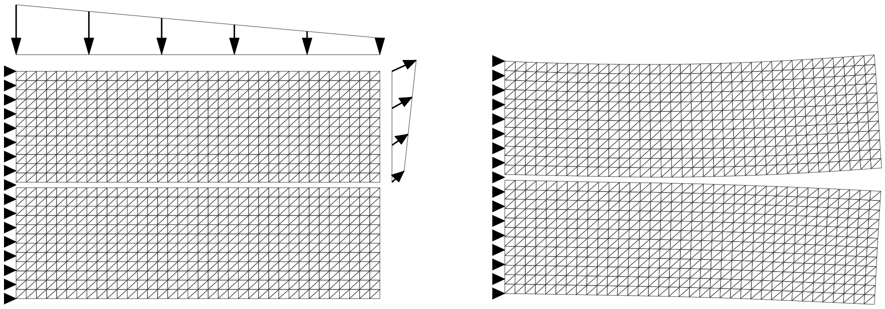


Figure 1: Contact of two elastic bodies Ω^1 (the upper body) and Ω^2, along the contact boundary Γ_c. The loading is due to surface tractions. On the right: Resulting displacements. The FEM data: $n = 1320$, $m = 30$.

We consider a particular geometry, see Figure 1. The FEM approximation (linear elements) yields the following *primal-dual* discrete state problem:

$$\boldsymbol{K}\boldsymbol{u} + \boldsymbol{N}^\top\boldsymbol{\lambda}_\nu + \boldsymbol{T}^\top\boldsymbol{\lambda}_t = \boldsymbol{f}, \tag{1}$$

$$\boldsymbol{N}\boldsymbol{u} \le 0,\ \boldsymbol{\lambda}_\nu \ge 0,\ \boldsymbol{\lambda}_\nu^\top\boldsymbol{N}\boldsymbol{u} = 0, \tag{2}$$

$$\left.\begin{array}{l} |\lambda_{t,i}| \le \mathcal{F}\lambda_{n,i}, \\ |\lambda_{t,i}| < \mathcal{F}\lambda_{n,i} \Rightarrow (\boldsymbol{T}\boldsymbol{u})_i = \boldsymbol{0}, \\ |\lambda_{t,i}| = \mathcal{F}\lambda_{n,i} \Rightarrow \exists\, c_{t,i} \ge 0 :\ (\boldsymbol{T}\boldsymbol{u})_i = c_{t,i}\lambda_{t,i}, \end{array}\right\}\quad i = 1,\dots,m, \tag{3}$$

where $(\boldsymbol{u},\boldsymbol{\lambda}_\nu,\boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$. Here $\boldsymbol{u}$ approximates a displacement field with n degrees of freedom. Further $\boldsymbol{\lambda}_\nu$, $\boldsymbol{\lambda}_t$ approximate normal and tangential stress components, respectively along the contact boundary Γ_c, m is the number of contact nodes. Data of the model: $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is a positive definite stiffness matrix, $\boldsymbol{N}, \boldsymbol{T} \in \mathbb{R}^{m\times n}$ are full rank matrices (the actions of distributed contact forces along normal and tangential directions), respectively, and $\boldsymbol{f} \in \mathbb{R}^n$ is a vector of nodal forces.

The inequalities (2) and (3) can be equivalently written as

$$\boldsymbol{\lambda}_\nu - P_{\mathbb{R}^m_+}(\boldsymbol{\lambda}_\nu + \rho \boldsymbol{N}\boldsymbol{u}) = \mathbf{0} \quad \text{and} \quad \boldsymbol{\lambda}_t - P_{[-\mathcal{F}(\boldsymbol{\lambda}_\nu+\rho\boldsymbol{N}\boldsymbol{u})_+,\mathcal{F}(\boldsymbol{\lambda}_\nu+\rho\boldsymbol{N}\boldsymbol{u})_+]}(\boldsymbol{\lambda}_t + \rho \boldsymbol{T}\boldsymbol{u}) = \mathbf{0},$$

respectively, where $P_{\mathbb{R}^m_+}$, $P_{[-\mathcal{F}(\boldsymbol{\lambda}_\nu+\rho\boldsymbol{N}\boldsymbol{u})_+,\mathcal{F}(\boldsymbol{\lambda}_\nu+\rho\boldsymbol{N}\boldsymbol{u})_+]}$ are suitable projectors and $(\cdot)_+$ denotes the non-negative part. Parameter $\rho > 0$ is arbitrary but fixed (e.g., $\rho = 1$). Therefore, solving (1)-(3) is equivalent to finding roots $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ of the equation

$$G(\boldsymbol{y}) \equiv \begin{pmatrix} \boldsymbol{K}\boldsymbol{u} + \boldsymbol{N}^\top \boldsymbol{\lambda}_\nu + \boldsymbol{T}^\top \boldsymbol{\lambda}_t \\ \boldsymbol{\lambda}_\nu - P_{\mathbb{R}^m_+}(\boldsymbol{\lambda}_\nu + \rho \boldsymbol{N}\boldsymbol{u}) \\ \boldsymbol{\lambda}_t - P_{[-\mathcal{F}(\boldsymbol{\lambda}_\nu+\rho\boldsymbol{N}\boldsymbol{u})_+,\mathcal{F}(\boldsymbol{\lambda}_\nu+\rho\boldsymbol{N}\boldsymbol{u})_+]}(\boldsymbol{\lambda}_t + \rho \boldsymbol{T}\boldsymbol{u}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \\ 0 \end{pmatrix}, \tag{4}$$

where $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$. The mapping $G : \mathbb{R}^{n+2m} \mapsto \mathbb{R}^{n+2m}$ is continuous and piecewise smooth. In particular, it is *piecewise affine*, see e.g. [7] for the notion.

2 The semi-smooth Newton method (SSNM)

For solving (4), we apply the Newton iterations. Due to nature of the operator G, semi-smooth methods are applicable, see e.g. [4]. Let $\mathcal{M} = \{1, 2, \ldots, m\}$ be the set of all indices of the contact points: Given $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{\lambda}_\nu, \boldsymbol{\lambda}_t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$, we define the *inactive* sets $\mathcal{I}_\nu := \mathcal{I}_\nu(\boldsymbol{y})$, $\mathcal{I}_t^+ := \mathcal{I}_t^+(\boldsymbol{y})$, $\mathcal{I}_t^- := \mathcal{I}_t^-(\boldsymbol{y})$ by

$$\begin{aligned} \mathcal{I}_\nu &= \{i \in \mathcal{M} : \lambda_{\nu,i} + \rho(\mathbf{Nu})_i < 0\}, \\ \mathcal{I}_t^+ &= \{i \in \mathcal{M} : \lambda_{t,i} + \rho(\mathbf{Tu})_i - \mathcal{F}\lambda_{\nu,i} > 0\}, \\ \mathcal{I}_t^- &= \{i \in \mathcal{M} : \lambda_{t,i} + \rho(\mathbf{Tu})_i + \mathcal{F}\lambda_{\nu,i} > 0\}, \end{aligned}$$

and the *active* sets $\mathcal{A}_\nu := \mathcal{A}_\nu(\boldsymbol{y})$, $\mathcal{A}_t := \mathcal{A}_t(\boldsymbol{y})$ as their complements:

$$\mathcal{A}_\nu = \mathcal{M} \setminus \mathcal{I}_\nu, \quad \mathcal{A}_t = \mathcal{M} \setminus (\mathcal{I}_t^+ \cup \mathcal{I}_t^-).$$

For details see [3].

3 Continuation

Consider the Coulomb friction model (1)-(3), i.e. (4). We assume that the model depends on parameters. In particular, we consider that

a) $\boldsymbol{f} := \boldsymbol{f}(\alpha)$ depends on a scalar parameter α which simulates loading changes, see [3]
b) the friction coefficient $\mathcal{F}$ is a positive parameter; we will call it β.

In this contribution we fix the load α and consider *continuous* changes of the friction parameter β. The resulting *solution path* is a curve in $\mathbb{R} \times \mathbb{R}^{n+2m}$, see a qualitative sketch in Figure 2. It consists of *oriented smooth branches*, connected by *transition points*.

- In order to follow the oriented smooth branches, we implemented *tangent continuation*, see [1], Algorithm 4.25, with *SSNM* as a corrector. We used an adaptive step-size control.

- In order to detect transition points, we introduced *branching* and *orientation* indicators. The idea is to modify the inactive sets $\mathcal{I}_\nu$, $\mathcal{I}_t^+$, $\mathcal{I}_t^-$ properly.

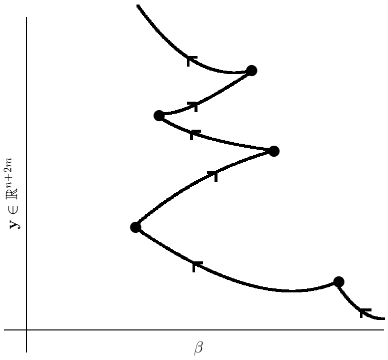


Figure 2: *Solution path* consists of *oriented smooth branches*, connected by *transition points*. Hence, refering to this particular sketch: For a fixed β, we may encounter up to five crossing points of the path. They are related to five different solutions.

4 Example

In an experiment in [3], we fixed the parameter $\mathcal{F} = 9$ and consider a linear loading parametrized by α. Applying continuation, see [3], we found three different solutions for $\alpha = 0$. They are shown in Figure 3 in a proper projection on the contact boundary.

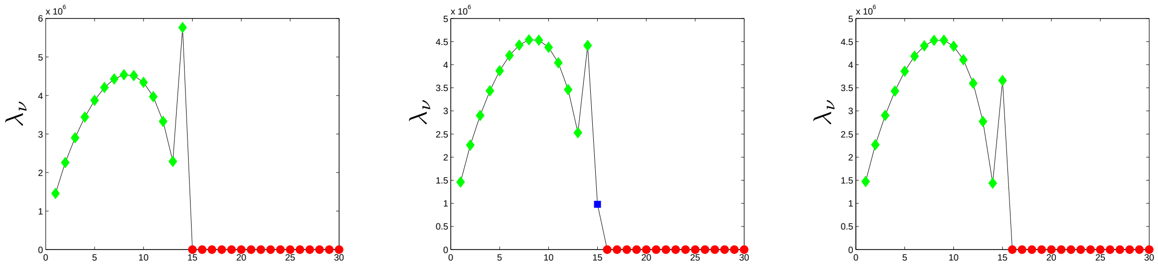


Figure 3: Three solutions labeled as state_0_9_case1, state_0_9_case2, state_0_9_case3 for parameters $\alpha = 0$ and $\beta = \mathcal{F} = 9$. Note that at the contact point No15 we have three different contact modes namely *no contact* (circle), *contact-slip* (square) and *contact-stick* (diamond).

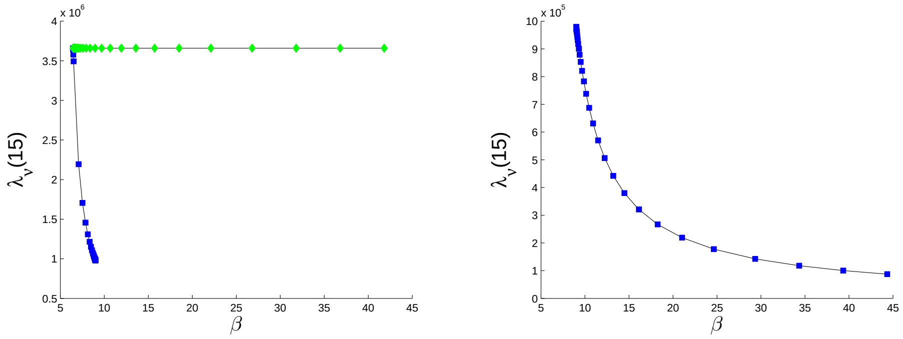


Figure 4: Path No1, the initial point state_0_9_case2. Solution path of the contact point No15 with the *orientation* = -1 (on the left) and the *orientation* = 1 (on the right). Observe the fold transition from the mode *contact-slip* (square) to the mode *contact-stick* (diamond), related to the ordinate $\beta = 6.4617$.

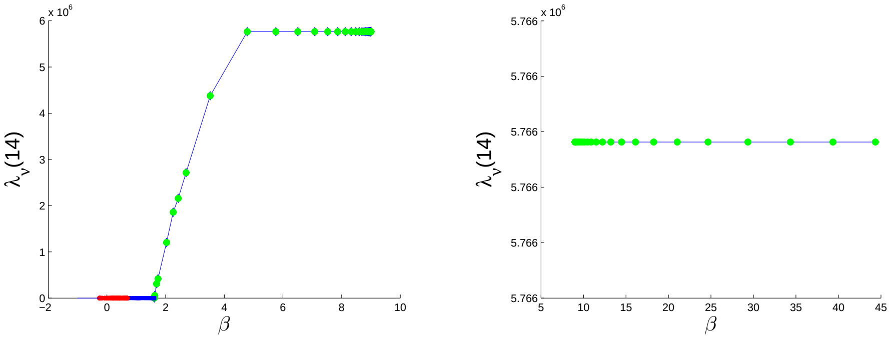


Figure 5: Path No2, the initial point state_0_9_case1. Solution path of the contact point No14 with the *orientation* = -1 (on the left) and the *orientation* = 1 (on the right). There is no fold transition point on the path.

We encountered two solution paths. Each path is initialized at the indicated point and followed with either the positive or negative orientation (orientation = 1, orientation = -1):

Path No1, see Figure 4: The path is initialized at the point state_0_9_case2. It contains also the point state_0_9_case3 with the same ordinate $\beta = \mathcal{F} = 9$. There is a fold transition point with the ordinate $\beta = \mathcal{F} = 6.4617$.

Path No2, see Figure 5: The path is initialized at the point state_0_9_case1.

Acknowledgment: This work was supported by the Grant Agency of the Czech Republic (grant No. P201/12/0671).

References

[1] P. Deuflhart, A. Hohmann: *Numerical analysis in modern scientific computing.* Texts in applied mathematics, Springer Verlag, New York, 2003.

[2] J. Haslinger, V. Janovský , T. Ligurský : *Qualitative analysis of solutions to discrete static contact problems with Coulomb friction.* Comp. Meth. Appl. Mech. Engrg., 205-–208, 149–161, 2012.

[3] J. Haslinger, V. Janovský, R. Kučera : *Path-following the static contact problem with Coulomb friction.* In: Proceedings of the International Conference Application of Mathematics 2013, Prague, May 15–17, 2013, J. Brandts, S. Korotov, M. Křížek, J. Šístek, T. Vejchodský, (Eds) Institute of Mathematics, Academy of Sciences of the Czech Republic, Prague 2013, 104–116.

[4] K. Ito, K. Kunisch: *Semi-smooth Newton methods for the Signorini problem.* Appl. Math., 53, 455–468, 2009.

[5] V. Janovský , T. Ligurský: *Computing non unique solutions of the Coulomb friction problem*, Math. Comput. Simul., 82, 2047–2061, 2012.

[6] T. Ligurský: *Theoretical analysis of discrete contact problems with Coulomb friction.* Appl. Math., 57, 263–295, 2012.

[7] Scholtes, S.: *Introduction to piecewise differentiable equations.* Springer Briefs in Optimalization, Springer, Berlin, 2012.

The core problem within a linear approximation problem with multiple right-hand sides

I. Hnětynková, M. Plešinger, D. M. Sima, Z. Strakoš, S. Van Huffel

Faculty of Mathematics and Physics, Charles University in Prague, Prague
Department of Mathematics, Technical University of Liberec, Liberec
Department of Electrical Engineering, ESAT-SCD, Katholieke Universiteit Leuven, Leuven
Faculty of Mathematics and Physics, Charles University in Prague, Prague
Department of Electrical Engineering, ESAT-SCD, Katholieke Universiteit Leuven, Leuven

Consider a linear (orthogonally invariant) approximation problem with multiple right-hand sides

$$AX \approx B, \qquad \text{where} \qquad A \in \mathbb{R}^{m\times n}, \quad X \in \mathbb{R}^{n\times d}, \quad B \in \mathbb{R}^{m\times d}.$$

The *total least squares* (TLS) formulation seeks for a solution X of

$$(A+E)X = B+G \qquad \text{such that} \qquad \min \|[G,E]\|_F.$$

A question of existence and uniqueness of this *TLS solution* has been studied for decades. Golub and Van Loan in the paper [1] showed that even with $d = 1$ the TLS solution may not exist, and when it exists, it may not be unique. The book [7] by Van Huffel and Vandewalle introduced the *nongeneric* approach, extended the Golub–Van Loan's analysis to *two special cases* with $d \geq 1$, and gave the so-called *classical TLS algorithm.* Wei further analyzed problems with nonunique solutions in [8, 9]. The necessary and sufficient condition for existence of TLS solution in the *general case* (with $d \geq 1$) was published in [2]. Our analysis, based on [1, 7], resulted in a new classification of TLS problems.

The single right-hand side case ($d = 1$) was revisited by Paige and Strakoš in [6]. They introduced a minimally dimensioned subproblem of $Ax \approx b$ called a *core problem* always having the unique TLS solution. We extended the core problem concept to the general case $d \geq 1$. Definition and detailed analysis of this core problem can be found in the recent paper [3]. Further analysis of the core problem, based on band generalization of Golub–Kahan iterative bidiagonalization, is prepared for publication; see [4].

In this contribution we concentrate on solvability of the core problem for $d > 1$. Using the properties of the right singular vector subspaces of the corresponding extended core problem matrix $[B_1, A_{11}]$, it will be shown that the core problem with multiple right-hand sides *may not have a TLS solution.* We show that core problems with multiple right-hand sides can have internal structure which allows to interpret the original problem as a direct sum of two (or more) *uncorrelated components.* In such case we call the core problem *reducible.* It will be shown that existence of a TLS solution of a reducible core problem depends on existence of a TLS solution of its components, but also on the relations among singular values of these components. Finally we show that also an *irreducible* core problem with multiple right-hand sides may not have a TLS solution. The analysis of solvability is still under development; see [5].

Acknowledgement: The research of Martin Plešinger has been supported by the ESF grant CZ.1.07/2.3.00/30.0065, "Support of the creation of excellent research and development teams at the Technical University of Liberec".

References

[1] G. H., Golub, C. F. Van Loan: *An analysis of the total least squares problem.* Numer. Anal., 17, 883–893, 1980.

[2] I. Hnětynková, M. Plešinger, D. M. Sima, Z. Strakoš, S. Van Huffel: *The total least squares problem in $AX \approx B$. A new classification with the relationship to the classical works.* SIAM J. Matrix Anal. Appl., 32, 748–770, 2011.

[3] I. Hnětynková, M. Plešinger, Z. Strakoš: *The core problem within a linear approximation problems $AX \approx B$ with multiple right-hand sides.* SIAM J. Matrix Anal. Appl., 34, 917–931, 2013.

[4] I. Hnětynková, M. Plešinger, Z. Strakoš: *Band generalization of the Golub–Kahan bidiagonalization, generalized Jacobi matrices, and the core problem.* In preparation.

[5] I. Hnětynková, M. Plešinger, D. M. Sima, Z. Strakoš, S. Van Huffel: *Remarks on solvability of the core problem with multple right-hand sides in the TLS sense.* In preparation.

[6] C. C. Paige, Z. Strakoš: *Core problem in linear algebraic systems.* SIAM J. Matrix Anal. Appl., 27, 861–875, 2006.

[7] S. Van Huffel, J. Vandewalle: *The total least squares problem: computational aspects and analysis.* SIAM Publications, Philadelphia, PA, 1991.

[8] M. Wei: *The analysis for the total least squares problem with more than one solution.* SIAM J. Matrix Anal. Appl., 13, 746–763, 1992.

[9] M. Wei: *Algebraic relations between the total least squares and least squares problems with more than one solution.* Numer. Math., 62, 123–148, 1992.

Příklady zpracování geometricky složitých modelů pro výpočty proudění podzemní vody

M. Hokr, D. Frydrych, A. Balvín

Technická univerzita v Liberci

1 Úvod

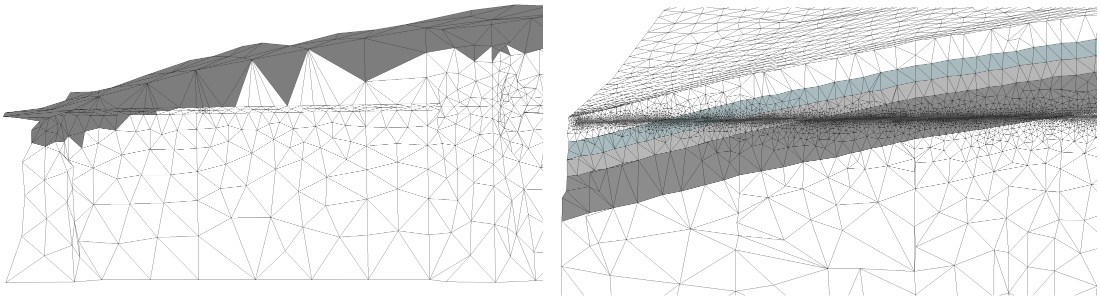
V dosavadních pracích autorů (např. [1]) byly řešeny úlohy průsaku podzemní vody do tunelu – hlavní přínos z hlediska numerických metod bylo použití kombinace 3D a 2D elementů (kontinua a puklin), z hlediska aplikace pak identifikaci parametrů horniny z měřeného průtoku v tunelu (inverzní úloha). Cílem této práce je prezentovat nové nástroje a postupy přípravy geometrie, které dokážou překonat omezení u dosavadních modelů, další řešené úlohy z aplikace na experimenty v podzemí, zmínit typické problémy a ukázat možnosti efektivnějšího řešení v souvislosti se způsoby reprezentace tunelu nebo vrtů v modelu – objektů válcového tvaru, které z hlediska měřítka celého modelu mají liniový charakter.

Ve všech případech je řešena úloha ustáleného filtračního proudění, tj. lineární eliptická parciální diferenciální rovnice s neznámým polem tlaku (piezometrické výšky), na příslušné oblasti, která je kombinací více subdomén stejné nebo různé dimenze (síť puklin-ploch nebo kombinace 2D puklin a 3D kontinua). Použita je smíšená-hybridní metoda konečných prvků implementovaná v softwaru Flow123D vyvíjeném na TU v Liberci [3].

2 Zpracování geometrie ploch ve 3D – puklin

Omezením open-source preprocesoru GMSH, který byl historicky využíván pro výpočetní software Flow123D (včetně převzetí některých datových formátů), je nutnost explicitního zadání všech význačných entit geometrie – tj. bodů, linií a ploch hranice oblasti, i vnitřích rozhraní (mezi materiály) a průniků podoblastí. Jejich určení je v případě většího počtu puklin (ploch v prostoru) obecného směru a přítomnosti tunelu a vrtů ručně téměř nemožné. Jako vhodný nástroj byl nalezen program SALOME [4], open-source projekt založený na principu geometrického modelování. Obsahuje knihovnu základních geometrických entit (bod, úsečka, krychle, koule, atd) a několik knihoven geometrických operací. Generující operace slouží k definování entit vyšších dimenzí pomocí entit nižších dimenzí (např. „vytažením“ úsečky je vygenerována plocha). Transformační operace zajišťují posuny, rotace, zrcadlení a změny měřítka daných geometrických entit. Boolovské operace zajišťují sjednocení, průnik a rozdíl geometrických entit. Z programu je možné po vytvoření geometrie přímo volat generátor diskretizace NETGEN jako jeden z možných alternativ.

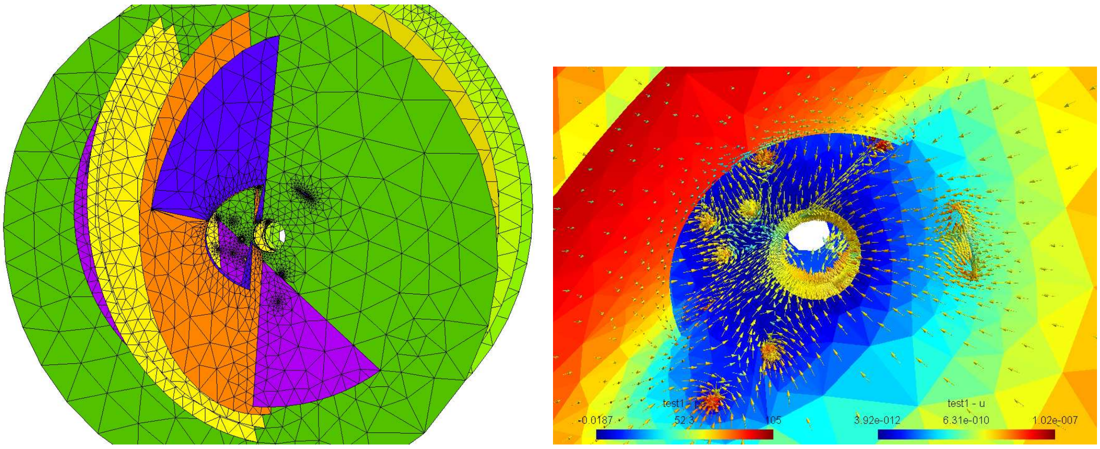
Pro úlohu průsaku do tunelu Bedřichov tak bylo možné jednak použít realističtější tvar tunelu (osmistěn), jednak předepsat pevnou polohu rozhraní mezi různými parametry horniny, zejména díky výpočtu průniků ploch a tunelu, ale i díky efektivnímu automatickému řízení kroku diskretizace (zjemnění kolem průniků a rozhraní). Srovnání modelů bylo předmětem práce [2] a ukázka je na obrázku 1. Síť jemnějšího modelu má přes milion elementů, objemový rozsah modelu je přitom přibližně čtvrtinový proti hrubší variantě s 220000 elementy.



Obrázek 1: Hrubá a jemná geometrie a diskretizace ve svislém řezu 3D úlohy průsaku do tunelu Bedřichov, odstíny jsou podoblasti různých materiálových koeficientů.

Úlohou vycházející z jiné aplikace je proudění v síti puklin v okolí experimentu VITA (vliv zahřívání na vlastnosti horniny) ve štole Josef, podzemním pracovišti ČVUT – jednak přirozený průsak do štoly, jednak při testech tlakování jednotlivých vrtů vedených ze štoly v různých směrech (průměr obvykle 76 mm). Model puklinové sítě vycházel z přímého mapování, určení poloh a orientací na stěně a ve vrtech. Pukliny jsou realizovány jako disky průměru od 9 do 12,5 m. Softwarem SALOME byly určeny liniové průniky mezi jednotlivými puklinami. Dále byly nalezeny linie představující vyústění jednotlivých puklin a to jak do jednotlivých vrtů (elipsy), tak i na hranici modelové oblasti. Síť vygenerovaná na geometrii zájmové oblasti je tvořena 16695 uzly a 33642 trojúhelníkovými elementy. Pohled na síť a ukázku výsledků je na obrázku 2.

Přes úspěšné vygenerování sítí pro uvedené úlohy různého charakteru a měřítka jsou i případy, kdy postup vyžaduje další úpravy geometrie – např. při více průsečících objektů blízko sebe (zejména pro síť puklin obecných směrů) dochází k extrémnímu zjemnění výsledné sítě, nepoužitelnému pro výpočet. Takový problém se typicky objevuje u stochasticky generovaných sítí puklin.



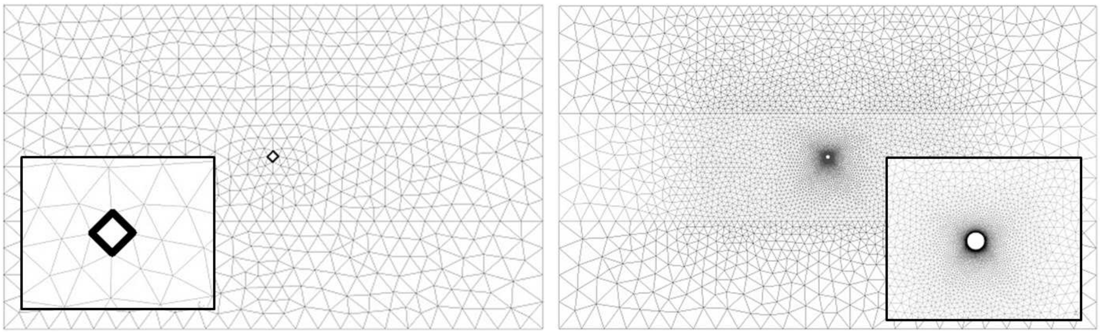
Obrázek 2: Tok v puklinové síti v okolí experimentu VITA ve štole Josef – geometrie, diskretizace a zobrazení výsledků.

3 Reprezentace liniových objektů – tunel, vrty

V úlohách podzemní vody se velmi často pracuje s vrty, které jsou typickým nástrojem jak pozorovat nebo ovlivňovat děje v hornině. Podobnou roli mají tunelové stavby ve větším měřítku. Fyzikálně jde o dutinu v modelové oblasti např. válcového tvaru s definovanou okrajovou podmínkou na jejím povrchu. Numerické řešení na takovéto oblasti komplikuje malý příčný rozměr vzhledem k měřítku úlohy, tj. nutnost velkého zjemnění diskretizace v okolí vrtu nebo tunelu. Přirozeným modelem vrtu by byl jednorozměrný objekt (úsečka), ale předepsání hodnoty odpovádající okrajové podmínce na úsečku uvnitř 3D oblasti vede na matematicky nekorektní úlohu (singularitu). V komerčních softwarech pro podzemní vodu je někdy takováto úloha „skrytě“ řešena – na úrovni diskretizované úlohy je možné předepsat hodnotu do konkrétního uzlu nebo elementu sítě, ale jde pak o řešení závislé na diskretizaci (např. u lineárních konečných prvků nebo konečných objemů se výsledek chová jako úloha s dutinou rozměru odpovídajícího měřítku kroku diskretizace). Tento efekt byl i prakticky testován na úloze bodového zdroje v ploše porovnáním s analytickým řešením radiálního toku, ale nelze čekat, že by bylo možno vztahy zobecnit pro různá numerická schémata a geometrie sítě.

I při reprezentaci tunelu nebo vrtu dutinou v geometrii je významný vliv přesnosti reprezentace tvaru a velikosti a použitého kroku diskretizace. Test byl proveden na několika úlohách, příklad sítí je na obrázku 3. Vztah mezi výsledkem úlohy s přesnou geometrií tunelu a jemnou diskretizací a úlohy s hrubší reprezentací (čtvercový profil větší velikosti použitý z důvodu omezení pro předchozí výpočty 3D úlohy [1]) pak byl použit pro dodatečnou korekci výsledků 3D úlohy o předpokládáný vliv velikosti tunelu a diskretizace [2]. V rámci 2D úlohy i v úloze sítě puklin (samotných ploch ve 3D) není proti 3D úloze (kombinace 2D a 3D) zjemnění tak omezující a úlohu experimentu VITA (obrázek 2) se sadou vrtů bylo možné diskretizovat a řešit s rozumnou výpočetní náročností.

Problém reprezentace vrtu lze řešit kombinací lokálního analytického řešení radiálního toku okolo vrtu (uvnitř jednoho elementu) a numerického řešení na hrubší síti [5], což je využíváno i v aplikacích. Tento přístup je výhodné formálně zapracovat do konceptu rozšířené metody konečných prvků (XFEM), což bude ukázáno na semináři v příspěvku P. Exnera, v návaznosti na dosavadní řešení v literatuře.



Obrázek 3: Sítě pro porovnání vlivu diskretizace profilu tunelu/vrtu ve 2D.

