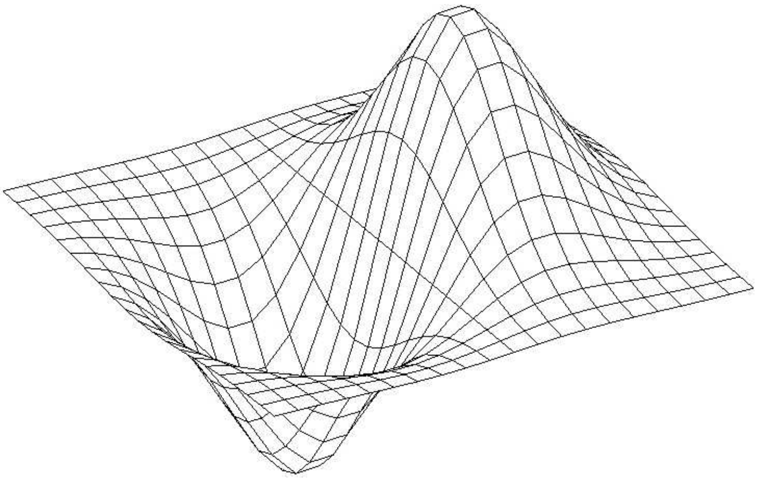
Institute of Geonics AS CR, Ostrava

SNA'11

Seminar on Numerical Analysis

Modelling and Simulation
of Challenging Engineering Problems



Winter School

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

Rožnov pod Radhoštěm, January 24 – 28, 2011

Programme committee:

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

Organizing committee:

Hana Bílková	Institute of Computer Science AS CR, Prague
Radim Blaheta	Institute of Geonics AS CR, Ostrava
Eva Dudková	Institute of Geonics AS CR, Ostrava
Jiří Starý	Institute of Geonics AS CR, Ostrava

Conference secretary:

Jaroslava Vávrová	Institute of Geonics AS CR, Ostrava

Institute of Geonics AS CR

ISBN 978-80-86407-19-7

Preface

Seminar on Numerical Analysis 2011 (SNA'11) is the eighth meeting in a series of events started in Ostrava 2003 and devoted to numerical methods necessary for mathematical modelling of problems in sciences and engineering. For the first time SNA'11 will be held in Rožnov pod Radhoštěm, a beautiful town with many attractions and friendly Beskydy mountains surrounding.

Since 2005, a part of SNA has been devoted to the so-called Winter school with tutorial lectures devoted to selected topics within the conference scope. In this year, the school part includes invited lectures devoted to operator splitting techniques for mutiphysics problems (Axelsson), scalable FETI algorithms for contact problems (Dostál, Kozubek, Vondrák, Brzobohatý, Markopoulos), ill posed problems in image processing (Hnětynková, Plešinger, Strakoš), principles of algebraic multigrid based on smoothed aggregations (Vaněk) and analysis and numerical approximation of non-local damage mechanics models (Zeman, Mielke, Roubíček).

The Winter school is complemented by contributed lectures devoted to many topics as aggregation based methods, computational mechanics, domain decomposition, efficient iterative solvers, finite element method, formulation of mathematical models, modelling of transport problems, parallel computations, etc.

We would like to wish SNA'11 to be, similarly to the previous SNA meetings, a fruitful event, providing interesting lectures, showing new ideas and starting or strengthening collaboration and friendship.

On behalf of the Programme and Organizing Committee of SNA'11,

Radim Blaheta and Jiří Starý

Contents

Winter school lectures

O. Axelsson
Operator splittings for solving nonlinear coupled multiphysics problems with an application for interface modeling

Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, A. Markopoulos
Scalable FETI based algorithms for contact problems: theory, implementation, and numerical experiments

I. Hnětynková, M. Plešinger, Z. Strakoš
Ill-posed inverse problems in image processing: introduction, structured matrices, spectral filtering, regularization, noise revealing

P. Vaněk:
Základy algebraického multigridu založeného na zhlazených agregacích

J. Zeman, A. Mielke, T. Roubíček
Analysis of a rate-independent model of non-local damage and its numerical approximation

An overview of aggregation techniques for two-level methods

R. Blaheta, V. Sokol

Institute of Geonics AS CR, Ostrava
VŠB - Technical University of Ostrava

1 Introduction

This paper is an effort to do an overview of aggregation techniques and compare their efficiency on model problem with heterogeneity when used for construction of coarse space for two-level Schwarz method. Aggregation techniques are usually used in context of multilevel and multigrid methods for construction of coarse levels. Initially coarse levels were obtained from hierarchy of meshes with different discretization parameters. Aggregation overcomes the need for this hierarchy of meshes and needs little or no information besides the matrix of the problem to be solved.

Aggregation techniques presented in this paper can be divided into two groups: node-wise and element-wise aggregations. The first group is somewhat larger and widely used, most likely because node-wise aggregations don't need any information about the mesh used for discretization of the method.

2 Aggregations

In this paper for the sake of simplicity we will restrict ourselves to the case of two-level methods only, multilevel methods can be devised by recursive use of the two-level scheme.

2.1 Two-level method with aggregation

Let us consider a problem discretized on triangulation $\mathcal{T}_h$ by finite element method and described by the linear system

$$A_h u_h = b_h, \tag{1}$$

solved by a two-level method. One iteration of two-level method is described in Algorithm 1. On lines 5 and 10 there are k_1 and k_2 steps of pre-smoothing and post-smoothing respectively by operator S, usually realized by one iteration of Gauss-Seidel or Jacobi method. In the case of classical multigrid methods, the prolongation operator P and restriction operator R are naturally induced by the hierarchy of triangulations $\mathcal{T}_h$ and $\mathcal{T}_H$ and the matrix A_H corresponds to the discretization on $\mathcal{T}_H$. However in the case of algebraic multigrid methods, the prolongation and restriction operators and the coarse space matrix A_H are created only by using a little information besides the matrix A_h thus avoiding the need to construct hierarchy of nested meshes.

Algorithm 1 One iteration of two-level method

```
**Input:**$A_h, b_h, u_h^i$
**Output:**$u_h^{i+1}$
$u = u_h^i$
**for** $j = 1$ **to** $k_1$ **do**
    $u = S(A_h, b_h, u)$
**end for**
$r_H = R\,(b_h - A_h u)$
$u = u + P\left(A_H^{-1} r_H\right)$
**for** $j = 1$ **to** $k_2$ **do**
    $u = S(A_h, b_h, u)$
**end for**
$u_h^{i+1} = u$
```

Aggregation technique divides set of unknowns $N = \{1, \dots, n\}$ into disjoint subsets C_i of aggregates of unknowns, so that $N = \bigcup_{i=1}^{k} C_i$, $C_i \bigcap_{i \neq j} C_j = \emptyset$. Then the prolongation and restriction operators are defined $R = \mathcal{R}$, $P = \mathcal{R}^T$ by boolean matrix $\mathcal{R}$:

$$(\mathcal{R})_{i,j} \begin{cases} = 1 & \text{if} \quad j \in C_i \\ = 0 & \text{otherwise} \end{cases} \tag{2}$$

2.2 Node-wise aggregations

In this subsection we focus on aggregation techniques that exploits the information directly stored in the matrix A_h, these include algorihms by Vaněk et al. [3], by Scheichl and Vainikko [2] and by Notay [1]. The aggregation algorithm by Notay was primarily designed to work with algebraic multilevel scheme based on a block approximate factorization of matrix, however it can also be used for algebraic multilevel methods. The algorithm firstly defines set of nodes S_i, to which node i is strongly negative connected:

$$S_i(\varepsilon) = \left\{ j \in N : j \neq i, a_{ij} < -\varepsilon \max_{a_{ik}<0} |a_{ik}| \right\}, \tag{3}$$

where parameter ε is used as threshold for strong coupling. The sets $S_i(\varepsilon)$ are used co construct pairs of nodes that are most strongly negative connected, and then used recursively for those pairs (and possibly few singletons) to create generalized quadruplets.

The algorithm by Vaněk et al. starts by defining strongly-connected neighborhood similar to (3) with thresholding parameter ε:

$$S_i(\varepsilon) = \left\{ j \in N : |a_{ij}| \geq \varepsilon \sqrt{a_{ii} a_{jj}} \right\}, \tag{4}$$

and then separates nodes that are not strongly connected to any other nodes. These nodes are isolated from others and are not aggregated. Rest of the nodes is used for initial covering by tentative aggregates C_i, the remaining nodes that does not belong to tentative aggregates forms set R. The main part of the algorithm can be described as follows:

step 1: enlarge aggregates C_i
move node j from R to aggregate C_i if there is strong connection

```
step 2: process unaggregated nodes
    create new aggregates: $C_i = S_j(\varepsilon) \cap R$, $R = R \setminus C_i$
```

Given this aggregation, tentative prolongation is created from (2), which can be further smoothed to get the final prolongation and restriction operators. To get the smoothed prolongation operator, simple damped Jacobi smoother was proposed in the form

$$P_s = (I - \omega\, (\mathrm{diag} A_h)^{-1} A_F) P\,, \tag{5}$$

where ω is damping parameter and A_F is filtered matrix.

The last aggregation of this subsection is that of Scheichl and Vainikko. The algorithm again starts by defining strongly connected nodes. Node j is strongly connected to i if the following condition is satisfied:

$$\left|\hat{A}_{ij}\right| \geq \varepsilon \max_{k \neq i} \left|\hat{A}_{ik}\right|\,, \tag{6}$$

where $\hat{A} = (\mathrm{diag} A_h)^{-\frac{1}{2}} A_h\, (\mathrm{diag} A_h)^{-\frac{1}{2}}$ and ε is again thresholding parameter for strong connection. To create set of aggregates $\{C_i\}$ strongly-connected graph r-neighborhood $S_{r,\varepsilon}(i)$ is used. $S_{r,\varepsilon}(i)$ is set of node i and all nodes j for which there exists a path of length r of strongly-connected nodes to node i. The algorithm creates aggregates by finding strongly-connected graph r-neighborhood of chosen seed node. To choose a good seed node advancing front in the graph induced by nodes and edges of triangulation $\mathcal{T}_h$ is used. Smoothed aggregation can be again obtained by applying damped Jacobi smoother with filtered matrix A_F (5).

2.3 Element-wise aggregations

The only aggregation of this subsection is of Fish and Belsky [4]. It uses the concept of stiff and weak element which is utilized in construction of aggregates. The element e_i is considered stiff if the spectral radius k_i of its stiffness matrix is relatively large compared to other elements. The spectral radius is estimated by Gershgorin theorem. This stiff and weak concept is element-wise counterpart of strong and weak connection of node-wise approach. The algorithm tries to place weak elements on the interface between aggregates of stiff elements.

```
start-up:
    set $E_A$ of elements to aggregate (less elements on boundary)
    set $E_I$ of interface elements, $E_I = \emptyset$
    seed element $e_s$ with minimum number of neighboring elements

step 1: create stiff aggregate $A_i$
    $A_i = \{e_s\} \cup \{e_j : e_l \in \mathrm{neighbor}(e_s) \cap E_A; k_j \geq \varepsilon k_s\}$

step 2: update sets $E_I$, $E_A$
    $E_I = E_I \cup \{e_k : (e_k \in \mathrm{neighbor}(e_j), e_j \in A_i) \cap (e_k \notin A_i)\}$
    $E_A = E_A \setminus \{e_k : (e_k \in \mathrm{neighbor}(e_j), e_j \in A_i) \cup A_i\}$

step 3: find new seed element $e_s$
    $E_F = \{e_k : (e_k \in \mathrm{neighbor}(e_j), e_j \in A_i) \cap E_A\}$
    find seed element $e_s : e_s \in E_F, k_s \geq k_i \quad \forall e_i \in E_F$

stopping criteria:
    if $E_F = \emptyset$ then stop
    else $i = i + 1$, go to step 2
```

The parameter ε is used as threshold for determining the stiffness of elements.

3 Model problem and two-level Schwarz preconditioner

The model problem on which we will test aggregation techniques will be Darcy flow described by following equations:

$$\left.\begin{aligned} v &= -k\nabla u \\ \nabla \cdot v &= f \end{aligned}\right\} \quad \text{in} \quad \Omega \tag{7}$$

The heterogeneity will be induced by the permeability coefficient k. In our model problem, the coefficient will be stochastically generated with log-normal distribution.

The method chosen to test aggregations will be two-level Schwarz preconditioner for CG. It uses decomposition of computational domain Ω into overlapping subdomains Ω_i^δ. The subdomains are then used to define decomposition of finite element space V_h:

$$\begin{aligned} V_h &= V_0 + V_1 + \ldots + V_k \\ V_i &= \left\{ v \in V_h, v \equiv 0 \text{ in } \Omega \setminus \Omega_i^\delta \right\}, \forall i \in \{1 \ldots k\} \ , \end{aligned}$$

where the FE space V_0 corresponds to a coarse triangulation $\mathcal{T}_H$. Then it is possible to construct various Schwarz-type preconditioners, the simplest and most commonly used is additive preconditioner (B_{AS}),

$$B_{AS} = \sum_{i=0}^{k} R_i^T A_i^{-1} R_i,$$

where $\{R_i\}_{i=1}^k$ are restriction operators mapping nodes from Ω to Ω_i^δ and A_i is FE matrix corresponding to problem on subdomain Ω_i^δ with homogeneous Dirichlet boundary condition on boundary. The multiplicative and various hybrid preconditioners can be found in [5]. The matrix A_0 corresponds to auxiliary coarse space V_0 with restriction operator R_0. This is the place where aggregation comes in the play, the restriction operator R_0 is defined by (2) and matrix A_0 by term $A_0 = R_0 A_h R_0^T$

4 Conclusion

In this paper overview of some aggregation techniques was presented. The aggregations were used for construction of coarse space for two-level Schwarz preconditioner for CG method. The motivation for using model problem with strong heterogeneity is development of robust solvers with respect to heterogeneity. These solvers are needed for e.g. investigation of (geo)composites where strong heterogeneity is present. When using two-level method as a preconditioner, the quality of auxiliary coarse space dramatically influences the number of iterations needed to solve the problem. The aggregation techniques represent one possible approach to get the coarse space of desired qualities. Note that an efficient application of a parallel aggregation-based solver for microstructure analysis is in [6].

Acknowledgement: This work has been supported by the grants GA ČR number 103/09/H078 and 105/09/1830.

References

[1] Y. Notay. Aggregation-Based Algebraic Multilevel Preconditioning. *SIAM J. Matrix Anal. Appl., Vol. 27, 998-1018.* 2006. ISSN 1095-7162

[2] R. Scheichl, E. Vainikko. Additive Schwarz with Aggregation-Based Coarsening for Elliptic Problems with Highly Variable Coefficients. *Computing, Vol. 80, 319-343.* 2007. ISSN 1436-5057

[3] P. Vaněk, J. Mandel, M. Brezina. Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems. *Computing, Vol. 56, 179-196.* 1996. ISSN 1436-5057

[4] J. Fish, V. Belsky. Generalized Aggregation Multilevel solver. *Int. J. for Numerical Methods in Engineering, Vol. 40, 4341-4361.* 1997. ISSN 1097-0207

[5] A. Toselli, O. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer, 2005, Berlin. ISBN 3-540-20696-5.

[6] P. Arbenz et al. A scalable multi-level preconditioner for matrix-free μ-finite element analysis of human bone structures. *Int. J. for Numerical Methods in Engineering, Vol. 73, 927-947.* 2008. ISSN 1097-0207

Macroscopic traffic flow models: requiem and ressurection

M. Brandner, J. Egermaier, H. Kopincová

NTIS – New Technologies for Information Society
Department of Mathematics University of West Bohemia in Pilsen

1 Introduction

We shortly describe basic ideas of macroscopic traffic flow modeling, discuss the features of these models critically, and give proposals for their improvements. We also propose three numerical schemes based on the finite volume approach and compare them.

2 First order macroscopic models

Traffic flow modeling has become a major problem in many countries after the Second World War. We can get different types of mathematical models depending on what scale we choose: from the microscopic to the macroscopic through the kinetic one. The first macroscopic mathematical models were developed in the 50's of the 20th century. The basic first order model (i.e., the model containing one equation) was formulated by Lighthill in 1955 and Whitham and Richards in 1956 as presented in [6] (LWR model). It is based on the analogy between vehicles in traffic flow and particles in a fluid. The basic equation represents the conservation law for the vehicles

$$\varrho_t + [f(\varrho)]_x = 0, \tag{1}$$

where $\varrho = \varrho(x,t)$ is the density of vehicles, $f = f(\varrho) = v\varrho$ is the flux, $v = v(\varrho)$ is the velocity. The function $f = f(\varrho)$ represents a constitutive relation and it is called the fundamental diagram. For example, we can put

$$f(\varrho) = v_{max}\left(1 - \frac{\varrho}{\varrho_{max}}\right), \tag{2}$$

where v_{max} is a given maximal velocity and ϱ_{max} is a given maximal density. This model is identical to the first-order fluid dynamics models of water flow in rivers and gas flow through pipes (except for the specific form of $f = f(\varrho)$ – see [3]). Daganzo [3] summarizes the shortcomings of this type of models: they are not suitable for light traffic, they are not describe correctly the motion of a vehicle through a shock, they don't predict some instabilities. Newell shows (see [12]), however, that the macroscopic LWR model is in agreement with some microscopic car-following models. LeVeque shows in [11] that problems can occur when the flux function $f = f(\varrho)$ is neither concave nor convex (the night time traffic flow). In this case the entropy solution of the Riemann problem (see [11]) does not address the real traffic flow. In this situation it is necessary to pay special attention to the anisotropy of the model, i.e., to the fact that the drivers make decicions according to the situation ahead of the vehicle, not behind it. Daganzo also argues that the concept of relaxation time or viscosity effects (and we add: numerical viscosity effects) is not a self-evident property of the traffic flow.

3 Second order macroscopic models

Some researchers have tried to eliminate the shortcomings of the above models so that they improved them by introducing relations that are analogous to the conservation of momentum in fluids. They obtained the second order models, i.e., the models containing two partial differential equations. For example, the Payne-Whitham model (1971, 1974) can be written as (for brevity, we present the simplified version without the relaxation term)

$$\begin{aligned} \varrho_t + (\varrho v)_x &= 0, \\ (\varrho v)_t + [\varrho v^2 + p(\varrho)]_x &= 0, \end{aligned} \tag{3}$$

where $\varrho = \varrho(x,t)$ is the density, $v = v(x,t)$ is the velocity and $p = p(\varrho)$ is a given constitutive relation. Daganzo [3] shows three basic weaknesses of this type of models:

1. A fluid particle responds to stimuli from the front and from behind, but a car is an anisotropic particle that mostly responds to frontal stimuli.
2. The width of a traffic shock only encompasses a few vehicles.
3. Unlike molecules, vehicles have personalities.

Other Daganzo's comments are also significant. The model described above is a system of two hyperbolic partial differential equations (for a suitable choice of $p = p(\varrho)$). But a characteristic speed can be greater than the macroscopic fluid velocity (future vehicle behavior is determined by what happened behind it). Furthermore, one must recognize the basic observation that the number of molecules in the fluid and the number of cars on road are radically different.

Another major contribution to this research is the work of Aw, Klar, Materne and Rascle [1, 2]. They propose the following model (again for brevity, we present a simplified version without the relaxation term):

$$\begin{aligned} \varrho_t + [q - \varrho p(\varrho)]_x &= 0, \\ q_t + [q^2/\varrho - p(\varrho)q]_x &= 0, \end{aligned} \tag{4}$$

where $\varrho = \varrho(x,t)$ is the density, $v = v(x,t)$ is the velocity and $p = p(\varrho)$ is a given constitutive relation. This model has two very interesting properties:

1. The eigenvalues of the Jacobi matrix of the flux vector are $\lambda_1(\varrho, v) = v - \varrho p'(\varrho)$ and $\lambda_2(\varrho, v) = v$. It means that if the function $p = p(\varrho)$ is increasing then the maximal characteristic speed is v.
2. The system (4) can be transformed into Lagrangian mass coordinates. If we use the Godunov method (or even the finite volume method with the Roe or HLL solver) to solve the transformed problem we obtain discrete relations that correspond to the microscopic follow-the-leader model (see [1]). In other words, we get a direct link between the continuous and discrete model. Notice that the previous model (3) is based on analogy with the description of fluid flow only.

4 Numerical schemes and experiments

We use three numerical methods to solve (4) – the central scheme (see [9]), central-upwind scheme (see [7]) and the scheme based on the Roe approximate Riemann solver (see [10]). It should be

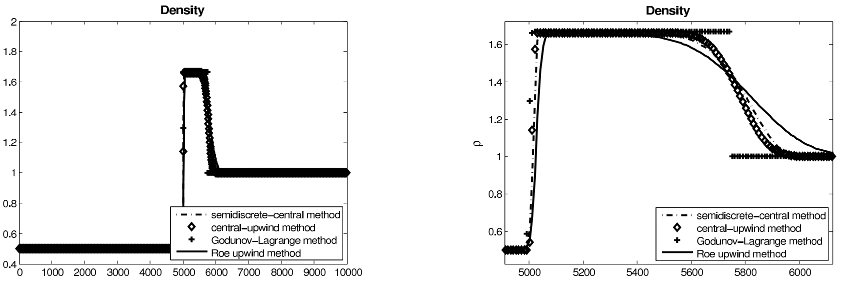


Figure 1: Solution of the Riemann problem 1 compared with the microscopic model represented by the Godunov method in Lagrangian coordinates (overall situation and a detailed view).

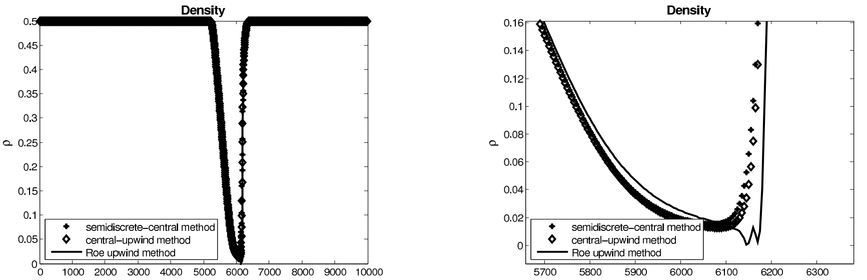


Figure 2: Solution of the Riemann problem 2 (overall situation and a detailed view).

noted that in the case of the Roe linearization we must determine the appropriate Roe matrix for the different constitutive relations $p = p(\varrho)$ separately. This itself may be a very difficult problem. We consider two Riemann problems with

1. left and right states given by $\varrho_L = 0.5$, $\varrho_R = 1$, $v_L = 10$, $v_R = 3$. The discretization steps are chosen as $\Delta x = 10$, $\Delta t = 0.25$ and $T = 250$ (initial number of cars: 7500);
2. left and right states given by $\varrho_L = 0.5$, $\varrho_R = 0.5$, $v_L = 6$, $v_R = 12$. The discretization steps are chosen as $\Delta x = 5$, $\Delta t = 0.1$ and $T = 100$ (initial number of cars: 5000). The vacuum state appears during the time evolution. In the case of the Roe method we can see instability caused by linearization.

5 Conclusion

The central and central-upwind schemes are Riemann-free methods. The central-upwind scheme may be interpreted as a method that uses the HLL solver. The HLL solver is based on the decomposition of the jump into two waves. Moreover, it does not use linearization, and thus it can be shown that the method that is based on this solver is positive. The scheme based on the Roe solver uses a special type of linearization - in the case of the single wave it approximates the shock speed exactly (in other cases, it is only an approximation). It seems therefore that in the case of the model based on two nonlinear partial differential equations the central-upwind method is the best approximation of the discrete follow-the-leader model. In conclusion, we note that it is very important to distinguish what is the error of model, the error of numerical methods and how to interpret the results of numerical simulations correctly. In the near future, we plan to compare our simulations with data obtained in real experiments, to use phase transition models and to develop numerical models for road networks.

Acknowledgement: This work has been supported by the Research Plan MSM 4977751301.

References

[1] A. Aw, A. Klar, T. Materne, M. Rascle: *Derivation of continuum traffic flow models from microscopic follow-the-leader models.* SIAM Journal on Applied Mathematics 63, 2002, 259–278.

[2] A. Aw, M. Rascle: *Ressurection of "second order" models of traffic flow?* SIAM Journal on Applied Mathematics 60, 2000, 916–938.

[3] C. F. Daganzo: *Requiem for second-order fluid approximations of traffic flow.* Transport Research 29B, 1995, 277–286.

[4] S. Darbha, K. R. Rajagopal, V. Tyagi: *A review of mathematical models for the flow of traffic and some recent results.* Nonlinear Analysis 69, 2008, 950–970.