4 Závěr

Představené příklady jednak ukazují možnosti standardních softwarových nástrojů a algoritmů pro generování geometrie a diskretizace, jednak v některých případech omezení vyplývající z malých rozměrů vrtů nebo tunelu vzhledem k měřítku úlohy. Pro takový případ je naznačen možný směr dalšího vývoje a použití numerických metod.

Poděkování: Práce byla podporována Ministerstvem průmyslu a obchodu v rámci projektů č. FR-TI3/325 (VITA) a FR-TI3/579 (Geostab). Na získání dat poloh puklin použitých v modelu a na měření ve vrtech se podílel celý řešitelský tým projektu FR-TI 3/325.

Reference

[1] M. Hokr, I. Škarydová, D. Frydrych: *Modelling of tunnel inflow with combination of discrete fractures and continuum.* Comp. Vis. Sci. 15, (1), 21–28, 2013.

[2] M. Hokr, A. Balvín, D. Frydrych, I. Škarydová: *Meshing issues in the numerical solution of the tunnel inflow problem.* In: V. Marascu-Klein (ed.), Mathematical Models in Engineering and Computer Science (Brasov, Romania, June 1–3, 2013), WSEAS Press, pp. 162–168, 2013.

[3] TUL: *FLOW123D version 1.6.5, Documentation of file formats and brief user manual.* NTI TUL, Online: `http://dev.nti.tul.cz/trac/flow123d`, 2012.

[4] *Salome – The open source integration platform for numerical simulation.* `http://www.salome-platform.org/`, 2012.

[5] Z. Chen, Y. Zhang: *Well flow models for various numerical methods.* Int. J. Numer. Anal. Mod., 6, 375–388, (2009).

The efficient reconstruction formula for the amplitudes of the rigid body modes in FETI for contact problems

D. Horák[1,2], *V. Hapla*[1,2], *L. Říha*[2]

[1] Dep. of Applied Math., VŠB-TUO, 17. listopadu 15, CZ 708 33 Ostrava,
[2] IT4Innovations, VŠB-TUO, 17. listopadu 15, CZ 708 33 Ostrava

1 Introduction

The presentation deals with the relation of the vector of amplitudes of the rigid body modes in FETI-1 or TFETI methods [1] and the multipliers generated by SMALSE algorithm [2] for the solution of nonlinear problems. Once the dual solution - vector of Lagrange multipliers is computed, to get the primal solution it is necessary to exclude those rows in the constraint matrix with inequalities which correspond to zero Lagrange multipliers, compute new coarse space matrix and coarse problem matrix, factorize it and solve this coarse problem [3]. Concerning e.g. elasto-plastic problems, we have to solve this reconstruction in each time step. The new formula computing this vector of the rigid body modes using a part of the residual and the SMALSE multipliers is presented and the scalability improvement is illustrated by numerical experiments done with FLLOP library [4] developed at our department.

Acknowledgement: This work was financially supported by the Projects of major infrastructures for research, development and innovation of Ministry of Education, Youth and Sports (LM2011033), the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), the PRACE project funded in part by the EUs 7th Framework Programme (FP7/2007-2013) under grant agreement no. RI-261557.

References

[1] Z. Dostal, D. Horak, R. Kucera: *Total FETI an easier implementable variant of the FETI method for numerical solution of elliptic PDE.* Commun. in Numerical Methods in Engineering, 22, 1155–1162, 2006.

[2] Z. Dostal: *Optimal quadratic programming algorithms: with applications to variational inequalities.* Springer Optimization and Its Applications, ISBN: 978-0-387-84805-1, 2009.

[3] V. Hapla, D. Horak: *TFETI coarse space projectors parallelization strategies.* In Proceedings of PPAM 2011, Lecture Notes in Computer Science, 7203 LNCS (PART 1), 152–162, ISSN: 03029743, ISBN: 978-364231463-6, DOI: 10.1007/978-3-642-31464-3 16, 2012.

[4] `http://industry.it4i.cz/produkty/fllop/`

LIF data evaluation – image processing algorithms

M. Isoz

Institute of Chemical Technology, Prague

1 Introduction

Measurements of the liquid films and related forms of flow are often performed via an optical experimental technique[1, 2]. One of such methods is so called Light Induced Fluorescence (LIF).

It is based on the principle of adding a marker to the measured liquid, illumination of the liquid by a monochromatic light and measurement of the intensities of light emitted by the marked liquid.

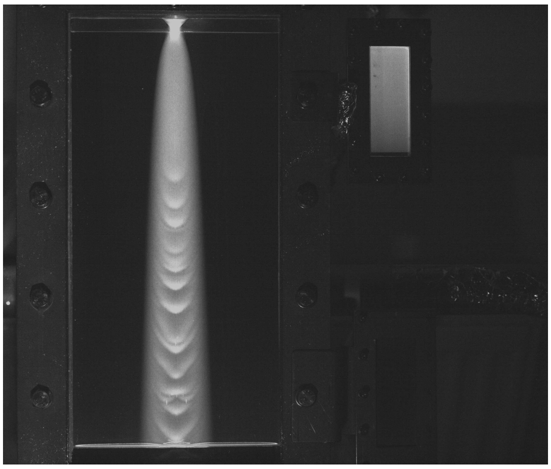


Figure 1: Example of experimental data obtained during LIF based measurement of gas–liquid interface of rivulet falling down on an inclined plate.

Example of LIF based experimental method data output can be seen in Figure 1. Crucial parts of image are the inclined plate on which the studied rivulet is positioned and the calibration cell in upper right corner of the image. Calibration cell serves as a scale for conversion of measured light intensities in local film thicknesses.

For automatization of the data evaluation process, locating these two objects in images obtained during experiments is of key importance.

Moreover, as the measurements are quite easy to perform and quick, usually more than 40 images with the same position of these two elements were available. This fact can be used to refine the found coordinates through simple statistics.

Each experimental image corresponds to a matrix A of type (m, n), where m and n are the vertical and horizontal resolutions of the image. Elements of matrix A, (a_{ij}) are the pixels of processed image, $(a_{ij}) \in \langle 0; 1 \rangle$. Case of $a_{ij} = 0$ corresponds to a black pixel and $a_{ij} = 1$ to the white one.

Both described algorithms work with black and white images. The transformation of grayscale image to black and white is done through comparison of matrix elements to a preset threshold.

Result of this transformation is matrix $\tilde{A}$ with

$$\tilde{a}_{ij} = \begin{cases} 1 & \text{if} \quad a_{ij} < \text{threshold} \\ 0 & \text{if} \quad a_{ij} \geq \text{threshold}\,. \end{cases} , \quad i = 1, \ldots, m, \; j = 1, \ldots, n \tag{1}$$

2 Calibration cell finding algorithm

The calibration cell is a very distinct object. Hence the algorithm does not have to be very complex. Besides, as the program was implemented in MATLAB, most of the steps have been performed via functions available in Image Processing Toolbox of this environment.

However, the most crucial step of the algorithm, selection of the calibration cell from all the candidate elements that passed through basic size based filtering, had to be developped. The proposed selection algorithm is based on comparing values of custom objective function.

From Figure 1, it is clear that the calibration cell is almost perfectly rectangular object. Additionally, it is always placed vertically. So the custom objective function penalizes the elements for not being rectangular and for not having edges parallel to image borders. This penalization is done through sum of two almost independent terms.

The first term of objective function penalizes checked object for not being rectangular and for not being oriented in the above described way.

Let us denote the term as $\Delta_R A$ and define it by relation,

$$\Delta_R A = \frac{\delta x \, \delta y - \int_S dS}{\delta x \, \delta y} \,, \tag{2}$$

where $\int_S dS$ stands for actual area of the element calculated directly from the number of pixels of which it consists. Other terms of the equation, δx and δy are the maximal distances between pixels in horizontal (x) and vertical (y) direction, respectively.

The product $\delta x \, \delta y$ always stands for the maximal possible area of tested element. Only for exactly vertically or horizontally placed rectangular elements, $\delta x \, \delta y = \int_S dS$, hence dividing their difference by $\delta x \, \delta y$ ensures

$$\Delta_R A \in \langle 0, 1) \,. \tag{3}$$

Other custom objective function term is penalizing the object for not being rectangular. It is based on the sum of dot products of the direction vectors of elements sides. Main idea is that the tangent vectors defined in the clockwise and counterclockwise directions in the top left and bottom right corner of the tested element should be perpendicular to each other. The second term of the objective function is denoted by DP and defined as follows,

$$DP = \sum_{i=1}^{2} \frac{\vec{u}_i}{\|\vec{u}_i\|} \cdot \frac{\vec{v}_i}{\|\vec{v}_i\|} \,, \tag{4}$$

where $\vec{u}_i$, $i = 1, 2$ are the tangent vectors defined in the clockwise direction in opposite corners and $\vec{v}_i$, $i = 1, 2$ are the vectors defined in the counterclockwise direction.

Value of the objective function for the i-th tested element is then calculated as sum of the above defined terms,

$$S_i = \Delta_R A_i + DP_i \,. \tag{5}$$

Algorithm can malfunction for the case of multiple distinct, roughly rectangular elements present in the image. However, testing proved it to be very dependable.

3 Plate finding algorithm

The second crucial element on the images from experiments is the inclined plate with measured rivulet itself. As it can be seen in Figure 1, the plate edges are much less distinct than the calibration cell. This resulted in need of much more sophisticated and computational time consuming algorithm.

First, a preprocessing had to be done, where the approximate position of the plate had to be specified manually, the rivulet itself was removed from the selected area and the contrasts in the resulting image were enhanced. The result was a matrix B, submatrix of A containing only the plate itself. Image represented by B was transformed in black and white using the relation (1). Finally the Hough transform was used to detect the line segments in $\tilde{B}$[3, 4].

The result of previously described procedure were line segments represented by coordinates of their starting and ending points. As it can be deduced from the Figure 1, only the vertical and horizontal lines near the borders of manually selected area were relevant for the plate position estimation.

All the other found line segments had to be excluded. Furthermore, the remaining lines had to be sorted with respect to the plate edge they were adjacent to. At last, the plate edges positions were estimated from the weighted mean of relevant line segments coordinates. Weights were based on the line segment length, with longer segments taken as more important.

The process is depicted in algorithm based on pseudo MATLAB syntax below.

```
% constants
Dx, Dy % maximal non-horizontality (non-verticality) of kept lines
Eh, Ev % maximal tolerated distance from the selection edges
Sh, Sv % horizontal and vertical sizes of selected area
for i = 1:number of found lines
% prepare the statements for processing the found segments
ishorizontal= abs(xStart-xEnd) < Dx; isvertical  = abs(yStart-yEnd) < Dy;
isleft      = abs(xStart-0)    < Eh; isright     = abs(xStart-Sh)   < Eh;
istop       = abs(yStart-0)    < Ev; isbottom    = abs(yStart-Sv)   < Ev;
% if the line is not vertical or horizontal, discard it
if ¬isvertical && ¬ishorizontal, discard tested segment; break; end
% sort lines with respect to selected area edges they are adjacent to
% discard the rest (lines in center of the selected area)
if isvertical && (isleft || isright)
        Vweight(i)    = abs(yStart-yEnd)/Sv; % calculate the weight of segment
    if isleft
        propLeft(i)   = mean([xStart xEnd]); % save its horizontal position
    else
        propLeft(i)   = mean([xStart xEnd]);
    end
elseif ishorizontal && (istop || isbottom)
    Hweight(i)        = abs(xStart-xEnd)/Sh; % calculate the weight of segment
    if istop
        propTop(i)    = mean([yStart yEnd]); % save its vertical position
    else
        propBottom(i) = mean([yStart yEnd]);
    end
else, discard tested segment;break;
end;end
```

After all the lines were tested, the plate edges position was calculated from proposed coordinates and their weights.

4 Statistical processing of algorithms results

Result of the above described algorithms is a matrix of found elements coordinates on experimental images, C. Each row of C corresponds to one processed image and each column to one found coordinate. As it was mentioned before, there are usually more than 40 images with the same experimental apparatus set up. This fact can be used to refine the found coordinates and to compensate for possible algorithms malfunctions.

Final position of the looked up elements is calculated as mean value of each column of C with previously excluded outliers.

Outliers exclusion is based on the data kurtosis[5]. From each column of matrix C are left out all the values not satisfying the equation

$$|(c_i)_j - \mu_j| \geq \alpha_j \sigma_j, \quad i = 1, \ldots, \text{number of images}\,. \tag{6}$$

In (6), μ_j is the mean value and σ_j is the standard deviation of the j-th column of C and the coefficient α_j is calculated from the columns kurtosis by the formula

$$\alpha_j = \frac{7}{\text{Kurt}((c_i)_j)}, \quad i = 1, \ldots, \text{number of images}\,. \tag{7}$$

For the case of standard distribution, the advised value of numerator in (7) is 9[3]. However, as the distrubution of found coordinate should be closer to δ-function (the experimental set up was not tempered with), value 7 was chosen and proven by testing as more appropritate.

5 Conclusion

Modern, LIF based measurements are very fast and with improving quality of digital capturing devices also accurate experimental techniques. Unfortunately, the obtained data are only as good as it is the image processing method used for their evaluation. With the above explained algorithms, it is possible to automatically and precisely locate the most important elements on experimental images and improve the quality of measured data.

References

[1] S. V. Alekseenko, V. A. Antipin, A. V. Bobylev, D. M. Markovich: *Application of PIV to velocity measurements in a liquid film flowing down an inclined cylinder.* In: Exp. Fluids, 43, 197–207, 2008.

[2] T. Hagemeier, M. Hartmann, M. Kühle, D. Thévenin, K. Zähringer: *Experimental characterization of thin films, droplets and rivulets using LED fluorescence.* In: Exp. Fluids, 52, 361–374, 2011.

[3] Mathworks, Inc. MATLAB Documentation. `http://www.mathworks.com/help/` (accessed Dec 15, 2013).

[4] R. O. Duda, P. E. Hart: *Use of the Hough transformation to detect lines and curves in pictures.* In: Comm. ACM, 15, 11–15, 1971.

[5] J. R. M. Hosking: *Moments or L-moments? An example comparing two measures of distributional shape.* In: The American Statistician, 46, (3), 186–189, 1992.

Different types of noncommutative algebras

D. Janovská, G. Opfer

Institute of Chemical Technology, Prague
University of Hamburg, Hamburg

We will study various types of noncommutative algebras as they were invented by Sir William Rowan Hamilton in 1843 and six years later, in 1849, by Sir James Cockle, namely we will study four algebraic systems: quaternions, coquaternions, tessarines, and hyperbolic quaternions.

Quaternions

Let us denote by $\mathbb{H}$ the field of quaternions, which is $\mathbb{R}^4$ equipped with a special multiplication rule which makes $\mathbb{R}^4$ a skew field. In order to explain that, let $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ be the four standard basis elements in $\mathbb{H}$. They obey the following multiplication rules, see (6):

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1; \quad \mathbf{ij} = \mathbf{k},\ \mathbf{jk} = \mathbf{i},\ \mathbf{ki} = \mathbf{j}. \tag{1}$$

Instead of $a := a_1 + a_2\mathbf{i} + a_3\mathbf{j} + a_4\mathbf{k}$ we write equivalently also $a = (a_1, a_2, a_3, a_4)$. Let $a := (a_1, a_2, a_3, a_4)$, $b := (b_1, b_2, b_3, b_4)$. Then, the multiplication rules (1) imply

$$\begin{aligned} ab \;:=\; & (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4,\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3, \\ & a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2,\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1). \end{aligned} \tag{2}$$

And we see from the above multiplication rule, that the set of quaternions of the form $a := (a_1, 0, 0, 0)$ is isomorphic to the field of real numbers, denoted by $\mathbb{R}$, and the set of quaternions of the form $a := (a_1, a_2, 0, 0)$ is isomorphic to the field of complex numbers, denoted by $\mathbb{C}$. Let $a := (a_1, a_2, a_3, a_4)$. Then, $\overline{a} := (a_1, -a_2, -a_3, -a_4)$ will be called conjugate of a. The absolute value of a is denoted by $|a|$ and defined by $|a| := \sqrt{a_1^2 + a_2^2 + a_3^2 + a_4^2}$. And for all $a, b \in \mathbb{H}$ there are the rules

$$|a|^2 = a\overline{a} = \overline{a}a, \quad |ab| = |ba| = |a||b|, \quad \overline{ab} = \overline{b}\,\overline{a}, \quad \Re(ab) = \Re(ba), \quad a^{-1} = \frac{\overline{a}}{|a|^2}, \tag{3}$$

where the last rule applies only for $a \neq 0$. Let us note that only real quaternions commute with all other quaternions, i.e. the center of $\mathbb{H}$ is $\mathbb{R}$.

The field $\mathbb{H}$ is isomorphic to a certain class of matrices in $\mathbb{C}^{2\times 2}$. Let $a = (a_1, a_2, a_3, a_4) \in \mathbb{H}$. Let us put $w = a_1 + a_2\mathbf{i}$, $z = a_3 + a_4\mathbf{i}$. Then the set of matrices of the form

$$\widetilde{\mathbf{H}} = \begin{pmatrix} w & z \\ -\overline{z} & \overline{w} \end{pmatrix}$$

with ordinary matrix addition and multiplication is isomorphic to $\mathbb{H}$, [8].

This leads to complex systems of equations with matrices in which a scalar element $a \in \mathbb{R}$ is replaced by a 2×2 matrix with complex elements. Due to the isomorphism it means that we can work with quaternionic matrices. It has some advantages (increased accuracy, economy of storage), but on the other hand it needs more computational effort.

The field $\mathbb{H}$ is also isomorphic to a certain class of matrices in $\mathbb{R}^{4\times 4}$.

Let $a = (a_1, a_2, a_3, a_4) \in \mathbb{H}$. We introduce the mapping $\omega : \mathbb{H} \longrightarrow \mathbb{R}^{4\times 4}$ by

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

The mapping ω represents the isomorphic image of a quaternion $a = (a_1, a_2, a_3, a_4)$ in the matrix space $\mathbb{R}^{4\times 4}$.

Coquaternions

The coquaternions or split–quaternions are elements of a 4-dimensional associative algebra introduced in 1849 by Sir James Cockle (1819–1895),[2], mathematician and lawyer in Australia. Like the quaternions, they form a four dimensional real vector space equipped with a multiplicative operation, see (6). Unlike the quaternion algebra, coquaternions contain zero divisors, nilpotent elements, and nontrivial idempotents. As a mathematical structure, they form an algebra over the real numbers, which is isomorphic to the algebra of all real 2×2 matrices.

The algebra of coquaternions will be abbreviated by $\mathbb{H}_{\text{coq}}$. Coquaternions obey multiplication rules given in (6). Let $a := (a_1, a_2, a_3, a_4)$, $b := (b_1, b_2, b_3, b_4)$. The explicit multiplication rule for the product ab is

$$\begin{aligned} ab \;\; := \;\; & a_1b_1 - a_2b_2 + a_3b_3 + a_4b_4 + (a_1b_2 + a_2b_1 - a_3b_4 + a_4b_3)\mathbf{i} + \\ & (a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2)\mathbf{j} + (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)\mathbf{k}\,. \end{aligned} \tag{5}$$

Also here, only real coquaternions commute with all other coquaternions, i.e. the center of $\mathbb{H}_{\text{coq}}$ is $\mathbb{R}$.

In the following table, the multiplication rules for quaternions and coquaternions are listed. Two tables differ only by signs of the red figures.

$\mathbb{H}$	**1**	**i**	**j**	**k**
1	**1**	**i**	**j**	**k**
i	**i**	−**1**	**k**	−**j**
j	**j**	−**k**	−**1**	**i**
k	**k**	**j**	−**i**	−**1**

$\mathbb{H}_{\text{coq}}$	**1**	**i**	**j**	**k**
1	**1**	**i**	**j**	**k**
i	**i**	−**1**	**k**	−**j**
j	**j**	−**k**	**1**	−**i**
k	**k**	**j**	**i**	**1**

(6)

Theorem 1 Let $a = a_1 + a_2\mathbf{i} + a_3\mathbf{j} + a_4\mathbf{k} \in \mathbb{H}_{\text{coq}}$ and define the matrix

$$\mathbf{C}_4 := \begin{pmatrix} a_1 & -a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & a_4 & a_1 & -a_2 \\ a_4 & -a_3 & a_2 & a_1 \end{pmatrix}. \tag{7}$$

Then, the set of all matrices of the type $\mathbf{C}_4$ forms an algebra, and this algebra is isomorphic to the algebra of coquaternions. □

For proof, see [4]. Let us remark, that two matrices $\omega(a)$ in (4) and $\mathbf{C}_4$ in (7) differ only by signs of the red elements.

Let $a = a_1 + a_2\mathbf{i} + a_3\mathbf{j} + a_4\mathbf{k} \in \mathbb{H}_{\rm coq}$. The algebra of coquaternions is also isomorphic to the algebra of real 2×2 matrices, [5]:

$$\begin{aligned} \mathbf{C}_2 &= a_1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + a_2 \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} + a_3 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + a_4 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad (8) \\ &= \begin{pmatrix} a_1 + a_4 & a_2 + a_3 \\ -a_2 + a_3 & a_1 - a_4 \end{pmatrix}, \quad \text{and} \\ \mathbf{C}_2^{-1} &= \frac{1}{d} \begin{pmatrix} a_1 - a_4 & -a_2 - a_3 \\ a_2 - a_3 & a_1 + a_4 \end{pmatrix}, \quad d := a_1^2 + a_2^2 - a_3^2 - a_4^2 \neq 0. \end{aligned}$$

If we denote the four basis elements in the order of the equation (8) by $\mathbf{E}, \mathbf{I}, \mathbf{J}, \mathbf{K}$, then they obey the same multiplication rules as $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ in (6). An algebra of this type, is also called a split algebra, in the current case the algebra of split quaternions, [5].

Let $a = (a_1, a_2, a_3, a_4)$ be a coquaternion. We define the conjugate of a (notation $\bar{a}$ or $\mathrm{conj}(a)$) and modulus abs_2 by

$$\bar{a} := (a_1, -a_2, -a_3, -a_4), \quad \mathrm{abs}_2(a) := a_1^2 + a_2^2 - a_3^2 - a_4^2. \qquad (9)$$

The quantity abs_2 may be negative, it is not the square of a norm. The coquaternions came to be called split-quaternions due to the division into positive and negative terms in the modulus function. Let b be another coquaternion. There are the following rules:

$$a\bar{a} = \bar{a}a = \mathrm{abs}_2(a), \quad \mathrm{abs}_2(ab) = \mathrm{abs}_2(ba) = \mathrm{abs}_2(a)\mathrm{abs}_2(b),$$
$$\overline{(ab)} = \bar{b}\bar{a}, \quad \Re(ab) = \Re(ba).$$

The coquaternion a will be called singular if $\mathrm{abs}_2(a) = 0$. If a is nonsingular (= not singular = invertible), then

$$aa^{-1} = a^{-1}a = (1, 0, 0, 0) \text{ holds for } a^{-1} = \frac{\bar{a}}{\mathrm{abs}_2(a)}.$$

Tessarines

A Tessarine System is a system in $\mathbb{R}^4$ equipped with the multiplicative operation defined in (10). The tessarines are best known for their subalgebra of real tessarines $t = a_1 + a_3\mathbf{j}$, also called split-complex numbers, which express the parametrization of the unit hyperbola. James Cockle introduced the tessarines in 1848 in a series of articles in Philosophical Magazine, [2].

In 2009 mathematicians proved a fundamental theorem of tessarine algebra: a polynomial of degree n with tessarine coefficients has n^2 roots, counting multiplicity, [7].

Linear representation:

Let $t = a_1 + a_2\mathbf{i} + a_3\mathbf{j} + a_4\mathbf{k}$ be a tessarine. Let us note that, since $\mathbf{ij} = \mathbf{k}$, we have $t = (a_1 + a_2\mathbf{i}) + (a_3 + a_4\mathbf{i})\mathbf{j}$. The mapping

$$t \mapsto \begin{pmatrix} w & z \\ z & w \end{pmatrix}, \quad w = a_1 + a_2\mathbf{i}, \quad z = a_3 + a_4\mathbf{i},$$

is a linear representation of the algebra of tessarines as a subalgebra of 2×2 complex matrices. For instance, $\mathbf{ik} = \mathbf{i}(\mathbf{ij}) = (\mathbf{ii})\mathbf{j} = -\mathbf{j}$ has the linear representation

$$\begin{pmatrix} \mathbf{i} & 0 \\ 0 & \mathbf{i} \end{pmatrix} \begin{pmatrix} 0 & \mathbf{i} \\ \mathbf{i} & 0 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

Note that unlike most matrix algebras, the algebra of tessarines is a commutative algebra.

The multiplication rules for tessarines and hyperbolic quaternions are listed in the following tables. These tables differ from the multiplication table for quaternions only by signs of the red figures.

Tess.	1	**i**	**j**	**k**
1	**1**	**i**	**j**	**k**
i	**i**	−**1**	**k**	−**j**
j	**j**	**k**	**1**	**i**
k	**k**	−**j**	**i**	−**1**

Hyp.quat.	**1**	**i**	**j**	**k**
1	**1**	**i**	**j**	**k**
i	**i**	**1**	**k**	−**j**
j	**j**	−**k**	**1**	**i**
k	**k**	**j**	−**i**	**1**

(10)

Hyperbolic quaternions

The system of hyperbolic quaternions is a nonassociative algebra in $\mathbb{R}^4$ equipped with multiplicative operation defined in (10). Unlike the ordinary quaternions, the hyperbolic quaternions are not associative. For example, $(\mathbf{ij})\mathbf{j} = \mathbf{kj} = -\mathbf{i}$, while $\mathbf{i}(\mathbf{jj}) = \mathbf{i}$.

In the fundamental work [6], A. Macfarlane defined the concepts of the theory of relativity of space and time using hyperbolic quaternions. More resent results can be found in [1].

Acknowledgement: The research was supported by the German Science Foundation, DFG, under the contract number OP 33/19-1.

References

[1] K. Carmody: *Circular and hyperbolic quaternions, octonions, and sedenions.* Journal Applied Mathematics and Computation, 28 47–72, 1988.

[2] J. Cockle: *On the symbols of algebra and on the theory of tessarines.* In London-Dublin-Edinburgh Philosophical Magazine, 34 406–410, 1849.

[3] K. Gürlebeck, W. Sprössig: *Quaternionic and Clifford calculus for physicists and engineers.* Wiley, Chichester, 371 p., 1997.

[4] D. Janovská, G. Opfer: *Linear equations and the Kronecker product in coquaternions.* Mitt. Math. Ges. Hamburg, 33, 1–16, 2013.

[5] T. Y. Lam: *The algebraic theory of quadratic forms.* W. A. Benjamin, Reading, Massachusetts, 343 pp., 1973.

[6] A. Macfarlane: *On the imaginary of algebra.* Proceedings of the American Association for the Advancement of Science, v. XLI, 33–55, 1892.

[7] R. D. Poodiack, K. J. LeClair: *Fundamental theorems of algebra for the perplexes.* The College Mathematics Journal 40, 322–335, 2009.

[8] B. L. van der Waerden: *Algebra I.* 5. Aufl., Springer, Berlin, 1960 (1st ed. 1936).

Slightly generalized regular space decompositions

A. Kolcun

Institute of Geonics AS CR, Ostrava

Space decomposition methods represent an important part of numerical modelling process. In many applications it is suitable to use the simplest case of decomposition – regular rectangular grid: the whole raster graphic concept can be seen from this point of view. Different raster concept, based on regular hexagonal mesh, is analyzed e.g. in [18]. Generalized task – decomposition of the space to the set of the identical elements have been presented in various research areas since a long time, and it is connected with Hilbert's 18th problem [9], [10].

Within the numerical methods development, the space discretization became an important tool of the shape expressivity. The domain of interest is decomposed to simple polyhedral elements. In the simplest case of linear approximation, the triangles for 2D tasks and tetrahedra for 3D tasks are used. There is a wide range of generators used for the decomposition, e.g. [7], [6], which produces meshes with irregular structure of nodes coincidency. For good interpolation properties, the condition like Delaunay one [3], [5] is required in the case of the isotropical environment. (In 2D case, the Delaunay triangulation maximizes the minimum angle. Compared to any other triangulation of the points, the smallest angle in the Delaunay triangulation is at least as large as the smallest angle in any other. However, the Delaunay triangulation does not necessarily minimize the maximum angle.)

Due to the fact, that for geometric modelling (CAD) parametrical models based on the tool of NURBS curves, surfaces and volumes are used, discretization with regular structure of node coincidency is important and wide spread. Moreover, the common basis for both geometric and physically based modelling can be founded [11], [4].

Using current methods, creating 3D models is an extremely time-consuming, unreliable, and labour-intensive process. So, when the geometry information is obtained e.g. from computer tomograph or similar devices, i.e. in the form of pixel/voxel grid, it is reasonable to create the space decomposition in the same or similar way. The discretization error – aliasing – has a very local character in this case only [2]. Moreover, this error can be eliminated as mentioned below.

Finer mesh This way, however, leads to substantial increasing of memory demands.

Adjusting the geometry Some of the grid nodes are shifted according to prescribed geometry. It is proved [16] that even in the simplest case of adjusting (local displacement of the grid nodes only, which are the most close to the prescribed shape) resulting mesh holds the Delaunay property.

Pixel/voxel partitioning This approach gives a wide variability of the shape expression: from four types of possible tetrahedra we can create 72 different conform decompositions [12], [1], [17]. On the other hand, not all configurations of diagonals are admissible: in some cases only nonconform decompositions are possible, and in some cases no decompositions are possible. This fact can be considered as a generalization of the decomposition of Schönhardts polyhedron [19], [20]. This drawback can be eliminated. We can move the node in such way, that planar quadrilateral, which is nonconformly partitioned, become a tetrahedron. In this case voxel can be decomposed to 6-13 tetrahedra. [13]. So, in this case both regularity of geometry and nodes coincidency is spoilt.

Goldberg's tiling There are several methods how to decompose 3D space into the set of the same tetrahedra [8], [9], [21]. Comparison of these decompositions in [14] shows the benefit of Goldbergs tiling. Within this class of decompositions we can find such one, based on tetrahedron, close to the regular one (each face is isoscelles triangle with edges ratio $\sqrt{(\frac{4}{3})} : 1 : 1$). Moreover, Cartesian indexation of nodes with three-indexes can be used and there are four different assembling schemes how we can compose the voxel form these six tetrahedra [15].

Acknowledgement:
This work has been supported by the grants FR-TI3/579 and GA13-18652S.

References

[1] T. Apel, N. Duvelmeyer: *Transformation of hexahedral FE-mesh into tetrahedral meshes according to quality criteria.* Computing, 71, 293–304, 2003.

[2] P. Arbenz, C. Flaig: *On smoothing in Voxel based finite element analysis of trabecular bone.* In: LSSC2007 Proc. (Lirkov, I., Margenov, S., Wasniewski, J. eds.), Lecture Notes in Computer Science 4818, Springer 2008, 69–77.

[3] M. de Berg: *Computational geometry – algorithms and applications.* Springer-Verlag, 2008.

[4] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs: *Isogeometric analysis: toward integration of CAD and FEA.* John Wiley & Sons, 2009.

[5] Delaunay Triangulation, `http://en.wikipedia.org/wiki/Delaunay_triangulation`. Retrieved 2014-01-06.

[6] H. Edelsbrunner: *Geometry and topology for mesh generation.* Cambridge University Press, 2001.

[7] P. J. Frey, P. L. George: *Mesh generation: application to finite elements.* Hermes Science, 2000.

[8] M. Goldberg: *Three infinite families of tetrahedral space/fillers.* Journal of Combinatorial Theory (A), 16, 348–354, 1978.