[5] D. Helbing, A. Johansson: *On the controversy around Daganzo's requiem for and Aw-Rascle's ressurection of second-order traffic flow models.* The European Physical Journal B 69, 2009, 549–562.

[6] S. P. Hoogendoorn, P H. L. Bovy: *State-of-the-art of vehicular traffic flow modelling.* Journal of Systems and Control Engineering 215, 2001, 283–303.

[7] A. Kurganov, S. Noelle, G. Petrova: *Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations.* SIAM Journal of Scientific Computation 23, 2001, 707–740.

[8] A. Kurganov, A. Polizzi: *Non-oscillatory central schemes for traffic flow models with Arrhenius look-ahead dynamics.* Networks and Heterogeneous Media 4, 2009, 431–451.

[9] A. Kurganov, E. Tadmor: *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations.* Journal of Computational Physics 160, 2000, 241–282.

[10] R. J. LeVeque: *Finite volume methods for hyperbolic problems.* Cambridge University Press, Cambridge, 2002.

[11] R. J. LeVeque: *Some traffic flow models illustrating interesting hyperbolic behavior.* Technical report, SIAM Annual Meeting, July 10, 2001.

[12] G. F. Newell: *Nonlinear effects in the dynamics of car following.* Operations Research 9, 1961, 209–229.

An efficient solution of elasto-plastic problems in mechanics

M. Čermák, T. Kozubek, A. Markopoulos

VŠB - Technical University of Ostrava

1 Introduction

The goal of this paper is to present an efficient algorithm for the numerical solution of elasto-plastic problems in mechanics. These problems with hardening lead to the so-called quasistatic problems, where each nonlinear and nonsmooth time step problem is solved by the semismooth Newton method. In each Newton iteration we have to solve an auxiliary (possibly of large size) linear system of algebraic equations. In this paper, we propose a new approach how to solve such system efficiently using in a sense optimal algorithm based on our Total-FETI variant of FETI (Finite Element Tearing and Interconnecting) domain decomposition method. The efficiency is illustrated by the results of 3D elasto-plastic model benchmark.

2 TFETI domain decomposition

To apply the TFETI domain decomposition, we tear each body from the part of the boundary with the Dirichlet boundary condition, decompose each body into subdomains, assign each subdomain a unique number, and introduce new "gluing" conditions on the artificial intersubdomain boundaries and on the boundaries with imposed Dirichlet condition. For the artificial intersubdomain boundaries, we introduce the following notation: Γ_G^{pq} denotes the part of Γ^p that is glued to Ω^q and Γ_G^p denotes the part of Γ^p that is glued to the other subdomains. Obviously $\Gamma_G^{pq} = \Gamma_G^{qp}$. An auxiliary decomposition of the problem with renumbered subdomains and artificial intersubdomain boundaries is in Fig. 1. The gluing conditions require continuity of the displacements and of their normal derivatives across the intersubdomain boundaries.

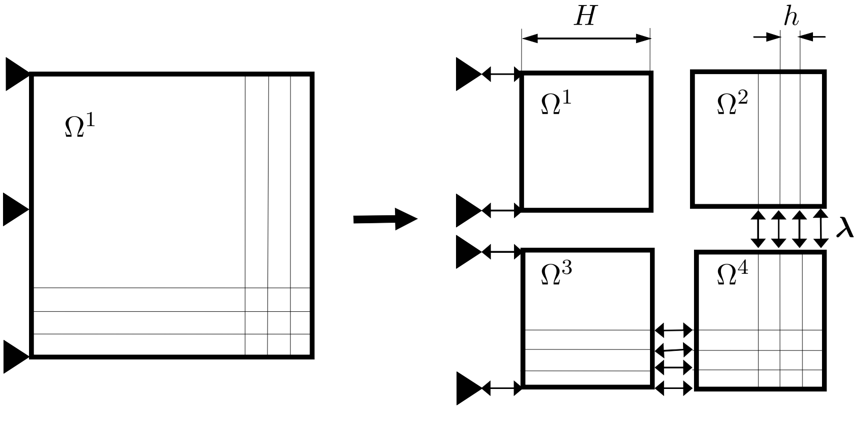


Figure 1: TFETI domain decomposition with subdomain renumbering.

The finite element discretization of $\overline{\Omega} = \overline{\Omega}^1 \cup \ldots \cup \overline{\Omega}^s$ with a suitable numbering of nodes results in the quadratic programming (QP) problem

$$\frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} - \mathbf{f}^\top \mathbf{u} \to \min \quad \text{subject to} \quad \mathbf{B}\mathbf{u} = \mathbf{c}, \tag{1}$$

where $\mathbf{K} = \mathrm{diag}(\mathbf{K}_1, \ldots, \mathbf{K}_s)$ denotes a symmetric positive semidefinite block-diagonal matrix of order n, $\mathbf{B}$ denotes an $m \times n$ full rank matrix, $\mathbf{f} \in \mathbb{R}^n$, and $\mathbf{c} \in \mathbb{R}^m$.

The diagonal blocks $\mathbf{K}_p$ that correspond to the subdomains Ω^p are positive semidefinite sparse matrices with known kernels, the rigid body modes. The blocks can be effectively decomposed using Cholesky factorization [1]. The vector $\mathbf{f}$ describes the nodal forces arising from the volume forces and/or some other imposed traction.

The matrix $\mathbf{B}$ with the rows $\mathbf{b}_i$ and the vector $\mathbf{c}$ with the entries c_i enforce the prescribed displacements on the part of the boundary with imposed Dirichlet condition and the continuity of the displacements across the auxiliary interfaces. The continuity requires that $\mathbf{b}_i\mathbf{u} = c_i = 0$, where $\mathbf{b}_i$ are vectors of the order n with zero entries except 1 and -1 at appropriate positions. Typically m is much smaller than n.

Even though (1) is a standard convex quadratic programming problem, its formulation is not suitable for numerical solution. The reasons are that $\mathbf{K}$ is typically ill-conditioned, singular, and the feasible set is in general so complex that projections into it can hardly be effectively computed.

The complications mentioned above may be essentially reduced by applying the duality theory of convex programming (see, e.g., Dostál [2]). The Lagrangian associated with problem (1) is

$$L(\mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} - \mathbf{f}^\top \mathbf{u} + \boldsymbol{\lambda}^\top(\mathbf{B}\mathbf{u} - \mathbf{c}). \tag{2}$$

It is well known [2] that (1) is equivalent to the saddle point problem

$$L(\overline{\mathbf{u}}, \overline{\boldsymbol{\lambda}}) = \sup_{\boldsymbol{\lambda}} \inf_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\lambda}). \tag{3}$$

For more details how to solve efficiently the resulting saddle-point system we recommend [2, 5].

3 Elasto-plasticity

Elasto-plastic problems are the so-called quasi-static problems, where the history of loading is taken into account. We consider the von Mises elasto-plasticity with the strain isotropic hardening and incremental finite element method with the return mapping concept. More details are in [3].

The elasto-plastic deformation of an body Ω after loading is described by the Cauchy stress tensor $\boldsymbol{\sigma}$, the small strain tensor $\boldsymbol{\varepsilon}$, the displacement $\mathbf{u}$, and the nonnegative hardening parameter $\boldsymbol{\kappa}$. Symmetric tensor is represented by the vectors and their deviatoric part is denoted by the symbol *dev*.

Let us denote the space of continuous and piecewise linear functions constructed over a regular partition of Ω into tetrahedrons with the discretization norm h by $V_h \subset V$, where $V = \left\{v \in [H^1(\Omega)]^3 : \quad v = 0 \text{ on } \Gamma_U\right\}$. Let

$$0 = t_0 < t_1 < \ldots t_k < \ldots < t_N = t^* \tag{4}$$

be a partition of the time interval $[0, t^*]$. Then the solution algorithm after time and space discretization has the form:

Algorithm 3.

1. Initial step: $\mathbf{u}_h^0 = 0,\ \boldsymbol{\sigma}_h^0 = 0,\ \boldsymbol{\kappa}_h^0 = 0,$
2. **for** $k = 0,\ \ldots, N-1$ **do** (load step)
3. From previous step we have: $\mathbf{u}_h^k,\ \boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k$ and compute $\triangle\mathbf{u}_h,\ \triangle\boldsymbol{\sigma}_h,\ \triangle\boldsymbol{\kappa}_h$
$$\triangle\boldsymbol{\varepsilon}_h = \varepsilon(\triangle\mathbf{u}_h), \quad \triangle\mathbf{u}_h \in V_h \tag{5}$$
$$\triangle\boldsymbol{\sigma}_h = T_\sigma(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k,\ \triangle\boldsymbol{\varepsilon}_h), \tag{6}$$
$$\triangle\boldsymbol{\kappa}_h = T_\kappa(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k,\ \triangle\boldsymbol{\varepsilon}_h), \tag{7}$$
4. Solution $\triangle\boldsymbol{\sigma}_h(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k, \varepsilon(\triangle\mathbf{u}_h))$ is substituted into equation of equilibrium:
$$\int_\Omega \triangle\boldsymbol{\sigma}_h^T(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k, \varepsilon(\triangle\mathbf{u}_h))\varepsilon(\mathbf{v}_h)dx = \langle \triangle\mathbf{F}^k,\ \mathbf{v}_h\rangle, \quad \forall\mathbf{v}_h \in V_h \tag{8}$$
leads to a nonlinear system of equations with unknown $\triangle\mathbf{u}_h$ which is solved using the Newton method [4]. The linearized problem arising in each Newton step is solved by TFETI algorithmic scheme proposed above.
5. Then we compute new aproximations: $\mathbf{u}_h^{k+1} = \mathbf{u}_h^k + \triangle\mathbf{u}_h,\ \boldsymbol{\sigma}_h^{k+1} = \boldsymbol{\sigma}_h^k + \triangle\boldsymbol{\sigma}_h,\ \boldsymbol{\kappa}_h^{k+1} = \boldsymbol{\kappa}_h^k + \triangle\boldsymbol{\kappa}_h$.
6. **enddo**

Above we consider the following notation. Let $\mathbf{C}$ denote the Hook's matrix, $\mathbf{E}$ represent linear operator dev, μ, λ be the Lamé coefficients, $\triangle\mathbf{f}_h^k$ be the increment of the right hand side and $\boldsymbol{\sigma}_h^t = \boldsymbol{\sigma}_h^k + \mathbf{C}\triangle\boldsymbol{\varepsilon}_h$. For return mapping concept we define

$$\triangle\boldsymbol{\sigma}_h \;=\; T_\sigma^{RM}(\boldsymbol{\sigma}_h^k, \boldsymbol{\kappa}_h^k, \triangle\boldsymbol{\varepsilon}_h) = \begin{cases} \mathbf{C}\triangle\boldsymbol{\varepsilon}_h & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) \le 0, \\ \mathbf{C}\triangle\boldsymbol{\varepsilon}_h - \gamma_R\widehat{\mathbf{n}} & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) > 0, \end{cases} \tag{9}$$

$$\triangle\boldsymbol{\kappa}_h \;=\; T_\kappa^{RM}(\boldsymbol{\sigma}_h^k, \boldsymbol{\kappa}_h^k, \triangle\boldsymbol{\varepsilon}_h) = \begin{cases} 0 & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) \le 0, \\ \gamma z = \gamma_R\|\mathbf{Cp}\|^{-1}z & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) > 0, \end{cases} \tag{10}$$

where

$$\gamma_R = \frac{3\mu}{3\mu + H_m}\sqrt{\frac{2}{3}}P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k), \quad \widehat{\mathbf{n}} = \frac{dev(\boldsymbol{\sigma}_h^t)}{\|dev(\boldsymbol{\sigma}_h^t)\|}, \quad \|\mathbf{Cp}\| = 2\mu\sqrt{\frac{3}{2}}, \quad z = 1 \tag{11}$$

and plasticity function

$$P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) = \sqrt{\frac{3}{2}}\|dev(\boldsymbol{\sigma}_h^t)\| - (Y + H_m\boldsymbol{\kappa}_h^k), \qquad Y, H_m > 0. \tag{12}$$

The function $\gamma_R\widehat{\mathbf{n}}$ is semismooth and potential. The derivative of T_σ^{RM} is

$$\begin{aligned} (T_\sigma^{RM})'(\triangle\boldsymbol{\varepsilon}) \;&=\; \mathbf{C} - 2\mu\tfrac{3\mu}{3\mu+H_m}\left[\mathbf{E} + \right. \\ &+\; \left.\sqrt{\tfrac{2}{3}}\tfrac{Y_0+H_m\boldsymbol{\kappa}_h^k}{\|dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon})\|}\left(\tfrac{dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon})(dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon}))^T}{\|dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon})\|^2} - \mathbf{E}\right)\right]. \end{aligned} \tag{13}$$

If we represent a function $\mathbf{v}_h \in V_h$ by the vector $\mathbf{v} \in \mathbb{R}^n$ and omit index k then (8) can be rewritten as the system of nonlinear equations

$$F(\triangle\mathbf{u}) = \triangle\mathbf{f}, \tag{14}$$

where

$$\begin{aligned} \langle F(\mathbf{v}), \mathbf{w}\rangle \;&=\; \textstyle\int_\Omega \langle T_\sigma^{RM}(\varepsilon(\mathbf{v}_h)), \varepsilon(\mathbf{w}_h)\rangle dx, \qquad \forall\mathbf{v}, \mathbf{w} \in \mathbb{R}^n \\ \langle \triangle\mathbf{f}, \mathbf{w}\rangle \;&=\; \triangle\mathbf{f}_h(v_h), \quad \forall\mathbf{w} \in \mathbb{R}^n. \end{aligned} \tag{15}$$

4 Numerical experiments

Let us consider a 3D plate with a hole in the center (due to symmetry only a quatre of the whole structure is used) with the geometry depicted in Fig. 2. Boundary conditions are specified in Fig. 3. Symmetry conditions are prescribed on the left and lower sides of Ω. The surface load $g(t) = 450 \sin(2\pi t)$ [MPa], $t \in [0, \frac{1}{4}]$ [sec], is applied to the upper side of Ω. The elasto-plastic material parameters are $E = 206900$ [MPa], $\nu = 0.29$, $Y = 450$, $H_m = 100$ and the time interval $[0, \frac{1}{4}]$ [sec] is divided into 50 steps. We consider a mesh with 9471 nodes and 48000 tetrahedrons.

In the nth Newton iteration we compute an approximation $\triangle \mathbf{u}^n$ by solving the linear problem of the form $\mathbf{K}^n \triangle \mathbf{u}^n = \triangle \mathbf{f}^n - \mathbf{B}^\top \boldsymbol{\lambda}^n$ using the TFETI algorithmic scheme proposed above. We stop the Newton method in every time step if $\|\triangle \mathbf{u}^{n+1} - \triangle \mathbf{u}^n\| / \left(\|\triangle \mathbf{u}^{n+1}\| + \|\triangle \mathbf{u}^n\|\right)$ is less than 10^{-9}.

Notice that the maximum number of the Newton iterations is small for all time steps, therefore the method is suitable for the problem. In remaining figures, we depict plastic and elastic elements, von Mises stress in the xy plane cross-section with the z coordinate -0.5 [mm] corresponding to the center of Ω. In Figs. 4, 5 6, we can see which elements are plastic (gray color) and which are elastic (white color). Particularly, in time steps 1-12 we observe only elastic behavior, and in time steps 13-50 plastic behavior of some elements. The von Mises stress on deformed mesh scaled 10x for better illustration is showed in Fig. 7.

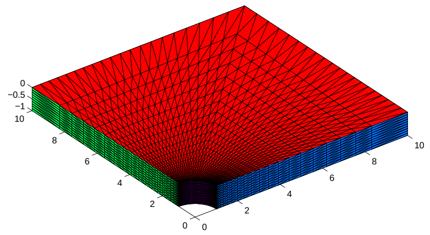


Figure 2: 3D plate geometry in [mm].

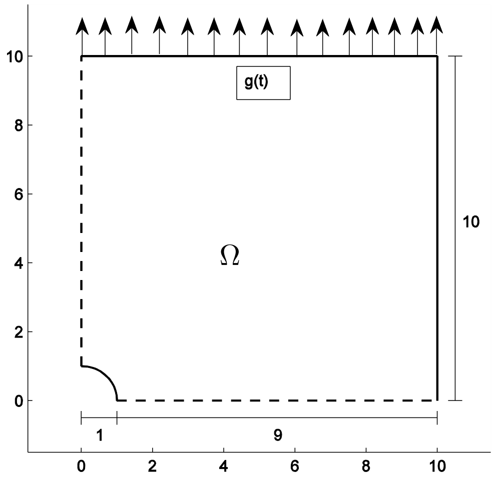


Figure 3: 2D plate geometry in [mm] and boundary conditions.

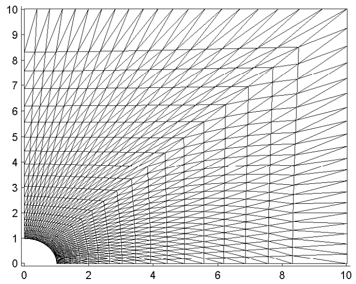


Figure 4: Plastic and elastic elements after 1 time step.

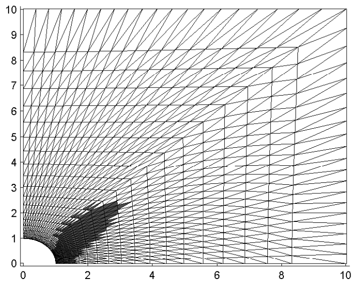


Figure 5: Plastic and elastic elements after 35 time steps.

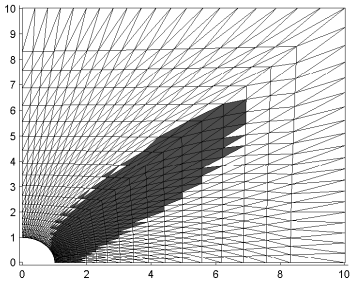


Figure 6: Plastic and elastic elements after 50 time steps.

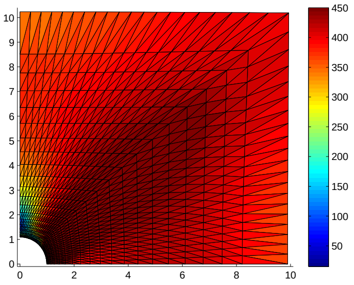


Figure 7: Von Mises stress on the deformed mesh.

5 Conclusion

We have presented an efficient algorithm for the numerical solution of elasto-plastic problems. These problems lead to the quasi-static problems, where each nonlinear and nonsmooth time step problem is solved by the semismooth Newton method. In each Newton iteration we have to solve an auxiliary (possibly of large size) linear system of algebraic equations. We proposed a new approach how to solve such system efficiently using in a sense optimal algorithm based on our Total-FETI variant of FETI domain decomposition method. The algorithm has been adapted also to the solution of contact problems [1].

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

References

[1] T. Brzobohatý, Z. Dostál, P. Kovář, T. Kozubek, A. Markopoulos: *Cholesky decomposition with fixing nodes to stable computation of a generalized inverse of the stiffness matrix of a floating structure.* Accepted for publishing in IJNME.

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

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

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

[5] R. Kučera, T. Kozubek, A. Markopoulos, J. Machalová: *On the Moore-Penrose inverse in solving saddle-point systems with singular diagonal blocks.* NLA submitted.

On optimally conditioned cubic spline wavelets on the interval

D. Černá, V. Finěk

Department of Mathematics and Didactics of Mathematics, Technical University of Liberec

1 Introduction

Wavelets are by now a widely accepted tool especially in signal and image processing. In the field of numerical mathematics, methods based on wavelets are successfully used for preconditioning of large systems arising from discretization of elliptic partial differential equations, sparse representations of some types of operators and adaptive solving of operator equations. Quantitative properties of these methods depends on the choice of the wavelet basis, in particular on its condition number.

Construction of wavelet bases on a bounded domain usually starts with the construction of wavelets on the real line. Then these wavelets are adapted to the interval and by tensor product to the n-dimensional cube. Finally splitting the domain into subdomains which are images of $(0,1)^n$ under appropriate parametric mappings one obtains wavelet bases on fairly general domains. Thus, the properties of the employed wavelet basis on the interval are crucial for the properties of the resulting bases on general domain.

The first biorthogonal spline-wavelet bases on the unit interval were constructed in [5]. However some of them are badly conditioned. Then several modifications were proposed. We will mention here only the recent construction by M. Primbs [6] which seems to outperform the previous constructions with respect to the condition number along with spectral properties of the corresponding stiffness matrices for linear and quadratic spline-wavelets. In this contribution, we present construction of cubic spline wavelets on the unit interval with a nearly optimal condition number (comparable with the condition number of the spline wavelet bases on the real line).

First of all, we summarize the desired properties:

- *Riesz basis property.* The functions form a Riesz basis of the space $L^2(\langle 0,1\rangle)$.
- *Locality.* The basis functions are local. Then the corresponding decomposition and reconstruction algorithms are simple and fast.
- *Biorthogonality.* The primal and dual wavelet bases form a biorthogonal pair.
- *Polymial exactness.* The primal bases have polynomial exactness of order N and the dual bases have polynomial exactness of order $\tilde{N}$. As in [4], $N+\tilde{N}$ has to be even and $\tilde{N} \geq N$.
- *Smoothness.* The smoothness of primal and dual wavelet bases is another desired property. It ensures the validity of norm equivalences.
- *Closed form.* The primal scaling functions and wavelets are known in the closed form. It is requested property for the fast computation of integrals involving primal scaling functions and wavelets.
- *Well-conditioned bases.* Our objective is to construct wavelet bases with improved condition number, especially for larger values of N and $\tilde{N}$.

From the viewpoint of numerical stability, ideal wavelet bases are orthogonal wavelet bases. However, they are usually avoided in numerical treatment of partial differential and integral equations, because they are not accessible analytically, the complementary boundary conditions can not be satisfied and it is not possible to increase the number of vanishing wavelet moments independent from the order of accuracy. Moreover, sufficiently smooth orthogonal wavelets typically have a large support.

2 Construction of wavelet bases on the interval

Majority of constructions of wavelets start with the construction of the primal scaling bases. Here, we use the primal scaling bases designed in [1], because they are known to be well-conditioned. Let N be the desired order of polynomial exactness of the primal scaling basis and let $\mathbf{t}^j = (t^j_k)_{k=-N+1}^{2^j+N-1}$ be a sequence of knots defined by

$$\begin{aligned} t^j_k &= 0 \quad \text{for} \quad k = -N+1, \dots, 0, \\ t^j_k &= \frac{k}{2^j} \quad \text{for} \quad k = 1, \dots 2^j - 1, \\ t^j_k &= 1 \quad \text{for} \quad k = 2^j, \dots, 2^j + N - 1. \end{aligned}$$

The corresponding B-splines of order N are defined by

$$B^j_{k,N}(x) := \left(t^j_{k+N} - t^j_k\right) \left[t^j_k, \dots, t^j_{k+N}\right]_t (t-x)^{N-1}_+ , \quad x \in [0,1], \tag{1}$$

where $(x)_+ := \max\{0, x\}$ and $[t_1, \dots t_N]_t f$ is the N-th divided difference of f. The set Φ_j of primal scaling functions is then simply defined as

$$\phi_{j,k} = 2^{j/2} B^j_{k,N}, \quad \text{for} \quad k = -N+1, \dots, 2^j - 1, \quad j \geq 0. \tag{2}$$

The inner functions are translations and dilations of a function ϕ which correspond to the primal scaling functions constructed by Cohen, Daubechies, Feauveau in [4]. In the following, we consider ϕ from [4] which is shifted so that its support is $[0, N]$.

The desired property of the dual scaling basis $\tilde\Phi$ is biorthogonality to Φ and polynomial exactness of order $\tilde N$. Let $\tilde\phi$ be dual scaling function designed in [4] which is shifted so that its support is $\left[-\tilde N + 1, N + \tilde N - 1\right]$. Then inner scaling functions are its translations and dilations of $\tilde\phi$:

$$\theta_{j,k} = 2^{j/2} \tilde\phi\left(2^j \cdot - k\right), \quad k = \tilde N - 1, \dots 2^j - N - \tilde N + 1. \tag{3}$$

Further, there will be two types of basis functions at each boundary. Basis functions of the first type are defined to preserve polynomial exactness in the same way as in [5]:

$$\theta_{j,k} = 2^{j/2} \sum_{l=-N-\tilde N+2}^{\tilde N-2} \left\langle p^{\tilde N-1}_{k+N-1}, \phi(\cdot - l)\right\rangle \tilde\phi\left(2^j \cdot - l\right)|_{[0,1]}, \quad k = 1 - N, \dots, \tilde N - N, \tag{4}$$

where $p^{\tilde N-1}_k$ are Bernstein polynomials defined by

$$p^{\tilde N-1}_k(x) := b^{-\tilde N+1} \binom{\tilde N - 1}{k} x^k (b-x)^{\tilde N-1-k}, \quad k = 0, \dots, \tilde N - 1. \tag{5}$$

The reason for the choice of Bernstein polynomials consists in their well-conditionality on $[0, b]$ relative to the supremum norm. In our numerical experiments, the constant $b = 10$ seems to be optimal.

The basis functions of the second type are defined as

$$\theta_{j,k} = 2^{\frac{j+1}{2}} \sum_{l=\tilde{N}-1-2k}^{N+\tilde{N}-1} \tilde{h}_l \tilde{\phi}\left(2^{j+1} \cdot -2k - l\right)|_{[0,1]}, \quad k = \tilde{N} - N + 1, \ldots, \tilde{N} - 2, \tag{6}$$

where $\tilde{h}_l$ are scaling coefficients corresponding to $\tilde{\phi}$. Then they are as much as possible similar to the inner functions.

The boundary functions at the right boundary are defined to be symmetrical with the left boundary functions:

$$\theta_{j,k} = \theta_{j,2^j-N+1-k}\left(1 - \cdot\right), \quad k = 2^j - N - \tilde{N} + 2, \ldots, 2^j - 1. \tag{7}$$

Since the set $\Theta_j := \left\{\theta_{j,k} : k = -N + 1, \ldots, 2^j - 1\right\}$ is not biorthogonal to Φ_j, we derive a new set $\tilde{\Phi}_j$ from Θ_j by biorthogonalization. Let $\mathbf{A}_j = \left(\langle \phi_{j,k}, \theta_{j,l} \rangle\right)_{j,l=-N+1}^{2^j-1}$, then viewing $\tilde{\Phi}_j$ and Θ_j as column vectors we define

$$\tilde{\Phi}_j := \mathbf{A}_j^{-T} \Theta_j, \tag{8}$$

assuming that $\mathbf{A}_j$ is invertible, which was the case for all tested choices of N, $\tilde{N}$.

The final step is to determine the corresponding wavelets. This problem can be transformed from functional analysis to linear algebra by a general principle called stable completion which was proposed in [2]. The initial stable completion was found by the method from [5] with some small changes.

For more details on the construction, the adaptation to complementary boundary conditions, properties of constructed bases, and the comparison of the quantitative behaviour in the adaptive wavelet method for cubic wavelet bases from [3] and [6], we refer to [3].

Acknowledgement: This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic through the Research Center 1M06047.

References

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

[2] J.M. Carnicer, W. Dahmen, J.M. Peña: *Local decompositions of refinable spaces.* Appl. Comp. Harm. Anal. 3, 1996, 127–153.

[3] D. Černá, V. Finěk: *Construction of optimally conditioned cubic spline-wavelets on the interval.* Advances in Computational Mathematics, DOI: 10.1007/s10444-010-9152-5, 2010.

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

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

[6] M. Primbs: *New stable biorthogonal spline-wavelets on the interval.* Results in Mathematics 57 1-2, 2010, 121–162.

On averaging in the domain decomposition methods

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

[1,2,3] Czech Technical University in Prague
[4] Institute of Mathematics AS CR, Prague

1 Introduction

Substructuring Domain Decompositon (DD) methods [1] are widely used as preconditioners for solving large systems of linear algebraic equations obtained by finite element discretization of second order elliptic problems. There are two main classes of the substructuring methods: *primal* methods (like classical Neumann-Neumann method, BDD or BDDC) and *dual* ones (like FETI or FETI-DP methods). Both classes can be regarded as equivalent in a sense that they can be described in a common framework and that a primal method and the corresponding dual one has the same convergence properties (see [2]). Both classes also use some sort of weighted *averaging* (or weighted distribution) of values across the interface.

Although we concentrate on BDDC in this paper, we believe that our ideas can be used for other primal and dual substructuring DD methods as well. It can be found in [2] that a primal (BDDC) and the correspondig dual (FETI-DP) method can be determined by a choice of two operators: the *injection* R and the *averaging* E, which also appear in the estimate of the condition number of the preconditioned operator. Operator R represents continuity conditions across the interface and thus also the choice of the coarse space. A lot of work has been devoted to investigation of influence of different choices of R on convergence properties. For significant results of this effort see for instance [1] or [3]. In this paper we focus on the averaging operator E, which seems to be left out of main direction of research so far. We introduce a general framework for derivation of the averaging operator, from which we recover the standard choice of the operator E found in literature and suggest some new proposals.

2 Primal and dual substructuring methods

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

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

where $\mathbf{K}$ is large, sparse, symmetric positive definite (SPD) matrix and $\mathbf{f}$ represents the load vector. Let us decompose the domain Ω into N non-overlapping subdomains Ω_i, $i = 1, \ldots, N$. Unknowns common to at least two subdomains form the *global interface* denoted as Γ. Remaining unknowns are classified as belonging to subdomain *interiors.* The global interface Γ can be expressed as union of *local interfaces* Γ_i, $i = 1, \ldots, N$, containing interface unknowns involved just in subdomain Ω_i.

The first step typical for substructuring DD methods is the ***reduction of the problem to the interface***. Without loss of generality, suppose that unknowns are ordered so that interior unknowns form the first part and the interface unknowns form the second part of the solution vector, i.e. $\mathbf{u} = \left[\begin{array}{cc} \mathbf{u}_\mathrm{o} & \widehat{\mathbf{u}} \end{array}\right]^T$, where $\mathbf{u}_\mathrm{o}$ stands for all interior unknowns and $\widehat{\mathbf{u}}$ for unknowns at interface. System (1) now can be formally rewritten to block form

$$\left[\begin{array}{cc} \mathbf{K}_{\mathrm{oo}} & \mathbf{K}_{\mathrm{or}} \\ \mathbf{K}_{\mathrm{ro}} & \mathbf{K}_{\mathrm{rr}} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_\mathrm{o} \\ \widehat{\mathbf{u}} \end{array}\right] = \left[\begin{array}{c} \mathbf{f}_\mathrm{o} \\ \widehat{\mathbf{f}} \end{array}\right]. \tag{2}$$

The hat symbol ($\widehat{\ }$) is used to denote global interface quantities. If we suppose the interior unknowns ordered subdomain after subdomain, then the submatrix $\mathbf{K}_{\mathrm{oo}}$ is block diagonal with each diagonal block corresponding to one subdomain. After eliminating all the interior unknowns from (2), we arrive at *Schur complement problem* for the interface unknowns

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

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

The main idea of the ***primal DD substructuring methods*** can be expressed as splitting the given residual of PCG method to subdomains, solving subdomain problems and projecting the result back to the global domain. A primal additive preconditioner of the Neumann-Neumann type can be written as $M_P = ES^{-1}E^T$, where operator E^T represents splitting of the residual to subdomains, S^{-1} stands for solution of subdomain problems, and E represents projection of subdomain solutions back to the global problem by some averaging. The condition number κ of the preconditioned operator $M_P\widehat{S}$ is bounded by

$$\kappa \leq ||RE||_S^2 = ||I - RE||_S^2\,, \tag{4}$$

where operator R splits the global interface into subdomains and relation $ER = I$ is assumed, which means that if the problem is split into subdomains and then projected back to the whole domain, the original problem is obtained. The energetic norm on the right-hand side of (4) is defined by the scalar product as $||u||_S^2 = \langle Su, u\rangle$. The estimate (4) can be found in [2].

The main idea of the BDDC ([2]) is to introduce a global *coarse problem* by imposing continuity conditions across the interface in selected *coarse unknowns*, in order to achieve better preconditioning and to fix 'floating subdomains' to guarantee invertibility of S. R now represents splitting of the global interface into subdomains except the coarse unknowns and E^T distributes residual among neighbouring subdomains only in those interface unknowns which are not coarse. Thus in BDDC, only part of the global residual is split into subdomains; residual at the coarse unknowns is left undivided – it is processed by the global coarse problem.

Dual methods can be described using the complementary projection to projection RE. It is usually expressed by composition of other two operators as $I - RE = B_D^TB$. Operator B specifies jump at interface values coming from adjacent subdomains and operator B_D^T (determined by E) distributes a given jump across the interface among adjacent subdomains. Relationship $BB_D^T = I$ is assumed. Instead of solving (3), linear system $\mathbf{B}\mathbf{S}^{-1}\mathbf{B}^T\,\lambda = \mathbf{B}\mathbf{S}^{-1}\mathbf{E}^T\,\widehat{\mathbf{f}}$ is solved for unknown λ using preconditioner $M_D = B_DSB_D{}^T$. For the condition number of the preconditioned operator M_DS^{-1}, the same upper estimate as for primal method is valid, see [2]: $\kappa \leq ||B_D^TB||_S^2 = ||I - RE||_S^2 = ||RE||_S^2\,.$

3 Choice of the averaging operator E

We assume that the operator R is given and our goal is to design the averaging operator E so that it in some sense minimizes the energetic norm on the right hand side of the estimate (4). Let us show the main ideas on the simple example derived from some scalar equation solved on the domain splitted to just two subdomains, without coarse unknowns (more detailed analysis can be found in [4]). In this case R and a standard choice of E have the matrix form

$$\mathbf{R} = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}, \qquad \mathbf{E} = \begin{bmatrix} \mathbf{A} & \mathbf{I} - \mathbf{A} \end{bmatrix}, \tag{5}$$

where $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a diagonal matrix of weights α_i at interface nodes of the first subdomain.

Our approach is to start with some fixed $\mathbf{u} = (\mathbf{u}_1,\ \mathbf{u}_2)^T$ with the interface jump $\mathbf{d} = \mathbf{u}_2 - \mathbf{u}_1$ and try to find $\mathbf{E}$ so that it minimizes energy norm of the projection $(\mathbf{I} - \mathbf{RE})\mathbf{u}$ of the given vector $\mathbf{u}$. The square of the energy norm can be expressed as $||(\mathbf{I} - \mathbf{RE})\mathbf{u}||_S^2 = \mathbf{u}^T(\mathbf{I} - \mathbf{RE})^T\mathbf{S}(\mathbf{I} - \mathbf{RE})\mathbf{u} = \mathbf{d}^T(\mathbf{A}^T\,\widehat{\mathbf{S}}\mathbf{A} - \mathbf{A}^T\mathbf{S}^1 - \mathbf{S}^1\mathbf{A} + \mathbf{S}^1\,)\,\mathbf{d}$, where $\mathbf{S}^{\mathrm{i}}$ are local Schur complements and we use the fact that $\mathbf{S} = \mathrm{diag}(\mathbf{S}^1, \mathbf{S}^2)$ and $\widehat{\mathbf{S}} = \mathbf{S}^1 + \mathbf{S}^2$ in the case of two subdomains. The formula above can be seen as a quadratic function of variables α_i, which can be minimised by computing all partial derivatives and equating them to zero:

$$\frac{\partial}{\partial \alpha_i} ||(\mathbf{I} - \mathbf{RE})\mathbf{u}||_S^2 = 2d_i \left(\sum_j \widehat{s}_{ij}\alpha_j d_j - \sum_j s^1_{ij} d_j \right) = 0 \quad \forall\, i\,. \tag{6}$$

Here d_i stands for the i-th component of the *jump vector* $\mathbf{d}$ and elements of the matrices $\widehat{\mathbf{S}}$ and $\mathbf{S}^1$ are denoted as $\widehat{s}_{ij}$ and s^1_{ij}, respectively. Values of α_i obtained from (6) are tailored to the interface jump $\mathbf{d}$ of the given $\mathbf{u}$. Let us take $\mathbf{d}$ as a test vector which can uncover hidden features of $\widehat{\mathbf{S}}$ and $\mathbf{R}$ and, moreover, which can be chosen so that it simplifies the system (6). One option is to choose all the cartesian basis vectors $\mathbf{e}_k$, one after another, which leads to the popular choice of

$$\alpha_i = s^1_{\mathrm{ii}} / (s^1_{\mathrm{ii}} + s^2_{\mathrm{ii}})\,. \tag{7}$$

For less elementary test vectors $\mathbf{d}$ we make an additional simplification: Let us assume that all α_i are equal to the same value of α for some set of nodes (so we are going to find some average value). Then, after adding all equations (6) together, we get

$$\alpha = \mathbf{d}^T\mathbf{S}^1\,\mathbf{d} / \mathbf{d}^T(\mathbf{S}^1 + \mathbf{S}^2)\,\mathbf{d}\,. \tag{8}$$

This formula can be generalized to more than 2 subdomains. Our proposition is to choose several test vectors with nonzero values at some selected nodes only, typically face or edge, and compute corresponding value of α for that face or edge.

3.1 Numerical results and conclusion

For a simple preliminary test a 2D Poisson equation on a rectangular domain was chosen. The domain was divided into two rectangular subdomains of the same size and shape, both of which touch the boundary with prescribed Dirichlet boundary condition. The problem was discretized by FEM with bilinear elements. BDDC was used just as an iteration method, not as a preconditioner combined with PCG. Four different methods for choice of E were tested:

iter.	Method I	Method II	Method III	Method IV	α
without coarse nodes					
1.	1.5001	1.4966	0.4235	1.5001	0.500
2.	0.3872	0.3854	0.0806	0.0001	0.276
3.	0.0999	0.0992	0.0153	2e-06	0.424
4.	0.0258	0.0255	0.0029	1e-09	0.492
5.	0.0066	0.0066	0.0006	4e-15	0.276
2 coarse nodes					
1.	0.7349	0.7332	0.2402	0.7349	0.500
2.	0.0929	0.0925	0.0140	0.0211	0.376
3.	0.0117	0.0117	0.0008	0.0012	0.376
4.	0.0015	0.0015	5e-05	7e-05	0.376
5.	0.0002	0.0002	3e-06	4e-06	0.376

Table 1: Comparisson of discussed methods.

I : arithmetic average, i.e. $\alpha = 0.5$,
II : weighted average (7), i.e. $\alpha_i = s^1_{\mathrm{ii}} / \, (s^1_{\mathrm{ii}} + s^2_{\mathrm{ii}})$,
III : proposition (8) with $d = (1, \ldots, 1)$, i.e. $\alpha = \sum_{i,j} s^1_{\mathrm{ij}} / \, \sum_{i,j} (s^1_{\mathrm{ij}} + s^2_{\mathrm{ij}})$,
IV : proposition (8) with d chosen as actual interface jump.

Table 1 contains norms of errors (differences from exact solution) at first 5 iterations. There are two different choices of coarse unknowns: either none (first part of the table), or 2 nodes at the opposite ends of the interface (second part). For Method II, computed values of α_i were between 0.4997 and 0.5000 in both cases (i.e. very close to the arithmetic average). For Method III, value of α was 0.276 for the first case and 0.397 for the second. For Method IV, values of α were recomputed in every step and are presented in the last column. Very similar results were obtained also in the case of two rectangular subdomains different in size.

For the simple test problem, it seems that Methods III and IV outperform Methods I and II. An interesting observation is that for the first three methods, using coarse unknowns leads to better performance (as one would expect), while it slightly worsens the convergence of Method IV. These are just preliminary results and numerical tests will be performed for 2D and 3D problems with more subdomains.

Acknowledgement: This research has been supported by the grant No. 106/08/0403 of the Czech Science Foundation and by project MSM 6840770010.

References

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

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

[3] J. Mandel, B. Sousedík: *Adaptive selection of face coarse degrees of dreedom in the BDDC and the FETI-DP iterative substructuring methods* Comput. Methods Appl. Mech. Engrg. 196 (8), 2007, 1389–1399.

[4] M. Čertíková, P. Burda, J. Novotný, J. Šístek: *Some remarks on averaging in the BDDC method.* Proceedings of PANM'15, Horní Maxov, 2010. To appear.

On two variants of incremental condition estimation

J. Duintjer Tebbens, M. Tůma

Institute of Computer Science AS CR, Prague

1 Introduction

Classical 2-norm condition estimators often assume a given triangular factorization and estimate the condition numbers of the triangular factors. For instance, if the matrix A is symmetric positive definite and $A = LL^T$ is its Cholesky decomposition, then $\kappa(A) = \kappa(L)^2$ is used. So-called *incremental* condition estimation for (lower) triangular matrices was proposed at the beginning of the nineties [1], [2]. It computes a sequence of approximate condition numbers of the leading upper left submatrices of growing dimension. The approximation for the current submatrix is obtained from an approximate singular vector constructed without accessing the previous submatrices. This makes the procedure relatively inexpensive and particularly suited when a triangular matrix is computed one row at a time. A similar strategy was proposed later [5] and recommended for sparse matrices.

In our talk we show that the two techniques may differ considerably with respect to their ability to find accurate approximations of either the minimal or the maximal singular value, although there is no general superiority of one technique for the condition number. We will also explain how the differences can be exploited when the inverse of the triangular matrix is computed along with the triangular matrix itself. This can be done at low expenses; see [4] for a discussion of well-known implementations and [3] for a recent strategy. Using the inverse, we obtain an incremental condition estimator which is significantly better than the estimators of [1] and [5].

In this extended abstract we give a brief description of the original incremental technique from [1] and a new interpretation of the alternative technique from [5]. Then we present experiments combining both techniques when the inverse of the triangular matrix is available.

2 The original incremental condition estimation technique

The incremental condition estimation of [1] for lower triangular matrices can be described as follows. Assume we have given a vector x which comes close to a maximum norm solution of $Lx = d$ with $\|d\| = 1$. Then $\sigma_{min}(L) \approx 1/\|x\|$ and $\tilde{\sigma}_{min}(L) = 1/\|x\|$ is used as an approximation. To find an approximation to the minimal singular value of

$$L' = \begin{pmatrix} L & 0 \\ v^T & \gamma \end{pmatrix}, \tag{1}$$

one searches for $s \equiv \sin\phi$ and $c \equiv \cos\phi$ such that

$$\begin{pmatrix} L & 0 \\ v^T & \gamma \end{pmatrix} \begin{pmatrix} sx \\ \frac{c - s\alpha}{\gamma} \end{pmatrix} = \begin{pmatrix} sd \\ c \end{pmatrix}, \tag{2}$$

where $\alpha = v^T x$. The parameters s and c are chosen such that the new approximate singular vector $\begin{pmatrix} sx \\ \frac{c-s\alpha}{\gamma} \end{pmatrix}$ has maximal norm. In other words, s and c solve

$$\max_{c,s} \quad s^2\|x\|^2 + \frac{(c-s\alpha)^2}{\gamma^2} \qquad \text{subject to} \qquad c^2+s^2=1. \tag{3}$$

The solution to this maximization problem can be found in [1]. With the chosen c and s, the resulting approximate minimal singular value is

$$\tilde{\sigma}_{min}(L') = \frac{1}{\sqrt{s^2\|x\|^2 + \frac{(c-s\alpha)^2}{\gamma^2}}} \approx \sigma_{min}(L').$$

One can estimate the largest singular value similarly. Assume we have given a vector x which comes close to a *minimum* norm solution of $Lx = d$ with $\|d\| = 1$. Then $\sigma_{max}(L) \approx 1/\|x\|$ and $\tilde{\sigma}_{max}(L) = 1/\|x\|$ is used as an approximation. To find an approximation of $\sigma_{max}(L')$, solve the *minimization* problem

$$\min_{c,s} \quad s^2\|x\|^2 + \frac{(c-s\alpha)^2}{\gamma^2} \qquad \text{subject to} \qquad c^2+s^2=1, \tag{4}$$

and define, with the resulting c and s, the estimate as

$$\tilde{\sigma}_{max}(L') = \frac{1}{\sqrt{s^2\|x\|^2 + \left(\frac{c-s\alpha}{\gamma}\right)^2}} \approx \sigma_{max}(L').$$

3 An alternative incremental condition estimation technique

Now suppose we want to estimate the condition number of an *upper* triangular matrix

$$R' = \begin{pmatrix} R & v \\ 0 & \gamma \end{pmatrix}. \tag{5}$$

Of course, one may apply the technique mentioned above to $(R')^T$, exploiting the fact that singular values are invariant under transposition. This would amount to approximating the extremal *right* singular vectors of $(R')^T$, although in some cases the extremal *left* singular vectors of $(R')^T$ may be easier to approach. To find left singular vectors (i.e. right singular vectors of R'), we set up the problem as follows. With an approximate singular vector x satisfying $Rx = d$, $\|d\| = 1$, we will search for numbers α, β such that

$$R' = \begin{pmatrix} R & v \\ 0 & \gamma \end{pmatrix} \begin{pmatrix} \beta x \\ \alpha \end{pmatrix} = \begin{pmatrix} \beta d + \alpha v \\ \gamma\alpha \end{pmatrix}. \tag{6}$$

Then we ask the numbers α, β to satisfy

$$\text{opt}_{c,s} \quad \beta^2\|x\|^2 + \alpha^2 \qquad \text{subject to} \qquad \beta^2 + \alpha^2\|v\|^2 + 2\alpha\beta v^T d + \gamma^2\alpha^2 = 1, \tag{7}$$

where opt stays for maximization if we approach $\sigma_{min}(R')$ and for minimization if we approach $\sigma_{max}(R')$.

Introduce the abbreviations $a = \|v\|^2 + \gamma^2$ and $b = v^T d$. According to elementary geometry the numbers α, β satisfying the constraint in (7) lie on an ellipse with the origin as center and semi-axes rotated by an angle of

$$\phi = 1/2 \arctan \frac{2b}{1-a};$$

the lengths of the semi-axes a' and b' are

$$a' = a\cos^2\phi - 2b\cos\phi\sin\phi + \sin^2\phi, \qquad b' = a\sin^2\phi + 2b\cos\phi\sin\phi + \cos^2\phi.$$

A parametrization of the ellipse (and of α and β) is

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\frac{\cos\phi}{\sqrt{a'}}\cos(t) + \frac{\sin\phi}{\sqrt{b'}}\sin(t)\\ \frac{\cos\phi}{\sqrt{b'}}\sin(t) + \frac{\sin\phi}{\sqrt{a'}}\cos(t)\end{pmatrix}, \qquad 0 \le t \le 2\pi, \tag{8}$$

and (7) can be written as

$$\mathrm{opt}_{0\le t\le 2\pi} \quad \begin{pmatrix}\sin(t) & \cos(t)\end{pmatrix} \begin{pmatrix} m_{11} & m_{12}\\ m_{21} & m_{22}\end{pmatrix} \begin{pmatrix}\sin(t)\\ \cos(t)\end{pmatrix}, \qquad \text{where}$$

$$\begin{pmatrix} m_{11} & m_{12}\\ m_{21} & m_{22}\end{pmatrix} = \begin{pmatrix} \frac{\cos^2\phi}{b'}\|x\|^2 + \frac{\sin^2\phi}{b'} & \frac{\cos\phi}{\sqrt{b'}}\frac{\sin\phi}{\sqrt{a'}}\|x\|^2 + \frac{\sin\phi}{\sqrt{b'}}\frac{\cos\phi}{\sqrt{a'}}\\ \frac{\cos\phi}{\sqrt{b'}}\frac{\sin\phi}{\sqrt{a'}}\|x\|^2 + \frac{\sin\phi}{\sqrt{b'}}\frac{\cos\phi}{\sqrt{a'}} & \frac{\sin^2\phi}{a'}\|x\|^2 + \frac{\cos^2\phi}{a'} \end{pmatrix}.$$

In case the optimization problem is a minimization problem for approximating the largest singular value, one determines the smallest eigenvalue of the matrix $M = (m_{ij})_{1\le i,j\le 2}$ and the corresponding normalized eigenvector is substituted in (8), yielding a solution of (7). When approximating the smallest singular value one determines the largest eigenvalue of M and the corresponding normalized eigenvector is substituted in (8), yielding a solution of (7). The eigenvalues of M are

$$\lambda_\pm = \frac{m_{11} + m_{22} \pm \sqrt{(m_{11} - m_{22})^2 + 4m_{12}^2}}{2}$$

and the corresponding normalized eigenvectors are

$$\frac{1}{\sqrt{1 + (\lambda_+ - m_{11})^2}} \begin{pmatrix}1\\ \lambda_+ - m_{11}\end{pmatrix}, \qquad \frac{1}{\sqrt{1 + (\lambda_+ - m_{22})^2}} \begin{pmatrix}1\\ -\lambda_+ + m_{22}\end{pmatrix}.$$

The technique of this section was proposed in [5]. Our description differs from [5] and represents an alternative derivation of this technique.

4 Combination of the two techniques

Clearly, the two described techniques do not give identical results in general. It is hard to say which one is better. The conclusion in [5] is that the newer technique is more suitable for sparse matrices, but otherwise superiority of a particular variant is not observed in the experiments.