[9] B. Grunbaum, G. C. Shepard: *Tilings with congruent tiles.* Bulletin of American Math. Soc., 3, (3), November 1980, 951–973.

[10] *Hilberts problems.* `http://en.wikipedia.org/wiki/Hilbert's_problems`. Retrieved 2014-01-06.

[11] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs: *Isogeometric analysis: CAD, finite elements, NURBS,exact geometry and mesh refinement.* Comput. Methods Appl. Mech. Engrg. (Elsevier), 194, 4135–4195, 2005.

[12] A. Kolcun: *Conform decompositions of cube.* In: Proceedings of Spring School on Computer Graphics, Comeniu University Bratislava, 1994, 185–191.

[13] A. Kolcun: *Non-conformity problem in 3D grid decomposition.* Journal of WSCG, 10, (1), 249–254, 2002.

[14] A. Kolcun: *(Semi) regular tetrahedral tilings.* In: WSCG 2013 Conference – Communication Papers Proceedings, ZČU Plzeň, 145–150.

[15] A. Kolcun: *Voxel representation in the context of Goldbergs tilings of 3D space.* In: Proceedings of 33-rd Conference on Geometry and Computer Graphics. VŠB-TU Ostrava 2013, 149–154.

[16] P. Kahánek, A. Kolcun: *Bresenham's regular mesh deformation and angle criteria.* In: Proc of the 25-th Conference on Geometry and Computer Graphics. VŠB-TU Ostrava 2005, 95–102.

[17] J. Lubojacký: *Basic algorithms of rasterization for generalized raster lattice* (in czech). Master Thesis, University of Ostrava, 2013.

[18] M. Middleton, J. Sivaswamy: *Hexagonal image processing.* Springer, 2005.

[19] J. Rambau: *On a generalization of Schönhardt's polyhedron.* Combinatorial and Computational Geometry, 52, 501–516, 2005.

[20] *Schönhardt's polyhedron.* `http://en.wikipedia.org/wiki/Schonhardt_polyhedron`. Retrieved 2014-01-06.

[21] M. Senechal: *Which tetrahedra Fill Space?* Mathematical Magazine, 54, (5), 227–243, 1981.

Noise revealing in Golub-Kahan bidiagonalization as a mean of regularization in discrete inverse problems

M. Kubínová, I. Hnětynková

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Computer Science, Academy of Sciences of the Czech Republic

1 Introduction

We consider an ill-posed linear system

$$Ax \approx b, \qquad A \in \mathbb{R}^{n\times n}, \qquad b = b^{\text{exact}} + b^{\text{noise}} \in \mathbb{R}^n, \tag{1}$$

where A is a nonsingular matrix and b^{noise} is an unknown perturbation of the right-hand side b^{exact}, $\|b^{\text{noise}}\| \ll \|b^{\text{exact}}\|$. Moreover, we assume that the matrix A is a discretized *smoothing* operator with singular values decaying gradually to zero and the vector b^{noise} represents noise (for simplicity, we assume white noise, that is, the noise has flat frequency characteristics). The aim is to approximate the exact solution

$$x^{\text{exact}} \equiv A^{-1}b^{\text{exact}}.$$

Since A has smoothing property, the operator A^{-1} amplifies high-frequencies. For noise significant enough, the discrete Picard condition is violated, which makes the naive solution $x^{\text{naive}} \equiv A^{-1}b$ completely meaningless, and problem (1) has to be regularized. A successful regularization method has to suppress the devastating effect of high-frequency noise while preserving sufficient information from the data. The amount of regularization is usually controlled by a regularization parameter and choice of this parameter represents the most difficult part of solving discrete inverse problems [2]. One can also attempt to eliminate (at least to some extent) the high-frequency part of the noise. Assume, we have an estimate $\tilde{b}^{\text{noise}}$ of the noise vector b^{noise}. Then, a straightforward approach to solve problem (1) is to subtract this estimate from the right-hand side b, and solve the system

$$Ax = b - \tilde{b}^{\text{noise}}. \tag{2}$$

We want system (2) to have better overall properties than the original problem (1). In our case, the aim is to dampen the high frequencies coming from noise. The key part of this approach is to find an estimate $\tilde{b}^{\text{noise}}$. In the following, we will present a cheap parameter-free method for finding such an estimate using Golub-Kahan bidiagonalization [1].

2 Estimating noise via noise propagation in Golub-Kahan bidiagonalization

Golub-Kahan bidiagonalization is an iterative procedure that is widely used in solving large linear systems. Given the initial vectors $w_0 \equiv 0$, $s_1 \equiv b/\beta_1$, $\beta_1 \equiv \|b\| \neq 0$, it computes

$$\begin{aligned} \alpha_k w_k &= A^T s_k - \beta_k w_{k-1}\,, \qquad \|w_k\| = 1\,, \\ \beta_{k+1} s_{k+1} &= A w_k - \alpha_j s_k\,, \quad \|s_{k+1}\| = 1, \end{aligned} \tag{3}$$

until $\alpha_k = 0$ or $\beta_{k+1} = 0$, or until $k = n$. Vectors s_k and w_k form the bases of Krylov subspaces $\mathcal{K}_K(AA^T, b)$ and $\mathcal{K}_K(A^TA, A^Tb)$ respectively.

In hybrid methods (see, e.g., [3, 4]), Golub-Kahan bidiagonalization is used as outer regularization (regularization of the original large problem by projection). Moreover, as shown in [5], due to the orthogonalization, one may also make use of the propagation of the noise through the bidiagonalization process. Since the starting vector s_1 is polluted by white noise, this noise is present in all subsequent left bidiagonalization vectors s_k. As shown in [5], the size of the noise in the vector s_{k+1} can be related to the amplification factor

$$\rho_k^{-1} \equiv \prod_{j=1}^{k} \frac{\alpha_j}{\beta_{j+1}}, \tag{4}$$

where α_j and β_{j+1} are the normalization coefficients from (3). It was also shown in [5] that if A is a discretized smoothing operator, then the factor ρ_k^{-1} has to grow (on average) until it reaches the point where the noise is revealed in the maximal way. This is illustrated in Figures 1 and 2.

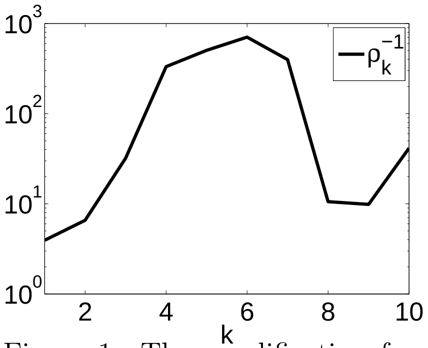


Figure 1: The amplification factor ρ_k^{-1} for problem `shaw(400)` from [6] with relative noise level 10^{-3}.

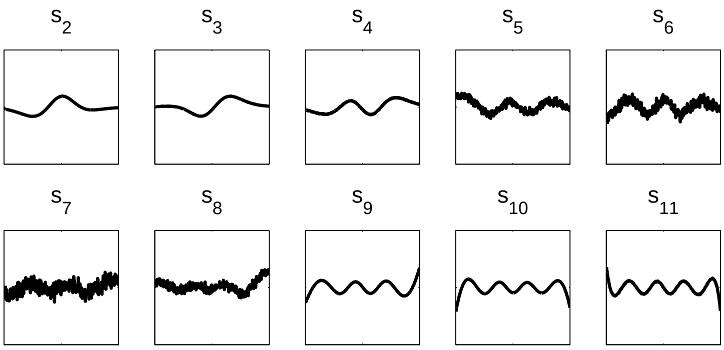


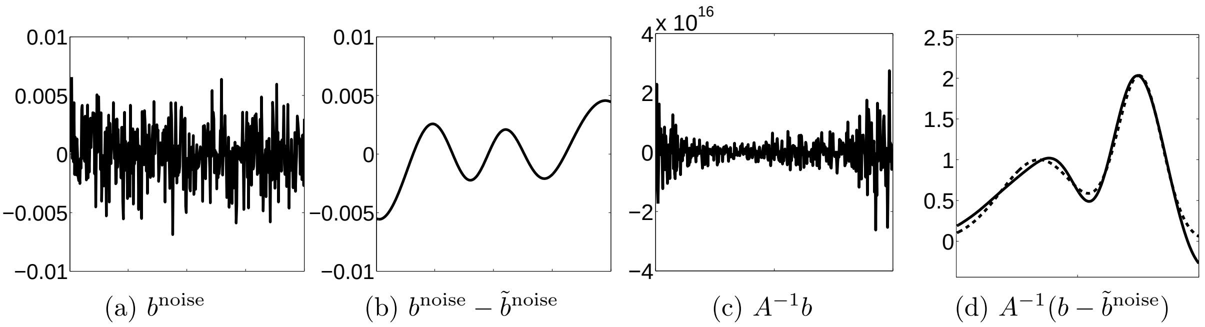
Figure 2: Corresponding left bidiagonalization vectors $s_k, k = 2, \ldots, 11$.

We see that for ρ_k^{-1} maximal, the bidiagonalization vector s_{k+1} may be fully dominated by the noise. This observation forms the basis of the proposed method for finding an estimate $\tilde{b}^{\text{noise}}$.

Let $\hat{k}+1$, where $\hat{k} \equiv \operatorname*{argmax}_k \rho_k^{-1}$, be the iteration of maximal noise revealing (in our example presented above, $\hat{k}+1 = 7$). Then, one may approximate the noise vector by the (properly scaled) left bidiagonalization vector $s_{\hat{k}+1}$. In [7] it was shown that the resulting right-hand side $b - \tilde{b}^{\text{noise}}$ lies in the span of smooth vectors (the troublesome high-frequencies coming from the noise are subtracted) and therefore the method has a regularization effect, as illustrated in Figure 3.

Despite being computationally undemanding, this method is, as shown in [7], competitive with standard methods for solving inverse problems such as truncated SVD or Tikhonov [4]. The method still needs to be tested on real-world examples and it has to be investigated, how to solve system (2) efficiently, or whether rounding errors and consecutive loss of orthogonality may harm the method significantly.

Acknowledgment: This work has been supported by GAUK Grant 695612 and GACR Grant 201/13-06684S. Marie Kubínová is supported by the city of Ostrava scholarship.



(a) b^{noise} (b) $b^{\text{noise}} - \tilde{b}^{\text{noise}}$ (c) $A^{-1}b$ (d) $A^{-1}(b - \tilde{b}^{\text{noise}})$

Figure 3: Regularizing effect of the proposed method – problem from Figure 1. Left to right: (a) original noise, (b) noise with reduced high-frequency part, (c) naive solution, (d) inverse operator applied to thr new right-hand side together with exact solution x^{exact} (dashed line).

References

[1] G. H. Golub, W. Kahan: *Calculating the singular values and the pseudo-inverse of a matrix.* SIAM J. Numer. Anal. Ser. B, 2, 205–224, 1965.

[2] M. E. Kilmer, D. P. O'Leary: *Choosing regularization parameters in iterative methods for ill-posed problems.* SIAM J. Matrix Anal. Appl., 22, 1204–1221, 2001.

[3] Å. Björck: *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations.* BIT Numerical Mathematics, 28, 659–670, 1988.

[4] P. C. Hansen: *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion.* SIAM, 2011.

[5] I. Hnětynková, M. Plešinger, Z. Strakoš: *The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data.* BIT Numerical Mathematics, 49, 669–696, 2009.

[6] P. C. Hansen: *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems.* Numerical Algorithms, 6, 1–35, 1994.

[7] M. Michenková: *Regularization techniques based on the least squares method.* Master's Thesis, Charles University in Prague, 2013.

Simultaneous transport of heat, moisture and salt in porous materials

J. Kruis

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

1 Introduction

Transport of heat, moisture and various species in civil engineering materials is studied more frequently. There are two reasons for this trend. First, durability and reliability of structures are investigated and especially salt transport plays very important role. Second, mechanical behaviour of structure is usually significantly influenced by temperature and moisture distribution. Chloride ions cause corrosion of reinforcement in concrete.

This contribution describes three models of transport processes and shows systems of partial differential equations which have to be solved. After spatial discretization by the finite element method, nonsymmetric systems of ordinary differential equations are obtained and they are solved by the generalized trapezoidal rule. The resulting system of algebraic equations is also nonsymmetric. It is solved by ***LU*** factorization or by GMRES method.

2 Selected material models

There are many material models describing moisture, heat and moisture, moisture and salt and heat, moisture and salt transport in porous material. For detailed explanation, see reference [1].

In 1995, Künzel proposed in [2] a model for coupled heat and moisture transport based on the following balance equations

$$\frac{\partial \rho_v}{\partial \varphi}\frac{\partial \varphi}{\partial t} = \mathrm{div}\left((D_\phi + \delta_p p_s)\mathrm{grad}\varphi + \delta_p \varphi \frac{\mathrm{d}p_s}{\mathrm{d}T}\mathrm{grad}T \right) \tag{1}$$

$$\frac{\partial H}{\partial T}\frac{\partial T}{\partial t} = \mathrm{div}\left((\lambda + L_v \delta_p \varphi \frac{\mathrm{d}p_s}{\mathrm{d}T})\mathrm{grad}T + L_v \delta_p p_s \mathrm{grad}\varphi \right) \tag{2}$$

where ρ_v is the partial moisture density, φ is the relative humidity, t (s) is the time, D_ϕ is the liquid water transport coefficient, δ_p (kg/m/s/Pa) is the water vapour permeability, p_s (Pa) is the partial pressure of saturated water vapour in the air, T (K) is the temperature, H (J/m^3) is the total enthalpy, λ (W/m/K) is the thermal conductivity, L_v (J/kg) is the latent heat of evaporation of water. This model is very popular in building physics.

Suwito, Cai and Xi published in [3] governing equations for the coupled problem of chloride ions and moisture diffusion

$$\frac{\partial C_t}{\partial C_f}\frac{\partial C_f}{\partial t} = \nabla(D_{Cl}\nabla C_f + \varepsilon D_\varphi \nabla \varphi) \tag{3}$$

$$\frac{\partial w}{\partial \varphi}\frac{\partial \varphi}{\partial t} = \nabla(\delta D_{Cl}\nabla C_f + D_\varphi \nabla \varphi) \tag{4}$$

where C_t is the total chloride concentration, C_f is the free chloride concentration, D_{Cl} (m^2/s) is the chloride diffusivity, D_φ is the humidity diffusivity, ε is the humidity gradient coefficient and δ is the chloride gradient coefficient.

Černý and coworkers published in [4] coupled salt-moisture-heat transport model which leads to the balance equation for the moisture

$$\left(\varrho_w + \frac{M}{RT}p_{sat}(\pi - w)\frac{\mathrm{d}\varphi}{\mathrm{d}w} - \frac{M}{RT}p_{sat}\varphi\right)\frac{\partial w}{\partial t} = \tag{5}$$

$$= \mathrm{div}\left(\left(\varrho_w\kappa + \varrho_s\delta_p\frac{\mathrm{d}\varphi}{\mathrm{d}w}\right)\nabla w\right) + \mathrm{div}\left(\delta_p\varphi\frac{\mathrm{d}p_s}{\mathrm{d}T}\nabla T\right) \tag{6}$$

balance equations for the salt

$$C_f H(C_{f,sat} - C_f)\frac{\partial w}{\partial t} + \left(wH(C_{f,sat} - C_f) + \frac{\partial C_b}{\partial C_f}\right)\frac{\partial C_f}{\partial t} + \frac{\partial C_c}{\partial t} = \tag{7}$$

$$= \mathrm{div}(wD\nabla C_f) + \mathrm{div}(C_f\kappa\nabla w) \tag{8}$$

$$\frac{\partial C_c}{\partial t} = H(C_f - C_{f,sat})\frac{\partial}{\partial t}(w(C_f - C_{f,sat})) \tag{9}$$

and balance equation for the heat

$$\frac{\partial H}{\partial T}\frac{\partial T}{\partial t} = \mathrm{div}\left(L_v\delta_p p_{sat}\frac{\mathrm{d}\varphi}{\mathrm{d}w}\nabla w\right) + \mathrm{div}\left(\left(\lambda + L_v\delta_p\frac{\mathrm{d}p_s}{\mathrm{d}T}\right)\nabla T\right) \tag{10}$$

where C_f the concentration of free salts in water (kg/m^3 of solution), C_b the concentration of bonded salts in the whole porous body (kg/m^3 of sample), $C_{f,sat}$ the saturated free salt concentration (kg/m^3 of solution), C_c the amount of crystallized salt (kg/m^3 of sample), D the salt diffusion coefficient (m^2/s), w the volumetric moisture content (m^3/m^3), κ the moisture diffusivity (m^2/s), $H(x)$ the Heaviside step unit function ($H(x < 0) = 0, H(x \geq 0) = 1$), δ the water vapour diffusion permeability (s), p_v the partial pressure of water vapour (Pa), ϱ_w the density of water (kg/m^3), L_v the latent heat of evaporation of water (J/kg), λ the thermal conductivity (W/m/K), ϱ the bulk density (kg/m^3), c the specific heat capacity (J/kg/K), T the temperature (K).

The balance equations (5–10) can be rewritten in the following form

$$\begin{aligned}
H_{ww}\frac{\partial w}{\partial t} &= \mathrm{div}(D_{ww}\boldsymbol{g}_w) + \mathrm{div}(D_{wT}\boldsymbol{g}_T) && (11)\\
H_{fw}\frac{\partial w}{\partial t} + H_{ff}\frac{\partial C_f}{\partial t} + H_{fc}\frac{\partial C_c}{\partial t} &= \mathrm{div}(D_{fw}\boldsymbol{g}_w) + \mathrm{div}(D_{ff}\boldsymbol{g}_f) && (12)\\
H_{cw}\frac{\partial w}{\partial t} + H_{cf}\frac{\partial C_f}{\partial t} + H_{cc}\frac{\partial C_c}{\partial t} &= 0 && (13)\\
H_{TT}\frac{\partial T}{\partial t} &= \mathrm{div}(D_{Tw}\boldsymbol{g}_w) + \mathrm{div}(D_{TT}\boldsymbol{g}_T) && (14)
\end{aligned}$$

One of the most difficult problem connected with coupled transports is definition of boundary conditions. Dirichlet boundary conditions (prescribed values) are barely available. Neumann boundary conditions (prescribed flux densities) are also usually not known. The most suitable boundary conditions are of the Newton type (transmission boundary conditions) but it can be very difficult to obtain appropriate transmission coefficients.

3 Discretization

Discretization of the balance equations is demonstrated on the model proposed by Černý in [4]. The unknown variables (the volumetric moisture content, the concentration of free salts in water, the amount of crystallized salt, the temperature) are discretized in the form

$$\begin{aligned} w &= \boldsymbol{N}_w \boldsymbol{d}_w && (15)\\ C_f &= \boldsymbol{N}_f \boldsymbol{d}_f && (16)\\ C_c &= \boldsymbol{N}_c \boldsymbol{d}_c && (17)\\ T &= \boldsymbol{N}_T \boldsymbol{d}_T && (18) \end{aligned}$$

where $\boldsymbol{N}_w, \boldsymbol{N}_f, \boldsymbol{N}_c, \boldsymbol{N}_T$ denotes the matrices of shape functions and $\boldsymbol{d}_w, \boldsymbol{d}_f, \boldsymbol{d}_c, \boldsymbol{d}_T$ are vectors of nodal values. The test functions are in the form

$$\begin{aligned} \eta_w &= \boldsymbol{N}_w \boldsymbol{b}_w && (19)\\ \eta_f &= \boldsymbol{N}_f \boldsymbol{b}_f && (20)\\ \eta_c &= \boldsymbol{N}_c \boldsymbol{b}_c && (21)\\ \eta_T &= \boldsymbol{N}_T \boldsymbol{b}_T && (22) \end{aligned}$$

and the gradients of unknown variables are expressed in the form

$$\begin{aligned} \boldsymbol{g}_w &= \boldsymbol{B}_w \boldsymbol{d}_w && (23)\\ \boldsymbol{g}_f &= \boldsymbol{B}_f \boldsymbol{d}_f && (24)\\ \boldsymbol{g}_c &= \boldsymbol{B}_c \boldsymbol{d}_c && (25)\\ \boldsymbol{g}_T &= \boldsymbol{B}_T \boldsymbol{d}_T && (26)\\ & && (27) \end{aligned}$$

where $\boldsymbol{B}_w, \boldsymbol{B}_f, \boldsymbol{B}_c, \boldsymbol{B}_T$ are matrices of partial derivatives of the shape functions. The shape functions are usually linear functions.

The resulting system of ordinary differential equations has the form

$$\begin{pmatrix} \boldsymbol{C}_{ww} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{C}_{fw} & \boldsymbol{C}_{ff} & \boldsymbol{C}_{fc} & \boldsymbol{0}\\ \boldsymbol{C}_{cw} & \boldsymbol{C}_{cf} & \boldsymbol{C}_{cc} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}_{TT} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{d}}_w\\ \dot{\boldsymbol{d}}_f\\ \dot{\boldsymbol{d}}_c\\ \dot{\boldsymbol{d}}_T \end{pmatrix} + \begin{pmatrix} \boldsymbol{K}_{ww} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{wT}\\ \boldsymbol{K}_{fw} & \boldsymbol{K}_{ff} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{K}_{Tw} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{TT} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_w\\ \boldsymbol{d}_f\\ \boldsymbol{d}_c\\ \boldsymbol{d}_T \end{pmatrix} = \begin{pmatrix} \boldsymbol{0}\\ \boldsymbol{0}\\ \boldsymbol{0}\\ \boldsymbol{0} \end{pmatrix} \quad (28)$$

Clearly, the system is nonsymmetric and nonlinear.

Time integration is based on the generalized trapezoidal rule [5] in the form

$$\boldsymbol{d}_{i+1} = \boldsymbol{d}_i + \Delta t \boldsymbol{v}_{i+\alpha} \ , \quad (29)$$

where the vector $\boldsymbol{v}_{i+\alpha}$ has the form

$$\boldsymbol{v}_{i+\alpha} = (1-\alpha)\boldsymbol{v}_i + \alpha \boldsymbol{v}_{i+1} \ . \quad (30)$$

The vector $\boldsymbol{v}$ contains time derivatives of unknown nodal variables, i.e. time derivatives of the vector $\boldsymbol{d}$. Substitution of expressions (29) and (30) to the system of balance equations results in well known form

$$\left(\boldsymbol{C} + \Delta t \alpha \boldsymbol{K}\right) \boldsymbol{v}_{i+1} = \boldsymbol{f}_{i+1} - \boldsymbol{K}\left(\boldsymbol{d}_i + \Delta t (1-\alpha) \boldsymbol{v}_i\right) \ . \quad (31)$$

The nonsymmetric and nonlinear system of equations (31) has to be solved within each time step.

4 Conclusion

Solution of transport processes in porous materials is generally very complicated task. Coupled transport processes are leading to nonsymmetric systems of algebraic equations and there are still questions about their solvability. This contribution summarizes three models of coupled transport processes.

Acknowledgement: Financial support for this work was provided by project number 13-18652S of Czech Science Foundation. The financial support is gratefully acknowledged.

References

[1] R. Černý, P. Rovnaníková: *Transport processes in concrete.* Spon Press, London and New York, 2002.

[2] H. M. Künzel: *Simultaneous heat and moisture transport in building components.* Technical Report, Fraunhofer Institute of Building Physics, Fraunhofer IRB Verlag Stuttgart, 1995.

[3] Suwito, X. C. Cai, Y. Xi: *Parallel finite element method for coupled chloride moisture diffusion in concrete.* International Journal of Numerical Analysis and Modeling, 3, (4), 481–503, 2006.

[4] V. Kočí, J. Maděra, R. Černý: *Deterministic physical and mathematical models of coupled heat, moisture and salt transport in multi-layered systems of building materials.* In 11th International Conference of Numerical Analysis and Applied Mathematics 2013. New York: American Institute of Physics, 960–963, 2013.

[5] T. J. R. Hughes: *The finite element method. Linear static and dynamic finite element analysis.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1987.

Optimization of parameters in SDFEM method for different spaces of parameter τ

P. Lukáš

Charles University in Prague, Faculty of Mathematics and Physics

The talk is devoted to the numerical solution of the scalar convection–diffusion equation. We present new results of an adaptive technique in finite element method based on minimizing a functional called error indicator $I_h : W_h \to \mathbb{R}$. The simplest form of such an indicator is

$$I_h(w_h) = \sum_{K\in\mathcal{T}_h, \overline{K}\cap\partial\Omega=\emptyset} h_K^2 \;\| -\varepsilon\Delta w_h + \boldsymbol{b}\cdot\nabla w_h + c w_h - f\|_{0,K}^2 \qquad \forall w_h \in W_h, \tag{1}$$

where we have used the notation from the article of V. John, P. Knobloch, S. B. Savescu [1]. It is possible to enrich this indicator by other terms, which favour less smeared solution to diffuse one. One example of such a term is $\|\phi(|\boldsymbol{b}^{\perp}\cdot\nabla w_h|)\|_{0,1,K}$, where ϕ is a function like square root. The suitability of added terms depends on the problem we solve.

The parameter we are changing in the optimization process is currently the parameter τ from SUPG (SDFEM) method. We use more different finite element spaces (space of piecewise constant functions, piecewise linear continuous functions, and piecewise linear discontinuous functions) for the parameter τ. The talk is based on the article of V. John, P. Knobloch, S. B. Savescu [1].

References

[1] V. John, P. Knobloch, S. B. Savescu: *A posteriori optimization of parameters in stabilized methods for convection-diffusion problems – Part I.* In: Computer Methods in Applied Mechanics and Engineering, 200, (41–44), 1 October 2011, 2916–2929.

Asymptotic expansion for convection–diffusion problems

J. Lamač[1], *J. Hozman*[2]

[1]Faculty of Mathematics and Physics, Charles University in Prague
[2]Technical University of Liberec

1 Introduction

While solving singularly perturbed problems, such as convection-diffusion equation or convection-diffusion-reaction equation, we would like to have some test solution of the respective differential equation (equipped with some simple boundary data) which can confirm or disprove our analysis or methods. This solution can be either exact or asymptotically exact. The same demand can we also have while construction anisotropic and adaptively refined meshes.

In this context, finding the asymptotically exact solution of the respective differential equation is more convenient. Although it seems that we loose the accuracy of the solution it is not the case, since we can choose the accuracy of the solution ourselves. The construction of the asymptotically exact solutions for differential equations – the method of matched asymptotic expansions – is well described for one-dimensional cases and several two-dimensional cases, see e.g. [1, 4] and the references cited therein. However, for multidimensional cases the construction of the asymptotic expansions of the solutions of partial differential equations is more complicated and in fact treated mostly on simple domains – squares and rectangles in 2D. And the analysis of the singularly perturbed problems is performed on these rectangular domains, as well. Therefore, the main goal of this paper is to extend the type of these domains to another convex polygons and enable the generalization of the above mentioned analysis of these problems.

2 Model equation and reduced problem

The model equation for our purposes will be a scalar convection-diffusion equation

$$Lu := -\varepsilon\Delta u(x,y) + \boldsymbol{b}^T(x,y)\nabla u(x,y) \;=\; f(x,y) \quad \text{in} \quad \Omega \subset \mathbb{R}^2, \tag{1}$$

$$u(x,y) \;=\; 0 \quad \text{on} \quad \partial\Omega, \tag{2}$$

where Ω is a convex polygonal domain with boundary $\partial\Omega$ satisfying

$$\overline{\partial\Omega} = \overline{\Gamma}_+ \cup \overline{\Gamma}_0 \cup \overline{\Gamma}_- \quad \text{and} \quad \overline{\Gamma}_+ \cap \overline{\Gamma}_0 = \overline{\Gamma}_0 \cap \overline{\Gamma}_- = \overline{\Gamma}_- \cap \overline{\Gamma}_+ = \emptyset \tag{3}$$

with $\overline{\Gamma}_+$, $\overline{\Gamma}_0$ and $\overline{\Gamma}_-$ defined as follows: $\Gamma_+ = \{(x,y) \in \partial\Omega, \boldsymbol{b}^T(x,y)\boldsymbol{n}(x,y) > 0\}$, $\Gamma_0 = \{(x,y) \in \partial\Omega, \boldsymbol{b}^T(x,y)\boldsymbol{n}(x,y) = 0\}$ and $\Gamma_- = \{(x,y) \in \partial\Omega, \boldsymbol{b}^T(x,y)\boldsymbol{n}(x,y) < 0\}$. Here $\boldsymbol{n}(x,y)$ denotes a unit normal vector at $(x,y) \in \partial\Omega$ orthogonal to the boundary $\partial\Omega$.

Since we are not interested in solving the equation (1) for general data but in finding some test solution for given domain, we can confine ourselves to sufficiently smooth data, namely $\boldsymbol{b} \in C^1(\overline{\Omega})^2$ and $f \in L^2(\overline{\Omega})$. In what follows we shall also consider that the vector $\boldsymbol{b}$ possesses the Taylor expansion in $\overline{\Omega}$, namely in the neighbourhood of $\partial\Omega$.

As $\varepsilon \to 0+$, the equation (1) becomes singularly perturbed and near the boundary Γ_+ it is usually difficult to compute the solution numerically. Thus we would like to determine the asymptotic

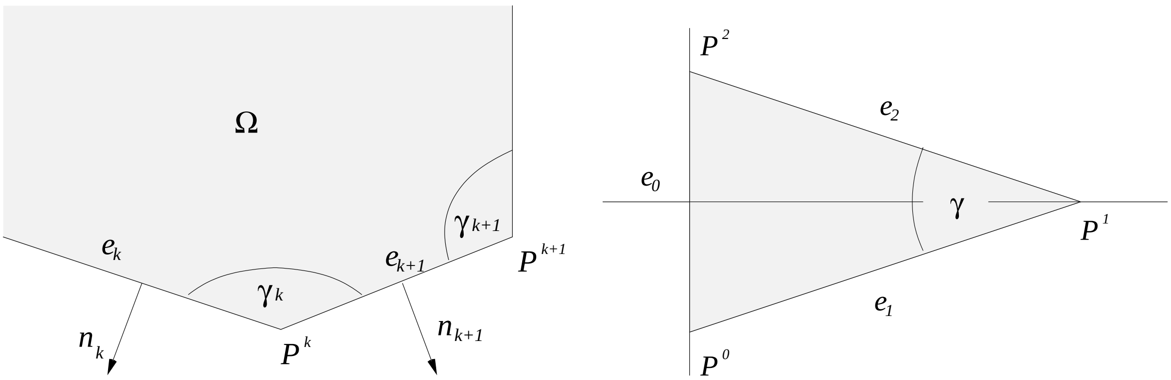


Figure 1: Part of the general convex domain Ω (left) and the simple triangular domain considered in numerical experiment (right).

expansion of the solution to the equation (1) near the boundary Γ_+. At first we formally set $\varepsilon = 0$ in the equation (1) and obtain the so-called *reduced problem*

$$\begin{aligned} \boldsymbol{b}^T(x,y)\nabla u_0(x,y) &= f(x,y) \quad \text{in} \quad \Omega \subset \mathbb{R}^2, && (4)\\ u_0(x,y) &= 0 \quad \text{on} \quad \Gamma_-, && (5) \end{aligned}$$

where we have to consider only the boundary condition on Γ_- due to the cancellation law (see [4] for details). The problem (4)–(5) is the hyperbolic problem and we assume that the solution of this problem is known, more specifically, we expect only the problems with the (analytically) computable reduced solution u_0. Some basic results on existence, uniqueness and regularity of the solution of (4)–(5) can be found in [2]. The reduced solution u_0 is, in fact, the first term of the so-called *global (or regular) expansion* of the solution u, which is a good approximation of u away from the layers. We call the function $E_g^m u$ the m-th order global expansion of the function u when $E_g^m u = \sum_{j=0}^m \varepsilon^j u_j$, where u_0 is the reduced solution and u_j, $j \in \{1, 2, \dots, m\}$ satisfy

$$\begin{aligned} \boldsymbol{b}^T(x,y)\nabla u_j(x,y) &= \Delta u_{j-1}(x,y) \quad \text{in} \quad \Omega \subset \mathbb{R}^2 && (6)\\ u_j(x,y) &= 0 \quad \text{on} \quad \Gamma_-. && (7) \end{aligned}$$

This definition immediately implies that $L(u - E_g^m u) = \varepsilon^{m+1}\Delta u_m$ in Ω, $u - E_g^m u = 0$ on Γ_- and $u - E_g^m u = -E_g^m u$ on $\Gamma_+ \cup \Gamma_0$. Due to the last property, considering $\varepsilon \ll \|E_g^m u\|_{\infty,\Omega}$, the comparison principle (or the maximum principle) yields only $\|u - E_g^m u\|_{\infty,\Omega} \le \|E_g^m u\|_{\infty,\Omega}$. This is the reason why the local correction terms must be introduced.