In the special case where besides the triangular factorization the inverses of these factors are available, we can derive an improved incremental condition estimator. Inverse factors are computed, for example, as a by-product of the recently introduced BIF method [3]. At first sight it may seem trivial that condition estimation works better when the inverse of the matrix is available. This is, however, not the case; in fact, the improvement consists of a carefully selected

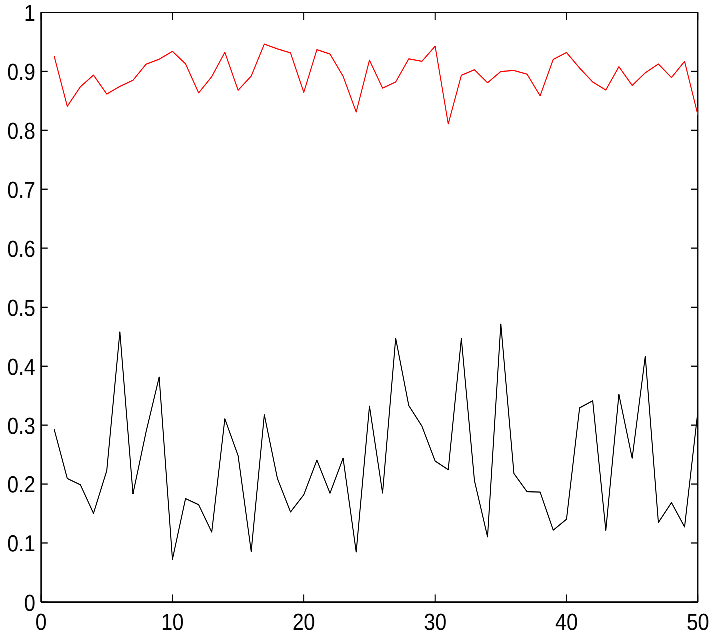


Figure 1: The values of $\frac{\tilde{\kappa}(L)^2}{\kappa(A)}$ (lower curve) and $\frac{\hat{\kappa}(L,L^{-1})^2}{\kappa(A)}$ (upper curve) for 50 random s.p.d. matrices of dimension 100.

combination of the techniques from [1] and [5]. Details on this combination are to be published in a forthcoming paper. Here we only present a numerical experiment.

We generated 50 random matrices B of dimension 100 with the command $\mathsf{B} = \mathsf{randn}(100, 100)$ in Matlab and we computed the Cholesky decompositions LL^T of the 50 symmetric positive definite matrices $A = BB^T$ with the BIF method, hence the factor L^{-1} was also computed. We first computed the condition number estimations $\tilde{\kappa}(L)$ obtained with the first technique (with (3)-(4)) from the factor L and then the improved condition number estimations $\hat{\kappa}(L, L^{-1})$ obtained with our combination of both techniques [1] and [5] from the factors L and L^{-1}. In Figure 1 we display the quality of these estimations through the number

$$\frac{\tilde{\kappa}(L)^2}{\kappa(A)}, \quad \text{resp.} \quad \frac{\hat{\kappa}(L, L^{-1})^2}{\kappa(A)}$$

where $\kappa(A)$ is the true condition number. Clearly, $\hat{\kappa}(L, L^{-1})$ is a much more accurate approximation.

Acknowledgement: This work is part of the Institutional Research Plan AV0Z10300504 and is supported by the project IAA100300802 of the Grant Agency of the Academy of Sciences of the Czech Republic.

References

[1] C.H. Bischof: *Incremental condition estimation.* SIAM J. Matrix Anal. Appl. 11, 1990, 312–322.

[2] C.H. Bischof, J.G. Lewis, D.J. Pierce: *Incremental condition estimation for sparse matrices.* SIAM J. Matrix Anal. Appl. 11, 1990, 644–659.,

[3] R. Bru, J. Marín, J. Mas, M. Tůma: *Balanced incomplete factorization.* SIAM J. Sci. Comput. 30, 2008, 2302–2318.

[4] J.J. Du Croz, N.J. Higham: *Stability of methods for matrix inversion.* IMA J. Numer. Anal. 12, 1992, 1–19.

[5] I.S. Duff, C. Vömel: *Incremental norm estimation for dense and sparse matrices.* BIT 42, 2002, 300–322.

Worst-case GMRES: characterization and examples

V. Faber, P. Tichý, J. Liesen

[2] Institute of Computer Science AS CR, Prague

Introduction

Let a nonsingular matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$ be given. Suppose that we apply the GMRES method with the initial guess $x_0 = 0$ to the linear system $Ax = b$. Then this method computes a sequence of iterates $x_k \in \mathcal{K}_k(A, b)$, so that the kth residual $r_k \equiv b - Ax_k$ satisfies

$$\|r_k\| \;=\; \min_{p\in\pi_k} \|\, p(A)b\,\| \,. \tag{1}$$

Here π_k denotes the set of polynomials of degree at most k and with value one at the origin, $\|\cdot\|$ denotes the Euclidean norm, and $\mathcal{K}_k(A, b) \equiv \text{span}\{b, Ab, \ldots A^{k-1}b\}$ is the kth Krylov subspace generated by A and b. Without loss of generality we will assume that $\|b\| = 1$.

A common approach for investigating the GMRES convergence behavior is to bound (1) independently of b, and thus to study the algorithm's worst-case behavior. In particular, for each iteration step k one may analyze the *worst-case GMRES approximation*

$$\psi_k(A) \;\equiv\; \max_{\|b\|=1} \min_{p\in\pi_k} \|\, p(A)b\,\| \,. \tag{2}$$

It is clear that there exists a starting vector $w = w(A, k)$ and the corresponding GMRES polynomial $p_{k,w} \in \pi_k$ such that $\psi_k(A) = \|p_{k,w}(A)w\|$. Such a vector and polynomial will be called a *worst-case GMRES starting vector* and a *worst-case GMRES polynomial for A and step k*.

Using the submultiplicativity of the Euclidean norm (or by changing the order of maximization and minimization in (2)), we can easily find the following upper bound on (2),

$$\psi_k(A) \;\leq\; \min_{p\in\pi_k} \|p(A)\| \;=\; \min_{p\in\pi_k} \max_{\|b\|=1} \|\, p(A)b\,\| \;\equiv\; \varphi_k(A)\,. \tag{3}$$

The quantity $\varphi_k(A)$, called the kth *ideal GMRES approximation*, has been introduced by Greenbaum and Trefethen [4]. The polynomial for which the minimum is attained in (3) is called the kth *ideal GMRES polynomial of A*.

After the 1994 paper [4], several studies have been devoted to the problem of characterizing the relation between $\psi_k(A)$ and $\varphi_k(A)$, and in particular the tightness of the inequality (3). The best known result is that (3) is an equality for all $k \geq 0$, whenever A is normal [3, 5]. Some nonnormal matrices A are known for which $\psi_k(A) < \varphi_k(A)$, even $\psi_k(A) \ll \varphi_k(A)$, for certain k, see [1, 8]. However, it is still an open problem whether for larger classes of nonnormal matrices the quantity $\varphi_k(A)$ indeed represents the essence of the GMRES process.

In this contribution we concentrate mainly on characterization of the worst-case GMRES problem (2), and present results of our recent paper [2]. We will show that worst-case starting vectors have some special properties. In particular, they satisfy the so called cross-equality and they are always right singular vectors of the matrix equal to the corresponding worst-case GMRES polynomial in the variable A. While the ideal GMRES polynomial is always unique, we will show that a worst-case GMRES polynomial need not be unique.

Special properties of worst-case starting vectors

The following theorem shows that if we apply GMRES to A and a worst-case starting vector w, and afterwards GMRES to A^T and the previous (normalized) residual vector, we obtain again the original starting vector w (up to a scaling factor). To emphasize that r_k is the kth GMRES residual for the matrix A and the starting vector b, we use the notation $r_k = \mathrm{GMRES}(A, b, k)$.

Theorem. *Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix, and k a positive integer, $k < d(A)$ where $d(A)$ denotes the degree of the minimal polynomial of A. Let $b^{(0)}$ be a unit norm worst-case GMRES starting vector for A and step k and consider the following process:*

$$\begin{aligned} r_k &= \mathrm{GMRES}(A, b^{(0)}, k) \\ b^{(1)} &= \frac{r_k}{\|r_k\|} \\ s_k &= \mathrm{GMRES}(A^T, b^{(1)}, k) \\ b^{(2)} &= \frac{s_k}{\|s_k\|}. \end{aligned}$$

Then

$$b^{(0)} = b^{(2)} \qquad \text{and} \qquad \|s_k\| = \|r_k\| = \psi_k(A).$$

This is an example of what we call the *cross-equality* (this term has been coined by Zavorin in an unpublished technical report [9]). Next, we will present and discuss the following result.

Theorem. *Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix, and k a positive integer, $k < d(A)$. If w is a unit norm worst-case GMRES starting vector for A and step k and $p_{k,w} \in \pi_k$ the corresponding GMRES polynomial, then $\psi_k(A)$ is a singular value of $p_{k,w}(A)$ and w is a corresponding right singular vector of $p_{k,w}(A)$.*

Uniqueness

We first summarize the known results on uniqueness of the solution of the worst-case GMRES problem (2) and the ideal GMRES problem (3).

Lemma. *Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix, and k a positive integer, $k < d(A)$. Then*

1. *the kth ideal GMRES polynomial is unique [4, 6];*
2. *if $\psi_k(A) = \varphi_k(A)$, then the kth worst-case GMRES polynomial is unique, and it is equal to the kth ideal GMRES polynomial of A [7].*

Based on the results of the previous theorems we will show that the kth worst-case GMRES polynomial need not be unique, if $\psi_k(A) < \varphi_k(A)$. Note that the condition $\psi_k(A) < \varphi_k(A)$ is a necessary but not a sufficient condition for the non-uniqueness of the kth worst-case GMRES polynomial. This phenomenon will be demonstrated numerically on a 4 × 4 matrix from [8].

Acknowledgements: The work of Petr Tichý was supported by the project M100300901 of the institutional support of ASCR, by the GAAS grant IAA100300802, and by the Institutional Research Plan AV0Z10300504. The work of Jörg Liesen was supported by the Heisenberg Program of the Deutsche Forschungsgemeinschaft.

References

[1] V. Faber, W. Joubert, E. Knill, T. Manteuffel: *Minimal residual method stronger than polynomial preconditioning.* SIAM J. Matrix Anal. Appl. 17, 1996, 707–729.

[2] V. Faber, P. Tichý, J. Liesen: *Worst-case GMRES: characterization and examples*, in preparation, 2011.

[3] A. Greenbaum, L. Gurvits: *Max-min properties of matrix factor norms.* SIAM J. Sci. Comput. 15, 1994, 348–358.

[4] A. Greenbaum, L.N. Trefethen: *GMRES/CR and Arnoldi/Lanczos as matrix approximation problems.* SIAM J. Sci. Comput. 15, 1994, 359–368.

[5] W. Joubert: *A robust GMRES-based adaptive polynomial preconditioning algorithm for nonsymmetric linear systems.* SIAM J. Sci. Comput. 15, 1994, 427–439.

[6] J. Liesen, P. Tichý: *On best approximations of polynomials in matrices in the matrix 2-norm.* SIAM J. Matrix Anal. Appl. 31, 2009, 853–863.

[7] P. Tichý, J. Liesen, V. Faber: *On worst-case GMRES, ideal GMRES, and the polynomial numerical hull of a Jordan block.* Electron. Trans. Numer. Anal. 26, 2007, 453–473.

[8] K.C. Toh: *GMRES vs. ideal GMRES.* SIAM J. Matrix Anal. Appl. 18, 1997, 30–36.

[9] I. Zavorin: *Spectral factorization of the Krylov matrix and convergence of GMRES.* Tech. Report CS-TR-4309, Computer Science Department, University of Maryland, 2001.

Backward error in linear least squares problems: estimates and their accuracy

S. Gratton, P. Jiránek, D. Titley-Peloquin

INPT-IRIT, University of Toulouse and ENSEEIHT
CERFACS, Toulouse
Mathematical Institute, University of Oxford

We consider a linear least squares (LS) problem

$$\text{find } \hat{x} \in \mathbb{R}^n \text{ such that } \quad \|b - A\hat{x}\|_2 = \min_{x\in\mathbb{R}^n} \|b - Ax\|_2, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, m and n are positive integers, $b \in \mathbb{R}^m$, both A and b are nonzero, and $\|v\|_2 = \sqrt{v^T v}$ denotes the Euclidean norm. The vector $\hat{x}$ is a solution of the LS problem (1) if and only if $\hat{x}$ satisfies the system of normal equations $A^T Ax = A^T b$ and provided that A has full column rank the problem (1) is uniquely solvable with $\hat{x} = (A^T A)^{-1} A^T b \equiv A^\dagger b$, where $A^\dagger$ is the pseudo-inverse of A. For more information, see, e.g., [1, 3, 10].

Let $x \in \mathbb{R}^n$ be an approximation to the solution $\hat{x}$ of the LS problem (1). We are interested in computing the backward error associated with the approximation x, i.e., we want to find the size of "smallest" perturbations E and f of the data A and b, respectively, such that x is the solution of the perturbed LS problem with the matrix $A + E$ and the right-hand side $b + f$. In [16] Waldén, Karlson, and Sun provide an explicit expression for the backward error μ defined by

$$\mu \equiv \min_{E,f}\{\|[E, \theta f]\|_F;\ (A + E)^T[b + f - (A + E)x] = 0\}, \tag{2}$$

where θ is a given positive weighting parameter and $\|\cdot\|_F$ denotes the Frobenius matrix norm. We denote by

$$\omega \equiv \min_{E,f}\{\|[E, \theta f]\|_F;\ (A + E)x = b + f\} = \frac{\theta\|r\|_2}{\sqrt{1 + \theta^2\|x\|_2^2}}, \qquad r \equiv b - Ax, \tag{3}$$

the backward error of x associated with the linear equations $Ax = b$ (see, e.g., [2, Theorem 2.2], [7, Problem 7.8]) and by $\sigma_{\min}(M)$ the minimal singular value of a matrix M. Then

$$\mu = \min\{\omega, \sigma_{\min}(M)\}, \tag{4}$$

where

$$M \equiv \begin{bmatrix} A^T \\ \omega(I - rr^\dagger) \end{bmatrix}, \tag{5}$$

see [16, Corollary 2.1] or [7, Theorem 20.5]. If the LS problem (1) is not compatible, then $\mu = \sigma_{\min}(M) < \omega$.

Computing the minimal singular value of the matrix M can be expensive and one can be rather interested in its good and cheaply computable estimate. First bounds of μ were given by Stewart [13, 14], which can be interpreted as Rayleigh quotient approximations to the minimal singular value of the matrix M in (5). The backward error μ can be bounded from above by $\bar{\mu}_1$ and $\bar{\mu}_2$ defined by

$$\bar{\mu}_1 \equiv \frac{\|Mr\|_2}{\|r\|_2} = \frac{\|A^T r\|_2}{\|r\|_2}, \qquad \bar{\mu}_2 \equiv \min_{0\neq s\perp\mathcal{R}(A)} \frac{\|Ms\|_2}{\|s\|_2} = \frac{\|M\hat{r}\|_2}{\|\hat{r}\|_2} = \frac{\theta\|P_A r\|_2}{\sqrt{1 + \theta^2\|x\|_2^2}}, \tag{6}$$

where $\hat{r} \equiv b - A\hat{x}$ is the residual associated with the solution of the LS problem (1) and $P_A \equiv AA^\dagger$ is the pseudo-inverse of A. Neither $\bar{\mu}_1$ nor $\bar{\mu}_2$ is however guaranteed to be a good estimate of the backward error μ. We have the bounds

$$\frac{1}{\sqrt{\sigma_{\max}^2(A)/\omega^2 + 1}}\,\bar{\mu}_1 \leq \mu \leq \bar{\mu}_1, \qquad \frac{1}{\sqrt{\omega^2/\sigma_{\min}^2(A) + 1}}\,\bar{\mu}_2 \leq \mu \leq \bar{\mu}_2.$$

Therefore $\min\{\bar{\mu}_1, \bar{\mu}_2\}$ is close to the backward error μ if the scaled residual norm ω is either larger than $\sigma_{\max}(A)$ or smaller than $\sigma_{\min}(A)$ (or at least of the same order of magnitude).

The literature suggests that the quantity

$$\nu \equiv \frac{\omega}{\|r\|_2} \|(A^T A + \omega^2 I)^{-1/2} A^T r\|_2 = \frac{\omega}{\|r\|_2} \left\| \begin{bmatrix} A \\ \omega I \end{bmatrix} \begin{bmatrix} A \\ \omega I \end{bmatrix}^\dagger \begin{bmatrix} r \\ 0 \end{bmatrix} \right\|_2$$

proposed by Karlson and Waldén [9] can be also used as an estimate of the backward error μ. In [6] Gu studies its accuracy and obtains (for A having full column rank and $r \neq 0$) the bounds, which can be expressed in the form

$$\frac{\|\hat{r}\|_2}{\|r\|_2} \leq \frac{\nu}{\mu} \leq \frac{1 + \sqrt{5}}{2} \tag{7}$$

(see also [5, Equation (1.5)]). In [4] Grcar shows that ν is asymptotically equal to μ in the sense that

$$\lim_{x \to \hat{x}} \frac{\nu}{\mu} = 1.$$

Methods of computing ν were considered by Grcar, Saunders, and Su [5] (see also [15]) and its efficient computation in the LSQR method [12, 11] was proposed in [8].

The bounds (7) show that ν is a good approximation to the backward error μ provided that x is a good approximation to $\hat{x}$ in the sense that the norms of their corresponding residuals are close to each other. The lower bound in (7) could suggest that ν might be a poor approximation of μ if $\|\hat{r}\|_2$ is much smaller than $\|r\|_2$. Numerical experience however shows that ν is a very good approximation of the LS backward error μ; see, e.g., [5, 15, 8]. Indeed, it appears that the estimate ν satisfies

$$\frac{1}{\sqrt{2}} \leq \frac{1}{\sqrt{2 - \left(\frac{\|\hat{r}\|_2^2}{\|r\|_2^2}\right)^2}} \leq \frac{\nu}{\mu} \leq 1.$$

Therefore the quantity ν is always an accurate estimate of the backward error μ.

References

[1] Å. Björck: *Numerical methods for least squares problems.* SIAM, Philadelphia, 1996.

[2] X.-W. Chang, C.C. Paige, and D. Titley-Peloquin: *Characterizing matrices that are consistent with given solutions.* SIAM J. Matrix Anal. Appl. 30 (4), 2008, 1406–1420.

[3] G.H. Golub and C.F. Van Loan: *Matrix computations.* The Johns Hopkins University Press, Baltimore, third edition, 1996.

[4] J.F. Grcar: *Optimal sensitivity analysis of linear least squares.* Technical Report LBNL-52434, Lawrence Berkeley National Laboratory, 2003.

[5] J.F. Grcar, M.A. Saunders, and Z. Su: *Estimates of optimal backward perturbations for linear least squares problems.* Technical Report SOL 2007-1, Department of Management Science and Engineering, Stanford University, 2007.

[6] M. Gu: *Backward perturbation bounds for linear least squares problems.* SIAM J. Matrix Anal. Appl. 20 (2), 1998, 363–372.

[7] N.J. Higham: *Accuracy and stability of numerical algorithms.* SIAM, Philadelphia, PA, 2nd edition, 2002.

[8] P. Jiránek and D. Titley-Peloquin: *Estimating the backward error in LSQR.* SIAM J. Matrix Anal. Appl. 31 (4), 2010, 2055–2074.

[9] R. Karlson and B. Waldén: *Estimation of optimal backward perturbation bounds for the linear least squares problem.* BIT 37 (4), 1997, 862–869.

[10] C.L. Lawson and R.J. Hanson: *Solving least squares problems.* Prentice-Hall, Englewood Cliffs, NJ, 1974.

[11] C.C. Paige and M.A. Saunders: *ALGORITHM 583; LSQR: Sparse linear equations and least-squares problems.* ACM Trans. Math. Software 8 (2), 1982, 195–209.

[12] C.C. Paige and M.A. Saunders: *LSQR: an algorithm for sparse linear equations and sparse least squares.* ACM Trans. Math. Software 8 (1), 1982, 43–71.

[13] G.W. Stewart: *An inverse perturbation theorem for the linear least squares problems.* SIGNUM Newsletter 10, 1975, 39–40.

[14] G.W. Stewart: *Research, development, and LINPACK.* In: Mathematical Software III, pages 1–14. Academic Press, New York, 1977.

[15] Z. Su: *Computational methods for least squares problems and clinical trials.* PhD thesis, Stanford University, 2005.

[16] B. Waldén, R. Karlson, and J.-G. Sun: *Optimal backward perturbation bounds for the linear least squares problem.* Numer. Linear Algebra Appl. 2 (3), 1995, 271–286.

Numerické metody vyššího řádu pro řešení transportních úloh

M. Hanuš, M. Smitková

Katedra matematiky, Západočeská univerzita, Plzeň

1 Úvod

Numerické modelování transportních procesů či v obecnejší rovině zákonů zachování se stále teší velké pozornosti, a to jak uživatelů (od biologů zajímajících se o proudění krve v cévách až např. po jaderné fyziky simulující šíření neutronového záření), tak vědecko-výzkumných pracovníků. Ti vytvářejí stále efektivnější a přesnější numerické metody schopné zachytit i složité fyzikální jevy, jimiž jsou úlohy tohoto typu často doprovázeny. Velmi oblíbené v této oblasti byly a stále jsou metody konečných objemů, v současnosti zejména moderní schémata s vysokým rozlišením. Dnes již však jejich dominantní postavení není zdaleka tak výrazné a do popředí se dostávají alternativní metody, jimiž se budeme zabývat v tomto příspěvku.

2 Testovací úloha

Pro účely testování a porovnání dále zmíněných metod byla vybrána úloha z čl. [2] a bylo pro ni metodou charakteristik sestrojeno přesné řešení.

- *Oblast:* čtverec $\Omega = [0,1] \times [0,1]$, s hranicí $\partial\Omega = \Gamma_- \cup \Gamma_+$, kde

$$\begin{aligned}\Gamma_- &= \{\mathbf{x} \in \partial\Omega : \mathbf{a}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\} \quad \text{(vtoková hrana)},\\ \Gamma_+ &= \{\mathbf{x} \in \partial\Omega : \mathbf{a}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \geq 0\} \quad \text{(odtoková hrana)},\end{aligned}$$

  $\mathbf{n}$ značí vektor vnější normály k $\partial\Omega$ a $\mathbf{x} = (x, y)$.

- *Rovnice:*

$$\nabla \cdot \big(\mathbf{a}(\mathbf{x})u(\mathbf{x})\big) + c(\mathbf{x})u(\mathbf{x}) = 0 \quad \text{v } \Omega, \qquad u(\mathbf{x}) = g(\mathbf{x}) \quad \text{na } \Gamma_-. \tag{1}$$

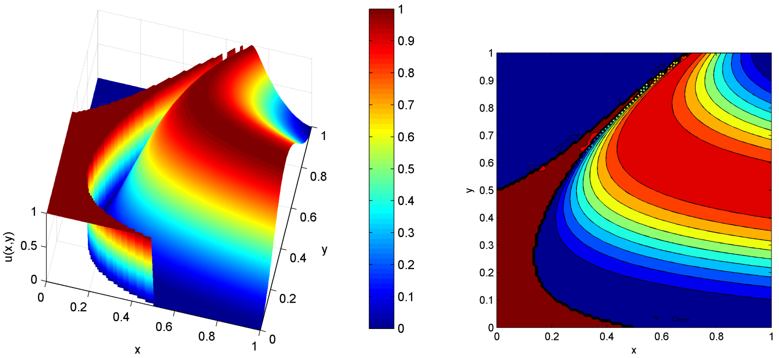
- *Parametry:*

$$\mathbf{a}(\mathbf{x}) = \begin{bmatrix} 10y^2 - 12x + 1 \\ 1 + y \end{bmatrix}, \qquad c(\mathbf{x}) = -\nabla \cdot \mathbf{a}(\mathbf{x}) \equiv 11.$$

- *Okrajové podmínky:*

$$g(\mathbf{x}) = \begin{cases} 0 & \text{pro } (x = 0 \wedge 0.5 < y \leq 1) \vee (0.5 < x \leq 1 \wedge y = 0),\\ 1 & \text{pro } (x = 0 \wedge 0 < y \leq 0.5) \vee (0 \leq x \leq 0.5 \wedge y = 0),\\ \sin^2(\pi y) & \text{pro } x = 1 \wedge 0 \leq y \leq 1.\end{cases}$$

- *Přesné řešení:* znázorněno na obr. 1.



Obrázek 1: Přesné řešení vyhodnocené ve 100×100 bodech čtverce Ω.