3 Results on exponential layers

In order to construct the local correction terms, new coordinates in the neighbourhood of Γ_+ must be introduced. For this purpose let us assume that there are only two vertices $\{P^0, P^H\} = \overline{\Gamma}_- \cap \overline{\Gamma}_+$ and consider $\overline{\Gamma}_+ = \cup_{k=1}^H e_k$, where e_k are the edges of Γ_+. Then $P^0 \in e_1$, $P^H \in e_H$ and the remaining vertices of Γ_+ satisfy $P^k = e_k \cap e_{k+1}, k = 1, \dots, H-1$.

The transformation of coordinates Ψ_k corresponding to the edge e_k, $k = 1, 2, \dots, H$, is now defined as $\Psi_k : (x,y) \to (\xi_k, \eta_k)$, where

$$\begin{aligned} \xi_k(x,y) &= (P_y^{k-1} - y)\cos\alpha_k - (P_x^{k-1} - x)\sin\alpha_k, && (8)\\ \eta_k(x,y) &= (P_x^{k-1} - x)\cos\alpha_k + (P_y^{k-1} - y)\sin\alpha_k. && (9) \end{aligned}$$

Here $P^{k-1} = [P_x^{k-1}, P_y^{k-1}]$ and $\boldsymbol{n}_k = (-\sin\alpha_k, \cos\alpha_k)^T$ is the normal vector, orthogonal to the edge e_k, see Figure 1 (left). Then let us denote $d_k = \eta_k(P^k)$ and due to the convexity of Ω, we may for simplicity assume that the domain Ω is oriented in such a way that $\alpha_k \in [0, 2\pi)$ and $\alpha_k < \alpha_{k+1}$ for all $k = 1, 2, \dots, H-1$. This notation also implies that the angle corresponding to the vertex P^k is equal to $\gamma_k = \pi + \alpha_k - \alpha_{k+1}$.

In what follows we recall result from [3], i.e. assume that all characteristics through points of $\overline{\Omega}$ leave $\overline{\Omega}$ at points of Γ_+ in finite time and $\Gamma_0 = \emptyset$, then the following estimate holds for sufficiently small ε,

$$|u(x,y) - u_{as}^m(x,y)| \le C\varepsilon^{m+1} \quad \text{in} \quad \overline{\Omega}, \tag{10}$$

where constant C is independent of Ω and ε, and u_{as}^m is the *asymptotic expansion* of the m-th order defined as

$$u_{as}^m(x,y) = \sum_{n=0}^{m} \varepsilon^n \left\{ u_n(x,y) + \sum_{k=1}^{H} V_n^k\left(\frac{\xi_k(x,y)}{\varepsilon}, \eta_k(x,y)\right) + \sum_{k=1}^{H-1} Z_n^k\left(\frac{\xi_k(x,y)}{\varepsilon}, \frac{\xi_{k+1}(x,y)}{\varepsilon}\right)\right\} \tag{11}$$

The functions V_n^k and Z_n^k in (11) are solutions of the differential equations independent of ε, e.g. for $n = 0$ we solve

$$-\frac{\partial^2 V_0^k}{\partial \xi_k^2} - \boldsymbol{b}\big(P^{k-1} + \tfrac{\eta_k}{d_k}(P^k - P^{k-1})\big) \cdot \boldsymbol{n}_k \, \frac{\partial V_0^k}{\partial \xi_k} = 0 \quad \text{in} \quad \mathbb{R}^+ \times (0, d_k) \tag{12}$$

$$-\frac{\partial^2 Z_0^k}{\partial \xi_{k-1}^2} + 2\cos\gamma_k \, \frac{\partial^2 Z_0^k}{\partial \xi_{k-1}\partial \xi_k} - \frac{\partial^2 Z_0^k}{\partial \xi_k^2} - \boldsymbol{b}\big(P^k\big)\cdot \boldsymbol{n}_{k-1} \, \frac{\partial Z_0^k}{\partial \xi_{k-1}} - \boldsymbol{b}\big(P^k\big)\cdot \boldsymbol{n}_k \, \frac{\partial Z_0^k}{\partial \xi_k} = 0 \ \text{ in } \ (\mathbb{R}^+)^2 \tag{13}$$

For $n > 0$, the functions V_n^k and Z_n^k are recursively defined from $V_0^k, \dots, V_{n-1}^k$ and $Z_0^k, \dots, Z_{n-1}^k$, the detailed description can be found in [3].

Now we shall numerically verify the theoretical estimate (10) for the first order asymptotic expansion of the solution of the equation (1) with simple data $\boldsymbol{b}^T = (1, 0)$ and $f = 1$ on a triangle with vertices $P^0 = [0, -\tan\frac{\gamma}{2}]$, $P^1 = [1, 0]$ and $P^2 = [0, \tan\frac{\gamma}{2}]$, see Figure 1 (right). Figure 2 (left) shows the particular case of the first order asymptotic expansion u_{as}^0 ($\gamma = \frac{\pi}{4}$ and $\varepsilon = 0.01$). The general form of the function u_{as}^0 for this simple domain is introduced in [3] as

$$\begin{aligned} u_{as}^0(x,y) = {} & u_0(x,y) - u_0\left(\Psi_1^{-1}(0, \eta_1(x,y))\right) \exp\left(\frac{\xi_1(x,y)}{\varepsilon} B_1^1(0, \eta_1(x,y))\right) \\ & - u_0\left(\Psi_2^{-1}(0, \eta_2(x,y))\right) \exp\left(\frac{\xi_2(x,y)}{\varepsilon} B_1^2(0, \eta_2(x,y))\right) \\ & + u_0(P^1)\left(\sum_{j=0}^{r} \exp\left(p_j^r \frac{\xi_1(x,y)}{\varepsilon} + q_j^r \frac{\xi_2(x,y)}{\varepsilon}\right) - \sum_{j=0}^{r-1} \exp\left(p_{j+1}^r \frac{\xi_1(x,y)}{\varepsilon} + q_j^r \frac{\xi_2(x,y)}{\varepsilon}\right)\right), \end{aligned} \tag{14}$$

where $u_0(x,y)$ is the solution of the reduced problem given by (4)–(5) and

$$\xi_1(x,y) = (1-x)\sin\tfrac{\gamma}{2} + y\cos\tfrac{\gamma}{2}, \qquad \xi_2(x,y) = (1-x)\sin\frac{\gamma}{2} - y\cos\frac{\gamma}{2}, \tag{15}$$

$$\eta_1(x,y) = x\cos\tfrac{\gamma}{2} + (y + \tan\tfrac{\gamma}{2})\sin\tfrac{\gamma}{2}, \quad \eta_2(x,y) = (1-x)\cos\frac{\gamma}{2} + y\sin\frac{\gamma}{2}, \tag{16}$$

$$p_j^r = \tfrac{\sin^2((j+1)\gamma)}{\sin^2\gamma}\left(B_1^1 + B_1^2 \tfrac{\sin(j\gamma)}{\sin((j+1)\gamma)}\right), \quad q_j^r = \tfrac{\sin^2((j+1)\gamma)}{\sin^2\gamma}\left(B_1^2 + B_1^1 \tfrac{\sin((j+2)\gamma)}{\sin((j+1)\gamma)}\right), \tag{17}$$

$$B_1^1 = B_1^1(0, d_1) = -\boldsymbol{b}(P^1)\cdot \boldsymbol{n}_1, \qquad B_1^2 = B_1^2(0,0) = -\boldsymbol{b}(P^1)\cdot \boldsymbol{n}_2. \tag{18}$$

Numerical experiments are carried out with the use of discontinuous Galerkin method (see, e.g. [5]) with piecewise linear approximations on uniformly refined meshes having approximately

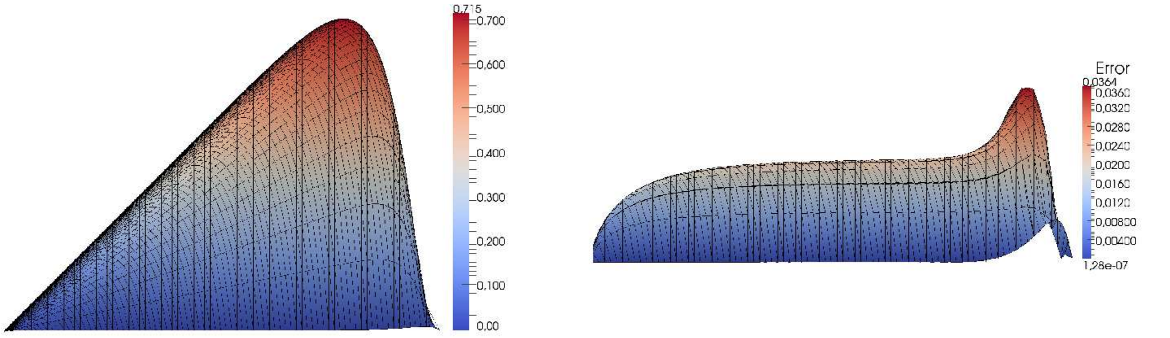


Figure 2: The 3D plots of the first order asymptotic expansion function (left) and the corresponding distribution of error (right) for case $\gamma = \frac{\pi}{4}$ and $\varepsilon = 0.01$.

5000 elements for several different values of γ and ε. The difference between the numerical solution u_h and asymptotic expansion u^0_{as} is depicted in Figure 2 (right). Table 1 records the corresponding errors $u_h - u^0_{as}$ in $L^\infty(\Omega)$-norm together with the experimental order of convergence (EOC) with respect to ε. We observe that $EOC \approx 1$ for all considered angles γ, which is in a good agreement with derived theoretical results of order $O(\varepsilon)$ according to (10).

ε	$\gamma = \frac{\pi}{2}$	$\gamma = \frac{\pi}{3}$	$\gamma = \frac{\pi}{4}$	$\gamma = \frac{\pi}{6}$
0.04	8.0211E-02	9.5164E-02	1.4183E-01	2.6822E-01
0.02	3.3935E-02	4.6841E-02	7.8396E-02	1.4062E-01
0.01	1.6045E-02	2.1667E-02	3.8123E-02	7.7320E-02
0.005	9.0778E-03	1.1417E-02	2.0733E-02	4.2789E-02
EOC	1.051	1.029	0.936	0.881

Table 1: Computational errors in $L^\infty(\Omega)$-norm and experimental orders of convergence for different values of γ and ε.

Acknowledgement: The authors would like to thank M. Hadrava and A. Kosík for their assistance with elaboration of numerical experiments. This work was partly supported by the Grant Agency of the Czech Republic, grant No. P201/11/1304.

References

[1] W. Eckhaus: *Asymptotic analysis of singular pertubations.* North-Holland,Amsterdam, 1979.

[2] H. Goering, A. Felgenhauer, G. Lube, H.-G. Roos, L. Tobiska: *Singularly perturbed differential equations.* Akademie-Verlag, Berlin, 1983.

[3] J. Lamač: *Asymptotic expansion of the solution of the singularly perturbed convection-diffusion equation in the 2D convex polygonal domain.* AIP Conf. Proc. 1558, 383–386, 2013.

[4] H.-G. Roos, M. Stynes, L. Tobiska: *Robust numerical methods for singularly perturbed differential equations.* Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 2008.

[5] B. Riviére: *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation.* Frontiers in Applied Mathematics, SIAM, Philadelphia, 2008.

Stability of suspension bridges in lateral wind

J. Malík

Institute of Geonics, Academy of Sciences of the Czech Republic,
Studentská 1768, 708 00 Ostrava–Poruba

1 Introduction

The collapse of the original Tacoma suspension bridge has been studied in many papers. On 7 November 1940 around 10 a.m. the torsional oscillations appeared on the deck of the original Tacoma bridge after the loosening of one midspan cable band, which resulted in the lateral asymmetry of the construction. It seems that the loosening of the midspan cable band had a significant impact on the behavior of the bridge and in the end it resulted in the collapse.

The model of the central span , depicted in Fig. 1,and the cable system studied in this paper is described by two functions corresponding with vertical and torsional motions of the central span and was formulated in [1]. The cable stays are modeled as a continuum. The model is based on the equilibrium state given by the gravitational forces acting on the whole construction. The two functions mentioned above describe the deflection from the equilibrium state. We analyze the action of lateral wind on the center span. These forces are relatively small comparing to the gravitational forces. The formulation describes the mutual reaction of the center span and the cable system as well as the reaction of the diagonal ties on the midspan cable bands. Three different types of evolution variational problems are formulated and analyzed.

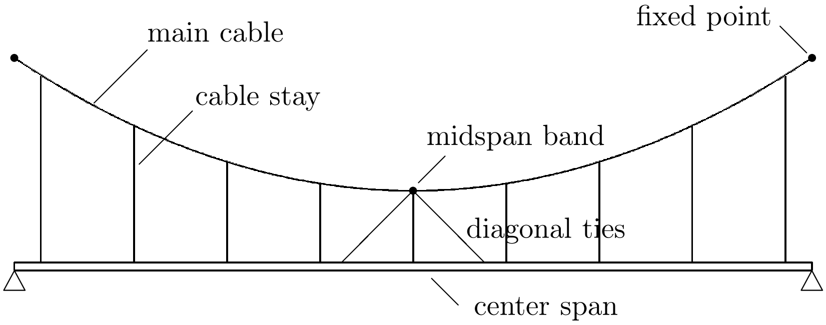


Figure 1: Specification of center span

The equations formulated here describe the deflections from the equilibrium state due to the forces induced by lateral wind. The analysis of the derived equations reveals that the action of lateral wind can cause torsional oscillations if just one midspan cable band loosens.

2 Formulation of problems and main results

The analysis is based on the variational equations derived in [1]. Let us remind the parameters of the deck and the cable system.

- The width of the deck is denoted $2D$.

- The length of the central span is L.
- The sag of the main cables is L_1.
- The mass of the deck per unit length along the span is M_D.
- The mass of the main cable per unit length is M_C.
- The modulus of elasticity of the deck is E_D.
- The moment of inertia of the deck cross section with respect to the horizontal line through its centroid is I_D.
- The polar mass moment of inertia of the deck is I_P.
- The shear modulus of the deck is G_D.
- The torsional constant of the deck is J_D.
- The gravitational acceleration is g.

The formulation of the linearized model is based on the Hamilton principle. The starting point is the equilibrium under gravitational forces. Then we look for a new equilibrium, which is a stationary point of the functional defined below. The deflection of the center span from the original equilibrium is described by functions $u(x,t), \theta(x,t)$, where $u(x,t)$ corresponds to vertical deformations and $\theta(x,t)$ corresponds to torsional deformations of the center span The formulation of the linearized models is based on the following hypotheses formulated in [1]. Let us define the bilinear form

$$a_c(u,v) = \int_{-\frac{L}{2}}^{\frac{L}{2}} H\left(1+\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2\right)\frac{\mathrm{d}u}{\mathrm{d}x}\frac{\mathrm{d}v}{\mathrm{d}x}\,\mathrm{d}x.$$

Then the potential energy of the main cables can be expressed in the form

$$a_c(u,u) + D^2 a_c(\theta,\theta)\,.$$

Let us define another two bilinear forms

$$a_{ver}(u,v) = \int_{-\frac{L}{2}}^{\frac{L}{2}} E_D I_D \frac{\mathrm{d}^2u}{\mathrm{d}x^2}\frac{\mathrm{d}^2v}{\mathrm{d}x^2}\mathrm{d}x\,, \quad a_{tor}(\theta,\varphi) = \int_{-\frac{L}{2}}^{\frac{L}{2}} G_D J_D \frac{\mathrm{d}\theta}{\mathrm{d}x}\frac{\mathrm{d}\varphi}{\mathrm{d}x}\mathrm{d}x\,,$$

which are connected with the bending and torsional deformation energy of the deck. To simplify our equations for the dynamic problems, we define the bilinear forms

$$m_{ver}(u,v) = \int_{-\frac{L}{2}}^{\frac{L}{2}} M_{ver} uv\mathrm{d}x, \quad m_{tor}(\theta,\varphi) = \int_{-\frac{L}{2}}^{\frac{L}{2}} M_{tor}\theta\varphi\mathrm{d}x,$$

where M_{ver}, M_{tor} are functions on $(-L/2, L/2)$ defined by

$$\begin{aligned} M_{ver}(x) =& 2M_C\left(1+\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2\right)^{\frac{1}{2}} + M_D\,,\\ M_{tor}(x) =& 2D^2M_C\left(1+\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2\right)^{\frac{1}{2}} + I_P\,. \end{aligned}$$

The equations for the dynamic problems will be derived from the Hamilton principle. The variational equation reads

$$m_{ver}(\ddot{u},v) + m_{tor}(\ddot{\theta},\varphi) + 2a_c(u,v) + 2D^2 a_c(\theta,\varphi) + a_{ver}(u,v) + a_{tor}(\theta,\varphi) = \int_{-\frac{L}{2}}^{\frac{L}{2}} F_{ver} v \,\mathrm{d}x + \int_{-\frac{L}{2}}^{\frac{L}{2}} F_{tor}\, \varphi \,\mathrm{d}x$$

and holds for all sufficiently smooth functions $v(x), \varphi(x)$ defined on $(-L/2, L/2)$. In our models we assume that the main span is hinged in its end points, so the functions u, θ satisfy the boundary conditions

$$u\,(-L/2,\, t) = u\,(L/2,\, t) = \theta\,(-L/2,\, t) = \theta\,(L/2,\, t) = 0\,.$$

So far we have not consider the fact that the main cables are inextensible and fixed at the end points and fastened at the midspan cable bands. Let us suppose that the deck deforms and the deformation transfers on the main cables via the inextensible suspenders. To simplify our considerations, we define three linear forms

$$h(u) = \int_{-\frac{L}{2}}^{\frac{L}{2}} \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{d}x\,, \quad h_r(u) = \int_{-\frac{L}{2}}^{0} \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{d}x\,, \quad h_l(u) = \int_{0}^{\frac{L}{2}} \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{d}x\,.$$

If both main cables are fixed in their end points, then u and θ satisfy the relations

$$h(u) = h(\theta) = 0\,.$$

In the case both main cables are fixed at the midspan cable bands as well, the following relations

$$h_r(u) = h_r(\theta) = h_l(u) = h_l(\theta) = 0$$

hold. In the end let us study the case, where both main cables are fixed at the end points and only one main cable is fixed at the midspan cable band, then the relations

$$h_r(u - D\theta) = h_l(u - D\theta) = h(u + D\theta) = 0$$

Now we are going to formulate three dynamic problems connected with the way how the main cables are fixed. The first dynamic problem describes oscillations of the center span if the main cables are fixed at the end points. The functions $u(t), \theta(t)$ are a solution to $\mathcal{D}_1$ if these functions satisfy the relations

$$h(u(t)) = h(\theta(t)) = 0$$

for all t, and the variational equation. The variational equation holds for all v, φ which satisfy the relations

$$h(v) = h(\varphi) = 0.$$

The initial conditions are compatible with $\mathcal{D}_1$, which is defined in [1].

The functions $u(t), \theta(t)$ are a solution to the dynamic problem $\mathcal{D}_2$ if these functions satisfy the relations

$$h_r(u(t)) = h_r(\theta(t)) = h_l(u(t)) = h_l(\theta(t)) = 0$$

for all t, the boundary conditions , and the variational equation. The variational equation holds for all v, φ which satisfy the relations

$$h_r(v) = h_r(\varphi) = h_l(v) = h_l(\varphi) = 0$$

and the boundary conditions. The initial conditions are compatible with $\mathcal{D}_2$, which is defined in [1].

The functions $u(t)$, $\theta(t)$ are a solution to the third dynamic problem $\mathcal{D}_3$ if these functions satisfy the relations

$$h_r(u(t) - D\theta(t)) = h_l(u(t) - D\theta(t)) = h(u(t) + D\theta(t)) = 0\,,$$

for all t, the boundary conditions, and the variational equation. The variational equation holds for all v, φ which satisfy the relations

$$h_r(v - D\varphi) = h_l(v - D\varphi) = h(v + D\varphi) = 0$$

and the boundary conditions. The initial conditions are compatible with $\mathcal{D}_3$, which is defined in [1].

The existence and continuous dependence on data is proved in [2].

3 Conclusion

The original Tacoma bridge exhibited relatively small vertical oscillations from the time that it was opened. The bridge was stable with respect to torsional oscillations until one midspan cable band loosened. This led to torsional oscillations which lasted for approximately one hour and then the deck broke. The new evolution variational equations were derived. These equations describe the behavior of the center span and main cables in the three different situations, where the both main cables have the fastened midspan cable bands, only one cable has the fastened midspan cable band, and the main cables have no fastened midspan cable bands. The analysis revealed that the behavior of the center span depends on the direction of lateral wind and vertical and torsional oscillations of the center span are connected if just one midspan cable band loosens.

Acknowledgment: The paper has been done in connection with project Institute of Clean Technologies, areg. no. CZ.1.05/2.1.00/03.0082 supported by Research and Development for Innovations Operational Program financed by Structural Funds of Europe Union and from the means of state budget of the Czech Republic.

References

[1] J. Malík: *Sudden lateral asymmetry and torsional oscillations in the original Tacoma suspension bridge.* Journal of Sound and Vibration, 332, 3772–3789, 2013.

[2] J. Malík: *Torsional asymmetry in suspension bridge systems.* Journal of Mathematical Analysis and Applications.

On three equivalent methods for parameter estimation problem based on spatio-temporal FRAP data

C. Matonoha [1], *Š. Papáček* [2]

[1] Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic

[2] University of South Bohemia in České Budějovice, Faculty of Fisheries and Protection of Waters,
South Bohemian Research Center of Aquaculture and Biodiversity of Hydrocenoses, Institute of Complex Systems, Zámek 136, 373 33 Nové Hrady, Czech Republic

1 Introduction

The FRAP (Fluorescence Recovery After Photobleaching) method is based on measurement of the change of fluorescence intensity in a region of interest (being usually an Euclidian 2D domain) in response to an external stimulus, a short period of high-intensity laser pulse provided by the CLSM.[2] Stimulus, the so-called *bleach*, causes irreversible loss in fluorescence of autofluorescence molecules or fluorescently tagged compounds (e.g. green fluorescence proteins – GFP) in living cells in bleached area without any damage in intracellular structures. After the bleach, the observed recovery in fluorescence presumably reflects the diffusion of fluorescence compounds from the area outside the bleach. Based on spatio-temporal FRAP images, the process is reconstructed using either a closed form model or simulation based model. In the latter case, beside a single diffusion coefficient D, also the sequence $\{D_j\}$ can be estimated as well. Let us underline that FRAP images are usually very noisy, with small signal to noise ratio (SNR), i.e. in order to get reliable results for the sequence $\{D_j\}$, an adequate technique residing in *regularization* is mandatory [1, 5, 6].

2 Inverse Problem Formulation

Assuming the special geometry residing in one-dimensional simplification, getting the unbleached particle concentration y as a function of dimensionless quantities $x := \frac{r}{L}$ (r is a spatial coordinate in physical units, L is a characteristic length), $\tau := \frac{t}{T}$ (t is time, T is a constant with some characteristic value, e.g. the time interval between the initial and the last measurements), and $p := D\frac{T}{L^2}$ (re-scaled diffusion coefficient), we obtain the following dimensionless diffusion equation

$$\frac{\partial y}{\partial \tau} - p\frac{\partial^2 y}{\partial x^2} = 0 \tag{1}$$

with the initial condition and Dirichlet boundary conditions

$$y(x, \tau_0) = f(x), \quad x \in [0, 1], \tag{2}$$

$$y(0, \tau) = g_0(\tau), \quad y(1, \tau) = g_1(\tau), \quad \tau \geq \tau_0. \tag{3}$$

[2] Confocal laser scanning microscopy (CLSM) allows the selection of a thin cross-section of the sample by rejecting the information coming from the out-of-focus planes. However, the small energy level emitted by the fluorophore and the amplification performed by the photon detector introduces a measurement noise.

Spatio-temporal FRAP data

Based on FRAP experiments, we have a 2D dataset in form of a table with $(N + 1)$ rows corresponding to the number of spatial points where the values are measured, and $(m + M + 1)$ columns with m pre-bleach and $M + 1$ post-bleach experimental values forming 1D profiles

$$y_{exp}(x_i, \tau_j), \quad i = 0 \ldots N, \quad j = -m \ldots M.$$

In fact, the process is determined by m columns of pre-bleach data containing the information about the steady state and optical distortion,[3] and $M+1$ columns of post-bleach data containing the information about the transport of unbleached particles (due to the diffusion) through the boundary.

Objective function

We construct an objective function $Y(p)$ representing the disparity between the experimental and simulated time-varying concentration profiles, and then within a suitable method we look for such a value $p \in \mathcal{R}^M$ minimizing Y.

The usual form of an objective function is the sum of squared differences between the experimentally measured and numerically simulated time-varying concentration profiles. Taking separately temporal (sub-index j) and spatial data points (sub-index i), we get:

$$Y(p) = \sum_{j=1}^{M} \sum_{i=0}^{N} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p_j) \right]^2 , \tag{4}$$

where $y_{sim}(x_i, \tau_j, p_j)$ are the simulated values resulting from the solution of problem (1)–(3), and $y_{exp}(x_i, \tau_0),\ i = 0 \ldots N,$ represent the initial condition $f(x)$. The left and right Dirichlet boundary conditions $g_0(\tau)$ and $g_1(\tau)$ are represented by $y_{exp}(0, \tau_j)$ and $y_{exp}(1, \tau_j),\ j = 1 \ldots M,$ respectively.

Ill-posedness

Our problem is ill-posed in the sense that the solution, i.e. the diffusion coefficients $p_1 \ldots p_M$, do not depend continuously on the initial experimental data. This led us to the necessity of using some stabilizing procedure in form of the following regularized cost functions:

$$Y_j(p_j, p_{reg}, \alpha) = \sum_{i=0}^{N} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j, p_j) \right]^2 + \alpha \left(p_j - p_{reg} \right)^2 \tag{5}$$

for $j = 1 \ldots M$, where $\alpha \geq 0$ is a regularization parameter and $p_{reg} \in \mathcal{R}$ is an expected value. Taking $\alpha = 0$, function $Y(p, p_{reg}, \alpha) = \sum_{j=1}^{M} Y_j(p_j, p_{reg}, \alpha)$ turns to (4).

Values $p_j^*(\alpha),\ j = 1 \ldots M,$ are approximate solutions of minimization problems [4]

$$p_j^*(\alpha) = \arg \min_{p_j, p_{reg}} Y_j(p_j, p_{reg}, \alpha). \tag{6}$$

[3] The noise identification can be performed using the pre-bleach data as well.

[4] Minimizing Y with respect to $p > 0$ represents a one-dimensional optimization problem. It was solved using variable metric method implemented in the UFO system [4].

It holds that $\lim_{\alpha\to 0} p_j^*(\alpha) = p_j^*(0)$. For $\alpha \to \infty$ we have that (i) $\|p^*(\alpha) - p_{reg}\|^2 \to 0$, i.e. the estimated parameter variance is diminishing or even $p_j^*(\alpha) \equiv p_{reg}\ \forall j$, and (ii) function values $Y(p^*(\alpha), p_{reg}, \alpha)$ become larger (although there is a *supremum*). The problem of choosing in some sense optimal parameter α^* is discussed in the next section.

3 Tikhonov regularization vs. Least squares with a quadratic constraint regularization

A useful tool to see the relation between the residuum for different values of regularization parameter α, and the norm of a solution or relative standard deviation of the solution or some other measure of variability of the solution, is the so-called *L-curve*. Usually, this parametric plot, in our case with $Y(p^*(\alpha), p_{reg}, 0)$ (without the regularization term) in the abscissa, and $\|p^*(\alpha) - p_{reg}\|^2$ in the ordinate, is L-shaped (hence the name). In the upper left part we have small values of α (under-smoothing, the solution is corrupted by the noise in data) and the lower right part corresponds to the over-smoothing (the regularization term dominates for large α). Let see Figure 1 for the just introduced plot corresponding to our FRAP problem with the synthetic noisy data.

Tikhonov regularization

Tikhonov regularization [6] is based on adding a regularization term in (4) getting (5) and solving the problem

$$p^*(\alpha) = \arg\min_{p, p_{reg}} Y(p, p_{reg}, \alpha), \quad \text{st.} \quad p \geq 0. \tag{7}$$

The question is how to choose a "right" (in some sense optimal) parameter α^*. In [2], it is preferred the so-called L-curve criterion consisting in finding the point of maximal curvature on the L-curve. This point with corresponding solution $p^*(\alpha^*)$ is called *L-curve optimal*. However, in most cases this point is hard to determine.

Constraint based on determination of estimated parameter variance

To avoid the above mentioned situation of non-unambiguous choice of the parameter α^*, another approach, consisting in prescribing the value of $\|p^* - p_{reg}\|^2$ in advance, can be used. As the norm of a solution $p^*(\alpha)$ becomes more and more smaller for $\alpha \to \infty$, assume that we have prescribed the variance in the solution with some value ξ. If we denote $Y(p) = \sum_{j=1}^{M} Y_j(p_j, p_{reg}, 0)$, then according to Hansen [2], we can solve the following equivalent optimization problem with a quadratic constraint

$$p^*(\xi) = \arg\min_{p} Y(p), \quad \text{st.} \quad \|p - p_{\text{reg}}\|^2 \leq \xi, \quad p \geq 0. \tag{8}$$

Measurement noise based constraint

Suppose that we either know or can estimate the noise in input data. If we denote $y_{exp}^{\delta}(x_i, \tau_j)$ as real noisy data and $y_{exp}(x_i, \tau_j)$ as ideal data that would be measured without the noise, then

$$\sum_{j=1}^{M} \sum_{i=0}^{N} \left[y_{exp}^{\delta}(x_i, \tau_j) - y_{exp}(x_i, \tau_j) \right]^2 \leq \delta$$

where δ specifies the noise level (for the normally distributed non-correlated additive noise with the variance σ_0^2, we have $\delta \approx M\ N\ \sigma_0^2$). This leads to another possibility to determine (6). As Hansen [2] claims, the following optimization problem is again equivalent to the previous ones

$$p^*(\delta) = \arg\min_p \|p - p_{\text{reg}}\|^2, \quad \text{st.} \quad Y(p) \leq \delta, \quad p \geq 0. \tag{9}$$

By theory, L-curve is continuous and decreasing which means that both constraints in (8) and (9) are attained on the boundary. Thus each value δ (specifying the noise level) corresponds the value $L(\delta) = \xi$ on the L-curve so that

$$Y(p) = \delta \quad \Leftrightarrow \quad \|p - p_{\text{reg}}\|^2 = L(\delta).$$

Moreover, this point also corresponds to a certain Tikhonov regularization parameter α, i.e. $\alpha \equiv [\delta, L(\delta)]$. The respective α^* for a given noise δ^* is called *noise optimal.* Then the solution $p^*(\delta^*)$ corresponds to the solution found by applying the discrepancy principle [3].