3 Metody konečných prvků (MKP)

MKP, populární zejména pro řešení diferenciálních úloh druhého a vyššího řádu, nebyly zpočátku pro transportní výpočty příliš atraktivní. Předpokládají totiž hladkost řešení, kterou nelze obecně v případě parciálních diferenciálních rovnic hyperbolického typu očekávat. Průlom učinil až článek [3], v němž byla představena metoda nespojitých konečných prvků ("Discontinuous Galerkin Method", dále jen DGM). Přestože DGM umožňuje využít příznivé vlastnosti MKP (geometrická flexibilita, snadno použitelná aproximace vysokého řádu atd.) i pro úlohy s nehladkým řešením, její použití je obvykle spojeno s většími výpočetními nároky než u klasické MKP. Zároveň proto probíhal vývoj tzv. stabilizovaných metod konečných prvků (SMKP), v nichž je zachována globálně spojitá aproximace řešení a problémy s jeho nízkou regularitou jsou adresovány úpravami diskrétní formulace.

DGM i SMKP využívají standardní rozklad (triangulaci) $\overline{\Omega} = \cup_{K \in \tau_h} \overline{K}$ oblasti Ω na množinu τ_h disjunktních elementů (v této práci čtverců) K a přibližné řešení u_h vyjadřují jako lineární kombinaci konečného počtu nad nimi definovaných bázových funkcí. Dosazením tohoto rozvoje do rovnice (1) a aplikací Galerkinovy metody je původní spojitá úloha v obou případech převedena na řešení soustavy lineárních rovnic pro neznámé koeficienty rozvoje. Praktické provedení tohoto postupu a tvar výsledné soustavy se však pro oba typy metod liší.

Z prostorových důvodů se zde budeme věnovat pouze nespojité Galerkinově metodě. Prostor bázových funkcí je pro ni definován jako

$$V_h = \{v \in L^2(\Omega); v|_K \in P^p(K)\ \forall K \in \tau_h\},$$

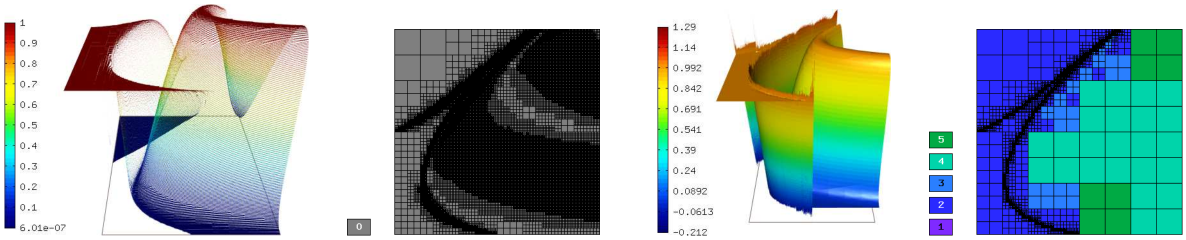
kde P^p představuje prostor polynomu stupně nejvýše p definovaných na elementu K. Klasický Galerkinův postup pro získání diskrétní verze dané úlohy vede v tomto případě (kdy je kvůli nedostatečné globální hladkosti funkcí z V_h nutné pro použití Greenovy věty integrovat po elementech) k jejímu následujícímu znění: Najdi $u_h \in V_h$ tak, aby $\forall v_h \in V_h$ platilo:

$$\sum_{K \in \tau_h} \int_K (-u_h \mathbf{a} \cdot \nabla v_h + c u_h v_h)\, \mathrm{d}\mathbf{x} + \sum_{e \not\subset \Gamma_-} \int_e \{\mathbf{a} u_h\}_{\mathbf{a}} \cdot [v_h]\, \mathrm{d}s = - \sum_{e \subset \Gamma_-} \int_e (\mathbf{a} \cdot \mathbf{n}) g v_h\, \mathrm{d}s,$$

$$\{\mathbf{a} u_h\}_{\mathbf{a}} = \begin{cases} \mathbf{a} u_h^L, & \text{když } \ \mathbf{a} \cdot \mathbf{n}^L > 0, \\ \mathbf{a} u_h^R, & \text{když } \ \mathbf{a} \cdot \mathbf{n}^L < 0, \\ \mathbf{a} \frac{u_h^L + u_h^R}{2}, & \text{když } \mathbf{a} \cdot \mathbf{n}^L = 0, \end{cases} \quad [v_h] = \begin{cases} v_h \mathbf{n}^L + v_h \mathbf{n}^R & \text{pro } e \not\subset \partial\Omega, \\ v_h \mathbf{n} & \text{pro } e \subset \partial\Omega, \end{cases}$$

kde e značí postupně hrany všech elementů τ_h a L, R sousední elementy na jejich stranách.

Na funkce $v_h \in V_h$ se nekladou žádné požadavky z hlediska spojitosti mezi elementy a teoreticky ani z hlediska maximálního stupně p. To umožňuje relativně snadnou implementaci adaptivního zjemňování sítě (h-adaptivita) a zvyšování řádu aproximace (p-adaptivita) bez starosti o konformitu elementů. Předběžné výsledky adaptivního výpočtu jsou na obr. 2. Byla použita jednoduchá automatická adaptivita, řízená velikostí L2 normy rozdílu řešení na dané síti a jeho L2-projekce na globálně zhrubenou síť. Na obrázcích je patrná dostatečná schopnost h-adaptivity zachytit nespojitosti v řešení. Při použití elementů vyššího řádu je lépe aproximováno řešení na okolí nespojitosti blízko odtokové hrany, objevují se v něm však nerealistické oscilace a ukazuje se, že upwinding zahrnutý v definici $\{\cdot\}_{\mathbf{a}}$ zde sám o sobě k zaručení stability nestačí.



(a) $p = 0$, h-adapt. → 83680 NDOF

(b) hp-adaptivita → 85220 NDOF

Obrázek 2: Adaptivní DGM. NDOF ... počet neznámých po konvergenci adaptačního procesu. Čísla příslušná barvám elementů odpovídají řádu na nich def. bázových funkcí.

4 Residual distribution schemes (RDS)

Další skupinou metod, jimž je v poslední době věnována značná pozornost, jsou metody typu RDS. Ty vznikly na základě myšlenek inspirovaných přístupy metody konečných objemů i MKP a přirozeně se snaží zachovat dobré vlastnosti obou. Z prvně jmenované tak např. robustnost danou silným vztahem k fyzikální podstatě řešeného problému, z druhé např. kompaktnost diskretizace i pro aproximaci vyššího řádu, jež umožňuje vývoj efektivních implicitních řešičů a jednoduchou paralelizaci (viz [1]).

Pro řešení testovací úlohy nestacionárním schématem typu RDS použijeme metodu ustalování. Pro nestacionární řešení vyššího řádu přesnosti v čase by bylo nutné použít konzistentní časovou diskretizaci, zde stačí nekonzistentní časová diskretizace (detaily viz [1]).

Uvažujme skalární zákon zachování $u_t + \nabla \cdot (\mathbf{a}u) = 0$ a libovolnou triangulaci oblasti Ω. Řešení je, obdobně jako v MKP 1. řádu, aproximováno spojitou funkcí lineární na každém trojúhelníku, $u(x, y, t) \approx \sum_i u_i(t) N_i(x, y)$, kde $u_i(t)$ je hodnota funkce u v uzlu i a N_i jsou standardní P1 bázové funkce.

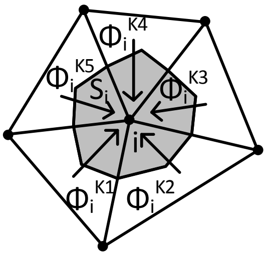
Definujeme reziduum na trojúhelníku K jako

$$\phi^K = -\int_K u_t \,\mathrm{d}\mathbf{x} = \oint_{\partial K} (\bar{\mathbf{a}}u) \cdot \mathrm{d}\mathbf{n}, \qquad \text{kde} \quad \bar{\mathbf{a}} = \frac{1}{K}\int_K \mathbf{a}\,\mathrm{d}\mathbf{x}.$$

Metoda RDS je založena na distribuci částí tohoto rezidua na sousední uzly. Vyjdeme-li z nekonzistentní formulace a Eulerovy explicitní integrace v čase, získáme následující schéma

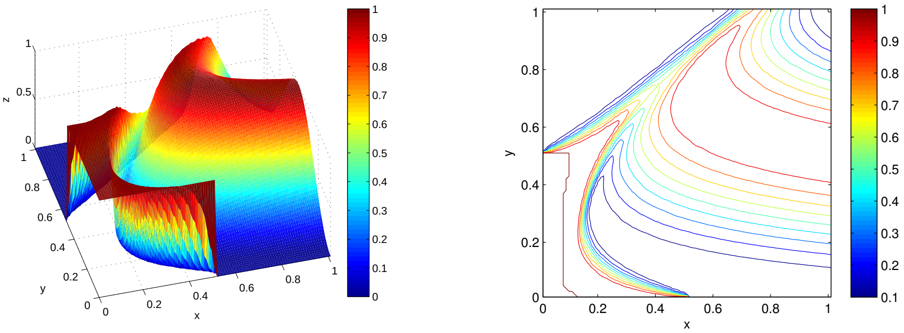
$$u_i^{n+1} = u_i^n - \frac{\Delta t}{S_i} \sum_T \beta_i^K \phi^K = u_i^n - \frac{\Delta t}{S_i} \sum_T \phi_i^K,$$

kde S_i je obsah duální buňky okolo uzlu i, tj. 1/3 obsahu všech trojúhelníků se společným vrcholem v uzlu i. Pro daný trojúhelník požadujeme, aby $\beta_1^K + \beta_2^K + \beta_3^K = 1$ (konzervativita). Distribuční koeficienty β mohou být stanoveny různými způsoby s ohledem na požadované vlastnosti monotónnosti a přesnosti řešení, kompaktní stencil zůstává zachován. Formálně definujeme distribuovaná rezidua jako $\phi_i^K = \beta_i^K \phi^K$.



Obrázek 3: Geometrické znázornění základních prvků RDS.

Z metod typu RDS jsme vybrali N (Narrow) schéma s $\phi_i^{K,N} = -\frac{k_i^+}{\sum_j k_j^+} \sum_j k_j^- (q_i^n - q_j^n)$ (monotónní lineární 1. řádu). Čísla k_i, definovaná jako $k_i = \frac{1}{2}\mathbf{a} \cdot \mathbf{n}_i$, nám dovolují rozlišit mezi vtokovými a odtokovými stranami a vrcholy trojúhelníka. Vektory $\mathbf{n}_i$ jsou definované jako vnitřní normály trojúhelníku o velikosti rovné délce příslušné strany. Pro více informací viz [1].



Obrázek 4: Numerické výsledky pro N schéma.

5 Závěr

Předběžné výsledky prezentované výše slibují použitelnost RDS i DGM pro řešení netriviálních transportních úloh. Obě metody však mají své neduhy (patrné při porovnání obr. 2 a 4 s obr. 1, na jejichž odstranění autoři textu v současné době pracují. Na semináři pak budou RDS, DGM i SMKP důkladněji porovnány.

Reference

[1] H. Deconinck, M. Ricchiuto, K. Sermeus: *Introduction to residual distribution schemes and comparison with stabilized finite elements.* In: H. Deconinck (Ed.), 33rd VKI Lecture Series CFD. Von Karman Institute, Sint-Genesius-Rode, 2003.

[2] P. Houston, R. Rannacher, E. Süli: *A posteriori error analysis for stabilised finite element approximations of transport problems.* In: Comput. Methods Appl. Mech. Engrg. 190, 2000, 1483–1508.

[3] W.H. Reed, T.R. Hill: *Triangular mesh methods for the neutron transport equation.* Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

Shape optimization in 2D contact problems with given friction and a solution-dependent coefficient of friction

J. Haslinger, J. V. Outrata, R. Pathó

[1,3] Charles University in Prague
[2] Institute of Information Theory and Automation AS CR, Prague

1 Introduction

The contribution deals with shape optimization of elastic bodies in unilateral contact. We aim at extending existing results (see [1] and [2]) to the case of contact problems, where the coefficient of friction depends on the solution. To this end, let us consider the two-dimensional Signorini problem, coupled with the physically less accurate model of given friction, but assume a solution-dependent coefficient of friction. For analysis of the shape optimization problem in the continuous, infinite-dimensional setting, its finite-dimensional approximation based on the finite-element method and for convergence analysis the reader is kindly referred to [4]. Our presentation starts with the so-called mixed formulation of the algebraic state problem, involving Lagrange multipliers for the normal contact displacement. It can be shown that if the coefficient of friction is Lipschitz continuous with a sufficiently small modulus, then the algebraic state problem is uniquely solvable and its solution is a Lipschitz continuous function of the control variable, describing the shape of the elastic body. In [2] its authors proposed the implicit programming approach (ImP) combined with sensitivity analysis based on the generalized differential calculus of Mordukhovich (see [5]) for the numerical solution of contact shape optimization problems involving the Coulomb law of friction. We shall adapt their approach to our case and point out the differences and difficulties compared to [2].

2 The state problem

Let an elastic body be represented by a domain $\Omega \subset \mathbb{R}^2$ with Lipschitz boundary $\partial\Omega$. Let $\partial\Omega$ be split into three non-empty, disjoint parts Γ_u, Γ_P and Γ_c with different boundary conditions: on Γ_u the body is fixed, while surface tractions of density $\boldsymbol{P} = (P_1, P_2)$ act along Γ_P. On Γ_c, representing the contact part of $\partial\Omega$, the body is *unilaterally supported* by the rigid foundation $O = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_2 \le 0\}$. In addition to the non-penetration conditions, we shall consider effects of friction between Ω and O. We use the friction law of Tresca type, i.e. with an a-priori given slip bound $g : \Gamma_c \to \mathbb{R}_+$, but with a coefficient of friction $\mathcal{F}$ which depends on the solution. Thus the friction conditions on Γ_c read as follows:

$$\left.\begin{array}{lcl} u_1 = 0 & \Longrightarrow & |T_1(\boldsymbol{u})| \le \mathcal{F}(0)g \\ u_1 \neq 0 & \Longrightarrow & T_1(\boldsymbol{u}) = -\mathrm{sgn}(u_1)\mathcal{F}(|u_1|)g \end{array}\right\} \text{ on } \Gamma_c,$$

where $T_1(\boldsymbol{u}) : \partial\Omega \to \mathbb{R}$ stands for the first component of the stress vector associated with $\boldsymbol{u}$. The equilibrium state of Ω is characterized by a displacement vector $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ which satisfies the system of linear equilibrium equations in Ω, the classical boundary conditions on Γ_u, Γ_P and the unilateral and friction conditions on Γ_c.

Let the contact boundary Γ_c be piecewise linear, given by a vector $\boldsymbol{\alpha} \in \mathcal{U}_{ad}$, where $\mathcal{U}_{ad} \subset \mathbb{R}^p_+$ is the set of admissible control variables (p corresponds to the number of contact nodes). Following the finite element approximation as described in [4], we define the discretized Signorini problem with given friction and a solution-dependent coefficient of friction as follows:

$$\left.\begin{aligned}&\text{Find } (\boldsymbol{u},\boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^p_+ \text{ such that:}\\ &\langle \mathbb{A}(\boldsymbol{\alpha})\boldsymbol{u}, \boldsymbol{v}-\boldsymbol{u}\rangle_n + \sum_{i=1}^p \omega_i(\boldsymbol{\alpha})\mathcal{F}(|(\boldsymbol{u}_\tau)_i|)\big(|(\boldsymbol{v}_\tau)_i| - |(\boldsymbol{u}_\tau)_i|\big)\\ &\qquad \geq \langle \boldsymbol{L}(\boldsymbol{\alpha}), \boldsymbol{v}-\boldsymbol{u}\rangle_n + \langle \boldsymbol{\lambda}, \boldsymbol{v}_\nu - \boldsymbol{u}_\nu\rangle_p \quad \forall \boldsymbol{v} \in \mathbb{R}^n,\\ &\langle \boldsymbol{\mu}-\boldsymbol{\lambda}, \boldsymbol{u}_\nu + \boldsymbol{\alpha}\rangle_p \geq 0 \quad \forall \boldsymbol{\mu} \in \mathbb{R}^p_+,\end{aligned}\right\} \qquad (\mathcal{M}(\boldsymbol{\alpha}))$$

where $\boldsymbol{v}_\nu \in \mathbb{R}^p$ stands for the subvector of $\boldsymbol{v} \in \mathbb{R}^n$ consisting of the second components of the displacement vector $\boldsymbol{v}$ at all contact nodes. Analogously, $\boldsymbol{v}_\tau \in \mathbb{R}^p$ consists of the first components of $\boldsymbol{v}$ at the contact nodes. Further, $\mathbb{A} \in C^1(\mathcal{U}_{ad}; \mathbb{R}^{n\times n})$ and $\boldsymbol{L} \in C^1(\mathcal{U}_{ad}; \mathbb{R}^n)$ denote the matrix and vector-valued functions associating with any $\boldsymbol{\alpha} \in \mathcal{U}_{ad}$ the stiffness matrix $\mathbb{A}(\boldsymbol{\alpha})$ and the load vector $\boldsymbol{L}(\boldsymbol{\alpha})$, respectively. Let us note that the functions ω_i depend on the weights of a quadrature rule and on the values of g at the contact nodes, as well. We assume that $\omega_i \in C^1(\mathcal{U}_{ad}; (0,\infty))$ $\forall i = 1,\dots,p$.

In the rest of this paper we shall be working with the reduced form of the state problem only. The reduction of $(\mathcal{M}(\boldsymbol{\alpha}))$ consists in eliminating all components of the displacement field $\boldsymbol{u}$ corresponding to the non-contact nodes of the finite element partition of the domain $\overline{\Omega}(\boldsymbol{\alpha})$. One obtains a variational inequality in terms of the state variable $\boldsymbol{y} = (\boldsymbol{u}_\tau, \boldsymbol{u}_\nu, \boldsymbol{\lambda})^T \in (\mathbb{R}^p)^3$, defined on the contact zone, which may be formulated as the following *generalized equation* (GE):

$$\boldsymbol{0} \in F(\boldsymbol{\alpha}, \boldsymbol{y}) + Q(\boldsymbol{\alpha}, \boldsymbol{y}), \tag{1}$$

where

$$F(\boldsymbol{\alpha},\boldsymbol{y}) := \begin{pmatrix} \mathbb{A}_{\tau\tau}(\boldsymbol{\alpha}) & \mathbb{A}_{\tau\nu}(\boldsymbol{\alpha}) & 0\\ \mathbb{A}_{\nu\tau}(\boldsymbol{\alpha}) & \mathbb{A}_{\nu\nu}(\boldsymbol{\alpha}) & -\mathbb{I}\\ 0 & \mathbb{I} & 0\end{pmatrix}\boldsymbol{y} - \begin{pmatrix}\boldsymbol{L}_\tau(\boldsymbol{\alpha})\\ \boldsymbol{L}_\nu(\boldsymbol{\alpha})\\ -\boldsymbol{\alpha}\end{pmatrix}, \quad Q(\boldsymbol{\alpha},\boldsymbol{y}) := \begin{pmatrix} Q_1(\boldsymbol{\alpha},\boldsymbol{y}_1)\\ 0\\ N_{\mathbb{R}^p_+}(\boldsymbol{y}_3)\end{pmatrix}.$$

The multifunction $Q_1 : \mathcal{U}_{ad} \times \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is defined as:

$$\big(Q_1(\boldsymbol{\alpha}, \boldsymbol{u}_\tau)\big)_i := \omega_i(\boldsymbol{\alpha})\mathcal{F}(|(\boldsymbol{u}_\tau)_i|)\partial|(\boldsymbol{u}_\tau)_i| \quad \forall i = 1,\dots,p,$$

where "∂" denotes the subdifferential of convex functions, $N_{\mathbb{R}^p_+}(\cdot)$ is the normal cone in the sense of convex analysis and the submatrices $\mathbb{A}_{\tau\tau}$, $\mathbb{A}_{\tau\nu}$, $\mathbb{A}_{\nu\nu} \in \mathbb{R}^{p\times p}$ are parts of the Schur complement to the stiffness matrix.

Note, that the multivalued part Q of our state problem (1) *depends* on the control variable $\boldsymbol{\alpha}$ as well. This is a major difference compared to the problem investigated in [2], making sensitivity analysis more involved.

Let us conclude this section with the following result concerning solvability of (1).

Theorem 1. *Let $S : \boldsymbol{\alpha} \mapsto \{\boldsymbol{y} \in (\mathbb{R}^p)^3 \,|\, \boldsymbol{0} \in F(\boldsymbol{\alpha},\boldsymbol{y}) + Q(\boldsymbol{\alpha},\boldsymbol{y})\}$ denote the control-to-state mapping and let $\mathcal{F} : \mathbb{R}_+ \to \mathbb{R}_+$ be Lipschitz continuous with a sufficiently small modulus. Then S is single-valued and Lipschitz continuous in $\mathcal{U}_{ad}$.*

Proof. It follows from Theorem 10 and Theorem 11 in [4]. □

3 ImP and sensitivity analysis

Let $J : \mathcal{U}_{ad} \times (\mathbb{R}^p)^3 \to \mathbb{R}$ be a continuously differentiable cost functional. Then the shape optimization problem reads as:

$$\left.\begin{array}{ll} \text{minimize} & J(\boldsymbol{\alpha}, \boldsymbol{y}) \\ \text{subj. to} & \mathbf{0} \in F(\boldsymbol{\alpha}, \boldsymbol{y}) + Q(\boldsymbol{\alpha}, \boldsymbol{y}) \\ & \boldsymbol{\alpha} \in \mathcal{U}_{ad}. \end{array}\right\} \tag{$\mathbb{P}$}$$

In the sequel we shall assume that the assumptions of Theorem 1 are satisfied. The ImP method consists in reformulating ($\mathbb{P}$) as the nonlinear program:

$$\left.\begin{array}{ll} \text{minimize} & \mathcal{J}(\boldsymbol{\alpha}) := J(\boldsymbol{\alpha}, S(\boldsymbol{\alpha})) \\ \text{subj. to} & \boldsymbol{\alpha} \in \mathcal{U}_{ad}, \end{array}\right\} \tag{$\tilde{\mathbb{P}}$}$$

which may be solved by standard algorithms of nonsmooth optimization. Such algorithms, however, require knowledge of some subgradient information, usually in the form of one (arbitrary) subgradient from the Clarke subdifferential $\overline{\partial}\mathcal{J}$ at each iteration step. Following [2], we are not going to use Clarke's calculus (cf. [3]) to obtain the desired subgradient, but the substantially richer calculus developed by B. Mordukhovich. A straightforward application of this theory is the next result. For the rest of this section let $\bar{\boldsymbol{\alpha}} \in \mathcal{U}_{ad}$ be arbitrary and put $\bar{\boldsymbol{y}} := S(\bar{\boldsymbol{\alpha}})$.

Lemma 1. $\overline{\partial}\mathcal{J}(\bar{\boldsymbol{\alpha}}) \subset \nabla_\alpha J(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}) + D^*S(\bar{\boldsymbol{\alpha}})(\nabla_y J(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}))$.

Therefore, we immediately see that it suffices to determine one element of the (limiting) coderivative $D^*S(\bar{\boldsymbol{\alpha}})(\nabla_y J(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}})) = \{\boldsymbol{p}^* \in \mathbb{R}^p \,|\, (\boldsymbol{p}^*, -\nabla_y J(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}})) \in N_{\operatorname{Gr} S}(\bar{\boldsymbol{\alpha}})\}$, where $N_{\operatorname{Gr} S}$ stands for the (limiting) normal cone to the graph of S. To facilitate the computation of this quantity, we have the following result at hand:

Theorem 2. *For every $\boldsymbol{p}^* \in D^*S(\bar{\boldsymbol{\alpha}})(\nabla_y J(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}))$ there exists a vector $\boldsymbol{v}^* \in (\mathbb{R}^p)^3$ such that $(\boldsymbol{p}^*, \boldsymbol{v}^*)$ is a solution of the (limiting) adjoint GE:*

$$\begin{pmatrix} \boldsymbol{p}^* \\ -\nabla_y J(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}) \end{pmatrix} \in \nabla F(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}})^T \boldsymbol{v}^* + D^*Q(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}, -F(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}))(\boldsymbol{v}^*). \tag{AGE}$$

Proof. See Lemma 8 and Theorem 13 in [4]. □

The assertion of Theorem 2 is analoguos to that of Theorem 4.1 in [2], but for its derivation we had to verify a calmness condition ([4, Lemma 8]) instead of strong regularity of the GE ([2, Theorem 3.13]).

In the rest of this section we show how one may express the coderivative D^*Q in terms of the data of the problem. First of all, note that the components of Q are *decoupled* (this fact is a consequence of the assumed model of given friction), hence its coderivative can be computed componentwise:

$$\forall \boldsymbol{q}^* \in (\mathbb{R}^p)^3 : \quad D^*Q(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{q}})(\boldsymbol{q}^*) = \begin{pmatrix} D^*Q_1(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}_1, \bar{\boldsymbol{q}}_1)(\boldsymbol{q}_1^*) \\ 0 \\ D^*N_{\mathbb{R}^p_+}(\bar{\boldsymbol{y}}_3, \bar{\boldsymbol{q}}_3)(\boldsymbol{q}_3^*) \end{pmatrix},$$

at any reference point $(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{q}}) \in \operatorname{Gr} Q$. The third component is standard, therefore we shall deal with the first component only. Let us write the multifunction $Q_1 : \mathbb{R}^p \times \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ as a composition of an outer multifunction Z_1 and an inner single-valued, smooth mapping Ψ:

$$Q_1(\boldsymbol{\alpha}, \boldsymbol{u}) = (Z_1 \circ \Psi)(\boldsymbol{\alpha}, \boldsymbol{u}), \tag{2}$$

where

$$\Psi = (\Psi_1, \ldots, \Psi_p) : \mathbb{R}^p \times \mathbb{R}^p \to \big((0,\infty) \times \mathbb{R}\big)^p, \qquad \Psi_j(\boldsymbol{\alpha}, \boldsymbol{u}) := \big(\omega_j(\boldsymbol{\alpha}), u_j\big),$$

and Z_1 is a composite multifunction itself:

$$Z_1 : \big((0,\infty) \times \mathbb{R}\big)^p \rightrightarrows \mathbb{R}^p, \quad \boldsymbol{y} \mapsto \big(Z(\boldsymbol{y}_1), \ldots, Z(\boldsymbol{y}_p)\big),$$

with

$$Z : (0,\infty) \times \mathbb{R} \rightrightarrows \mathbb{R}, \quad (x_1, x_2) \mapsto x_1 \mathcal{F}(|x_2|) \partial |x_2|.$$

Now the chain rule from [6, Theorem 10.40] allows us to compute the coderivative of the composite multifunction (2) as follows:

Theorem 3. *Let $(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{q}}) \in Gr\, Q_1$ be such that the following condition holds:*

$$Ker \nabla \Psi(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{u}})^T \cap D^* Z_1(\Psi(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{u}}), \bar{\boldsymbol{q}})(\boldsymbol{0}) = \{\boldsymbol{0}\}. \tag{3}$$

Then:

$$\forall \boldsymbol{q}^* \in \mathbb{R}^p : \quad D^* Q_1(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{q}})(\boldsymbol{q}^*) \subset \nabla \Psi(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{u}})^T D^* Z_1(\Psi(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{u}}), \bar{\boldsymbol{q}})(\boldsymbol{q}^*) \tag{4}$$

Since the components of Z_1 are also decoupled, one may compute the coderivative on the right-hand side of (4) componentwise, i.e. in terms of coderivatives of the mapping Z. This is done in detail in Section 6.2 of [4], from which the validity of the qualification condition (3) follows as well.

Acknowledgement: The work of the first two authors was supported by the Grant Agency of the Czech Academy of Sciences, project no. IAA100750802. The third author would like to thank for support of the grant no. SVV-2010-261316.

References

[1] P. Beremlijski, J. Haslinger, M. Kočvara, J. V. Outrata: *Shape optimization in contact problems with Coulomb friction.* SIAM J. Opt. 13, 2002, 561–587.

[2] P. Beremlijski, J. Haslinger, M. Kočvara, R. Kučera, J. V. Outrata: *Shape optimization in three-dimensional contact problems with Coulomb friction.* SIAM J. Opt. 20, 2009, 416–444.

[3] F.F. Clarke: *Optimization and nonsmooth analysis.* John Wiley & Sons, New York, 1983.

[4] J. Haslinger, J.V. Outrata, R. Pathó: *Shape optimization in 2D contact problems with given friction and a solution-dependent coefficient of friction.* Submitted to Set-Valued and Variational Analysis.

[5] B.S. Mordukhovich: *Variational analysis and generalized differentiation, I: Basic Theory, II: Applications.* Grundlehren Series (Fundamental Principles of Mathematical Sciences), Vols. 330 and 331, Springer-Verlag, Berlin-Heidelberg, 2006.

[6] R.T. Rockafellar, R. Wets: *Variational analysis.* Springer-Verlag, Berlin, 1998.

Řešení benchmarkové úlohy transportu látky v diskrétní puklinové síti

M. Hokr, J. Havlíček

Technická univerzita v Liberci

1 Úvod

Téma článku vychází z potřeb studia vlastností horninového prostředí pro hodnocení bezpečnosti hlubinného úložiště vyhořelého jaderného paliva, hlavními výzvami pro matematické modelování v této oblasti jsou sdružené fyzikální procesy a složitost geometrické struktury prostředí.

Jedním z hlavních faktorů na bezpečnost úložiště izolační schopnost horniny, která je obvykle hodnocena přes její (ekvivaletní) hydraulickou vodivost, tj. celkový průtok vody přes určitý průřez. Skutečnou hodnocenou veličinou je ale rychlost průchodu rozpuštěných radionuklidů, která je sice pro porézní prostředí úměrná průtoku, ale pro nehomogenity typu puklin závisí na rozložení toku v objemu - rychle proudící „kanály“ proti méně vodivým puklinám s pomalým tokem. v této práci jsou na modelové úloze puklinové sítě určovány průnikové křivky a střední hodnota a rozptyl tzv. doby zdržení – srovnány jsou výpočty pomocí sledování částic (particle tracking) a pomocí rovnice advekčního transportu.

2 Popis úlohy a řešení

Úloha byla definována v projektu Decovalex [4], kde je tímto způsobem hodnocen vliv napjatosti na charakter toku a dobu zdržení částic. Výpočty navazují na dříve prezentované výpočty proudění pro různé stavy napjatosti [1, 2, 3], mimojiné i srovnáním způsobu hodnocení pomocí ekvivalentní vodivosti a pomocí rychlosti průchodu látky. V tomto textu se nezabýváme přímo vlivem napjatosti, jednotlivé varianty jsou chápány jako různé parametry puklinové sítě pro vyhodnocení proudění a transportu (v prezentaci bude popsáno v plném kontextu).

Geometrie úlohy je zadána seznamem 7797 puklin se souřadnicemi koncových bodů a velikostí rozevření (šířky) ve čtverci v rozsahu $-10 < x < 10$, $-10 < y < 10$. Okrajové podmínky pro proudění jsou zadány hodnotami tlaku (Dirichlet) po celém obvodu nebo na protilehlé stěny tak, aby generoval konstantní gradient 10^4 Pa/m (dvě varianty: vodorovně zprava doleva a svisle shora dolů) – obrázek 1. Úloha transportu látky je zadána okamžitým pulsním vstupem (vtok daného celkového množství látky za velmi krátký čas) do všech puklin na přítokové straně modelového čtverce.

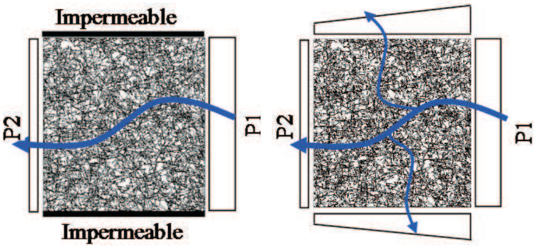
Úlohy proudění i transportu byly vypočteny softwarem FLOW123D vyvíjeným na pracovišti autorů [6]. Rovnice proudění je řešena smíšenou-hybridní metodou konečných prvků, jejíž výsledkem jsou diskrétní toky jednotlivými puklinami. Segmenty puklin mezi průsečíky jsou zároveň elementy diskretizace (z důvodů linearity v 1D segmentech není další dělení potřebné). Rovnice advektivního transportu je řešena metodou konečných objemů, s upwind vážením a explicitními časovými kroky. Volba časových kroků je řízena CFL podmínkou. Doba zdržení je určena jako vážený průměr z času pro jednotlivé části hmoty (=váhy) na výstupu za každý časový krok výpočtu.

Srovnávací výpočty pomocí softwaru NAPSAC využívají standardní metodu konečných prvků pro proudění (srovnání např. z hlediska splnění bilance hmoty je provedeno v [3]) a výpočet transportu byl proveden pomocí metody sledování částic (particle tracking). Doba zdržení je přímo výsledkem výpočtu pro každou jednotlivou částici. Jednotlivé výsledky byly zpracovány autory výpočtů ve zprávách [7, 5].

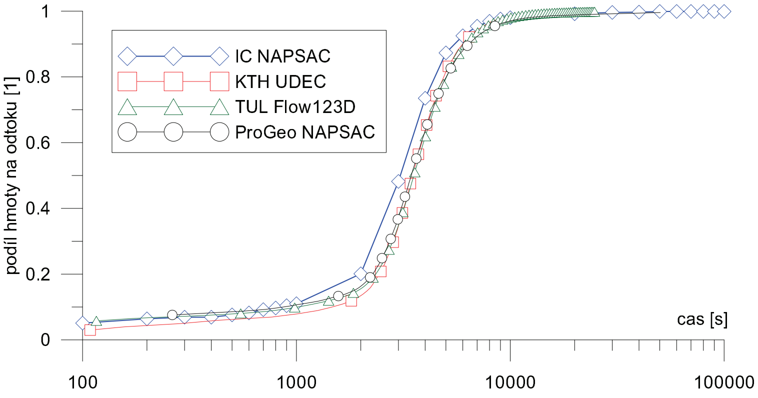
3 Výsledky

Výsledné průnikové křivky jsou určovány jako průběh v čase pro podíl hmoty (resp. počtu částic) proteklé odtokovými hranami modelového čtverce a celkové zadané hmoty (resp. počtu částic). Výsledky pro horizontální gradient jsou uvedeny na obrázku 2, kde jsou porovnány jednotlivé metody a softwary. Je vidět dobrá vzájemná shoda kromě výsledků IC. Zajímavým výsledkem je, že se ve sklonu křivky nijak neprojevuje numerická difuze z upwind metody (Flow123D) proti částicovým metodám (NAPSAC), což lze vysvětlit tím, že dominantním difuzním jevem je mísení roztoku resp. částic mezi různě „rychlými“ trajektoriemi v síti puklin.

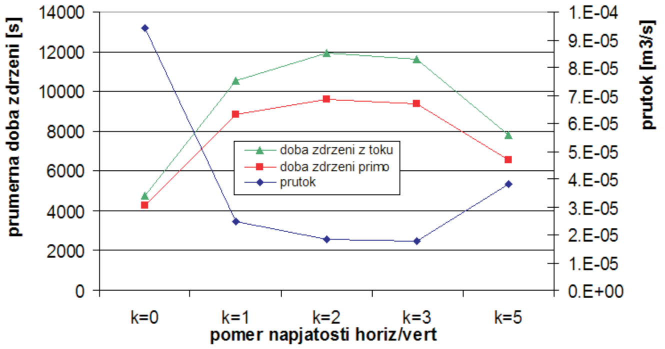
Na obr. 3 je další srovnání – hodnocení horniny přes celkový průtok a přes dobu zdržení (čas transportu). Jiný způsob vyjádření doby zdržení je možné dostat přímo z průtoku jako doba ideální výměny celkového objemu vody (celkový objem ku průtoku). Přestože tok se mění mezi jednotlivými variantami v mnohem vyšším poměru, obě vyjádření času zdržení dávají podobný průběh (mění se výrazně objem vody mezi variantami). Výsledky potvrzují předpoklad vzniku vodivých kanálů z několika konkrétních puklin, které i při snížení toku snižují dobu zdržení (tzv. channeling).



Obrázek 1: Schéma okrajových podmínek určujících tlakový gradient pro úlohu proudění – dvě varianty s propustnými nebo nepropustnými bočními stěnami.



Obrázek 2: Porovnání průnikových křivek (závislost hmoty na výstupu na čase) mezi jednotlivými modely a řešitelskými týmy Decovalex.



Obrázek 3: Porovnání doby zdržení (času transportu) určené přímo z výpočtu transportu a určené z celkového průtoku, proti průtoku samotnému, pro různé varianty parametrů puklin (vlivem napjatosti).

Poděkování: Tato práce byla realizována za podpory státních prostředků České republiky v rámci projektu VaV „Pokročilé sanační technologie a procesy“ č. 1M0554 v programu MŠMT Výzkumná centra a v rámci projektu č. FR-TI1/362 v programu MPO TIP. Dále byla financována SÚRAO, smlouva č. 2010/019/Slo.

Práce popsané v tomto článku byly provedeny v rámci mezinárodního projektu DECOVALEX (DEmonstration of COupled models and their VALidation against EXperiments). Názory vyjádřené v tomto článku jsou však názory autorů a nemusí být nutně názory financujících organizací.

Reference

[1] A. Baghbanan, L. Jing: *Hydraulic properties of fractured rock masses with correlated fracture length and aperture.* Int. J. Rock Mech. Min. Sci. 44, 2007, 704–719.

[2] M. Hokr, J. Kopal, J. Havlíček: *Řešení úlohy proudění v rozsáhlé diskrétní síti puklin v kontextu sdružených úloh proudění-mechanika.* In: SNA'09 Modelling and Simulation of Chalenging Engineering Problems (Blaheta, Starý, eds.), Ústav geoniky AV ČR, Ostrava, 2009.

[3] M. Hokr, J. Kopal, J. Březina, P. Rálek: *Sensitivity of results of the water flow problem in a discrete fracture network with large coefficient differences.* In: I. Dimov, S. Dimova, and N. Kolkovska (Eds.): NMA 2010, LNCS 6046, pp. 420–427, 2011. Springer-Verlag Berlin Heidelberg 2011.

[4] J. Hudson, L. Jing, I. Neretnieks: *Technical definition of the 2-D BMT problem for Task C*, DECOVALEX-2011 project, 5 May 2008.

[5] PROGEO s.r.o.: *Simulace transportu pomocí metody particle tracking.* Technická zpráva 2010.

[6] O. Severýn, M. Hokr, J. Královcová, J. Kopal, M. Tauchman: *Flow123D: Numerical simulation software for flow and solute transport problems in combination of fracture network and continuum.* Technical Report, TU Liberec, 2008.

[7] J. Hudson, L. Jing (eds.): *Task C: Integrated assessment of THMC coupled processes in single fractures and fractured rocks.* DECOVALEX-2011 Project Progress Report, Stage2 (in preparation).

Parallel implementations of Total-FETI-1 algorithm for contact problems using PETSc

D. Horák, Z. Dostál

VŠB-Technical University of Ostrava

1 Introduction

Domain decomposition method is one of the most successful methods of solution of elliptic partial differential equations describing many technical problems, which is based on “divide and conquer” strategy. The FETI (Finite Element Tearing and Interconnecting) method proposed by Farhat and Roux turned out to be one of the most successful algorithms for parallel solution of these problems. The FETI-1 method is based on the decomposition of the spatial domain into non-overlapping subdomains that are "glued" by Lagrange multipliers. Efficiency of the FETI-1 method was further improved by introducing special projectors and preconditioners. By projecting the Lagrange multipliers in each iteration onto an auxiliary space to enforce continuity of the primal solutions at the crosspoints, Farhat, Mandel and Tezaur obtained a faster converging FETI method for plate and shell problems - FETI-2. Similar effect was achieved by a variant called the Dual-Primal FETI method FETI-DP, introduced by Farhat et al., where the continuity of the primal solution at crosspoints is implemented directly into the formulation of the primal problem. The FETI-DPC algorithm for nonlinear problems is based on active set strategies and additional planning steps. Total-FETI-1 (TFETI-1) by Dostal simplifies the inversion of stiffness matrices of subdomains by using Lagrange multipliers not only for gluing the subdomains along the auxiliary interfaces, but also for implementation of the Dirichlet boundary conditions. This method may be even more efficient than the original FETI-1.

FETI methods are even more successful for the solution of variational inequalities. The reason is that duality reduces not only large primal problem to smaller dual, relatively well conditioned strictly convex iteratively solved QP problem but also transforms the general inequality constraints into the nonnegativity constraints so that efficient algorithms that exploit cheap projections and other tools may be exploited. Our research concerns development of the scalable FETI-based methods for contact problems combining FETI approach with algorithms for bound constrained quadratic programming problems with a known rate of convergence given in terms of the spectral condition number (QPMPGP, SMALBE) and their testing in parallel environment. The most difficult part - solution of subdomain problems - may be usually carried out in parallel without any coordination, so that high parallel scalability is enjoyed. The increasing number of subdomains decreases the subdomain problem size resulting in shorter time for subdomain stiffnes matrix factorizations and subsequently forward and backward substitutions during pseudoinverse application, but on the other hand the increasing number of subdomains assuming fixed discretization parameter increases the dual dimension and coarse problem size resulting in longer time for all dual vector operations and projector application. Three types of parallelization strategies and their impact to parallel scalability level will be discussed.

2 FETI-1 and TFETI-1

Let us consider contact boundary value problem. To apply the FETI-1 based domain decomposition let us partition domain Ω into N_s subdomains Ω^s and we denote by K^s, f^s, u^s and B^s, respectively the subdomain stiffness matrix, the subdomain force and displacement vectors and the signed matrix with entries -1, 0, 1 describing the subdomain interconnectivity (gluing or nonpenetration). We shall get the discretized problem

$$\min \frac{1}{2} u^T K u - u^T f \ \text{ s. t. } \ Bu \leq 0 \tag{1}$$

$$K = \begin{bmatrix} K^1 & & \\ & \ddots & \\ & & K^{N_s} \end{bmatrix}, \quad f = \begin{bmatrix} f^1 \\ \vdots \\ f^{N_s} \end{bmatrix}, u = \begin{bmatrix} u^1 \\ \vdots \\ u^{N_s} \end{bmatrix}, \quad B = [B^1, \ldots, B^{N_s}]. \tag{2}$$

The basic idea of TFETI is to keep all the subdomain stiffness matrices K^s as if there were no prescribed displacements and to enhance the prescribed displacements into the matrix of constraints B. To enhance the boundary conditions like $u_i = 0$, just append the row b with all the entries equal to zero except $b_i = 1$. The prescribed displacements will be enforced by the Lagrange multipliers which may be interpreted as forces. An immediate result of this procedure is that all the subdomain stiffness matrices will have known and typically the same defect. The remaining procedure is exactly the same as described for FETI-1, the key point is that the kernels R^s of the locall stiffness matrices K^s are known and can be formed directly. We can easily assemble the block–diagonal basis R of the kernel of K as

$$R = \begin{bmatrix} R^1 & & \\ & \ddots & \\ & & R^{N_s} \end{bmatrix}. \tag{3}$$

Let's establish following notation

$$F = BK^\dagger B^T, \ \widetilde{G} = R^T B^T, \ \widetilde{d} = BK^\dagger f, \ \widetilde{e} = R^T f, \ G = T\widetilde{G}, \ e = T\widetilde{e}$$

where $K^\dagger$ denotes matrix satisfying $KK^\dagger K = K$ such as generalized inverse or Moore-Penrose pseudoinverse, T denotes a nonsingular matrix, that defines the orthonormalization of the rows of $\widetilde{G}$. The critical point of evaluation of $K^\dagger$, the determination of the ranks of the subdomain stiffness matrices K^s is trivial when the TFETI-1 procedure is applied. Our minimization problem reads

$$\min \ \frac{1}{2} \lambda^T F \lambda - \lambda^T \widetilde{d} \ \text{ s.t. } \ \lambda_I \geq 0 \ \text{ and } \ \widetilde{G}\lambda = \widetilde{e}. \tag{4}$$

The problem of minimization on the subset of the affine space is transformed to the problem on subset of vector space by means of arbitrary $\widetilde{\lambda}$ which satisfies $G\widetilde{\lambda} = e$ while the solution is looked for in the form $\lambda + \widetilde{\lambda}$. Using old notation and denoting $d = \widetilde{d} - F\widetilde{\lambda}$, the problem (4) is equivalent to

$$\min \ \frac{1}{2} \lambda^T F \lambda - \lambda^T d \ \text{ s.t. } \ \lambda_I \geq -\widetilde{\lambda_I} \ \text{ and } \ G\lambda = 0. \tag{5}$$

Further improvement is based on the observation, that the augmented Lagrangian for problem (5) can be decomposed by orthogonal projectors

$$Q = \widetilde{G}^T (\widetilde{G}\widetilde{G}^T)^{-1} \widetilde{G} = G^T G \qquad \text{and} \qquad P = I - Q$$

on the kernel of G and on the image space of G^T ($\mathrm{Im}Q = \mathrm{Ker}G$ and $\mathrm{Im}P = \mathrm{Im}G^T$), so that the final problem reads

$$\min \frac{1}{2}\lambda^T PFP\lambda - \lambda^T Pd \text{ s.t. } \lambda_I \geq -\widetilde{\lambda_I} \text{ and } G\lambda = 0, \tag{6}$$

and may be solved effectively by a scalable algorithm SMALBE (Semi-Monotonic Augmented Lagrangians with Bound and Equality) using QPMPGP (Quadratic Programming with Modified Proportioning and Gradient Projection) in inner loop or just by QPMPGP for convex quadratic programming problems with simple bounds enforcing equality constraint by dual penalty as the proof of the classical estimate by Farhat, Mandel and Roux

$$\kappa(PFP|\mathrm{Im}P) \leq C\frac{H}{h} \tag{7}$$

of the spectral condition number κ of the restriction of PFP to the range of P by the ratio of the decomposition parameter H and the discretization parameter h remains valid for TFETI-1.

3 Parallelization strategies

Programmes were implemented using PETSc 3.0.0 (Portable Extensible Toolkit for Scientific Computation), developed by Argonne National Laboratory. PETSc is a suite of data structures and routines that provide the building blocks for the implementation of large-scale application codes on high-performance computers.

The supercomputer for numerical experiments was HECToR at EPCC. Its architecture: two Cray supercomputing facilities: the phase 2a (XT5h) machine and the phase 2b (XT6) machine; and an archiving facility, the main service (phase 2a) uses a Cray XT4 system as its major compute engine offering a total of 3072 AMD 2.3 GHz quad-core Opteron processors - 12,288 cores offering a theoretical peak performance of 113 Tflops, 8 GB of main memory available per Opteron processor, which is shared between its four cores, HECToR's total memory is 24.6 TB, processors are connected with a high bandwidth interconnect using Cray SeaStar2 communication chips.

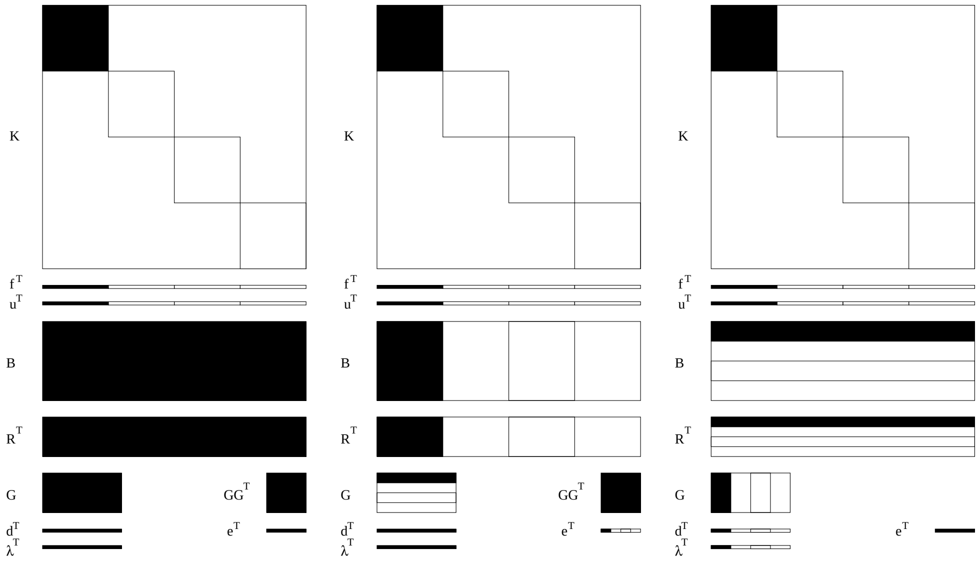


Figure 1: Three types of compared data distributions of primal and dual data.

Most of computations appearing in these programmes are purely local and therefore parallelizable (subdomains problems), but some operations require data transfers. The level of communication depends first of all on distribution of B and R, G and GG^T computation and GG^T factorization or G orthonormalization (see Figure 1).

Acknowledgement: This research has been supported by the grants: GA CR 101/09/P601, Ministry of Education of the CR No. MSM 6198910027 and EU project PRACE-1IP.