The practical confirmation of the Hansen's conjecture about the equivalency of above three methods is shown in Figure 1.

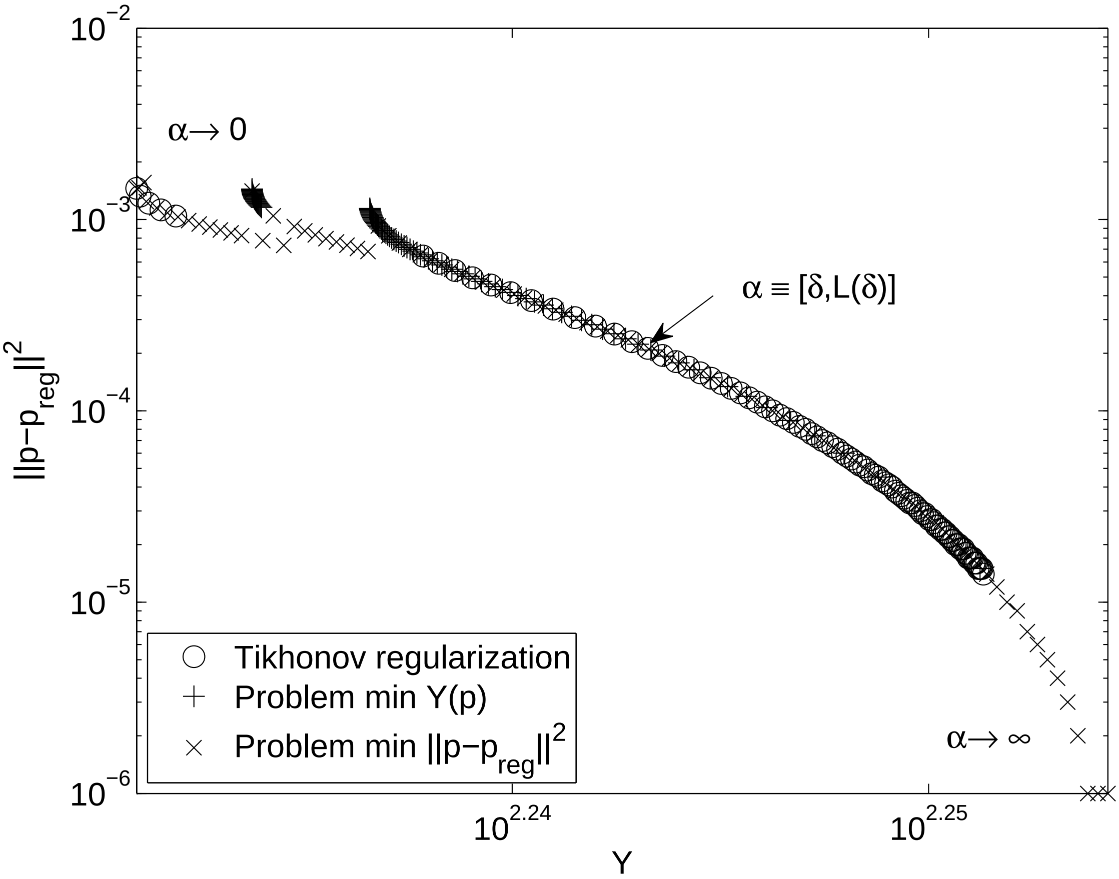


Figure 1: L-curves for three different regularization methods, i.e. the log-log-plot of the solution norm versus the residual norm, with α as the parameter.

4 Conclusions

We have presented three methods for the solution of the apparently simple parameter estimation problem. However, due to the noisy data from the spatio-temporal FRAP measurement, we have to look for a stabile numerical process. The most usual method is the Tikhonov regularization. Nevertheless, in our specific problem we had to deal with the complicated problem of determining the optimal regularization parameter α. Fortunately, there are two equivalent methods based on least squares with a quadratic constraint regularization enabling the application of the UFO system [4]. While the first method constrains the estimated parameter variance, the second is based on the measurement noise determination and constraining the residuum (proportional to the noise level). This latter approach naturally takes into account the noise level in the data and corresponds to the discrepancy principle as well. Furthermore, all three approaches were implemented into our software CA-FRAPwith satisfactory results on synthetic data.

Acknowledgement: This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807), and by the project “Jihočeské výzkumné centrum akvakultury a biodiverzity hydrocenóz” (CENAKVA CZ.1.05/2.1.00/01.0024), OP VaVpI. The results of the project LO1205 were obtained with a financial support from the MEYS of the CR under the NPU I program.

References

[1] H. W. Engl, M. Hanke, A. Neubauer: *Regularization of inverse problems.* Kluwer, Dordrecht, 1996.

[2] P. C. Hansen: *Rank-deficient and discrete ill-posed problems.* SIAM, 1998.

[3] V. A. Morozov: *On the solution of functional equations by the method of regularization.* Soviet Math. Dokl., 7, 414–417, 1966.

[4] L. Lukšan, M. Tuma, J. Vlček, N. Ramešová, M. Šiška, J. Hartman, C. Matonoha: *UFO 2011 – Interactive system for universal functional optimization.* Technical Report V-1151, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague 2011 (`http://www.cs.cas.cz/luksan/ufo.html`).

[5] Š. Papáček, R. Kaňa, C. Matonoha: *Estimation of diffusivity of phycobilisomes on thylakoid membrane based on spatio-temporal FRAP images.* Mathematical and Computer Modelling, 57 1907–1912, 2013.

[6] A. N. Tychonoff, V. Y. Arsenin: *Solution of Ill-posed problems.* Washington, Winston & Sons, 1977.

A posteriori algebraic error estimation in numerical solution of linear diffusion PDEs

J. Papež[1], *M. Vohralík*[2]

[1]Institute of Computer Science, AS CR, Prague
[1]Charles University in Prague
[2]INRIA, Paris-Rocquencourt

1 Introduction

The paper [1], see also the references therein, proposes an adaptive method with a posteriori stopping criteria for numerical solution of nonlinear partial differential equations of diffusion type. The main idea in [1] is to distinguish different components of the error, namely the discretization, the linearization, and the algebraic ones, and to design stopping criteria based on balancing these error components. The estimates rely on quasi-equilibrated flux reconstructions and yield a general framework which can be applied to various discretization schemes.

In the present contribution we tightly follow [1] and concentrate specifically on estimating the algebraic part of the error. We show that, with an additional assumption on the flux reconstructions, the algebraic error can be bounded using the algebraic a posteriori error estimator. This justifies the distinction of error components presented in [1]. For simplicity we restrict ourselves to a linear model problem discretized using the conforming finite element method. We show that the flux reconstruction given in [1] can be modified such that the newly introduced assumption is satisfied. We believe that an analogous modification is possible also for other discretization schemes, as well as for the nonlinear setting considered in [1].

2 Model problem and discrete setting

Let $\Omega \subset \mathbb{R}^d, d \geq 2$, be a polygonal (polyhedral) domain. We consider the Poisson model problem: find $u : \Omega \to \mathbb{R}$ such that

$$\Delta u = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega, \tag{1}$$

where $f : \Omega \to \mathbb{R}$ is the source term. Assuming $f \in L^2(\Omega)$, the model problem (1) can be casted into the weak form: find $u \in V \equiv H_0^1(\Omega)$ such that

$$(\nabla u, \nabla v) = (f, v) \qquad \forall v \in V, \tag{2}$$

where $H_0^1(\Omega)$ denotes the standard Hilbert space of $L^2(\Omega)$ functions whose weak derivatives are in $L^2(\Omega)$ and with trace vanishing on $\partial\Omega$. Owing to (2), the flux $-\nabla u$ is in the space $\mathbf{H}(\mathrm{div}, \Omega)$ spanned by the functions in $[L^2(\Omega)]^d$ with weak divergences in $L^2(\Omega)$.

Let $\mathcal{T}_h$ be a simplicial mesh of Ω. We suppose that the mesh is conforming in the sense that, for two distinct elements of $\mathcal{T}_h$, their intersection is either an empty set or a common l-dimensional face, $0 \leq l \leq d-1$. We denote a generic element of $\mathcal{T}_h$ by K and its diameter by h_K. We denote by $\mathbb{P}_m(K)$ the space of m-th order polynomial functions on an element K and by $\mathbb{P}_m(\mathcal{T}_h)$ the broken polynomial space spanned by $v_h|_K \in \mathbb{P}_m(K)$ for all $K \in \mathcal{T}_h$. Let

$$V_h \equiv H_0^1(\Omega) \cap \mathbb{P}_m(\mathcal{T}_h) = \left\{ v \in H_0^1(\Omega), v|_K \in \mathbb{P}_m(K) \quad \forall K \in \mathcal{T}_h \right\} \tag{3}$$

be the usual finite element space of continuous, piecewise m-th order polynomial functions, $m \geq 1$. The corresponding discrete formulation of problem (2) reads: find $u_h \in V_h$ such that

$$(\nabla u_h, \nabla v_h) = (f, v_h) \qquad \forall v_h \in V_h\,. \tag{4}$$

Let $\psi_j \in V_h,\ j \in \mathcal{C} \equiv \{1, \dots, \dim(V_h)\}$, denote the usual Lagrange basis of V_h. Employing this basis in (4) gives rise to the system of linear algebraic equations

$$\mathsf{A}\mathsf{U} = \mathsf{F}\,. \tag{5}$$

At the i-th step, $i = 1, 2, \dots$, of an iterative solver applied to the algebraic system (5), we obtain the approximation $\mathsf{U}^i = [\mathsf{U}^i_j]_{j\in\mathcal{C}}$ to the solution U and the algebraic residual vector $\mathsf{R}^i = [\mathsf{R}^i_j]_{j\in\mathcal{C}}$ such that

$$\mathsf{A}\mathsf{U}^i = \mathsf{F} - \mathsf{R}^i\,. \tag{6}$$

Finally, by u_h^i we denote the approximation to the solution u determined by the coefficient vector U^i,

$$u_h^i \equiv \sum_{j\in\mathcal{C}} \mathsf{U}^i_j \psi_j\,. \tag{7}$$

3 Error measure and a posteriori error estimates for total error and for the algebraic error

The (total) error between the exact solution u of the weak formulation (2) and the approximate solution $u_h^i \in V_h$ given by (7) is measured as

$$\|\nabla(u - u_h^i)\| = \sup_{\varphi\in V, \|\nabla\varphi\|=1} \big(\nabla(u - u_h^i), \nabla\varphi\big)\,. \tag{8}$$

The following assumption is the starting point for a posteriori error estimation proposed in [1].

Assumption 3.1 (Quasi-equilibrated flux reconstructions). *There exist vector-valued functions* $\mathbf{t}_h^i \in \mathbf{H}(\mathrm{div}, \Omega)$, $\mathbf{d}_h^i, \mathbf{a}_h^i \in [L^2(\Omega)]^d$, *and a scalar-valued function* $\rho_h^i \in L^2(\Omega)$ *such that*

1. $\nabla\cdot \mathbf{t}_h^i = f_h - \rho_h^i\,,$
2. $\mathbf{t}_h^i = \mathbf{d}_h^i + \mathbf{a}_h^i\,,$
3. *as the linear solver converges,* $\|\mathbf{a}_h^i\| \to 0\,.$

Here f_h is a piecewise polynomial approximation of the source term f verifying $(f_h, 1)_K = (f, 1)_K$ for all $K \in \mathcal{T}_h$.

For any $K \in \mathcal{T}_h$, the Poincaré inequality states that

$$\|\varphi - \varphi_K\|_K \leq C_{\mathrm{P}} h_K \|\nabla\varphi\|_K \qquad \forall \varphi \in H^1(K)\,, \tag{9}$$

where φ_K denotes the mean value of φ in K. Since the simplices K are convex, there holds $C_{\mathrm{P}} = 1/\pi$; see, e.g., [2, 3]. The Friedrichs inequality states that

$$\|\varphi\| \leq h_\Omega \|\nabla\varphi\| \qquad \forall \varphi \in V\,, \tag{10}$$

where h_Ω denotes the diameter of the domain Ω. The following theorem is a simple application of [1, Theorems 3.4 and 3.6] to our model problem. We denote local estimators in the form $\eta^i_{\Box,K}$, where $i = 1, 2, \ldots$ stands for the algebraic iteration step and $K \in \mathcal{T}_h$ for the mesh element. The global versions of these estimators are defined as $\eta^i_\Box \equiv \left\{\sum_{K\in\mathcal{T}_h} (\eta^i_{\Box,K})^2\right\}^{1/2}$.

Theorem 3.2 (Total error a posteriori estimate distinguishing error components)**.** *Let $u \in V$ solve (2), let $u^i_h \in V_h$ be given by (7), and let Assumption 3.1 hold. For any $K \in \mathcal{T}_h$, define respectively the* discretization estimator*, the* algebraic estimator*, the* algebraic remainder*, and the* data oscillation estimator *as*

$$\begin{aligned} \eta^i_{\mathrm{disc},K} &\equiv \|\nabla u^i_h + \mathbf{d}^i_h\|_K\,, && (11)\\ \eta^i_{\mathrm{alg},K} &\equiv \|\mathbf{a}^i_h\|_K\,, && (12)\\ \eta^i_{\mathrm{rem},K} &\equiv h_\Omega\|\rho^i_h\|_K\,, && (13)\\ \eta^i_{\mathrm{osc},K} &\equiv C_{\mathrm{P}} h_K\|f - f_h\|_K\,. && (14)\end{aligned}$$

Then

$$\|\nabla(u - u^i_h)\| \le \eta^i_{\mathrm{disc}} + \eta^i_{\mathrm{alg}} + \eta^i_{\mathrm{rem}} + \eta^i_{\mathrm{osc}}\,. \qquad (15)$$

In the adaptive algorithm proposed in [1] the flux reconstruction $\mathbf{d}^i_h$ is constructed using the approximate algebraic solution U^i given at the i-th step of algebraic iterative solver. Then one performs $\nu > 0$ additional iteration steps yielding the vector $\mathsf{U}^{i+\nu}$ and the corresponding flux reconstruction $\mathbf{d}^{i+\nu}_h$. The algebraic error flux reconstruction is defined as $\mathbf{a}^i_h \equiv \mathbf{d}^{i+\nu}_h - \mathbf{d}^i_h$. The number ν of the additional iteration steps and the convergence of the algebraic solver are controlled using the (global) criteria

$$\begin{aligned} \eta^i_{\mathrm{rem}} &\le \gamma_{\mathrm{rem}} \max\left\{\eta^i_{\mathrm{disc}}, \eta^i_{\mathrm{alg}}\right\}, && (16)\\ \eta^i_{\mathrm{alg}} &\le \gamma_{\mathrm{alg}}\, \eta^i_{\mathrm{disc}}\,, && (17)\end{aligned}$$

or using the elementwise equivalents

$$\begin{aligned} \eta^i_{\mathrm{rem},K} &\le \gamma_{\mathrm{rem},K} \max\left\{\eta^i_{\mathrm{disc},K}, \eta^i_{\mathrm{alg},K}\right\}, && && (18)\\ \eta^i_{\mathrm{alg},K} &\le \gamma_{\mathrm{alg},K}\, \eta^i_{\mathrm{disc},K}\,, && \forall K \in \mathcal{T}_h\,. && (19)\end{aligned}$$

Here $\gamma_{\mathrm{rem}}, \gamma_{\mathrm{alg}}$ (respectively $\gamma_{\mathrm{rem},K}, \gamma_{\mathrm{alg},K}$) are the user-given weights (typically of order 0.1). The criteria (16)–(17) are sufficient to establish the global efficiency of the total error estimator; the *local* criteria (18)–(19) assure the *local* efficiency; see [1, Section 5].

Elaborating on the results from [1], our goal is to bound also the algebraic error

$$\|\nabla(u_h - u^i_h)\| = \sup_{\varphi_h\in V_h, \|\nabla\varphi_h\|=1} \left(\nabla(u_h - u^i_h), \nabla\varphi_h\right),$$

where u_h is the (unknown) solution of the discrete formulation (4) and $u^i_h \in V_h$ is an approximation to u_h as given by (7). We introduce for this purpose an additional assumption on the flux reconstruction.

Assumption 3.3 (Quasi-equilibration of $\mathbf{d}^i_h$)**.** *The function $\mathbf{d}^i_h$ satisfies $\mathbf{d}^i_h \in \mathbf{H}(\mathrm{div}, \Omega)$ and there exists a scalar-valued function $r^i_h \in L^2(\Omega)$ such that*

$$\begin{aligned} \nabla\cdot \mathbf{d}^i_h &= f_h - r^i_h\,, && (20)\\ (r^i_h, \psi_j) &= \mathsf{R}^i_j \qquad \forall j \in \mathcal{C}\,. && (21)\end{aligned}$$

Assuming (20) and setting $\mathbf{a}_h^i = \mathbf{d}_h^{i+\nu} - \mathbf{d}_h^i$ as above, Assumption 3.1 is satisfied with $\rho_h^i \equiv r_h^{i+\nu}$.

Theorem 3.4 (Algebraic error a posteriori estimate)**.** *Let u_h be the solution of* (4) *and $u_h^i \in V_h$ be given by* (7)*. Let $\eta_{\mathrm{alg}}^i, \eta_{\mathrm{rem}}^i$ be defined respectively by* (12) *and* (13)*. Let Assumption 3.3 hold. Then*

$$\|\nabla(u_h - u_h^i)\| \leq \eta_{\mathrm{alg}}^i + \eta_{\mathrm{rem}}^i \,. \tag{22}$$

Therefore, using the criteria (16) or (18), the algebraic estimator η_{alg}^i provides an upper bound on the algebraic error. The efficiency of this estimator is a subject of further study — the techniques used for the proof of global and local efficiency of the total error estimator (see [1, Section 5]) are not applicable in this case.

4 Flux reconstructions

The paper [1] presents flux reconstruction in various discretization schemes that fulfill Assumption 3.1 and the first part (20) of Assumption 3.3. In this contribution we restrict ourselves to the conforming finite element method. We show that we can easily modify the flux reconstruction from [1] such that the relation (21) required for proving the bound (22) is also satisfied. The flux reconstruction is sought in the Raviart–Thomas–Nédélec finite element space and it is constructed using (mutually independent) local homogeneous Neumann mixed finite element problems posed on patches around mesh vertices.

5 Conclusion

Following [1] we presented a posteriori error estimate for the total error that distinguishes its different components. The estimate yields a guaranteed upper bound on the total error. Additionally, we showed that the parts of the estimate denoted as algebraic estimator and algebraic reminder provide an upper bound on the algebraic error. This justifies the distinction of error components and the stopping criteria presented in [1]. We applied the general framework from [1] to a linear problem and the conforming finite element discretization. The application for other discretization schemes and nonlinear problems and the efficiency of the estimate are subjects of further study.

Acknowledgement: The work was supported by the project LL1202 in the programme ERC-CZ funded by the Ministry of Education, Youth and Sports of the Czech Republic, and by the project 201/13-06684S of the Grant Agency of the Czech Republic.

References

[1] A. Ern, M. Vohralík: *Adaptive inexact Newton methods with a posteriori stopping criteria for nonlinear diffusion PDEs.* SIAM J. Sci. Comput., 35, (4), A1761–A1791, 2013.

[2] L. E. Payne, H. F. Weinberger: *An optimal Poincaré inequality for convex domains.* Arch. Rational Mech. Anal., 5, 286–292, 1960.

[3] R. Verfürth: *A note on polynomial approximation in Sobolev spaces.* M2AN Math. Model. Numer. Anal., 33, (4), 715–719, 1999.

Variability of Turing patterns in reaction-diffusion systems

V. Rybář, T. Vejchodský

Institute of Mathematics AS CR, Prague

1 Introduction

Systems of reaction-diffusion equations have been used for several decades in biology and ecology to explain phenomena concerning symmetry breaking, spatial variations, and formation of patterns. For example, let us mention predator-prey models as two (or more) species spatial ecological models, biochemical reaction-diffusion systems of morphogenes in developmental biology, formation of skin patterns, and vascularization of tumours. Typical reaction-diffusion system consists of two equations

$$\frac{\partial u}{\partial t} = D_1 \Delta u + f(u,v) \quad \text{in } (0,\infty) \times \Omega, \tag{1}$$

$$\frac{\partial v}{\partial t} = D_2 \Delta v + g(u,v) \quad \text{in } (0,\infty) \times \Omega, \tag{2}$$

where $u = u(t,x),\ v = v(t,x),\ \Omega \subset \mathbb{R}^2$ is a domain, D_1, D_2 are diffusion coefficients and $f(u,v)$, $g(u,v)$ are nonlinear reaction terms.

Turing showed [4] that if u and v are in a linearly stable uniform steady state in case $D_1 = D_2 = 0$, then this state can, under certain conditions, become unstable for $D_1 \neq 0$, $D_2 \neq 0$, and spatially inhomogeneous stationary solution can evolve. Such solutions are called patterns. The set of parameters that yield patterns is known as the Turing domain. Linear analysis can help with identification of the Turing domain, but in general system (1)–(2) is a source of non-trivial problems in the fields of bifurcation analysis, theory of partial differential equations and others.

In this brief contribution we study the non-uniqueness of stationary solutions to problem (1)–(2) with periodic boundary conditions. For simplicity, let us consider Ω to be a square $(0,L)^2$ and define the following periodic boundary conditions

$$u(0,y) = u(L,y) \quad \forall y \in (0,L) \quad \text{and} \quad u(x,0) = u(x,L) \quad \forall x \in (0,L), \tag{3}$$

$$v(0,y) = v(L,y) \quad \forall y \in (0,L) \quad \text{and} \quad v(x,0) = v(x,L) \quad \forall x \in (0,L). \tag{4}$$

We first show that any shift of a stationary solution to problem (1)–(4) is again a stationary solution. Therefore, we define a periodic shift of a function by a vector (r,s). Let $u \in C([0,L]^2)$ satisfy the periodic boundary condition (3). The periodic shift $\tilde{u} \in C([0,L]^2)$ of u by $(r,s) \in (0,L)^2$ is defined as

$$\tilde{u}(x,y) = \begin{cases} u(x+r, y+s) & \text{for } x \in (0, L-r), y \in (0, L-s), \\ u(x+r, y+s-L) & \text{for } x \in (0, L-r), y \in (L-s, L), \\ u(x+r-L, y+s) & \text{for } x \in (L-r, L), y \in (0, L-s), \\ u(x+r-L, y+s-L) & \text{for } x \in (L-r, L), y \in (L-s, L). \end{cases} \tag{5}$$

Note that values $\tilde{u}(x,y)$ for $x = L-r$ or $y = L-s$ are determined by the continuity.

Lemma 1. *Let $u, v \in C^2([0, L]^2)$ be a stationary solution to* (1)*–*(2)*. Let r,$s \in (0, L)$ be fixed and let $\tilde{u}$ and $\tilde{v}$ be periodic shifts of u and v, respectively, by the vector (r, s). Then $\tilde{u}, \tilde{v}$ is a stationary solution to problem* (1)*–*(4)*.*

The proof of this lemma is easy and we skip it. Lemma 1 implies that there are classes of stationary solutions to problem (1)–(4) that are equivalent up to a shift. Thus, there is a question, how many classes of solutions there exist for a given nonlinear system. Therefore, we focus on a particular system from [2] and try to answer this question numerically.

2 Model problem and numerical scheme

Liu, Liaw, and Maini use in [2] the following reaction-diffusion system

$$\frac{\partial u}{\partial t} = D\delta\Delta u + \alpha u + v - r_2 uv - \alpha r_3 uv^2, \tag{6}$$

$$\frac{\partial v}{\partial t} = \delta\Delta v - \alpha u + \beta v + r_2 uv + \alpha r_3 uv^2. \tag{7}$$

to model the formation of pigment patterns on coats of leopards and jaguars. As opposed to [2], we equip system (6)–(7) with periodic boundary conditions (3)–(4). Due to unstable behaviour of this system, we compute its stationary solutions by sufficiently long time evolutions starting from initial conditions that mimic small amplitude random fluctuations around the spatially constant steady state. We use the fourth order Runge-Kutta method [1] for time discretization. Fourier collocation spectral method, implemented according to [3], was used for spatial discretization and we present its brief description.

Let us consider a function z sampled on the spatial discretization grid $\{x_1, \dots, x_N\}$ with $z_j = z(x_j)$. Let z be periodic, i.e. $z_1 = z_N$. Using definitions of discrete Fourier transform (DFT) and inverse discrete Fourier transform (both properly defined and discussed in [3]), we can compute the derivatives $w_j = z'(x_j)$, $j = 1, \dots, N$, by the following procedure:

1. given $z_j,\ j = 1, \dots, N$, compute its DFT $\hat{z}_k = \sum_{j=1}^{N} \mathrm{e}^{-\mathrm{i}kx_j} z_j,\ k = -N/2 + 1, \dots, N/2,$
2. define $\hat{w}_k = \mathrm{i}k\hat{z}_k,\ k = -N/2 + 1, \dots, N/2,$
3. compute $w_j = \frac{1}{2\pi} \sum_{k=-N/2+1}^{N/2} \mathrm{e}^{\mathrm{i}kx_j} \hat{w}_k,\ j = 1, \dots, N.$

Applying this procedure two times yields second derivatives. Thus, the diffusion terms in (1) and (2) can be transformed into $-Dk^2\hat{u}_k$ and $-Dk^2\hat{v}_k$, respectively, and partial differential equations (1)–(2) transform to a system of ordinary differential equations

$$\frac{\mathrm{d}\hat{u}_k}{\mathrm{d}t} = -D_1 k^2 \hat{u}_k + \widehat{f(u, v)}, \quad k = -N/2 + 1, \dots, N/2, \tag{8}$$

$$\frac{\mathrm{d}\hat{v}_k}{\mathrm{d}t} = -D_2 k^2 \hat{v}_k + \widehat{g(u, v)}, \quad k = -N/2 + 1, \dots, N/2. \tag{9}$$

This system can be efficiently solved for example by the fourth order Runge-Kutta method [1].

3 Numerical experiments

We use system (6)–(7) with boundary conditions (3)–(4) and with parameters taken from [2], $D = 0.45$, $\delta = 6$, $\alpha = 0.899$, $\beta = -0.91$, $r_2 = 2$, and $r_3 = 3.5$. These parameters yield stationary solutions that correspond to spotted patterns. The components u and v of a stationary solution are complementary in the sense that local maxima of u (centres of spots) correspond to local minima of v. Therefore, we concentrate on the component v only in what follows.

The experiments are performed in domain $\Omega = (0, L)^2$ with $L = 50$ which was divided into 48×48 vertices of uniform discretization grid. As an initial condition, we generate a uniformly distributed random number in $(-0.05, 0.05)$ for every node of the grid. The time stepping is performed with step $\Delta t = 1$, and it is terminated as soon as the relative l^2-norm of two consecutive approximate stationary solutions v_k and v_{k+1} in times $t_k = k\Delta t$ and $t_{k+1} = (k+1)\Delta t$ is smaller than 10^{-4}, i.e. when

$$\frac{\|v_k - v_{k+1}\|_{l^2}}{\|v_k\|_{l^2}} < 10^{-4}. \tag{10}$$

Note that the discrete l^2-norms are computed over the grid nodes.

We solved the problem with this setup 6000 times. Every time with different (random) initial condition. In the resulting sample of 6000 stationary solutions we try to identify classes of solutions that are identical up to a shift in the sense of Lemma 1. We do this by successive building of a database of representatives of solution classes and numbers of solutions in every class. At the beginning the database is empty. For every stationary solution in the sample, we check whether it is equivalent to a representative from the database. If it is, we increase the number of solutions in this class by one. If not, we insert this solution into the database as a representative of a new class and initialise the number of solutions in it to one.

The crucial step in this algorithm is the check of equivalence of two stationary solutions. Given two computed stationary solutions v_1 and v_2, we determine their equivalence according to the following procedure. We first shift v_1 and v_2 to $\tilde{v}_1$ and $\tilde{v}_2$ according to (5) such that minima of $\tilde{v}_1$ and $\tilde{v}_2$ are attained in the centre of the square $(0, L)^2$, i.e. in the point $(25, 25)$. Then we test if the relative l^2-norm of the difference of $\tilde{v}_1$ and $\tilde{v}_2$ is below a tolerance TOL. This means that if v_1 and v_2 are equivalent

$$\frac{\|\tilde{v}_1 - \tilde{v}_2\|_{l^2}}{\|\tilde{v}_1\|_{l^2}} < \text{TOL}.$$

In this numerical experiment we have chosen TOL $= 0.16$. This value corresponds to the observed sizes of differences between solutions within the same class. These differences are caused mainly by the discretization error on the relatively coarse grid and by the chosen time step. With TOL set to this level, the algorithm identified 9 different classes of solutions in the sample of 6000 stationary solutions. Table 1 presents numbers of solutions in these classes. Figure 1 shows representatives of these classes with minima centred to $(25, 25)$. In this figure, we observe certain symmetries. For example, rotating the representative of class 1 by 90°, we obtain the representative of class 2. Representative of class 3 is representative of class 4 reflected over the horizontal or vertical axis. Similarly, representatives of classes 5, 6, 7 and 8 differ by suitable reflections and rotations. These symmetries are not surprising due to symmetries of the domain.

Class	1	2	3	4	5	6	7	8	9
# Solutions	2997	2898	46	32	9	5	2	8	3

Table 1: Number of solutions in solution classes.

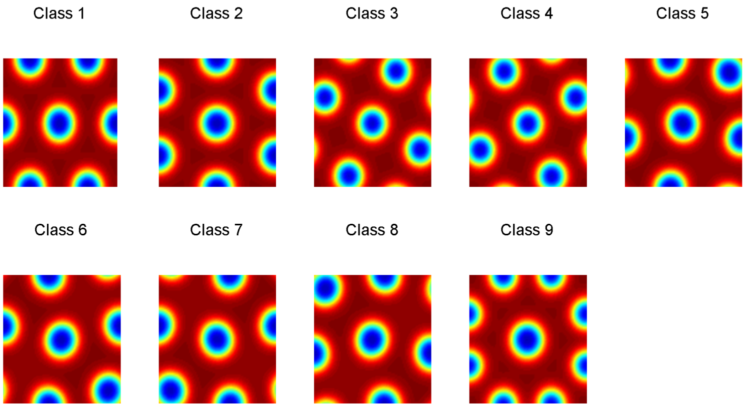


Figure 1: Representatives of solution classes.

4 Conclusion

The performed numerical study confirms that identification of the number of distinct stationary solutions for reaction-diffusion problems undergoing Turing instability is non-trivial. Besides the general fact that a shift of a periodic solution remains a solution, we have identified a number of equivalence classes of distinct stationary solutions for the particular system. Data in Table 1 show that stationary solutions in classes 1 and 2 are much more frequent than solutions in the remaining classes. This means that most of the random initial conditions lead to stationary solutions from classes 1 and 2. Only a small fraction of initial conditions leads to a stationary solution of another class. An interesting point is that the number of time steps to reach the steady state according to the criterion (10) is considerably smaller for the solutions from classes 1 and 2 in comparison to the other classes. This indicates that solutions from classes 1 and 2 are more natural and robust steady states of the system, while solutions from the other classes are exceptional, but still existing steady states.