References

[1] Z. Dostál, D. Horák: *Scalability and FETI based algorithm for large discretized variational inequalities.* Math. and Comp. in Simulation 61, (3-6), 2003, 347–357.

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

A remark on the optimal mesh and the optimal polynomial degree distribution in solving 1D boundary value problems by the *hp*-FEM

J. Chleboun

Faculty of Civil Engineering, Czech Technical University in Prague

1 Introduction

This contribution deals with an optimal distribution of mesh nodes as well as an optimal distribution of polynomial degrees in the *hp*-version of the finite element method (FEM) applied to solving concrete 1D boundary value problems.

Unlike the *h*-version of the FEM, where the polynomial degree distribution is fixed and only the mesh can be adaptively changed to improve the accuracy of the FE solution, the *hp*-FEM offers more flexibility in the process of minimizing the difference between the exact and the FE solution.

Indeed, one can add and/or redistribute mesh nodes as well as change p, the degree of polynomials forming the FE basis functions. Moreover, the degree need not be uniformly distributed over the mesh. Although this diversity of changes is advantageous, it also recoils upon the analyst who then faces the problem of establishing a good (or, better, an optimal) strategy of mesh and polynomial degree modifications.

An extensive literature on adaptive methods in the *hp*-FEM shows that many efforts have been made to minimize the error of approximation. Nevertheless, even for 1D boundary value problems, results on optimal meshing and optimal FE basis are rather limited and directed towards asymptotical optimality, see [1, 2, 3], which is not the topic we will pursue.

This work does not deal with general purpose error estimate approaches and h or p adaptivity algorithms. We will focus on the optimal use of a fixed number of degrees of freedom (DOF) in a given 1D boundary value problem. The obtained optimal h and p distributions then can serve in defining benchmark problems for practical *hp*-adaptive algorithms in one and (in special cases) more spatial dimensions.

2 *hp*-FEM in 1D

Let us consider a boundary value problem defined on an interval (α, β), that is,

$$-u''(x) + c(x)u(x) = f(x) \text{ in } (\alpha, \beta), \tag{1}$$

$$u(\alpha) = 0,\ u(\beta) = 0, \quad \text{or} \quad u'(\alpha) = 0,\ u'(\beta) = 0, \tag{2}$$

where the function c is such that the bilinear form

$$a(u, v) \equiv \int_\alpha^\beta (u'(x)v'(x) + c(x)u(x)v(x))\,\mathrm{d}x$$

is continuous and V-elliptic for $u, v \in V$. The space V is a subspace of the Sobolev space $H^1(\alpha, \beta)$ and it is determined by the boundary conditions (2) (mixed boundary conditions could also be introduced in (2)).

Assuming $f \in L^2(\alpha, \beta)$, we arrive at the weak formulation of (1)-(2): Find $u \in V$ such that

$$a(u, v) = \int_{\alpha}^{\beta} f(x)v(x)\,\mathrm{d}x \quad \forall v \in V; \tag{3}$$

by virtue of the assumptions, (3) is uniquely solvable.

To find an approximate solution, we substitute a finite-dimensional subspace $V_{n,X_n}^{P_n} \subset V$ for V in (3). The space $V_{n,X_n}^{P_n}$ is constructed as follows: (i) a mesh determined by nodes $x_0 = \alpha < x_1 < \cdots < x_{n+1} = \beta$ is defined; (ii) a set of basis functions is introduced. For each interval $I_i \equiv [x_{i-1}, x_i]$, a maximum polynomial degree p_i is given, $i = 1, \ldots, n$. Let us define X_n, a vector comprising x_i (inner mesh nodes), and P_n, a vector comprising p_i; in both cases $i = 1, \ldots, n$.

The support of each basis function is either $[x_i, x_{i+2}]$ (hat function) or $[x_i, x_{i+1}]$ (Lobatto polynomials of degree two up to p_i; see [4] for the details). The dimension of $V_{n,X_n}^{P_n}$ is also known as the number of DOF (NDOF).

3 Mesh and polynomial degree optimization

Let us assume that the NDOF is equal to N. Let us define $\mathcal{F}_{n,X_n,P_n}^{N}$, a family of all FE spaces $V_{n,X_n}^{P_n}$ whose NDOF equals N. To avoid formal and computational difficulties caused by degenerated mesh intervals, $\mathcal{F}_{n,X_n,P_n}^{N}$ is constrained through a positive minimum length the mesh intervals must not break through. Each space $V_{n,X_n}^{P_n}$ is determined by a configuration of n, X_n, and P_n, that is, by the total number of mesh intervals, by their length and position, and by the respective maximum polynomial degree on each interval. A configuration is called N-admissible if the related FE space has dimension N.

The difference between the solution of (3) and its FE counterpart u_{n,X_n,P_n} is measured by

$$\Psi(n, X_n, P_n) = \|u - u_{n,X_n,P_n}\|_{H^1(\alpha,\beta)}.$$

The optimization problem is set as follows: For a fixed positive integer N,

$$\text{minimize } \Psi(n, X_n, P_n) \text{ over } \mathcal{F}_{n,X_n,P_n}^{N}. \tag{4}$$

The core of solving problem (4) lies in solving a continuous optimization subproblems. Indeed, for a fixed N, n, and P_n, we minimize $\Psi(n, X_n, P_n)$ via searching for the optimal position of the nodes $x_1, \ldots, x_n$. These subproblems have to be solved for each N-admissible configuration of n and P_n, that is, for each configuration that results in N degrees of freedom. Thus problem (4) has combinatorial features and, as a consequence, it is computationally demanding.

Numerical experiments were performed for a few low values of N. To this end, a chosen function was substituted for u in (1), the right-hand side f was calculated, and then used as the known right-hand side in the FE problems determined by (3) and the N-admissible configurations. The calculations were performed in the MATLAB® environment.

Acknowledgement: This work has been supported by grant No. P105/10/1682 of the Czech Science Foundation.

References

[1] I. Babuška, T. Strouboulis: *The finite element method and its reliability.* Oxford University Press, 2001.

[2] I. Babuška, T. Strouboulis, K. Copps: *hp Optimization of finite element approximations: Analysis of the optimal mesh sequences in one dimension.* Comput. Methods Appl. Mech. Engrg. 150, 1997, 89-108.

[3] W. Gui, I. Babuška: *The h, p and $h-p$ versions of the finite element method in 1 dimension. I, II, III.* Numer. Math. 49, 1986, 577–612, 613–657, 659–683.

[4] P. Šolín, K. Segeth, I. Doležel: *Higher-order finite element methods.* Chapman & Hall/CRC, 2004.

Bézier form of S–Patches

A. Kolcun

Institute of Geonics AS CR, Ostrava

1 Introduction

Parametric Cartesian surface, e.g. [1], is a wide-spread tool for data interpolation and approximation. However, for simple modeling systems there is not strong requirement to control all possible geometric parameters of resulting surface. Moreover, due to the fact, that nonplanar rectangular patches are very often tessellated to triangles, it is useful to require the same degree of all boundary curves of tessellated triangles. In [3] the concept of Smart-patches (S–Patches) is introduced. Its main benefits are:

1. the same degree of both diagonal and boundary curves,
2. the number of independent control points is smaller than n^2.

In this paper the main properties for the biquadratic case of S–Patches are described. Bézier form of patches is used. It gives us the possibility to find the correlation between triangular and quadrilateral patches. Condition for smooth concatenation of biquadratic BS–Patches is formulated. Proves can be find in [2].

2 S–Patch

Let us consider biquadratic parametric patch

$$X(u,v) = \mathbf{u}\ \mathbf{R}\ \mathbf{v}^T = (1\ u\ u^2) \begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{pmatrix} (1\ v\ v^2)^T \tag{1}$$

It is obvious that all boundary curves are quadratic polynomial ones.

Let us consider S–Patch [3], i.e. such patch where both main diagonals $D_1(u)$, $D_2(u)$

$$D_1(u) = X(u,u) = \mathbf{u}\ \mathbf{R}\ \mathbf{u}^T$$

$$D_2(u) = X(u, 1-u) = \mathbf{u}\ \mathbf{R}(\mathbf{1\text{-}u})^T = \mathbf{u}\ \mathbf{R} \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{pmatrix} \mathbf{u}^T$$

are quadratic polynomial curves too.

Theorem 1. Biquadratic patch (1) is S–Patch iff $R_{12} = R_{21} = R_{22} = 0$.

Corollary. All parametric lines of biquadratic S–Patch $L(u) = X(u, a+bu)$ are curves of degree $d \leq 2$.

Cartesian Bézier patch is defined as

$$B(u,v) = (b_{0,n}(u) \ldots b_{n,n}(u)) \begin{pmatrix} P_{00} & \ldots & P_{0n} \\ \vdots & & \vdots \\ P_{n0} & \ldots & P_{nn} \end{pmatrix} (b_{0,n}(v) \ldots b_{n,n}(v))^T$$

where $b_{i,n}(u) = \begin{pmatrix} n \\ i \end{pmatrix} (1-u)^{n-i}u^i$ are the Bernstein polynomials and P_{ij} are the control points of the patch. Let us express the biquadratic S–patch in Bézier form. Control points P_{ij} can be found according to the relations below.

$$\mathbf{P} = \begin{pmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & 0 \\ R_{20} & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \tag{2}$$

3 BS–Patch

Let us analyze the relations between main diagonals $D_1(u), D_2(u)$ of S–Patch and proper Bézier diagonals – i.e. the curves defined on the set of diagonal control points P_{00}, P_{11}, P_{22} and P_{20}, P_{11}, P_{02} respectively

$$D_{1B}(u) = \mathbf{u} \begin{pmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} P_{00} \\ P_{11} \\ P_{22} \end{pmatrix}, \quad D_{2B}(u) = \mathbf{u} \begin{pmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} P_{20} \\ P_{11} \\ P_{02} \end{pmatrix},$$

where $\mathbf{P}$ and $\mathbf{R}$ are connected with relation (2).
These relations can be formulated as the theorem below.

Theorem 2. $D_1(u) = D_{1B}(u)$ if and only if $R_{11} = 0$. Moreover, equality of these diagonals automatically implies the equality of $D_2(u) = D_{2B}(u)$.

On the base of the Theorem 2 we can introduce biquadratic BS–Patch, i.e. patch in the form as follows

$$X(u,v) = \mathbf{u} \begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & 0 & 0 \\ R_{20} & 0 & 0 \end{pmatrix} \mathbf{v}^T .$$

In this case mutual relations among Bézier control points P_{ij} and S–patch control points R_{ij} are valid

$$(P_{00}P_{01}P_{02}P_{10}P_{11}P_{12}P_{20}P_{20}P_{21}P_{22}) = (R_{00}R_{01}R_{10}R_{02}R_{20})\frac{1}{2} \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{pmatrix} .$$

We can see that in this case the patch is defined by 5-element set of control points. Examples of non independent and independent 5-element sets of control points of BS–Patches are presented in Fig. 1.

The rest of control points e.g. for the independent pentad $P_{01}, P_{11}, P_{21}, P_{10}, P_{20}$ from Fig. 1e) may be represented as follows

$$\begin{aligned} P_{00} &= P_{01} + P_{10} - P_{11} \quad & P_{02} &= P_{01} + P_{12} - P_{11} \\ P_{20} &= P_{21} + P_{10} - P_{11} \quad & P_{22} &= P_{21} + P_{12} - P_{11} \end{aligned} \tag{3}$$

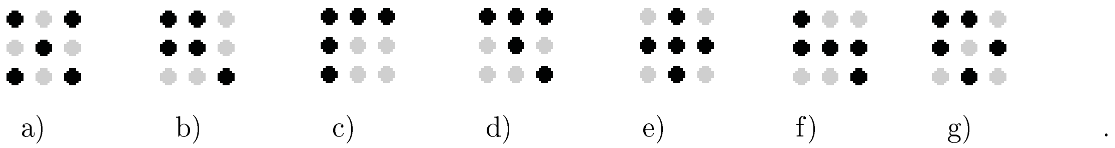


Figure 1: 5-element sets of control-points. a),b) non independent sets, c)–g) independent sets.

4 BS–Patch and Bézier triangles

As both diagonal and boundary curves of BS–Patches are B ezier curves, it is meaningful to analyze the triangle patches. There is a very close connection between the Cartesian BS–patch and a pair of triangular Bézier patches. Let us consider triangular mesh of nodes

$$P_{ijk},\ 0 \le i,j,k \le n,\ i+j+k = n\,,$$

where nodes $P_{i_1j_1k_1}$, $P_{i_2j_2k_2}$ are neighbour, if $\mid i_1 - i_2 \mid + \mid j_1 - j_2 \mid + \mid k_1 - k_2 \mid = 2\,.$

Bézier triangular patch is defined as

$$B_{\triangle}(u,v,w) = \sum_{(i,j,k)} \frac{n!}{i!j!k!} u^i v^j w^k P_{ijk}$$

where $0 \le u,v,w \le 1$, $u+v=w=1$, $0 \le i,j,k \le n$, $i+j+k=n\,.$

Let us consider Cartesian and triangular indexing of control points according to Fig. 2.

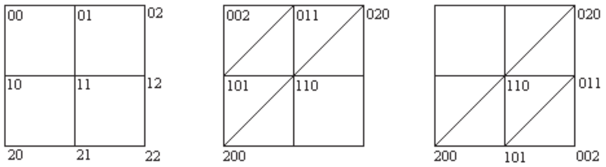


Figure 2: Cartesian and triangular indexing of control nodes for n = 2.

Theorem 3. BS–Patch defined on control points P_{ij}, $0 \le i,j \le 2$ is the same surface as the pair of triangular Bézier patches, defined on the sets of proper control points.

This theorem gives us generalization of the trivial fact that bilinear patch can be decomposed to two triangles iff the quaternion of control points is planar.

5 Smooth concatenation of BS–Patches

Let us consider 5-element set of independent control points from Fig. 1e). Condition (3) says that the set of control points creates four rhomboids, see Fig. 3. Here we can distinguish three types of control points: central crosswise and dependent.

The conditions for concatenation of the patches can be formulated in the following way.

Definition. Let there are two open polylines $\Lambda_1 = (P_0P_1 \dots P_n)$ and $\Lambda_2 = (R_0R_1 \dots R_m)$. Let
a) $m+1$ copies of polyline Λ_1 are created, each of it started in a node of Λ_2,
b) $n+1$ copies of polyline Λ_2 are created, each of it started in a node of Λ_1 .

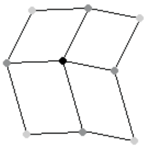


Figure 3: Control points for BS–Patch. Different types of them are distinguished: black – central one, dark – crosswise ones, light – dependent ones.

Resulting set of rhomboids we call 'product of polylines' $\Lambda_1 \bullet \Lambda_2$.

Theorem 4. Surface is set of smooth BS–patches iff the set of central control points of BS–Patches is a product of polylines.

Construction

Given two polylines given two sets (sets of ratios)

$$\pi = (p_0, p_1, \ldots, p_{n-1}),\ \rho = (r_0, r_1, \ldots, r_{m-1}),\ 0 < p_i, r_j < 1\,,$$

we can construct smooth concatenation of BS–Patches according to the steps below.

a) The central control points of BS–Patches are the product of polylines $\Lambda_1 \bullet \Lambda_2$.

b) Crosswise control points can be found as a ratios of neighbour central control points.

c) Dependent control points (corners of BS–patches) are found according to the (3).

d) Concatenation consists of full-defined BS–patches.

Fig. 4 demonstrates the above described construction.

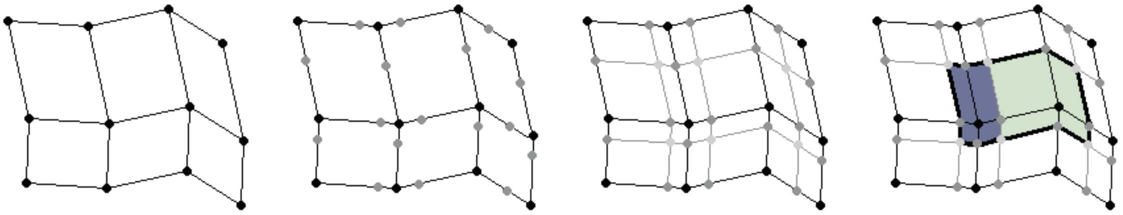


Figure 4: Smooth concatenation of BS–patches according to the steps a)–d) above.

Acknowledgement: This work is supported by grant GACR 105/09/1830 of the Grant Agency CR and the research plan AVOZ 30860518 of the Academy of Sciences of the Czech Republic.

References

[1] G. Farin: *Curves and surfaces for CAGD: A practical guide.* Academic Press, 1988.

[2] A. Kolcun: *Biquadratic S–Patch in Bézier form.* In: V. Skala (ed.): Proceedings of the Conference WSCG 2011, Plzeň 2011.

[3] V. Skala, V. Ondračka: *S–Patch: Modification of the Hermite parametric patch.* In: Proceedings of Conference ICGG 2010, Kyoto 2010.

Orthogonalization with a non-standard inner product and approximate inverse preconditioning

J. Kopal, M. Rozložník, M. Tůma

[1] Institute of Novel Technologies and Applied Informatics, Technical University of Liberec
[2,3] Institute of Computer Science AS CR, Prague

1 Introduction

One of the most important and frequently used preconditioning techniques for solving symmetric positive definite systems $Ax = b$ is based on computing the approximate inverse factorization in the form $A^{-1} = ZZ^T$, where Z is upper triangular [1]. It is also a well-known fact that the columns of the factor Z can be computed by means of the A-orthogonalization process applied to the unit basis vectors $e_1, \dots, e_n$. As noted in [3] such A-orthogonalization also produces the Cholesky factor of the matrix $A = U^T U$, where $U^{-1} = Z$. This fact has been exploited to construct efficient sparse approximate inverse preconditioners [1, 2, 3]. In a more general setting, given the symmetric positive definite matrix A and the nonsingular matrix $Z^{(0)}$, we look for the factors Z and U so that $Z^{(0)} = ZU$ with $Z^T AZ = I$ and the upper triangular matrix U is a Cholesky factor of the matrix $(Z^{(0)})^T AZ^{(0)} = U^T U$.

2 Ortogonalization techniques

One can use a lot of algorithms to calculate matrices Z and U. Straightforward and probably the most expensive way is the computation based on spectral decoposition. Assume spectral decomposition of the matrix A in the form $A = V\Lambda V^T$. We can get the factor U as the upper triangular factor from QR decomposition (with standard inner product) of the matrix $\Lambda^{1/2} V^T Z^{(0)} = QU$ and the factor Z can be then obtained simply as the product $Z = V\Lambda^{-1/2}Q$. This approach (called EIG here) is due to computation cost useful only for small dimensional matrices. For the real-world problems it is more suitable and likely the most common way to compute matrices Z and U using on the generalized Gram-Schmidt orthogonalization (the A-orthogonalization), which forms the columns of the matrix Z. The orthogonalization coefficients form the upper triangular factor U. There are several versions of the Gram-Schmidt algorithm, which lead to the same result in exact arithmetic. The classical Gram-Schmidt (CGS) algorithm employs a lot of parallelism, because the scalar products can be computed separately. Rearraging of this scheme has led to the modified Gram-Schmidt algorithm (MGS), which partly lost parallel properties, but provides better numerical results. Except CGS and MGS algorithms there is a specific combination of these schemes, which originates from AINV preconditoner [3]. This scheme will be further referred as the AINV orthogonalization. The papers on approximate inverse factorization are mainly focused on the construction of the algorithms and do not study their numerical properties. Therefore it is necessary to study incomplete algorithms also from the numerical point of view and understand well their numerical behavior. The development of algorithms for constructing approximate inverse has led from oblique projections based AINV and CGS orthogonalizations [3] to their stabilized version represented by SAINV algorithm [2], which uses MGS orthogonalization algorithm.

3 Theoretical analysis

Assume computed quantitie $\bar{Z}$ which approximate Z so that $A^{-1} \approx \bar{Z}\bar{Z}^T$. Our analysis has focused in particular on the bound for the loss of orthogonality which can be completely different for various algorithm as it will be presented later. With the loss of orthogonality we mean the 2-norm of the matrix $\bar{Z}^T A \bar{Z} - I$. The orthogonality between computed vectors has a cardinal significance for the quality of the preconditioner computed by the orthogonalization process. It is a well-known fact that the eigenvalues of $\bar{Z}^T A \bar{Z}$ affect the convergence rate of preconditioned conjugate gradient method applied to $\bar{Z}^T A \bar{Z} y = \bar{Z} b$, where $x = \bar{Z} y$. Therefore our primary goal is to solve the orthogonal basis problem in this application. There exist complete rounding error analysis [4, 5, 7] for all main schemes for the QR decomposition with the standard inner product, but the situation is completely different for the non-standard inner product (induced by matrix A).

In this contribution we review the most important schemes used for orthogonalization with respect to the non-standard inner product and give the worst-case bounds for corresponding quantities computed in finite precision arithmetic. We formulate our results on the loss of orthogonality, on the factorization error, and on Cholesky factorization error (measured by $\|\bar{Z}^T A \bar{Z} - I\|$, $\|Z^{(0)} - \bar{Z}\bar{U}\|$, and $\|A - \bar{U}^T \bar{U}\|$) in terms of quantities proportional to the roundoff unit u, in terms of the condition number $\kappa(A)$ which represents an upper bound for the relative error in computing the A-inner product as well as the condition number of the matrix $A^{1/2} Z^{(0)}$ which plays an important role in the factorization $(Z^{(0)})^T A Z^{(0)} \approx \bar{U}^T \bar{U}$.

4 Numerical experiments

We consider a test problem defined as a sequence of matrices A_i with dimension $n = 10$ which are generated as powers of the Pascal matrix $A = \text{pascal}(10) = V \Lambda V^T$ ($\kappa(A) \approx 10^9$) such that $A_i = V \Lambda^{i/9} V^T$ with $\kappa(A_i) \approx 10^i, i = 0, \ldots, 17$. The matrix $Z_i^{(0)}$ is equal to $Z_i^{(0)} = I$.

We can see from figure 1, that the loss of orthogonality for all these algorithms is proportional to $u\kappa(A)$. This problem does not reach the worst-case bound, obtained by CGS, AINV, and MGS in the form $\|\bar{Z} A \bar{Z} - I\| \leq O(u)\kappa^{3/2}(A)$. The factorization error corresponds to theoretical analysis $\|I - \bar{Z}\bar{U}\|$; its bound for the algorithms is proportional to $u\kappa^{1/2}(A)$ [6]. On figure 2 we can see the Cholesky factorization error $\|A - \bar{U}^T \bar{U}\|$, for the EIG implementation it is proportional to $u\|A\|$ and for other algorithms it is proportional to $u\kappa^{1/2}(A)\|A\|$, that are worst-case bounds for Cholesky factorization error [6].

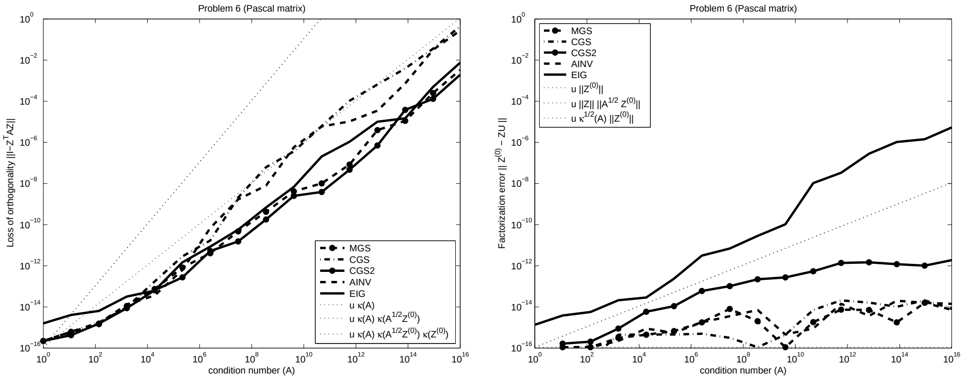


Figure 1: Loss of orthogonality and factorization error for the test problem.

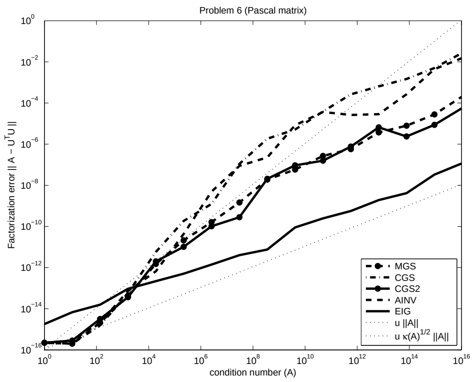


Figure 2: Cholesky factorization error for the test problem.