Of course, it is not clear whether we succeeded to identify all classes of stationary solutions for system (6)–(7) with boundary conditions (3)–(4). There is still a possibility that there are other extremely rare stationary solutions. Theoretically, it would be interesting to link the obtained classes of solutions to possible stationary solutions of linearised system which can be obtained analytically. Further, we plan to investigate the influence of the domain size on the patterns and try to find a natural period of the spotted patterns. Finally, we plan to perform a similar study in the case of homogeneous Neumann boundary conditions.

References

[1] A. K. Kassam: *Solving reaction-diffusion equations 10 times faster.* Numerical Analysis Group Research Report, 16, Mathematical Institute, Oxford, 2003.

[2] R. T. Liu, S. S. Liaw, P. K. Maini: *Two-stage Turing model for generating pigment patterns on the leopard and the jaguar.* Physical Review, 74, 011914, 2006.

[3] L. N. Trefethen: *Spectral methods in MATLAB.* SIAM, Philadelphia, 2000.

[4] A. M. Turing: *The chemical basis of morphogenesis.* Philosophical Transactions of the Royal Society B, 237, 37–72, 1952.

Numerical results of plunger cavity optimal design

P. Salač

Technical University of Liberec, Liberec

1 Introduction

This work concerns the numerical approximation of continuous problem of the shape optimization presented in [1], where a rotationally symmetric system of mould, glass piece, plunger and plunger cavity is considered. The state problem is given as a stationary head conduction-convection process. The system has given heat source and is cooled by flowing water inside the cavity and outside by environment. The design variable is taken to be the shape of inner surface of the plunger cavity. Existence and uniqueness of the state problem solution and existence of a solution of the optimization problem are proved in [1].

The results of the numerical optimization to required target temperature 800 [°C] of the outward surface of the plunger Γ_1 together with the distribution of temperatures along the interface Γ_1 between the plunger and the glass piece before and after the optimization process are presented.

2 Numerical results

The scheme of the system with dashed design function is visualized on the Fig. 1. For detail formulation of the problem, proofs of existence and uniqueness of the state problem solution and existence of a solution of the optimization problem see [1].

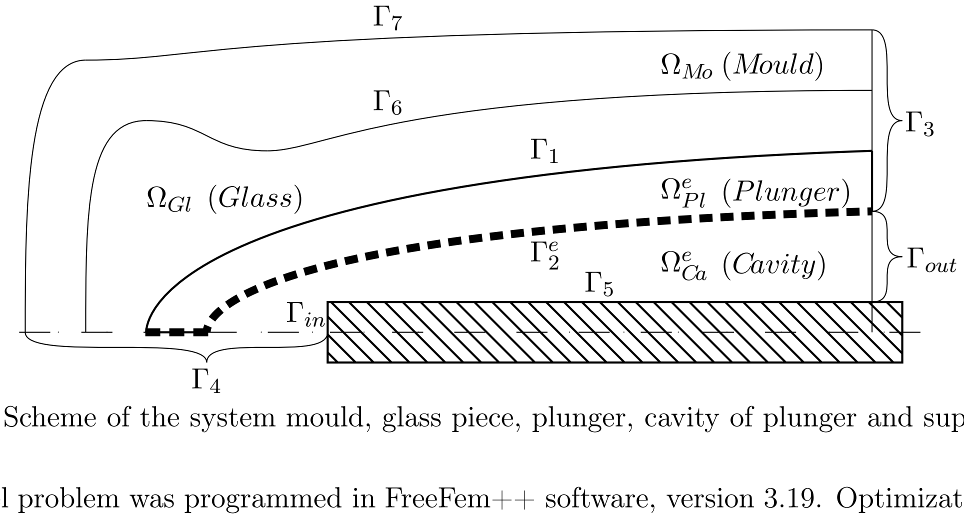


Figure 1: Scheme of the system mould, glass piece, plunger, cavity of plunger and supply tube.

The model problem was programmed in FreeFem++ software, version 3.19. Optimization of the plunger cavity shape was implemented on the system for pressing glass vases of high 267 [mm] of a weight $1,55$ [kg]. The heat source was determined as the average power derived from the solution of mixed problem for heat conduction with forced linear decreasing Dirichlet boundary conditions to the given surface temperature of 800 [°C] at the moment of separation pressing tools and glass piece, i.e. at the time of 13 [s] on the inner side and at the time of 88 [s] on the outward of the glass piece and with constant initial temperature of 1150 [°C]= 1423 [K] throughout the region Ω_{Gl}, i.e. the problem

$$
\begin{aligned}
c_v \varrho \frac{\partial \vartheta}{\partial t} &= k \Delta \vartheta && \text{in } [0;13] \times \Omega_{Gl}\ , && (1)\\
\vartheta(0,x,r) &= 1423 && \text{in } \Omega_{Gl}\ , && (2)\\
\vartheta(t,x,r) &= \frac{1073-1423}{13} t + 1423 && \text{in } [0;13] \times \Gamma_1\ , && (3)\\
\vartheta(t,x,r) &= \frac{1073-1423}{88} t + 1423 && \text{in } [0;13] \times \Gamma_6\ , && (4)
\end{aligned}
$$

where we set up for specific heat capacity of glass $c_v = 796$ [J kg^{-1} K^{-1}], density $\varrho = 2500$ [kg m^3]. We solve this mixed problem by the method of time discretization. Stationary heat source for the state problem is determined from the solution of this problem in time $t = 13$ [s] according to the relation

$$
q(x,r) = \frac{c_v}{13}(\vartheta(0,x,r) - \vartheta(13,x,r))\ . \qquad (5)
$$

Cooling of the plunger cavity was realized by the potential flow of cooling water with constant mass flow 1 [kg min^{-1}], inlet temperature of 15 [°C] and outlet temperature of 100 [°C]. We obtain velocity field of flowing water as a solution of the Neumann boundary value problem for the Laplace equation in the plunger cavity Ω^e_{Ca}. For the detail variational formulation of the problem of potential flow of water see (1.3)–(1.5) in [1] pages 408–409. We used FEM by FreeFem++ with automatic mesh generator for the numerical solution.

Cooling of the mould from the outward was realized by outside environment of temperature 60 [°C] with considered coefficient of heat transfer 14 [W m^{-2} K^{-1}]. We look for the temperature distribution in the entire system by solving of the state problem in the form of a mixed boundary value problem for the energy equation in which we employ the heat source (5) determined on the basis of the solution of the problem (1)–(4) and potential flow from the problem (1.3)–(1.5). For the detail variational formulation of the problem for energy equation see [1] pages 410–414. We again used FEM by FreeFem++ with automatic mesh generator for the numerical solution.

We solve the problem of the optimal design formulated in [1] page 414. The cost functional of the continuous problem is in the form

$$
\mathcal{J}^S(F^e_2) = \|\vartheta(F^e_2)|_{\Gamma_1} - 1073\|^2_{0,r,\Gamma_1}\ , \qquad (6)
$$

where $\vartheta(F^e_2)|_{\Gamma_1}$ is the trace of solution $\vartheta(F^e_2)$ of continuous state problem in the region Ω^e_{Pl} on the boundary Γ_1, and the optimal surface plunger temperature is $T_{\Gamma_1} = 1073$ [K]= 800 [°C]. The surface integral in the cost functional is computed numerically by the midpoint method with equidistant division to 1000 subintervals according to the length of arc.

Temperatures in 11 pilot points on the surface Γ_1 of the plunger were monitoring during optimization. To these 11 points, the 11 "shadow" points were found using gradient lines of temperatures at the boundary Γ^e_2 between the plunger and water. The shadow points were moved in the directions corresponding to the required changes of temperatures in the pilot points. The shape of the plunger cavity surface is created by natural cubic spline functions inset by the 11 shadow points.

Remark. A sensitivity analysis can be performed on the basis of temperature evaluation along the boundary Γ_1. Let us introduce a homeomorphism between the outward plunger boundary Γ_1 and the plunger cavity boundary Γ^e_2 defined by the gradient lines of the temperature field in the plunger. In the parts of Γ_1 where we need to decrease the temperature, we narrow "the wall" by moving the points of Γ_2^c along the gradient lines to locally achieve more intensive cooling. On the other hand, in places of Γ_1 where we need higher temperature, we increase "the wall thickness" to locally decrease the intensity of cooling. By the term "the wall thickness" we understand the length of the temperature gradient line that connects the related points of Γ_1 and Γ_2^c.

In the numerical realization of the solution we first determine the stationary heat source for the state problem by application of time-discretization to the mixed heat conduction problem (1)–(4), then we start to iterate. In each iteration, first we find a cubic spline functions passing through the shadow points which form the inner wall of the plunger cavity. In the calculation we use the rotation of the coordinate system about 60°. After we obtain plunger cavity we solve the problem for finding potential flow of cooling water, then we deal with the state problem for temperature throughout the system. Next, we determine the cost functional and the coordinates of shadow points for the next iteration.

100 iterations was carried to find the optimal shape of the cavity. Before the beginning of the iteration process the approximated cost functional took on the value of $796,982$, gradually declined to values around ten and then fluctuated. The minimum value of $3,123$ has been achieved in the 72-th iteration. Halting the decline in value of the cost functional was caused by a small number of shadow points (11 points), for further refinement would be needed to increase their number.

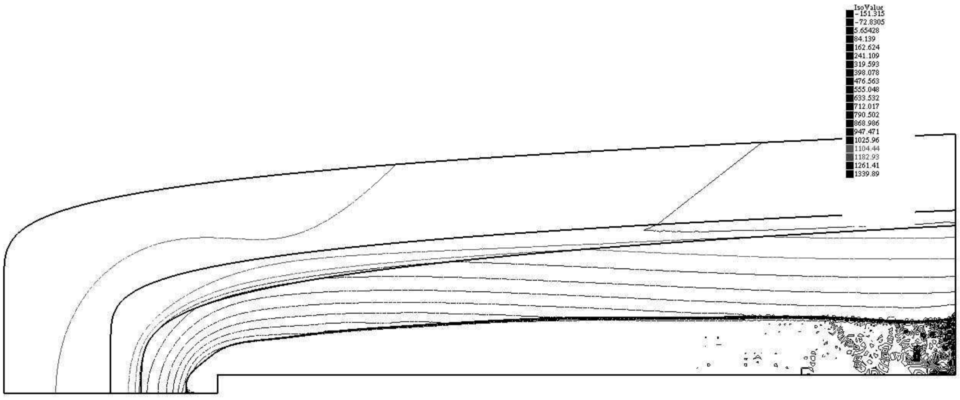


Figure 2: Contours of temperature at the beginning of the optimization.

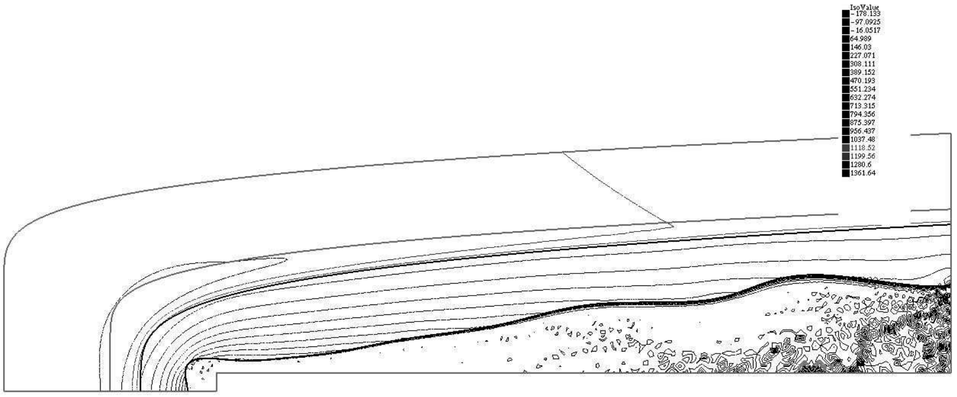


Figure 3: Contours of temperature and the final shape of the plunger cavity optimized under surface target temperature 800 [°C]= 1073 [K].

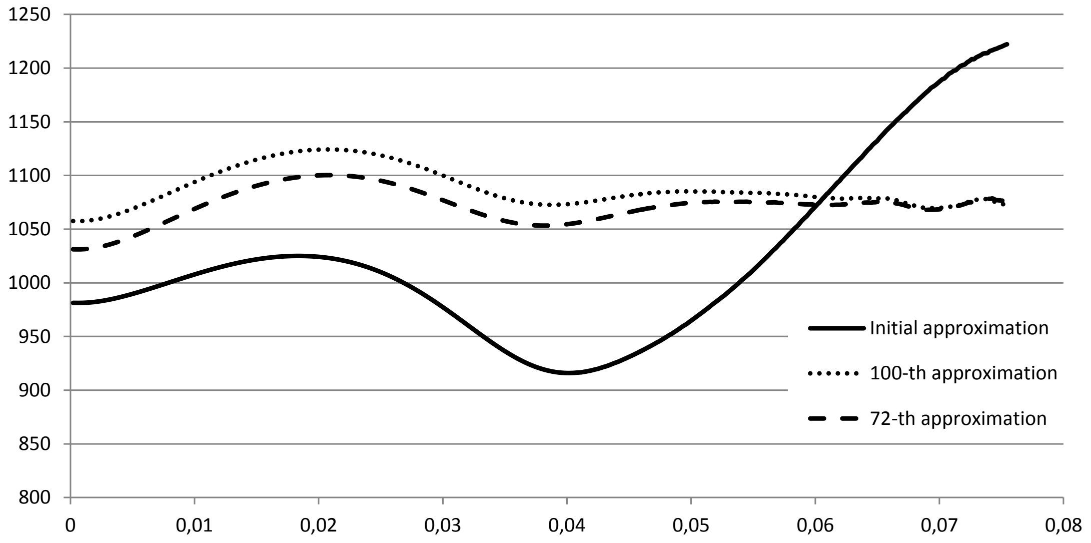


Figure 4: Distribution of temperature along the surface Γ_1 before and after the optimization.

Fig. 4 shows distributions of temperatures along plunger outward surface before and after the optimization to the required target temperature of the plunger surface 800 [°C]= 1073 [K].

3 Conclusion

The problem of cooling of the plunger by stationary flowing water through its cavity was introduced. Results of the numerical plunger cavity shape optimization with a view to achievement required temperature 800 [°C]= 1073 [K] at surface Γ_1 between the glass and the plunger were performed. Numerical results shows limited capability of approximation by cubic splines and suggest using of order approximation (for example B-splines).

Acknowledgement: This work was realized with financial support by the Technological Agency of the Czech Republic, project No. TA03010852.

References

[1] P. Salač: *Optimal design of the cooling plunger cavity.* In: Appl. Math., 58, 405–422, 2013.

[2] P. Salač: *Problem of identification of heat transfer coefficients.* In: Conference Proceedings of Seminar on Numerical Analysis'13, January 21.–25. 2013, 97–100, Rožnov pod Radhoštěm, ISBN 978-80-86407-34-0.

Max-min and min-max approximation problems for normal matrices revisited

P. Tichý

Institute of Computer Science AS CR, Prague

We give a new proof for an equality of certain max-min and min-max approximation problems involving normal matrices. The previously published proofs of this equality apply tools from matrix theory, (analytic) optimization theory and constrained convex optimization. Our proof uses a classical characterization theorem from approximation theory and thus exploits the link between the two approximation problems with normal matrices on the one hand and approximation problems on compact sets in the complex plane on the other.

Solution of algebraic systems arising from the discontinuous Galerkin discretization of PDEs by the p-multigrid technique

A. Živčák, V. Dolejší

Faculty of Mathematics and Physics
Charles University in Prague

We deal with the numerical solution of partial differential equations with the aid of the discontinuous Galerkin (DG) method. This technique is based on piecewise polynomial but discontinuous approximation. Therefore, we can simply construct hierarchical basis functions locally for each element.

The DG discretization leads to the necessity to solve large (non-)linear algebraic systems. Among the most efficient techniques solving algebraic systems belong the so-called *multigrid methods*, which aim to attain the so called *textbook multigrid efficiency.*

Multigrid methods are based on coarser representations of the discretized problem. Can be used for solving linear and also nonlinear problems. Very well known and widely used h-multigrid is based on geometrical hierarchy of computational meshes. However, for the DG discretization, more suitable is the so-called p-variant of multigrid, where a hierarchy of discretization spaces with respect to polynomial approximation degree p is considered.

Projection operators, which carry out the restriction and prolongation depends on choice of basis function. Due to the locality of basis function in the DG method we get local projection operators. Their form is very simple in the case of orthonormal basis function and therefore effortless implementation can be used.

We describe the application of the p-multigrid to the DG method, namely the restriction and prolongation operators. We discuss several solution strategies and present first preliminary numerical results in comparison with iterative solvers. Moreover, we mention some weakness of the presented algorithm and also give some outline of a possible use of a non-linear multigrid.

Winter school lectures

V. Kučera
Discontinous Galerkin method

Z. Strakoš
Operator preconditioning

M. Šorel
Mathematics in image processing

J. Vondřejc, J. Zeman, I. Marek
FFT-based Galerkin method for homogenization of periodic media

Discontinuous Galerkin method

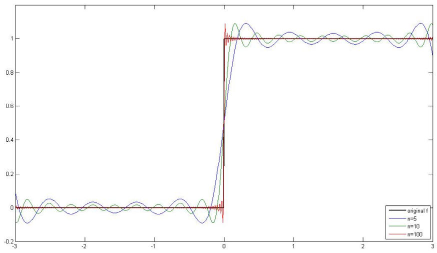
Václav Kučera

Department of Numerical Mathematics
Charles University in Prague

Gibbs phenomenon

Gibbs phenomenon

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx} \quad \approx \quad f_N(x) = \sum_{n=-N}^{N} c_n e^{inx}$$



Gibbs phenomenon

- Stigler’s law of eponymy: *”No scientific discovery is named after its original discoverer”*.
- Discovered and explained by Henry Wilbraham 1848, i.e. 51 years before Josiah Willard Gibbs.
- Similar phenomenon observed in FEM, however nobody ever proved any deeper connection with classical Gibbs.
- Well understood for Fourier series.
- In FEM, we essentially cure the symptoms and not the cause (stabilizations, filtering, postprocessing,...).
- Observation: approximating discontinuous functions by continuous (or even smooth) functions is not a good idea.
- Approximate by piecewise continuous functions instead.

Discontinuous approximations

Finite element method
- Globally continuous piecewise polynomial approximations.
- Arbitrarily high orders of convergence.
- Gibbs phenomenon ruins everything.

Finite volume method
- Solution approximated by piecewise constant functions.
- Does not suffer from Gibbs phenomenon (usually).
- Lowest possible order.
- Very dissipative.

Discontinuous Galerkin (DG)
- Piecewise polynomial solutions.
- Global continuity in some weak sense (penalization).
- Arbitrarily high orders of convergence.
- Gibbs phenomenon stays localized (unlike FEM).
- Expensive.

The concept of penalization

Poisson problem

$$-\Delta u = f \quad \text{in } \Omega.$$

- Multiply by *test function* $v \in H^1(\Omega)$, integrate over Ω and apply Green's theorem:

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x - \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x. \qquad (1)$$

- Seek $u \in H^1(\Omega)$ such that (1) holds for all $v \in H^1(\Omega)$.

Boundary conditions

- Neumann: $\nabla u \cdot \mathbf{n} = g_N$ on $\Gamma_N \subset \partial\Omega$.
- Dirichlet: $u = u_D$ on $\Gamma_D \subset \partial\Omega$.

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x - \int_{\Gamma_D} \nabla u \cdot \mathbf{n} v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x + \int_{\Gamma_N} g_N v \, \mathrm{d}S.$$

Dirichlet boundary conditions

$$-\Delta u = f \text{ in } \Omega, \qquad u = u_D \text{ on } \partial\Omega.$$

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x - \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x.$$

- Unlike the Neumann condition, there is no way how to incorporate $u|_{\partial\Omega} = u_D$ directly into the equation itself.
- We write $u = u_0 + \tilde{u}_D$, where

$$u_0|_{\partial\Omega} = 0,$$
$$\tilde{u}_D|_{\partial\Omega} = u_D.$$

- $\tilde{u}_D \in H^1(\Omega)$ chosen arbitrarily.
- New equation for $u_0 \in H^1_0(\Omega)$, weak formulation holds for all $v \in H^1_0(\Omega)$.

Dirichlet BCs by penalization

Courant '43, Lions'68, Babuška '73...

$$u = u_D \quad \longmapsto \quad u + \varepsilon \nabla u \cdot \mathbf{n} = u_D, \ \varepsilon \ll 1.$$

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x - \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x$$

$$\Downarrow$$

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x + \frac{1}{\varepsilon} \int_{\partial\Omega} (u - u_D) v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x$$

We seek $u_\varepsilon \in H^1(\Omega)$ such that

$$\int_\Omega \nabla u_\varepsilon \cdot \nabla v \, \mathrm{d}x + \frac{1}{\varepsilon} \int_{\partial\Omega} (u_\varepsilon - u_D) v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x, \quad \forall v \in H^1(\Omega).$$

Dirichlet BCs by penalization

$$\underbrace{\int_\Omega \nabla u_\varepsilon \cdot \nabla v \, \mathrm{d}x}_{(1)} + \underbrace{\frac{1}{\varepsilon} \int_{\partial\Omega} (u_\varepsilon - u_D) v \, \mathrm{d}S}_{(2)} = \underbrace{\int_\Omega f v \, \mathrm{d}x}_{(1)}.$$

- $u_\varepsilon, v \in H^1(\Omega)$.
- (1) = standard formulation of the equation.
- (2) = penalization of non-satisfaction of Dirichlet BC.
- $u_\varepsilon|_{\partial\Omega} \neq u_D$, but $u_\varepsilon|_{\partial\Omega} \to u_D$ for $\varepsilon \to 0$.
- u_ε is the unique minimiser over $H^1(\Omega)$ of the functional

$$J_\varepsilon(v) := |v|^2_{H^1(\Omega)} - 2\int_\Omega f v \, \mathrm{d}x + \frac{1}{\varepsilon} \int_{\partial\Omega} (v - u_D)^2 \, \mathrm{d}S.$$

- The following estimate holds:

$$|u - u_\varepsilon|^2_{H^1(\Omega)} + \|u_D - u_\varepsilon\|^2_{L^2(\partial\Omega)} = O(\varepsilon).$$

$$\int_\Omega \nabla u_\varepsilon \cdot \nabla v \, \mathrm{d}x + \frac{1}{\varepsilon} \int_{\partial\Omega} (u_\varepsilon - u_D) v \, \mathrm{d}S = \int_\Omega f v \, \mathrm{d}x.$$

Penalization - general recipe

- Take your equation in weak form.
- Add your requirement (eg. $u - u_D = 0$) to your equation, with some weight (e.g. $\frac{1}{\varepsilon}$) and tested by some expression involving the test function (e.g. v).
- Resulting left-hand side term should be semi-elliptic, e.g. $\frac{1}{\varepsilon} \int_{\partial\Omega} u_\varepsilon^2 \, \mathrm{d}S \geq 0$.

DG method for ordinary differential equations

Abstract ODE

We seek $u : [0,T] \to H$ such that

$$u'(t) + Au(t) = f(t),\ \forall t \in (0,T), \quad u(0) = u^0.$$

- H is a Hilbert space with scalar product $(\cdot,\cdot)_H$.
- $A : H \to V$ is a given operator.
- $f : [0,T] \to V$ is a given right-hand side.
- V is a Hilbert space with scalar product $(\cdot,\cdot)$.
- Define $a(u,v) := (Au, v)$ for $u, v \in H$.

Examples:

- System of ODEs: $H = V := (R)^n$.
- Heat equation: $H := H_0^1(\Omega)$, $V := L^2(\Omega)$ and $A := -\Delta$, hence $a(u,v) = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x$.
-

- $0 = t_0 < t_1 < \ldots < t_n = T$.
- $I_k := (t_{k-1}, t_k]$.
- For a function $\varphi : \bigcup_{k=1}^n I_k \to H$ we denote

$$\varphi_k^\pm = \varphi(t_k\pm) := \lim_{t \to t_k\pm} \varphi(t), \quad [\varphi]_k := \varphi(t_k+) - \varphi(t_k-).$$

- Space of piecewise polynomial functions of order q with values in H:

$$S_\tau = \{\varphi : [0,T] \to H;\quad (\varphi|_{I_k})(t) = \sum_{j=0}^q \varphi_j t^j, \text{ where } \varphi_j \in H\}.$$

DG formulation

$$u'(t) + Au(t) = f(t) \qquad \Big| \,.\varphi \in S_\tau,\ \int_{I_k} \mathrm{d}t.$$

$$\int_{I_k} (u',\varphi) + a(u,\varphi)\,\mathrm{d}t = \int_{I_k} (f,\varphi)\,\mathrm{d}t.$$

Integrate per partes twice:

$$\int_{I_k} (u',\varphi)\,\mathrm{d}t = \big(u(t_k), \varphi_k^-\big) - \big(\underbrace{u(t_{k-1})}_{=u_{k-1}^-}, \varphi_{k-1}^+\big) - \int_{I_k} (u,\varphi')\,\mathrm{d}t$$

$$= \big(\underbrace{u_{k-1}^+ - u_{k-1}^-}_{=[u]_{k-1}}, \varphi_{k-1}^+\big) + \int_{I_k} (u',\varphi)\,\mathrm{d}t.$$

DG formulation

DG scheme

We seek $U \in S_\tau$ such that $U_0^- := u^0$ and for all $k = 1, \ldots, n$,

$$\int_{I_k} (U',\varphi) + a(U,\varphi)\,\mathrm{d}t + \big([U]_{k-1}, \varphi_{k-1}^+\big) = \int_{I_k} (f,\varphi)\,\mathrm{d}t, \quad \forall \varphi \in S_\tau.$$

- One-step method: Given U_{k-1}^-, we can compute U on I_k.
- Initial condition u^0 is not satisfied exactly, only by penalization.
- Continuity at t_k is not exact, only by penalization.
- For $q = 0$, i.e. piecewise constants w.r.t. time, we define $U|_{I_k} := U^k \in H$, $\varphi := 1.\varphi_0 \in H$. Thus

$$|I_k| a(U^k, \varphi_0) + (U^k - U^{k-1}, \varphi_0) = \Big(\int_{I_k} f\,\mathrm{d}t, \varphi_0\Big), \quad \forall \varphi_0 \in H,$$

which is the implicit Euler method.

Properties

- A-stability.
- Is the method stable for $u'(t) + Au(t) = 0$ with A positive semi-definite?
- Useful for stiff systems.

A-stability

Let $f = 0$ and $a(\varphi,\varphi) \geq \alpha \|\varphi\|_H^2$ for all $\varphi \in H$, $\alpha \geq 0$. Then for all $k = 1, \ldots, n$,

$$\|U_k^-\|^2 + 2\alpha \int_0^{t_k} \|U\|_H^2\,\mathrm{d}t + \sum_{j=0}^{k-1} \big\|[U]_j\big\|^2 \leq \|u^0\|^2.$$

A-stability

Proof:
We take $\varphi := 2U$ on I_{t_k} and zero elsewhere:

$$\underbrace{2\int_{I_k} (U',U)\,\mathrm{d}t}_{(i)} + \underbrace{2\int_{I_k} a(U,U)\,\mathrm{d}t}_{(ii)} + \underbrace{2\big([U]_{k-1}, U_{k-1}^+\big)}_{(iii)} = 0.$$

$$(i) = 2\int_{I_k} \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|U\|^2\,\mathrm{d}t = \|U_k^-\|^2 - \|U_{k-1}^+\|^2,$$

$$(ii) \geq 2\alpha \int_{I_k} \|U\|_H^2\,\mathrm{d}t,$$

$$(iii) = \big\|[U]_{k-1}\big\|^2 + \|U_{k-1}^+\|^2 - \|U_{k-1}^-\|^2.$$

By gathering all the above estimates, one obtains

$$\|U_k^-\|^2 - \|U_{k-1}^-\|^2 + 2\alpha \int_{I_k} \|U\|_H^2\,\mathrm{d}t + \big\|[U]_{k-1}\big\|^2 \leq 0.$$

Sum over all k.

Under certain technical assumptions (A self-adjoint, etc.), the following results hold:

Error estimate

$$\sup_{t\in(0,T)} \|u(t)-U(t)\| \le \tau^{q+1}\Big(\int_0^T \Big\|\frac{\partial^{q+1}u}{\partial t^{q+1}}(s)\Big\|_H^2 \mathrm{d}s\Big)^{1/2}.$$

Nodal superconvergence

$$\max_{k=1,\dots,n} \|u(t_k)-U_k^-\| \le \tau^{2q+1}\Big(\int_0^T \Big\|\frac{\partial^{q}(A^{q+1/2}u)}{\partial t^{q}}(s)\Big\|_H^2 \mathrm{d}s\Big)^{1/2}.$$

Concluding remarks

- One-step implicit scheme.
- Discontinuous piecewise-polynomial approximation.
- Initial condition and continuity enforced weakly by penalization.
- Arbitrary orders of convergence.
- Unconditionally stable for all orders.
- Suitable for stiff ODEs.
- Expensive.

Convective problems

Continuous problem

Let $Q_T := \Omega \times (0,T)$. We seek a function $u : Q_T \to \mathbb{R}$ such that

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = g \quad \text{in } Q_T,$$
$$u|_{\Gamma_D\times(0,T)} = u_D,$$
$$u(x,0) = u^0(x), \quad x \in \Omega.$$

- $f_1,\dots,f_d \in C^1(\mathbb{R})$ are *convective fluxes*. In theoretical work, f_s usually *globally Lipschitz continuous*
- Describes things that flow: fluids, electrons in semiconductors, city traffic, etc.
- linear=advection, nonlinear=convection.
- typical solution contains discontinuities (shock waves etc.).

Continuous problem

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = g$$

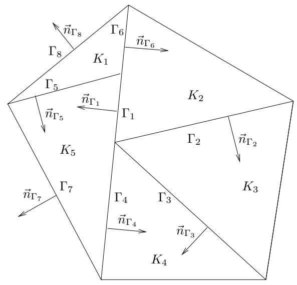
- u is a conserved quantity: Integrate equation over $\widetilde{\Omega} \subset \Omega$,

$$\underbrace{\frac{d}{dt}\int_{\widetilde{\Omega}} u\,\mathrm{d}x}_{(1)} + \underbrace{\int_{\partial\widetilde{\Omega}} \sum_{s=1}^{d} f_s(u) n_s^{(K)}\,\mathrm{d}S}_{(2)} = \underbrace{\int_{\widetilde{\Omega}} g\,\mathrm{d}x}_{(3)}.$$

- (1) = amount of u contained in $\widetilde{\Omega}$,
- (2) = flow of u through boundary of $\widetilde{\Omega}$,
- (3) = sources inside $\widetilde{\Omega}$,

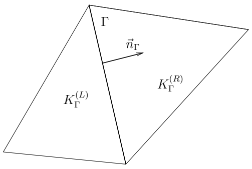
DG formulation

- Let $\mathscr{T}_h$ be a partition of the closure $\overline{\Omega}$ into a finite number of closed triangles $K \in \mathscr{T}_h$.
- By $\mathscr{F}_h$ we denote the set of all edges of $\mathscr{T}_h$. For a given edge $\Gamma \in \mathscr{F}_h$ we define a unit normal $\mathbf{n}_\Gamma$.



For each interior face $\Gamma \in \mathscr{F}_h$ there exist two neighbours $K_\Gamma^{(L)}, K_\Gamma^{(R)} \in \mathscr{T}_h$. We use the convention that $\mathbf{n}_\Gamma$ is the outer normal to the element $K_\Gamma^{(L)}$.

$$
\begin{aligned}
v^{(L)} &= \text{ trace of } v|_{K_\Gamma^{(L)}} \text{ on } \Gamma,\\
v^{(R)} &= \text{ trace of } v|_{K_\Gamma^{(L)}} \text{ on } \Gamma,\\
[v]_\Gamma &= v^{(L)} - v^{(R)},\\
\langle v\rangle_\Gamma &= \tfrac{1}{2}\big(v^{(L)} + v^{(R)}\big).
\end{aligned}
$$



- Over $\mathscr{T}_h$ we define the *broken Sobolev space*

$$H^k(\Omega, \mathscr{T}_h) = \{v; v|_K \in H^k(K)\,\forall K \in \mathscr{T}_h\}$$

- We discretize the continuous problem in the space of discontinuous piecewise polynomial functions

$$S_h = \{v; v|_K \in P_p(K)\,\forall K \in \mathscr{T}_h\},$$

  where $P_p(K)$ is the space of polynomials on K of degree $\leq p$.
- In order to derive a variational formulation, we multiply our equations by a test function $\varphi \in H^1(\Omega, \mathscr{T}_h)$, integrate over some element $K \in \mathscr{T}_h$ and apply Green's theorem.

DG formulation

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = g \quad \Big|\, .\varphi \in S_h, \int_K \mathrm{d}x$$

$$\int_K \frac{\partial u}{\partial t}\varphi\,\mathrm{d}x + \int_{\partial K}\sum_{s=1}^{d} f_s(u)n_s^{(K)}\varphi\,\mathrm{d}S - \int_K \sum_{s=1}^{d} f_s(u)\frac{\partial \varphi}{\partial x_s}\,\mathrm{d}x = \int_K g\varphi\,\mathrm{d}x.$$

Sum over all $K \in \mathscr{T}_h$, rearrange edge terms

$$\int_\Omega \frac{\partial u}{\partial t}\varphi\,\mathrm{d}x + \int_{\mathscr{F}_h}\sum_{s=1}^{d} f_s(u)n_s[\varphi]\,\mathrm{d}S - \sum_{K\in\mathscr{T}_h}\int_K \sum_{s=1}^{d} f_s(u)\frac{\partial \varphi}{\partial x_s}\,\mathrm{d}x = \int_\Omega g\varphi\,\mathrm{d}x.$$

Since u will eventually be replaced by a discontinuous discrete approximation $u_h \in S_h$,we approximate

$$\int_\Gamma \sum_{s=1}^{d} f_s(u)n_s[\varphi]\,\mathrm{d}S \approx \int_\Gamma H(u^{(L)}, u^{(R)}, \mathbf{n})[\varphi]\,\mathrm{d}S.$$

$$\frac{\partial u}{\partial t} + \sum_{s=1}^{d} \frac{\partial f_s(u)}{\partial x_s} = g$$

- Convective form

$$b_h(u,\varphi) = -\sum_{K\in\mathscr{T}_h}\int_K \sum_{s=1}^{d} f_s(u)\frac{\partial \varphi}{\partial x_s}\,dx + \int_{\mathscr{F}_h} H(u^{(L)}, u^{(R)}, \mathbf{n})[\varphi]\,dS.$$

- Right-hand side form

$$\ell_h(\varphi)(t) = \int_\Omega g(t)\,\varphi\,\mathrm{d}x.$$

DG scheme

We seek $u_h \in C^1([0,T]; S_h)$ such that

$$\frac{d}{dt}(u_h(t), \varphi_h) + b_h(u_h(t), \varphi_h) = \ell_h(\varphi_h)(t), \quad \forall \varphi_h \in S_h, \forall t \in (0,T).$$

DG formulation

$$\frac{d}{dt}(u_h(t), \varphi_h) + b_h(u_h(t), \varphi_h) = \ell_h(\varphi_h)(t), \quad \forall \varphi_h \in S_h. \qquad (2)$$

- Take a basis $\mathscr{B} = \{\varphi_\alpha\}_{\alpha=1}^n$ of the space S_h and write

$$u_h(t) = \sum_{\alpha=1}^{n} \xi_\alpha(t)\varphi_\alpha.$$

  Test by $\varphi_h := \varphi_\alpha, \alpha = 1, \ldots, n$. Then (2) is a system of ODEs for unknown functions $\{\xi_\alpha(t)\}_{\alpha=1}^n$. Solve using your favorite method.
- For $p = 0$, DG = finite volume method.

Numerical flux

$$\int_\Gamma \sum_{s=1}^{d} f_s(u)n_s[\varphi]\,\mathrm{d}S \approx \int_\Gamma H(u_h^{(L)}, u_h^{(R)}, \mathbf{n})[\varphi]\,\mathrm{d}S.$$

- We have approximated the *physical flux* $\sum_{s=1}^d f_s(u)n_s$ of quantity u through edge Γ, by a numerical approximation, the so-called *numerical flux* $H(u_h^{(L)}, u_h^{(R)}, \mathbf{n})$.
- Straightforward choices

$$H(u_h^{(L)}, u_h^{(R)}, \mathbf{n}) = \sum_{s=1}^{d} f_s(\langle u_h\rangle)n_s \quad \text{or} \quad \sum_{s=1}^{d} \langle f_s(u_h)\rangle n_s$$

  lead to unstable schemes.
- Averaging is natural for a diffusive problem. For convective problems, information is transported, not diffused.
- In the finite volume method, H uses some information about the flow of u through Γ (characteristics etc.).

Lax-Friedrichs

- For a constant $\lambda > 0$ define

$$H(u_h^{(L)}, u_h^{(R)}, \mathbf{n}) = \underbrace{\sum_{s=1}^{d} \langle f_s(u_h) \rangle n_s}_{H_1} + \underbrace{\lambda \big(u_h^{(L)} - u_h^{(R)}\big)}_{H_2}.$$

- H_1 = 'naive' choice.
- In the resulting formulation, H_2 leads to the term

$$\int_{\mathscr{F}_h} \lambda [u_h][\varphi]\, dS,$$

  which imposes, by penalization, $[u_h] = 0$, i.e. continuity on all edges.
- For a certain choice of λ, this is the famous Lax-Friedrichs numerical flux used in the finite volume method.

Upwinding

$$H(u_h^{(L)}, u_h^{(R)}, \mathbf{n}) = \begin{cases} \sum_{s=1}^{d} f_s(u_h^{(L)}) n_s, & \text{if } A > 0, \\ \sum_{s=1}^{d} f_s(u_h^{(R)}) n_s, & \text{if } A \le 0, \end{cases}$$

where $A = \sum_{s=1}^{d} f_s'(\langle u_h \rangle) n_s$ is the direction of information propagation w.r.t. Γ.
Upwinding can be rewritten as

$$H\big(u_h^{(L)}, u_h^{(R)}, \mathbf{n}\big) = \underbrace{\sum_{s=1}^{d} \langle f_s(u_h) \rangle n_s}_{H_1} + \underbrace{\frac{\mathrm{sgn(A)}}{2} \sum_{s=1}^{d} \big[f_s(u_h)\big] n_s}_{H_2}.$$

H_1 = 'naive' choice, but H_2 leads to the term

$$\sum_{s=1}^{d} \int_{\mathscr{F}_h} \frac{\mathrm{sgn(A)}}{2} \big[f_s(u_h) n_s\big][\varphi]\, dS,$$

which imposes, by penalization, $\big[f_s(u_h) n_s\big] = 0$ on edges, i.e. continuity of fluxes.

Theory

Smooth solutions - Zhang & Shu, V.K.

Let $u, \frac{\partial u}{\partial t} \in L^2(0, T; H^{p+1}(\Omega))$ and $f_s \in C^2(\mathbb{R}), s = 1, \dots, d$. Let H be an *E-flux*. Let $u_h^k, k = 0, 1, \dots$ be the DG solution obtained by the explicit Euler method with the time step restriction $\tau = max_{k=0,1,\dots} \tau_k = O(h^{4/3})$. Then we have the estimate

$$\max_{k=0,1,\dots} \|u(t_k) - u_h^k\|_{L^2(\Omega)} = O(h^{p+1/2} + \tau).$$

General solution - Cockburn & Gremaud

Let u be the exact entropy solution, let H be the Lax-Friedrichs numerical flux along with shock capturing streamline diffusion. Then for compactly supported solutions

$$\|u - u_h\|_{L^\infty(0,T;L^1(\Omega))} = O(h^{1/8}).$$

Final remarks

- Combination of Finite volume and Finite element techniques.
- Higher orders straightforward (unlike FV).
- Works for convective problems (unlike FEM).
- Piecewise polynomial solutions, continuity imposed weakly by penalization (numerical fluxes).
- Basis functions can have a support of one element.
- Local stabilizations.
- Parallelization.
- More degrees of freedom !!!!!

Examples

Flow in GAMM channel, $M_\infty = 0.67$

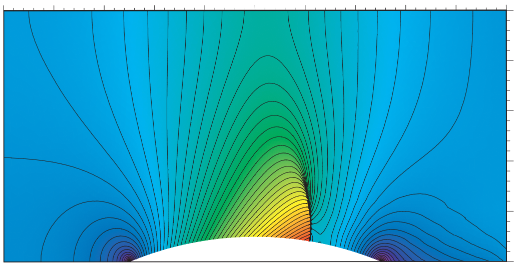


Figure: Mach number isolines.

Flow in GAMM channel, $M_\infty = 0.67$

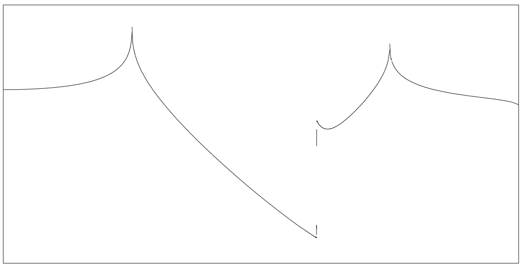


Figure: Densit.

Flow in GAMM channel, $M_\infty = 0.67$

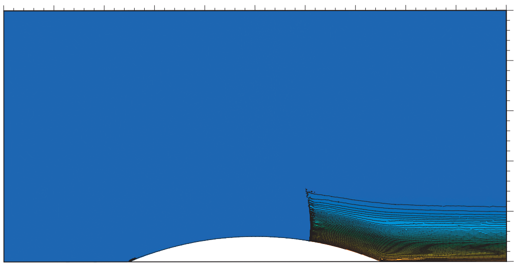


Figure: Entropy isolines.

Flow in GAMM channel, $M_\infty = 0.67$

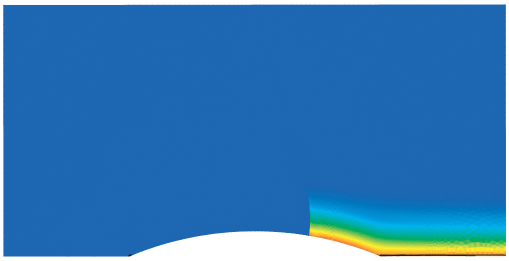


Figure: Entropy.

Flow around cylinder, $M_\infty = 10^{-4}$

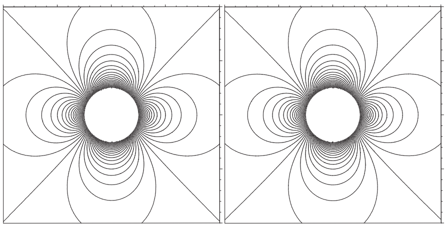


Figure: Velocity isolines of exact and numerical solution, respectively.

Flow around cylinder, $M_\infty = 10^{-4}$

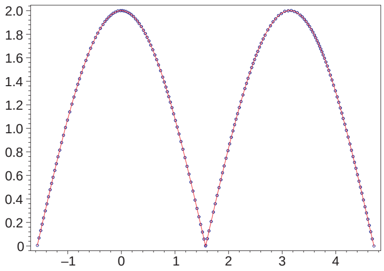


Figure: Velocity distribution on cylinder surface.

Corner eddies near cylinder, $M_\infty = 10^{-4}$

L.E. Fraenkel: On Corner Eddies in Plane Inviscid Shear Flow, 1961

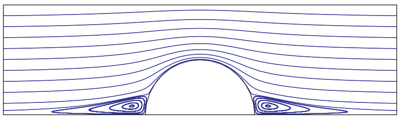


Figure: Exact solution streamlines.

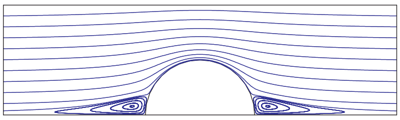


Figure: Approximate solution streamlines.

Corner eddies near cylinder, $M_\infty = 10^{-4}$

L.E. Fraenkel: On Corner Eddies in Plane Inviscid Shear Flow, 1961

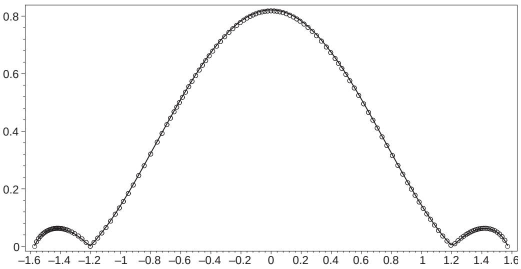


Figure: Velocity distribution on cylinder surface: ∘∘∘ – exact solution of the incompressible equations, —— – numerical solution.

Flow around Zhukovsky profile, low Mach number

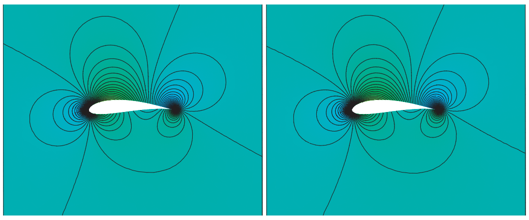


Figure: $M_\infty = 10^{-4}$, velocity isolines: exact solution of incompressible flow (left), numerical solution (right).

Flow around Zhukovsky profile, low Mach number

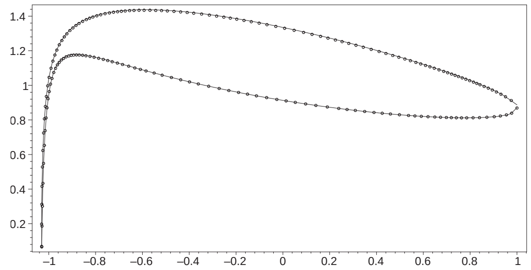


Figure: Distribution of velocity on the profile surface: ∘∘∘ – exact solution of incompressible flow, —— – numerical solution.

Flow around Zhukovsky profile, low Mach number

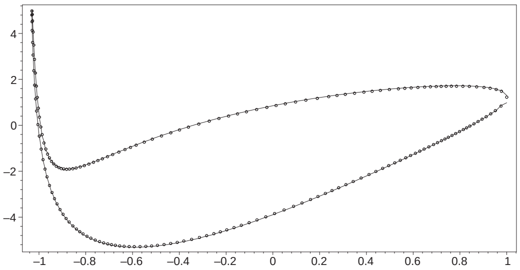


Figure: Distribution of pressure on the profile surface: ∘∘∘ – exact solution of incompressible flow, —— – numerical solution.

Supersonic flow around Zhukovsky profile

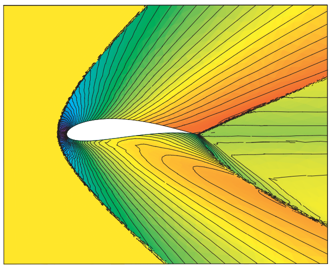


Figure: $M_\infty = 2.0$, Mach number isolines.

NACA 0012 viscous flow

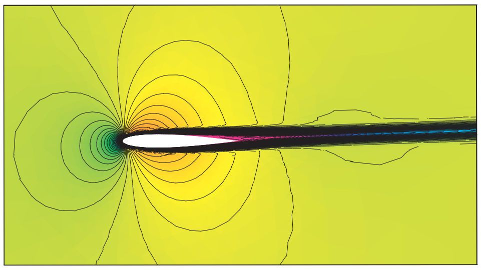


Figure: $M_\infty = 0.5$, $Re = 5000$, $\alpha = 2^\circ$, Mach number isolines.

NACA 0012 viscous flow

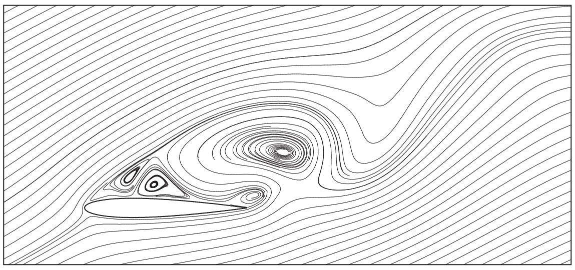


Figure: $M_\infty = 0.5$, $Re = 5000$, $\alpha = 25°$, streamlines

Diffusive problems

Poisson problem

We seek a function $u : \Omega \to (R)$ such that

$$\begin{aligned} -\Delta u &= g \quad \text{in } \Omega, \\ u &= u_D \quad \text{on } \Gamma_D \subset \partial\Omega, \\ \frac{\partial u}{\partial n} &= g_N \quad \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{aligned}$$

- Corresponds to diffusion, smoothing, averaging.
- Ideal for classical FEM.
- Unnatural for DG.

DG formulation

$$-\Delta u = g \quad \big| \,.\varphi \in H^2(\Omega, \mathscr{T}_h),\ \int_K \mathrm{d}x,\ \text{Green}$$

$$\int_K \nabla u \cdot \nabla\varphi \,\mathrm{d}x - \int_{\partial K} \nabla u \cdot \mathbf{n}^{(K)} \varphi \,\mathrm{d}S = \int_K g\varphi \,\mathrm{d}x.$$

Sum over all $K \in \mathscr{T}_h$, rearrange edge terms using $\nabla u = \langle \nabla u \rangle$,

$$\sum_{K\in\mathscr{T}_h} \int_K \nabla u \cdot \nabla\varphi \,\mathrm{d}x - \int_{\mathscr{F}_h^I} \langle \nabla u \rangle \cdot \mathbf{n}[\varphi] \,\mathrm{d}S - \int_{\mathscr{F}_h^D} \nabla u \cdot \mathbf{n}\varphi \,\mathrm{d}S = \int_\Omega g\varphi \,\mathrm{d}x + \int_{\mathscr{F}_h^N} g_N \varphi \,\mathrm{d}S.$$

Similar to classical weak formulation, except for 2^{nd} term. This is due to the discontinuity of φ and u_h.

DG formulation

$$\sum_{K\in\mathscr{T}_h} \int_K \nabla u \cdot \nabla\varphi \,\mathrm{d}x - \int_{\mathscr{F}_h^I} \langle \nabla u \rangle \cdot \mathbf{n}[\varphi] \,\mathrm{d}S - \int_{\mathscr{F}_h^D} \nabla u \cdot \mathbf{n}\varphi \,\mathrm{d}S = \int_\Omega g\varphi \,\mathrm{d}x + \int_{\mathscr{F}_h^N} g_N \varphi \,\mathrm{d}S.$$

- The left-hand side is not symmetric with respect to u, φ, unlike the standard weak formulation.
- The left-hand side is not elliptic with respect to some suitable energy norm.
- There is no means of imposing Dirichlet boundary conditions.
- In the DG method, these requirements are mutually exclusive.

$$\sum_{K\in\mathscr{T}_h} \int_K \nabla u \cdot \nabla\varphi \,\mathrm{d}x - \int_{\mathscr{F}_h^I} \langle \nabla u \rangle \cdot \mathbf{n}[\varphi] \,\mathrm{d}S - \int_{\mathscr{F}_h^D} \nabla u \cdot \mathbf{n}\varphi \,\mathrm{d}S = \int_\Omega g\varphi \,\mathrm{d}x + \int_{\mathscr{F}_h^N} g_N \varphi \,\mathrm{d}S.$$

Fixing symmetry:

- Since $[u] = 0$ on edges, we add the term
  $$-\Theta \int_{\mathscr{F}_h^I} \langle \nabla\varphi \rangle \cdot \mathbf{n}[u] \,\mathrm{d}S.$$
  For $\Theta = 1$, we obtain symmetry w.r.t. 2^{nd} term.
- Since $u = u_D$ on $\partial\Omega$, we add the term
  $$-\Theta \int_{\mathscr{F}_h^D} \nabla\varphi \cdot \mathbf{n}(u - u_D) \,\mathrm{d}S.$$
  For $\Theta = 1$, we obtain symmetry w.r.t. 3^{rd} term AND impose Dirichlet boundary conditions by penalization.
- Such a formulation is symmetric for $\Theta = 1$, but not elliptic. It is semi-elliptic for $\Theta = -1$, but not symmetric.

$$\sum_{K\in\mathscr{T}_h}\int_K \nabla u\cdot\nabla\varphi\,\mathrm{d}x-\int_{\mathscr{F}_h^I}\langle\nabla u\rangle\cdot\mathbf{n}[\varphi]\,\mathrm{d}S-\int_{\mathscr{F}_h^D}\nabla u\cdot\mathbf{n}\varphi\,\mathrm{d}S=\int_\Omega g\varphi\,\mathrm{d}x+\int_{\mathscr{F}_h^N}g_N\varphi\,\mathrm{d}S.$$

Fixing ellipticity and Dirichlet conditions:

- We add the *interior and boundary penalty terms*

$$\int_{\mathscr{F}_h^I}\frac{C_W}{|\Gamma|}[u][\varphi]\,\mathrm{d}S+\int_{\mathscr{F}_h^D}\frac{C_W}{|\Gamma|}(u-u_D)\varphi\,\mathrm{d}S,$$

  where $C_W>0$ is an appropriate constant.
- These terms impose, in a weak sense, continuity of u_h on edges and satisfaction of the Dirichlet BC.
- For C_W large enough, we get ellipticity of the resulting bilinear form.

- *Diffusion form*

$$a_h(u,\varphi)=\sum_{K\in\mathscr{T}_h}\int_K\nabla u\cdot\nabla\varphi\,\mathrm{d}x-\int_{\mathscr{F}_h^I}\langle\nabla u\rangle\cdot\mathbf{n}[\varphi]\,\mathrm{d}S-\int_{\mathscr{F}_h^D}\nabla u\cdot\mathbf{n}\varphi\,\mathrm{d}S$$
$$-\Theta\int_{\mathscr{F}_h^I}\langle\nabla\varphi\rangle\cdot\mathbf{n}[u]\,\mathrm{d}S-\Theta\int_{\mathscr{F}_h^D}\nabla\varphi\cdot\mathbf{n}u\,\mathrm{d}S.$$

- *Interior and boundary penalty form*

$$J_h(u,\varphi)=\int_{\mathscr{F}_h^I}\frac{C_W}{|\Gamma|}[u][\varphi]\,\mathrm{d}S+\int_{\mathscr{F}_h^D}\frac{C_W}{|\Gamma|}u\varphi\,\mathrm{d}S.$$

- *Right-hand side form*

$$\ell_h(\varphi)=\int_\Omega g\varphi\,\mathrm{d}x+\int_{\mathscr{F}_h^N}g_N\varphi\,\mathrm{d}S-\Theta\int_{\mathscr{F}_h^D}\nabla\varphi\cdot\mathbf{n}u_D\,\mathrm{d}S+\int_{\mathscr{F}_h^D}\frac{C_W}{|\Gamma|}u_D\varphi\,\mathrm{d}S.$$

We seek $u_h\in S_h$ such that

$$a_h(u_h,\varphi_h)+J_h(u_h,\varphi_h)=\ell_h(\varphi_h),\quad\forall\varphi_h\in S_h.$$

Properties of the DG formulation

$$a_h(u_h,\varphi_h)+J_h(u_h,\varphi_h)=\ell_h(\varphi_h),\quad\forall\varphi_h\in S_h.$$

- In fact, the specific scheme depends on the choice of Θ:

$$\Theta=\begin{cases}1, & \text{Symmetric variant (SIPG)},\\ 0, & \text{Incomplete variant (IIPG)},\\ -1, & \text{Nonsymmetric variant (NIPG.)}\end{cases}$$

- Despite the complexity of the scheme, we still have consistency:

$$a_h(u,\varphi_h)+J_h(u,\varphi_h)=\ell_h(\varphi_h),\quad\forall\varphi_h\in S_h.$$

- Galerkin orthogonality.

Properties of the DG formulation

Ellipticity and boundedness

There exists a constant $C_W^0>0$, such that if

$$C_W>\begin{cases}2C_W^0, & \text{for the SIPG variant},\\ C_W^0, & \text{for the IIPG variant},\\ 0, & \text{for the NIPG variant},\end{cases}$$

then we have ellipticity and boundedness of $a_h(.,.)+J_h(.,.)$ w.r.t. the DG-norm

$$\|\varphi_h\|_{DG}:=\Big(\tfrac{1}{2}\big(|\varphi_h|^2_{H^1(\Omega,\mathscr{T}_h)}+J_h(\varphi_h,\varphi_h)\big)\Big)^{1/2}.$$

Corollary

$\exists!u_h$.

Properties of the DG formulation

Error estimates

Let $u\in H^{p+1}(\Omega)$. There exists a constant C independent of h, such that

$$\|u-u_h\|_{DG}\le Ch^p|u|_{H^{p+1}(\Omega)}.$$

- Convergence for $u\in H^1(\Omega)$ has been proved only recently (*medius analysis*, T. Gudi 2010).
- Optimal $O(h^{p+1})$ convergence in the $L^2(\Omega)$-norm can be done for SIPG using duality tricks on convex domains.
- For other variants this is an open problem even in 1D (partially answered for IIPG by O. Havle 2010).

Unified approach of Arnold et al. 2002

$$-\Delta u=g\quad\Longleftrightarrow\quad\begin{cases}-\mathrm{div}\chi=g,\\ \nabla u=\chi,\end{cases}$$

- Discretize using DG as a convective problem

$$\sum_{K\in\mathscr{T}_h}\int_K\chi\cdot\nabla\varphi\,\mathrm{d}x-\int_{\mathscr{F}_h}\chi\cdot\mathbf{n}[\varphi]\,\mathrm{d}S=\int_\Omega g\varphi\,\mathrm{d}S,\quad\forall\varphi\in S_h,$$
$$-\sum_{K\in\mathscr{T}_h}\int_K u\,\mathrm{div}\psi\,\mathrm{d}x+\int_{\mathscr{F}_h}u\mathbf{n}\cdot[\psi]\,\mathrm{d}S=\int_\Omega\chi\cdot\psi\,\mathrm{d}x,\quad\forall\psi\in(S_h)^d.$$

- Approximate using numerical fluxes

$$\int_\Gamma\chi\cdot\mathbf{n}[\varphi]\,\mathrm{d}S\approx\int_\Gamma H_\chi(u^{(L)},u^{(R)},\chi^{(L)},\chi^{(R)},\mathbf{n})[\varphi]\,\mathrm{d}S,$$
$$\int_\Gamma u\mathbf{n}\cdot[\psi]\,\mathrm{d}S\approx\int_\Gamma H_u(u^{(L)},u^{(R)},\chi^{(L)},\chi^{(R)},\mathbf{n})[\psi]\,\mathrm{d}S,$$

Unified approach of Arnold et al. 2002

- A natural choice is e.g.

$$\begin{aligned} H_\chi(u^{(L)}, u^{(R)}, \chi^{(L)}, \chi^{(R)}, \mathbf{n}) &:= \langle \nabla u \rangle \cdot \mathbf{n},\\ H_u(u^{(L)}, u^{(R)}, \chi^{(L)}, \chi^{(R)}, \mathbf{n}) &:= \langle u \rangle \mathbf{n}. \end{aligned}$$

  We can eliminate the auxiliary variable χ and obtain SIPG.
- Other choices of H_χ, H_u lead to other variants of the DG method for Poisson's equation.
- Arnold et al. list *nine* basic possibilities, not including IIPG.

Concluding remarks

- DG formulation of $-\Delta u$ is a mess.
- Technical, counterintuitive, leads to very complicated formulations, computationally expensive (more DOFs, nonsymmetric systems, etc.), theory is full of open problems, lots of possible formulations.
- No one should use DG to discretize Poisson's problem.

However...

- It works.
- Sometimes we are forced to use it.
- If diffusion terms are present in complicated nonlinear convection-dominated problems, where finite elements fail, we need to discretize them using DG along with the rest of the equation.

Mathematics in Image Processing

Michal Šorel et al.
Department of Image Processing
Institute of Information Theory and Automation (ÚTIA)
Academy of Sciences of the Czech Republic

Image processing and related fields

- Image processing
  - Image restoration (denoising, deblurring, SR)
  - Computational photography (includes restoration)
  - Segmentation
  - Registration
  - Pattern recognition
- Computer vision – recognition and 3D reconstruction but growing overlap with image processing
- Machine learning
- Compressive sensing (also sub-field of computational photography)

Image reconstruction (inverse problems)

- Denoising
- Deblurring
- Tomography

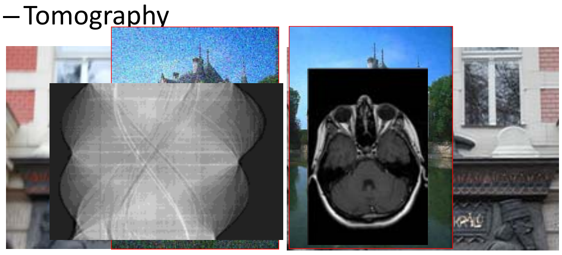


Image segmentation and classification

- Separating objects, categories, foreground/background, cells or organs in biomedical applications etc.

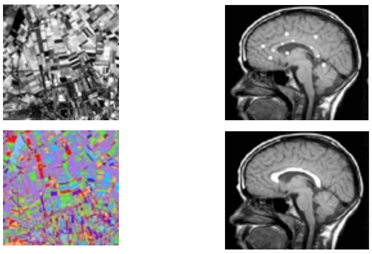
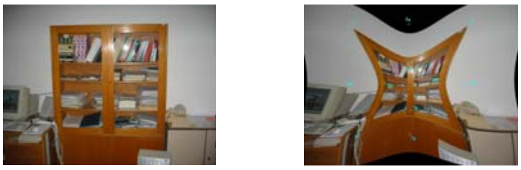
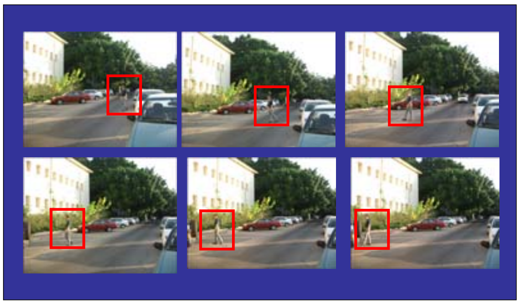


Image Registration

- Transforming different sets of data into one coordinate system
- Transform is constrained to have a specific form (rotation, affine, projective, splines etc.)

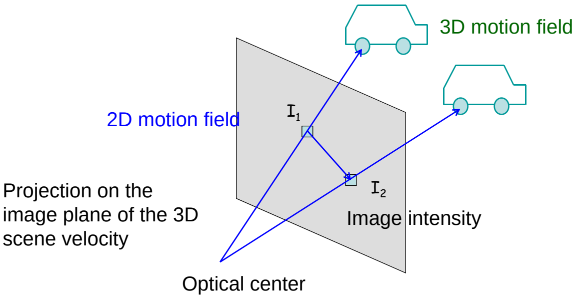


Optical flow



Sequence of images contains information about the scene,
We want to estimate motion – special case of image registration

2D Motion Field = Optical Flow



3D motion field

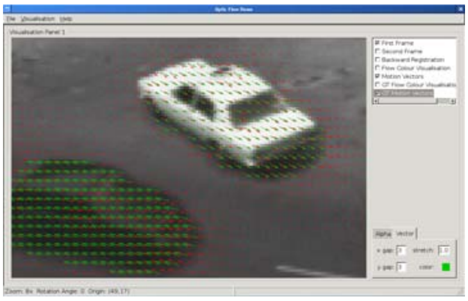
2D motion field

Projection on the image plane of the 3D scene velocity

Image intensity

Optical center

Optical flow example



☑ First Frame
☐ Second Frame
☐ Backward Registration
☐ Flow Colour Visualisation
☑ Motion Vectors
☐ GT Flow Colour Visualisatio

x gap: 1 stretch: 1.0

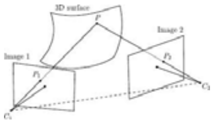
y gap: 1 color: 🟩

Zoom: 8x Rotation Angle: 0 Origin: (49,17)

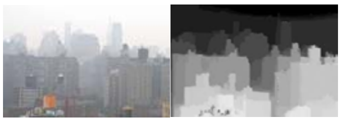
Source: CBIA Brno, http://cbia.fi.muni.cz

Stereo reconstruction

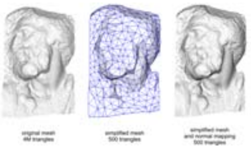
Principle



Result (**depth map** or **disparity map**)



Result (3D model)



Source: http://lcav.epfl.ch

Mathematics in image processing

Mathematics in image processing , CV etc.	My subjective estimate
Linear algebra	90%
Numerical mathematics	70%
Statistics and probability	30%
Analysis (including convex analysis and variational calculus)	used in all above
Graph theory (mainly graph algorithms)	15%
Universal algebra	not much

Probably similar for most engineering fields…

Presentation outline

- Mathematical formulations of image processing problems
- Bayesian view of inverse problems in (not only) image restoration, sparsity
- Discrete labeling problems and Markov random fields (MRF)
  - Surprising result – a large family of non-convex MRF problems can be solved exactly in polynomial time/ reformulated as convex optimization problems

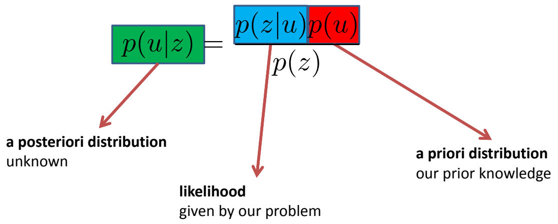
Image

- Greyscale image
  - Continuous representation – 2D function
  - Discrete – matrix
  - Both can be extended to 3D
- Color image = set of 3 or more greyscale images
  - RGB channels are highly correlated → many algorithm work with greyscale only

Inverse problems in image restoration

- Denoising
- Deconvolution and deblurring
- Super-resolution
- JPEG Decompression
- CT, MRI, PET etc. reconstruction (reconstruction from projections)

Bayesian Paradigm



$$p(u|z) = \frac{p(z|u)p(u)}{p(z)}$$

a posteriori distribution
unknown

likelihood
given by our problem

a priori distribution
our prior knowledge

- z … observation, u … unknown original image
- Maximum a posteriori (MAP): max p(u|z)
- Maximum likelihood (MLE): max p(z|u)

MAP corresponds to regularization

$$\max_u p(u|z) \propto p(z|u)p(u)$$

$$\min_u -\log p(u|z) \propto -\log p(z|u) - \log p(u)$$

data term

regularization term

Data term for image denoising

$$p(u|z) \propto p(z|u)p(u)$$

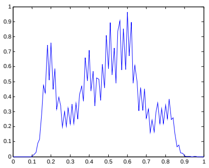
$$-\ln p(u|z) = -\ln p(z|u) - \ln p(u)$$

$$-\ln p(z|u) = -\ln k \prod_i e^{\frac{(z_i-u_i)^2}{2\sigma^2}} = \frac{1}{2\sigma^2}\sum_i (z_i - u_i)^2 + c$$

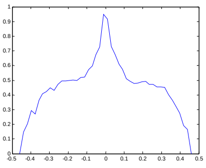
$$n \ldots N(0,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \prod_{i=1}^{N} e^{\frac{n_i^2}{2\sigma^2}}$$

Image Prior

$$\ln p(\mathbf{u}) = \ln \prod_i p(\mathbf{u}_i) = \sum_i \ln p(\mathbf{u}_i)$$

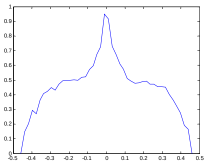


Intensity histogram



Gradient histogram

Image Prior



Gradient histogram

$$p(\mathbf{u}) \propto \prod_i e^{-\lambda\phi(\nabla u_i)}$$

$$\ln p(\mathbf{u}) = -\lambda \sum_i \phi(\nabla u_i) + c$$

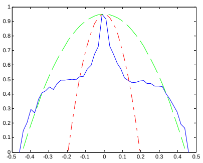
Theory on when we can do this will be given later (CRF)

Image Prior

$$Q(u) = \lambda \int |\nabla u|^2$$

Tikhonov regularization

$$p(\mathbf{u}) \propto \prod_i e^{-\lambda |\nabla u_i|^2} = e^{-\lambda \mathbf{u}^T \mathbf{L} \mathbf{u}}$$



$$Q(u) = \lambda \int |\nabla u|$$

TV regularization

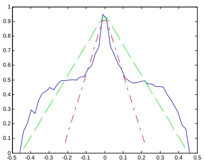
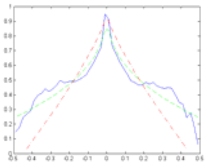


Image Prior

$$Q(u) = \lambda \int |\nabla u|^{0.8}$$

$$Q(u) = \lambda \int |\nabla u|^{0.4}$$

Non-convex regularization



Bayesian MAP approach for denoising

$$-\ln p(u|z) = -\ln p(z|u) - \ln p(u)$$

$$\frac{1}{2\sigma^2} \sum_i (z_i - u_i)^2 + \lambda \sum_i \phi(|\nabla u_i|)$$

$$\min_{\mathbf{u}} \frac{1}{2} \sum_i (z_i - u_i)^2 + \lambda \sum_i \phi(|\nabla u_i|)$$

Sparsity

- TV regularization can be extended to other sparse representations

$$\min_u \frac{1}{2} \|z - u\|^2 + \lambda \|\nabla u\|_{2,1}$$

$$\min_u \frac{1}{2} \|z - u\|^2 + \lambda \|W u\|_1$$

- W often a set of convolutions with highpass filters
  - Wavelets
  - Learned by PCA

Measure of Sparsity

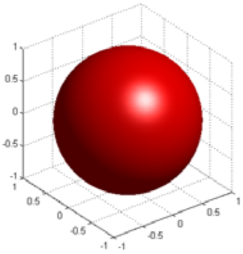
- $l_p, \quad 0 < p \le 1$ norms ($\|\mathbf{a}\|_p^p$)

$$\|\mathbf{a}\|_p = \left(\sum_i |a_i|^p \right)^{\frac{1}{p}}$$

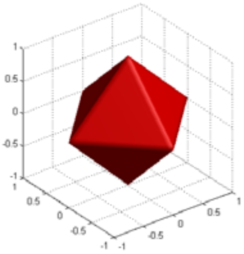
- l_0 norm, counts nonzero elements

- many other sparsity measures
  - smooth l_1 $\quad \rho(\mathbf{a}) = \|\mathbf{a}\|_1 - \epsilon \log \left(1 + \frac{\|\mathbf{a}\|_1}{\epsilon} \right)$

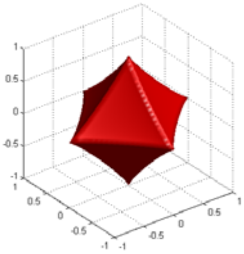
l_2 unit ball



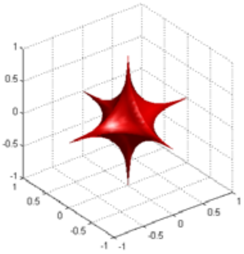
l_1 unit ball



$l_{0.9}$ unit ball

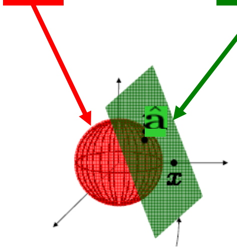


$l_{0.5}$ unit ball



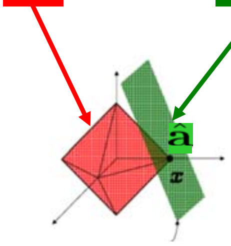
l_2-norm

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\mathbf{a}\|_2^2 \quad \text{subject to} \quad \mathbf{Da} = \mathbf{x}$$



l_1-norm

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\mathbf{a}\|_1 \quad \text{subject to} \quad \mathbf{Da} = \mathbf{x}$$



Deblurring

- Denoising $\quad z = u + N(0, \sigma^2 I)$

$$\min_u \frac{1}{2}\|z - u\|^2 + \lambda \|\nabla u\|_{2,1}$$

- Deblurring

$$z = h * u + N(0, \sigma^2 I) = Hu + N(0, \sigma^2 I)$$

$$\min_u \frac{1}{2}\|z - Hu\|^2 + \lambda \|\nabla u\|_{2,1}$$

Super-resolution (with deblurring)

Several possibly shifted blurred images

$$z_i = DH_i u + N(0, \sigma^2 I)$$

$$\min_u \frac{1}{2} \sum_i \|z_i - DH_i u\|^2 + \lambda \|\nabla u\|_{2,1}$$

D_i … downsampling operator

Convolutions represent also the shift

Optical flow

- Based on the assumption of constant brightness

$$I(t, x(t)) = I(t_0, x(t_0))$$

$$\frac{dx}{dt} \cdot \nabla I + \frac{\partial I}{\partial t} = 0 \text{ at } t = t_0$$

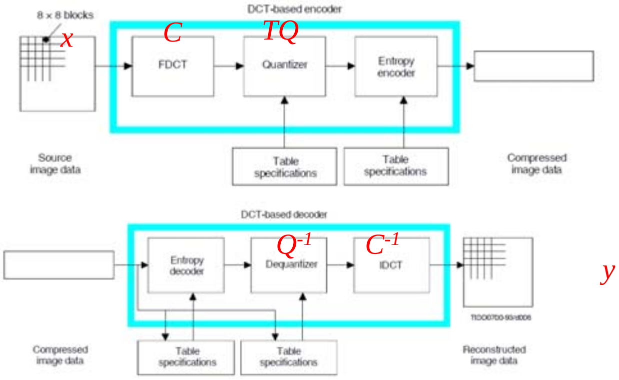
- Optical flow is the velocity field

$$\mathbf{v}(t_0) = \frac{d\mathbf{x}}{dt}(t_0)$$

Optical flow

$$\min_{\mathbf{v}} \underbrace{\frac{1}{2} \int_\Omega (\nabla I \cdot \mathbf{v} + I_t)^2 dx}_{\text{Data term}} + \underbrace{\lambda}_{\text{Weighting parameter}} \underbrace{\sum_{i=1}^{2} \|\nabla \mathbf{v}_i\|_{2,1}}_{\text{Regularization term}}$$

JPEG compression



Bayesian MAP restoration

MAP – maximum a posteriori probability

$$\max_x p(x|y) \propto p(y|x)p(x)$$

$$\min_x -\log p(y|x) - \log p(x)$$

$$-\log p(x) = \tau \left\|\Phi^T x\right\|_1$$

$$-\log p(y|x) = \begin{cases} 0 & for\ QCx \in \langle QCy - 0.5, QCy + 0.5\rangle \\ \infty & otherwise \end{cases}$$

Bayesian JPEG decompression

Using total variation (TV)

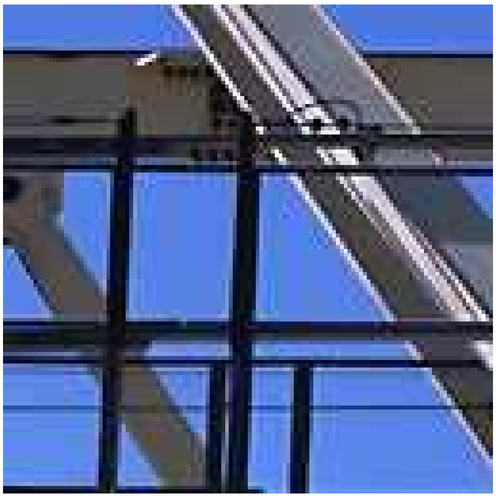
$$\min_x \|\nabla x\|_{2,1} \ s.t. \ \ QCx \in \langle QCy - 0.5, QCy + 0.5\rangle$$

(Bredies and Holler, 2012)

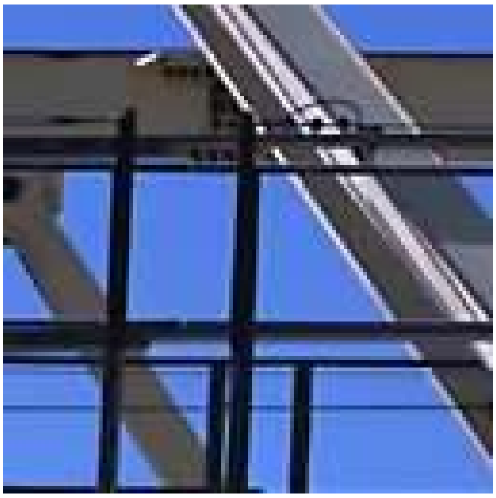
Or using redundant wavelets

$$\min_x \|\Phi^T x\|_1 \ s.t. \ \ QCx \in \langle QCy - 0.5, QCy + 0.5\rangle$$

50
jpg



50
est

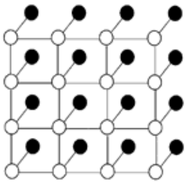


Convex variational problems

- Denoising, deblurring, SR, optical flow, JPEG decompression …
- Solution by convex optimization (interior point, proximal methods)
- What to do for discrete or non-convex problems?

Discrete labeling problems

- For each site (pixel) we look for a label (or a vector of labels)
- Labels depend on local image content and a smoothness constraint
- Image restoration, segmentation, stereo, and optical flow are all labeling problems



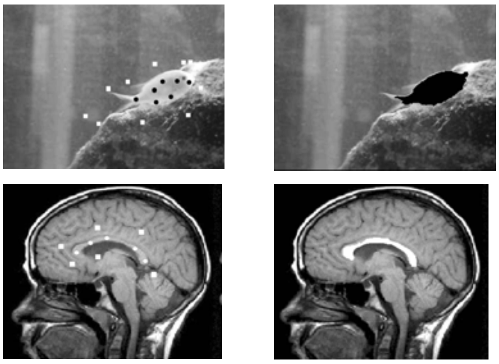
Discrete labeling problems

- For each site (pixel) we look for a label (or a vector of labels)
- Labels depend on local image content and a smoothness constraint

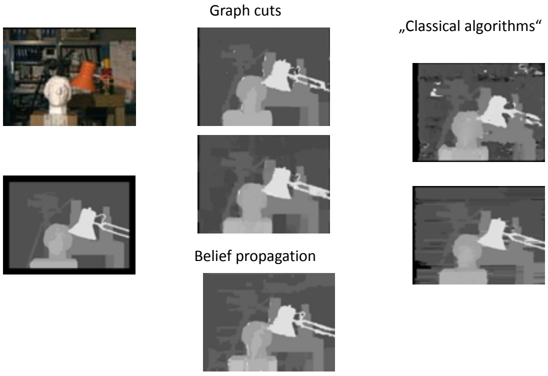
Segmentation	foreground/background
	or object number
		{0,1}
		{1.. k}

Stereo	Disparity (inverse depth)	-k..k
Optical flow	local motion	(-k..k) x (-k..k)
Restoration	Intensity	0..255

Segmentation



Graph cuts & Belief propagation



Markov Random Fields (MRFs)

- Markov Random Field
- Gibbs Random Field
- MRF ⇔ GRF (Hammersley-Clifford theorem)
- Smoothness priors
- MRF models in
  - stereo
  - segmentation
  - restoration (denoising, deblurring)

Markov Random Field (MRF)

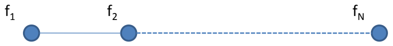
- sites S = {1, ... , m}
- F ... set of random variables defined on S
- N ... **neighborhood system**
- $f_i \in \mathcal{L}$ ... (possibly discrete) label
- **configuration** f = {f_1 ... f_K},

$$P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i})$$

$$P(f) > 0$$

- Other (possible) properties – homogeneity, isotropy

MAP in a chain



$$P(f) = \frac{1}{Z} \prod_{i \in \mathcal{S}} \varphi_i(f_i) \prod_{i \in \mathcal{S}} \prod_{i' \in \mathcal{N}_i} \psi_{i,i'}(f_i, f_{i'})$$

- [illegible] n)

$$\max_f p(f) = \max_{f_1} \max_{f_2} \cdots \max_{f_N} p(f_1, f_2, \ldots, f_N)$$

$$\max_f p(f) = \frac{1}{Z} \max_{f_1} \phi_1(f_1) \max_{f_2} \psi_{12}(f_1, f_2) \phi_2(f_2) \max_{f_3} \cdots$$

Gibbs Random Field

$$P(f) = \frac{1}{Z} e^{-\frac{1}{T} U(f)}$$

P(f) > 0 !

Partition function $$Z = \sum_f e^{-\frac{1}{T} U(f)}$$

Energy function U(f)

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f) = \sum_{i \in \mathcal{S}} V_1(f_i) + \sum_{i \in \mathcal{S}} \sum_{i' \in \mathcal{N}_i} V_2(f_i, f_{i'})$$

V_c(f) ... clique potentials

Hammersley-Clifford theorem

MRF = GRF

F is an MRF on S with respect to N

if and only if

F is a Gibbs random field on S with respect to N

MRF ... conditional independence of non-neighbor nodes (variables)

GRF ... global function depending on local “compatability functions”

Hammersley-Clifford theorem - proof

- An MRF is also a GRF – complicated, introduction of cano $P(f_i|f_{S-\{i\}}) = P(f_i|f_{N_i})$ d
- A GRF is a MRF

$$P(f_i|f_{S-\{i\}}) = \frac{P(f)}{\sum_{f_i\in\mathcal{L}} P(f')} = \frac{e^{-\sum_{c\in\mathcal{C}} V_c(f)}}{\sum_{f'_i} e^{-\sum_{c\in\mathcal{C}} V_c(f)}}$$

$$P(f_i|f_{S-\{i\}}) = \frac{e^{-\sum_{\{c,i\in c\}} V_c(f)}}{\sum_{f'_i} e^{-\sum_{\{c;i\in c\}} V_c(f)}}$$

MRF = GRF

- MAP-MRF

$$\max_f p(f) = \frac{1}{Z}e^{-E(f)}$$

$$\min_f(-\ln p(f)) = \min_f E(f) + const$$

- How to incorporate smoothness?
  - Penalties/potentials similar for most applications

Smoothness prior

Priors on derivatives, usually first derivative

$$V(f_i, f_j) = \kappa_{ij}\;\delta(f_i - f_j)$$ segmentation, sometimes in stereo

$$V(f_i, f_j) = \kappa_{ij}\;(f_i - f_j)^2$$ Tikhonov regularization

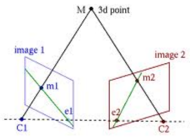
Discontinuity preserving penalties

$$V(f_i, f_j) = \kappa_{ij}\;|f_i - f_j|$$ TV regularization

$$V(f_i, f_j) = \kappa_{ij}\min((f_i - f_j)^2, const)$$ line process, Mumford-Shah functional

MAP-MRF for stereo (Boykov & al.)

2 images d^1,d^2 on the input



$$E(f) = \sum_i V_1(f_i, d^1, d^2) + \kappa \sum_i \delta(f_{i+1} - f_i)$$

Birchfield-Tomasi matching cost – insensitivite to sampling:

$$V_1(f_i, d^1, d^2) = \min(\min_{\triangle\in(f_i-\frac{1}{2}, f_i+\frac{1}{2})} |d_i^1 - d_{i+\triangle}^2|, \ldots, const)^2$$

MAP-MRF for segmentation

- " "GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts", C. Rother, V. Kolmogorov, A. Blake, SIGGRAPH 2004

MAP-MRF for segmentation

- "Grab cut" example

$$V_2(f_i, f_j) = \kappa_{ij}\;\delta(f_i - f_j) = \gamma e^{-\frac{||d_i - d_j||^2}{2\sigma^2}}\delta(f_i - f_j)$$

$$V_1(f_i, d_i) \cong -\ln p(f_i|d_i) \cong -\ln p(d_i|f_i) - \ln p(f_i)$$

$V_1(f_i,d_i)$ ~ probability to be in fg/bg based on a feature space (intensities, texture features etc...) – modeled for example as a mixture of Gaussians

MAP-MRF for restoration

- Denoising (with anisotropic TV regularization)
  - 2D indexing - only this slide

$$E(f) = \frac{1}{\sigma^2}\sum_{ij}(f_{ij}-d_{ij})^2 + \kappa\sum_{ij}|f_{i+1,j}-f_{ij}| + \kappa\sum_{ij}|f_{i,j+1}-f_{ij}|$$

- Deblurring (with TV regularization)

$$E(f) = \frac{1}{\sigma^2}||f*h-d||^2 + \kappa\sum_{ij}|f_{i+1,j}-f_{ij}| + \kappa\sum_{ij}|f_{i,j+1}-f_{ij}|$$

- Discrete methods not efficient for restoration!

MRFs - Summary

- Common framework for many image processing a CV problems
- MAP-MRF approach results in similar functionals
- MRF = GRF

MAP-MRF using graph cuts

- MAP – Maximum a posteriori probability

$$\max_f p(f) = \frac{1}{Z}e^{-E(f)}$$

$$\min_f(-\ln p(f)) = \min_f E(f) + const$$

- Graph cuts = min-cut ~ max-flow (Ford-Fulkerson theorem)
- Much better than simulated annealing based methods, often very close to global optimum

Graph cuts minimization

$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$

For $V_2 \geq 0$ metric

- $V_2(a,b) = 0 \Leftrightarrow a = b$
- $V_2(a,b) = V_2(b,a)$ (actually not necessary)
- $V_2(a,b) \leq V_2(a,c) + V_2(c,b)$

or semimetric (without Δ-inequality)

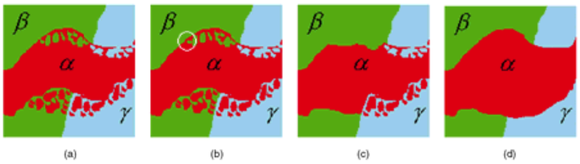
Metric: $\delta(f_i - f_j)$
$\min(|f_i - f_j|, const)$ for any norm |.|

Semimetric: $\min((f_i - f_j)^2, const)$

Graph cuts minimization

- Local minimization – minimum if no possible decrease of E(f) in one “move”
- Iterated conditional modes (ICM) iteratively minimizes each node (pixel) ➡ easily gets trapped in a local minimum (~ gradient descent)
- Simulated annealing – global moves but without any specific direction ➡ slow
- Graph cuts – use much larger set of “moves” so that the minimum over the whole set can be found in a reasonable (polynomial) time

α-β swap and α-expansion moves



initial labeling

standard move

α-β swap move

α-expansion move

α-expansion algorithm

```
1. Start with an arbitrary labeling $f$
2. Set success := 0
3. For each label $\alpha \in \mathcal{L}$
    3.1. Find $\hat{f} = \arg\min E(f')$ among $f'$ within one $\alpha$-expansion of $f$
    3.2. If $E(\hat{f}) < E(f)$, set $f$ := $\hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return $f$
```

- Arbitrary **metric** $V_2(\alpha,\beta)$ (Δ-inequality)
- Not worse than 2x optimum

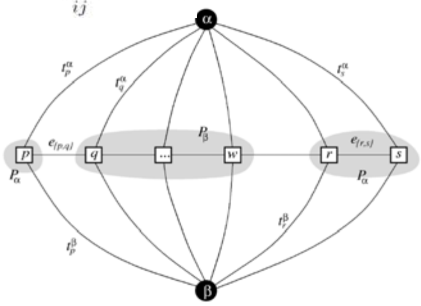
α-β swap algorithm

```
1. Start with an arbitrary labeling $f$
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$
    3.1. Find $\hat{f} = \arg\min E(f')$ among $f'$ within one $\alpha$-$\beta$ swap of $f$
    3.2. If $E(\hat{f}) < E(f)$, set $f$ := $\hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return $f$
```

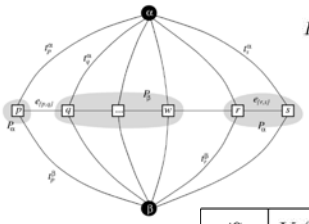
- Arbitrary **semimetric** $V_2(\alpha,\beta)$ (without Δ-inequality)
- No optimality guaranteed

α-β swap move graph

$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$



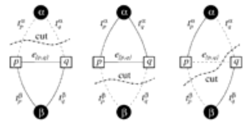
α-β swap move graph



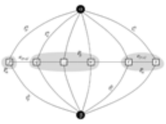
$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$

t_p^α	$V_p(\alpha) + \sum_{\substack{q\in\mathcal{N}_p \\ q\notin\mathcal{P}_{\alpha\beta}}} V(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$V_p(\beta) + \sum_{\substack{q\in\mathcal{N}_p \\ q\notin\mathcal{P}_{\alpha\beta}}} V(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V(\alpha,\beta)$	$\{p,q\}\in\mathcal{N}$ $p,q\in\mathcal{P}_{\alpha\beta}$

α-β swap move graph

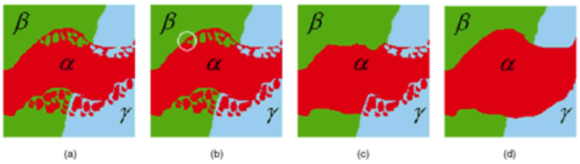


$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$



t_p^α	$V_p(\alpha) + \sum_{\substack{q\in\mathcal{N}_p \\ q\notin\mathcal{P}_{\alpha\beta}}} V(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$V_p(\beta) + \sum_{\substack{q\in\mathcal{N}_p \\ q\notin\mathcal{P}_{\alpha\beta}}} V(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V(\alpha,\beta)$	$\{p,q\}\in\mathcal{N}$ $p,q\in\mathcal{P}_{\alpha\beta}$

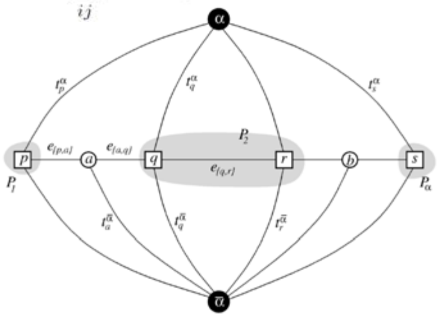
α-β swap - summary



- We know how to transform minimization of E(f) over all possible α-β swap moves to graph cut problem

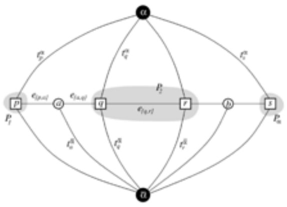
α-expansion move graph

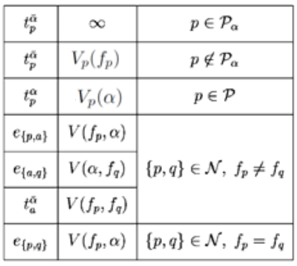
$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$



α-expansion move graph

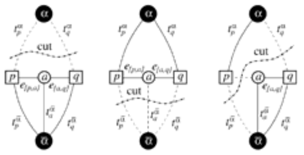
$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$



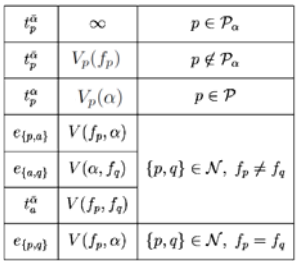


$t_p^{\bar\alpha}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar\alpha}$	$V_p(f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$V_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$V(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p \neq f_q$
$e_{\{a,q\}}$	$V(\alpha, f_q)$	
$t_a^{\bar\alpha}$	$V(f_p, f_q)$	
$e_{\{p,q\}}$	$V(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p = f_q$

α-expansion graph - cuts



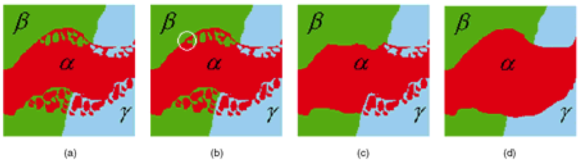
$$E(f) = \sum_i V_1(f_i) + \sum_{ij} V_2(f_i, f_j)$$



$t_p^{\bar\alpha}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar\alpha}$	$V_p(f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$V_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$V(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p \neq f_q$
$e_{\{a,q\}}$	$V(\alpha, f_q)$	
$t_a^{\bar\alpha}$	$V(f_p, f_q)$	
$e_{\{p,q\}}$	$V(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p = f_q$

Δ - inequality !

α-expansion - summary



- We know how to transform minimization of E(f) over all possible α-expansion moves to graph cut problem
- What remains? - how to find the minimum cut

Graph cuts algorithm

- “Augmenting path” type algorithm with simple heuristics
  - Looks for a non-saturated path ~ path in residual graph
  - Simultaneously builds trees from α and β
- Maximum complexity O(n^2mC_{max}), C_{max} cost of the minimum cut
- Actually typically linear with respect to the number of pixels
- On our problems faster than good combinatorial algorithms - Dinic O(n^2m), Push-relabel O($n^2\sqrt{m}$)

Graph cuts - summary

- Minimization of E(f) by finding min-cut in a graph in polynomial time

2 label minimization can be done in polynomial (and typically linear) time with respect to the number of pixels

- K>2 labels – NP hard
  - Equivalent to Multiway Cut Problem
  - α-expansion finds a solution ≤ 2*optimum
  - In practice both α-β swap and α-expansion algorithms get very close to global minimum

Graph cuts – additional example

- " "GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts", C. Rother, V. Kolmogorov, A. Blake, SIGGRAPH 2004

Discrete optimization in MRFs - summary

- Conditional independence is strong structural information that can be exploited
- Gives useful approximations for difficult (NP-hard) problems
- For convex problem mostly better to use continuous methods

References

- Graph Cuts
  - "Fast Approximate Energy Minimization via Graph Cuts" - Y. Boykov, O. Veksler, R. Zabih, PAMI 2001 (Augmenting path min-cut algorithm)
  - "An Experimental Comparison of Min-Cut/Max-flow Algorithms for Energy Minimization in Vision" – Y. Boykov, V. Kolmogorov, PAMI 2004 (Graph construction for α-β swap and α-expansion moves)
  - " "GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts", C. Rother, V. Kolmogorov, A. Blake, SIGGRAPH 2004
- Belief propagation
  - "Understanding Belief Propagation and its Generalizations" - J.S. Yedidia, W.T.Freeman, Y.Weiss (Mitsubishi electric research laboratories, Technical report, 2002)

Title: Seminar on Numerical Analysis & Winter School
Proceedings of the Conference SNA'14
Nymburk
January 27 – 31, 2014

Editors: Hana Bílková, Miroslav Rozložník, Petr Tichý

Published by: Institute of Computer Science AS CR, Prague

Printed by: Tigris spol. s r. o., Nábřeží 599, 760 01 Zlín – Prštné

First edition
Praha, 2014
copies 70

ISBN 978-80-87136-16-4