5 Conclusion

As it was noted, from all given Gram-Schmidt algorithms we can get significantly different numerical results, but the factorization error is essentially the same. The bound for the loss of orthogonality depends linearly on the condition number $\kappa(A)$ for the case of eigenvalue based implementation (EIG) and classical Gram-Schmidt with reorthogonalization (CGS2). For the modified Gram-Schmidt it is also true, although, also besides $\kappa(A)$ it depends on the condition number $\kappa(A^{1/2}Z^{(0)})$. The loss of orthogonality is similar for the classical Gram-Schmidt (CGS) and AINV orthogonalization, no matter that theoretically the bound depends on $\kappa(A)\kappa(A^{1/2}Z^{(0)})\kappa(Z^{(0)})$. From the numerical point of view and due to the computation cost, MGS seems to be a good compromise between all these algorithms. For all these results and details we refer to [6]. We believe that these results may initialize a detailed research of the schemes which leads to some sparse approximation of the matrices Z and U. For a overview of such schemes we refer to [1].

Acknowledgement: This work has been supported by Grant Agency of the Academy of Sciences of the Czech Republic under the project IAA100300802, by the international collaboration support M100300902 of AS CR and by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications" and by the grant No. 102/08/H081 of the Grant Agency of the Czech Republic.

References

[1] M. Benzi: *Preconditioning techniques for large linear systems: a survey.* J. Comput. Phys. 182 (2), 2002, 418–477.

[2] M. Benzi, J. K. Cullum, and M. Tuma: *Robust approximate inverse preconditioning for the conjugate gradient method*, SIAM J. Sci. Comput. 22 (4), 2000, 1318–1332.

[3] M. Benzi, C. D. Meyer, and M. Tuma: *A sparse approximate inverse preconditioner for the conjugate gradient method.* SIAM J. Sci. Comput. 17 (5), 1996, 1135–1149.

[4] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof: *Rounding error analysis of the classical Gram-Schmidt orthogonalization process.* Num. Math. 101, 2005, 87–100.

[5] L. Giraud, J. Langou, and M. Rozložník: *On the loss of orthogonality in the Gram-Schmidt orthogonalization process.* Comput. Math. Appl. 50 (7), 2005, 1069–1075.

[6] J. Kopal, M. Rozložník, A. Smoktunowicz, and M. Tuma: *Rounding error analysis of orthogonalization with a non-standard inner product*. Submited to Num. Math., 2010.

[7] A. Smoktunowicz, J.L. Barlow, and J. Langou: *A note on the error analysis of the classical Gram-Schmidt*. Num. Math. 105 (2), 2006, 299–313.

Numerical algorithms on multicore architectures

P. Kotas

VŠB - Technical University of Ostrava

1 Introduction

Many tools such as Matlab (and its open-source counterpart, Octave) are often used for algorithm prototyping and development. Matlab & Octave have easy to learn syntax and provide the easiest way of implementing numerical algorithms. However both Matlab & Octave share problems that limits their usefulness. The problems include :

- Matlab is a proprietary software with an expensive license. This fact limits the use of programs written in Matlab. Octave partially solves this problem, however not all Matlab functionality is yet implemented in Octave.
- Both Matlab & Octave are weakly dynamically typed languages, which means type checking is performed at runtime as opposed to compile-time. As such, it is possible to write type unsafe programs that could break during deployment stage. Furthermore, there is performance penalty associated with runtime check.
- It is possible to call function written in other languages (such as C and Fortran) from Matlab. However functions written in Matlab can not be easily called from other languages.
- Increasing availability of multi-core CPUs has opened a possibility to increase performance via parallelization. However parallelization is non-trivial within Matlab, which is in contrast to the ease of parallelization via OpenMP or Thread building blocks.

Due to above issues, most programs (or algorithms) written in Matlab are often converted to another programming language (C++, Java, etc.), when targeting commercial deployment or large scale parallel environments. This leads to another set of problems, such as reimplementing Matlab functions that are necessary requirement for run of implemented algorithm.

This work considers alternative approach to algorithm prototyping. The main area of my research are parallel algorithms for computer vision. Therefore, I focused on libraries for computer vision, linear algebra packages and parallel libraries.

2 Computer vision libraries

There are two suitable image libraries, OpenCV [1] and cImg [2]. CImg library is basically the only template providing basic routines for handling images. Because I need a replacement of the Matlab image processing toolbox I have chosen the OpenCV library. OpenCV is an extensive set o functions for image processing and computer vision with neat implementation of matrix operations. OpenCV also possess basic implementation of graphical user interface.

3 Linear algebra packages

Armadillo [3] is easy to learn linear algebra package. It is build on top of LAPACK and ATLAS and it is designed to have syntax similar to Matlab. This features makes Armadillo perfect library for numerical algorithm prototyping.

4 Parallel libraries

There are two widely used libraries for parallelization on multi-core architectures. Thread building blocks (TBB) is library developed by Intel. It is suited for developing parallel algorithms in C++. TBB uses object oriented approach and is based on template algorithms. On the other hand, OpenMP is set of compiler pragmas and set of parallel instruction is built-in most todays compilers. OpenMP is well suited for parallelization of existing sequential algorithms that spent most of the time iterating over arrays.

5 Discussion

Both OpenMP and TBB are reliable libraries and could do similar job. Because my work is done in C++ and all algorithms are mainly iterating over large arrays, the choice of parallel library is not simple. Also library needs to incorporate with OpenCV. Therefore choice of parallelization library will be based on experience with implementing simple image processing algorithm. The Armadillo and OpenCV integration will also be tested.

References

[1] G. Bradski: *The OpenCV library.* Dr. Dobb's Journal of Software Tools, 2000.

[2] *cImg library.* http://cimg.sourceforge.net/

[3] C. Sanderson: *Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments.* NICTA Technical Report, 2010.

[4] J. Reinders: *Intel threading building blocks: outfitting C++ for multi-core processor parallelism.* Sebastopol: O'Reilly Media, 2007.

[5] B. Chapman, G. Jost, R. van der Pas: *Using OpenMP: portable shared memory parallel programming.* MIT Press, 2008.

Solution of non-linear algebraic systems in coupled thermo-mechanical analysis

J. Kruis, T. Koudelka

Faculty of Civil Engineering, Czech Technical University in Prague

1 Introduction

This contribution concentrates on mechanical analyses based on damage models coupled with heat transfer. The damage models are used for description of concrete and rock materials. Typical feature of such models is softening branch after the peak stress value. In order to follow the softening behaviour, the methods of arc-length are used [1], [2]. Three of them are compared in this paper.

2 Solution of non-linear algebraic systems of equations

The equilibrium condition of a structure after discretization by the finite element method has the form

$$\boldsymbol{f}_{int}(\boldsymbol{d}) = \boldsymbol{f}_c + \lambda \boldsymbol{f}_p \tag{1}$$

where $\boldsymbol{d}$ denotes the vector of nodal displacements, $\boldsymbol{f}_{int}$ denotes the vector of internal forces, $\boldsymbol{f}_c$ denotes the vector of constant prescribed forces, $\lambda \boldsymbol{f}_p$ denotes the vector of proportionally changing forces and λ denotes the scalar load-level multiplier. The vector of unbalanced forces has the form

$$\boldsymbol{r}(\boldsymbol{d}, \lambda) = \boldsymbol{f}_c + \lambda \boldsymbol{f}_p - \boldsymbol{f}_{int}(\boldsymbol{d}) \tag{2}$$

and it is the residual. The dependence of $\boldsymbol{d}$ on λ has to be obtained by an iterative process. Let the i-th step be known, i.e. the vector $\boldsymbol{d}_i$ and the parameter λ_i are known and $\boldsymbol{r}(\boldsymbol{d}_i, \lambda_i) = \boldsymbol{0}$. Expansion of the residual has the form

$$\boldsymbol{r}(\boldsymbol{d}_{i+1}, \lambda_{i+1}) = \boldsymbol{r}(\boldsymbol{d}_i, \lambda_i) + \frac{\partial \boldsymbol{r}(\boldsymbol{d}_i, \lambda_i)}{\partial \boldsymbol{d}} \delta \boldsymbol{d}_i + \frac{\partial \boldsymbol{r}(\boldsymbol{d}_i, \lambda_i)}{\partial \lambda} \delta \lambda_i = -\boldsymbol{K}_{i,0} \delta \boldsymbol{d}_{i,1} + \boldsymbol{f}_p \delta \lambda_{i,1} = \boldsymbol{0} \tag{3}$$

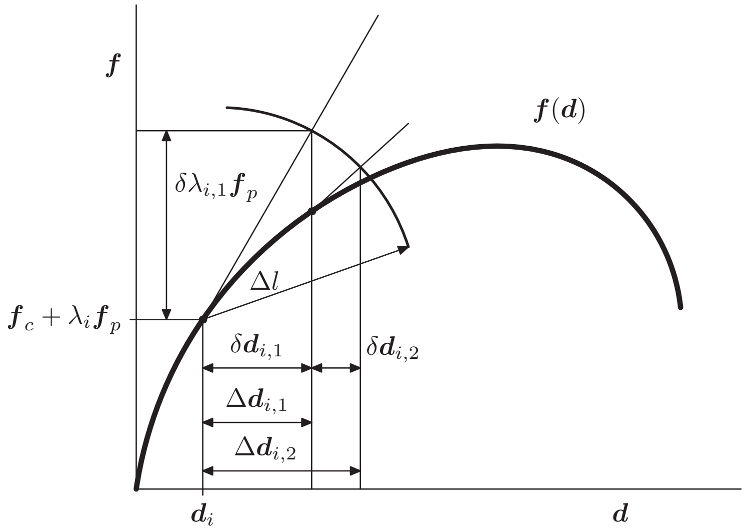
where the following notation

$$\frac{\partial \boldsymbol{r}(\boldsymbol{d}_i, \lambda_i)}{\partial \boldsymbol{d}} = -\boldsymbol{K}_{i,0} \tag{4}$$

$$\frac{\partial \boldsymbol{r}(\boldsymbol{d}_i, \lambda_i)}{\partial \lambda} = \boldsymbol{f}_p \tag{5}$$

is used. Let the vector $\delta \boldsymbol{d}_{i,1}$ be in the form

$$\delta \boldsymbol{d}_{i,1} = \delta \lambda_{i,1} \boldsymbol{v}_{i,1} \tag{6}$$



$\boldsymbol{f}$
$\boldsymbol{f}(\boldsymbol{d})$
$\delta\lambda_{i,1}\boldsymbol{f}_p$
Δl
$\boldsymbol{f}_c + \lambda_i \boldsymbol{f}_p$
$\delta\boldsymbol{d}_{i,1}$
$\delta\boldsymbol{d}_{i,2}$
$\Delta\boldsymbol{d}_{i,1}$
$\Delta\boldsymbol{d}_{i,2}$
$\boldsymbol{d}_i$
$\boldsymbol{d}$

Figure 1: Load–deflection curve.

Substitution of the assumption (6) to (3) leads to the expression

$$\boldsymbol{v}_{i,1} = \boldsymbol{K}_{i,0}^{-1}\boldsymbol{f}_p \tag{7}$$

The length of arc can be written

$$(\delta\boldsymbol{d}_{i,1})^T\delta\boldsymbol{d}_{i,1} + \psi^2(\delta\lambda_{i,1})^2\boldsymbol{f}_p^T\boldsymbol{f}_p = (\delta\lambda_{i,1})^2\boldsymbol{v}_{i,1}^T\boldsymbol{v}_{i,1} + \psi^2(\delta\lambda_{i,1})^2\boldsymbol{f}_p^T\boldsymbol{f}_p = (\Delta l)^2 \tag{8}$$

where the scaling parameter ψ was defined. The increment of the scalar load multiplier has the form

$$\delta\lambda_{i,1} = \pm\frac{\Delta l}{\sqrt{\boldsymbol{v}_{i,1}^T\boldsymbol{v}_{i,1} + \psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p}} \tag{9}$$

Substitution of (9) and (7) to the assumption (6) leads to the modified vector of displacements. Generally, the residual is not equal to the zero vector

$$\boldsymbol{r}(\boldsymbol{d}_i + \delta\boldsymbol{d}_{i,1}, \lambda_i + \delta\lambda_{i,1}) = \boldsymbol{f}_c + (\lambda_i + \delta\lambda_{i,1})\boldsymbol{f}_p - \boldsymbol{f}_{int}(\boldsymbol{d}_i + \delta\boldsymbol{d}_{i,1}) \neq \boldsymbol{0} \tag{10}$$

and new system has to be solved

$$\begin{aligned} \boldsymbol{r}(\boldsymbol{d}_{i+1}, \lambda_{i+1}) &= \boldsymbol{r}_{i,1} - \boldsymbol{K}_{i,1}\delta\boldsymbol{d}_{i,2} + \boldsymbol{f}_p\delta\lambda_{i,2} = \boldsymbol{f}_c + (\lambda_i + \delta\lambda_{i,1})\boldsymbol{f}_p - \\ &- \boldsymbol{f}_{int}(\boldsymbol{d}_i + \delta\boldsymbol{d}_{i,1}) - \boldsymbol{K}_{i,1}\delta\boldsymbol{d}_{i,2} + \boldsymbol{f}_p\delta\lambda_{i,2} = \boldsymbol{0} \end{aligned} \tag{11}$$

where the notation

$$\boldsymbol{r}_{i,1} = \boldsymbol{r}(\boldsymbol{d}_i + \delta\boldsymbol{d}_{i,1}, \lambda_i + \delta\lambda_{i,1}) \tag{12}$$

is used.

Cumulative quantities are defined

$$\Delta\boldsymbol{d}_{i,j} = \Delta\boldsymbol{d}_{i,j-1} + \delta\boldsymbol{d}_{i,j} \qquad (\Delta\boldsymbol{d}_{i,1} = \delta\boldsymbol{d}_{i,1}) \tag{13}$$

$$\Delta\lambda_{i,j} = \Delta\lambda_{i,j-1} + \delta\lambda_{i,j} \qquad (\Delta\lambda_{i,1} = \delta\lambda_{i,1}) \tag{14}$$

and they are schematically depicted in Figure 1. Equation (11) can be rewritten to the form

$$\boldsymbol{K}_{i,1}\delta\boldsymbol{d}_{i,2} = \boldsymbol{f}_c + (\lambda_i + \Delta\lambda_{i,1})\boldsymbol{f}_p - \boldsymbol{f}_{int}(\boldsymbol{d}_i + \Delta\boldsymbol{d}_{i,1}) + \boldsymbol{f}_p\delta\lambda_{i,2} \tag{15}$$

The system of equations (15) can be split into two systems

$$\begin{aligned} \boldsymbol{K}_{i,1}\boldsymbol{u}_{i,2} &= \boldsymbol{f}_c + (\lambda_i + \Delta\lambda_{i,1})\boldsymbol{f}_p - \boldsymbol{f}_{int}(\boldsymbol{d}_i + \Delta\boldsymbol{d}_{i,1}) & (16) \\ \boldsymbol{K}_{i,1}\boldsymbol{v}_{i,2} &= \boldsymbol{f}_p & (17) \end{aligned}$$

and the decomposition

$$\delta\boldsymbol{d}_{i,2} = \boldsymbol{u}_{i,2} + \delta\lambda_{i,2}\boldsymbol{v}_{i,2} \tag{18}$$

is assumed. The length of arc has now the form

$$\|\Delta\boldsymbol{d}_{i,1} + \boldsymbol{u}_{i,2} + \delta\lambda_{i,2}\boldsymbol{v}_{i,2}\|^2 + \psi^2\|\Delta\lambda_{i,1}\boldsymbol{f}_p + \delta\lambda_{i,2}\boldsymbol{f}_p\|^2 = (\Delta l)^2 \tag{19}$$

which is the quadratic equation

$$a_1(\delta\lambda_{i,2})^2 + a_2(\delta\lambda_{i,2}) + a_3 = 0 \tag{20}$$

with coefficients

$$\begin{aligned} a_1 &= \boldsymbol{v}_{i,2}^T\boldsymbol{v}_{i,2} + \psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p & (21) \\ a_2 &= 2\boldsymbol{v}_{i,2}^T(\Delta\boldsymbol{d}_{i,1} + \boldsymbol{u}_{i,2}) + 2\Delta\lambda_{i,1}\psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p & (22) \\ a_3 &= (\Delta\boldsymbol{d}_{i,1} + \boldsymbol{u}_{i,2})^T(\Delta\boldsymbol{d}_{i,1} + \boldsymbol{u}_{i,2}) + (\Delta\lambda_{i,1})^2\psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p - (\Delta l)^2 & (23) \end{aligned}$$

The increment $\delta\lambda_{i,2}$ is obtained from the quadratic equation (20) and it is substituted to (18). New values are again substituted to the residual and equality to the zero vector is checked. The algorithm is summarized in Table 1 and it is called the spherical arc-length method. If the scaling parameter ψ is equal to zero, the method is called the cylindrical arc-length method.

Solution of the quadratic equation (20) is straightforward but only one root has to be used for next computation. One of the criteria used has the form

$$\cos\theta = \frac{\Delta\boldsymbol{d}_{i,j+1}^T\Delta\boldsymbol{d}_{i,j}}{(\Delta l)^2} \to \max \tag{24}$$

Substitution of (13) and (18) leads to the form

$$\cos\theta = \frac{1}{(\Delta l)^2}\Delta\boldsymbol{d}_{i,j}^T(\Delta\boldsymbol{d}_{i,j} + \boldsymbol{u}_{i,j+1} + \delta\lambda_{i,j+1}\boldsymbol{v}_{i,j+1}) \tag{25}$$

New notation

$$\begin{aligned} a_4 &= \Delta\boldsymbol{d}_{i,j}^T(\Delta\boldsymbol{d}_{i,j} + \boldsymbol{u}_{i,j+1}) & (26) \\ a_5 &= \Delta\boldsymbol{d}_{i,j}^T\boldsymbol{v}_{i,j+1} \end{aligned}$$

results to the concise form

$$\cos\theta = \frac{a_4 + \delta\lambda_{i,j+1}a_5}{(\Delta l)^2} \tag{27}$$

Both roots of the equation (20) are substituted to the expression (27) and the root leading to the larger value is selected.

Linearized form of the arc-length leads to the expression

$$\delta\lambda_{i,j+1} = \frac{-\frac{1}{2}l_{i,j} - \Delta\boldsymbol{d}_{i,j}^T\boldsymbol{u}_{i,j+1}}{\Delta\boldsymbol{d}_{i,j}^T\boldsymbol{v}_{i,j+1} + \psi^2\Delta\lambda_{i,j}\boldsymbol{f}_p^T\boldsymbol{f}_p} \tag{28}$$

and no root selection procedure is needed.

```
$\lambda_0 = 0, \boldsymbol{d}_0 = \boldsymbol{0}$
For $i = 0, 1, 2, \ldots$
    $\Delta\lambda_{i,0} = 0$, $\Delta\boldsymbol{d}_{i,0} = \boldsymbol{0}$, $\boldsymbol{r}_{i,0} = \boldsymbol{0}$
    For $j = 0, 1, 2, \ldots$
    $\boldsymbol{u}_{i,j+1} = \boldsymbol{K}_{i,j}^{-1}\boldsymbol{r}_{i,j}$
    $\boldsymbol{v}_{i,j+1} = \boldsymbol{K}_{i,j}^{-1}\boldsymbol{f}_p$
    $a_1 = \boldsymbol{v}_{i,j+1}^T\boldsymbol{v}_{i,j+1} + \psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p$
    $a_2 = 2\boldsymbol{v}_{i,j+1}^T(\Delta\boldsymbol{d}_{i,j} + \boldsymbol{u}_{i,j+1}) + 2\Delta\lambda_{i,j}\psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p$
    $a_3 = \|\Delta\boldsymbol{d}_{i,j} + \boldsymbol{u}_{i,j+1}\|^2 + (\Delta\lambda_{i,j})^2\psi^2\boldsymbol{f}_p^T\boldsymbol{f}_p - (\Delta l)^2$
    $a_1(\delta\lambda_{i,j+1})^2 + a_2(\delta\lambda_{i,j+1}) + a_3 = 0 \quad\Rightarrow\quad \delta\lambda_{i,j+1}$
    $\delta\boldsymbol{d}_{i,j+1} = \boldsymbol{u}_{i,j+1} + \delta\lambda_{i,j+1}\boldsymbol{v}_{i,j+1}$
    $\Delta\boldsymbol{d}_{i,j+1} = \Delta\boldsymbol{d}_{i,j} + \delta\boldsymbol{d}_{i,j+1}$
    $\Delta\lambda_{i,j+1} = \Delta\lambda_{i,j} + \delta\lambda_{i,j+1}$
    $\boldsymbol{r}_{i,j+1} = \boldsymbol{f}_c + (\lambda_i + \Delta\lambda_{i,j})\boldsymbol{f}_p - \boldsymbol{f}_{int}(\boldsymbol{d}_i + \Delta\boldsymbol{d}_{i,j})$
    if $\|\boldsymbol{r}_{i,j+1}\| < \varepsilon$, stop
$\lambda_{i+1} = \lambda_i + \Delta\lambda_i$
$\boldsymbol{d}_{i+1} = \boldsymbol{d}_i + \Delta\boldsymbol{d}_i$
```

Table 1: Algorithm of the Arc-length Method.

3 Conclusions

Numerical experiments based on damage models of rock show that the linearized version of the arc-length method converges faster than the spherical or cylindrical methods but on the other hand, it sometimes performs spurious loading and unloading cycles.

Acknowledgement: Financial support for this work was provided by project number P105/10/1682 of Czech Science Foundation. The financial support is gratefully acknowledged.

References

[1] Z. Bittnar, J. Šejnoha: *Numerical Methods in Structural Mechanics.* ASCE Press, New York, USA, 1996.

[2] M. A. Crisfield: *Non-linear Finite Element Analysis of Solids and Structures.* John Wiley & Sons Ltd, Chichester, UK, 1991.

Construction of higher-order basis functions on meshes with hanging nodes in 3D

P. Kůs

Institute of Thermomechanics AS CR, Prague

1 Introduction

Finite element method using higher-order basis functions and meshes with hanging nodes (*hp*-FEM) became very popular thanks to it's ability to achieve fast (exponential) convergence. The reason of it's qualities is its ability to perform both h (division of element in space) and p (increase of the polynomial order) refinements in the adaptivity process. This approach has been described in several books, see e.g. [1], [3].

In a practical computer implementation, however, many serious technical and theoretical difficulties arise. In this presentation we want to address one of the crucial parts, which is construction of conforming higher-order basis functions on meshes with arbitrary-level hanging nodes.

2 Arbitrary-level hanging nodes

Main feature of introduction of irregular meshes is that faces, edges or vertices of elements can lie inside faces and edges of other elements in the mesh. This situation is not allowed in standard FEM, where adjacent elements either share a single vertex, a single edge, or a single face. With the technique of arbitrary-level hanging nodes, very small elements can be neighbors of very large ones while keeping an undistorted regular shape – this is impossible in standard FEM. Further, this technique makes element refinements completely local – refinement of an element never causes refinements in adjacent elements.

Some authors try to avoid implementational complexity of fully irregular meshes by introducing 1-irregular mesh. It allows hanging nodes, but of only first level. Comparison can be seen in

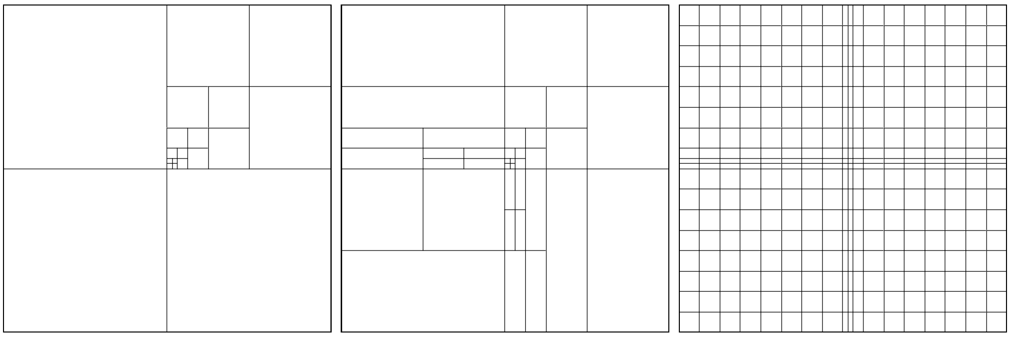


Figure 1: Meshes resulting from an automatic mesh adaptive procedure for problem with singularity slightly to the right and up from the square center. Arbitrary-level hanging nodes (left), level-one hanging nodes (center), no hanging nodes – regular mesh (right).

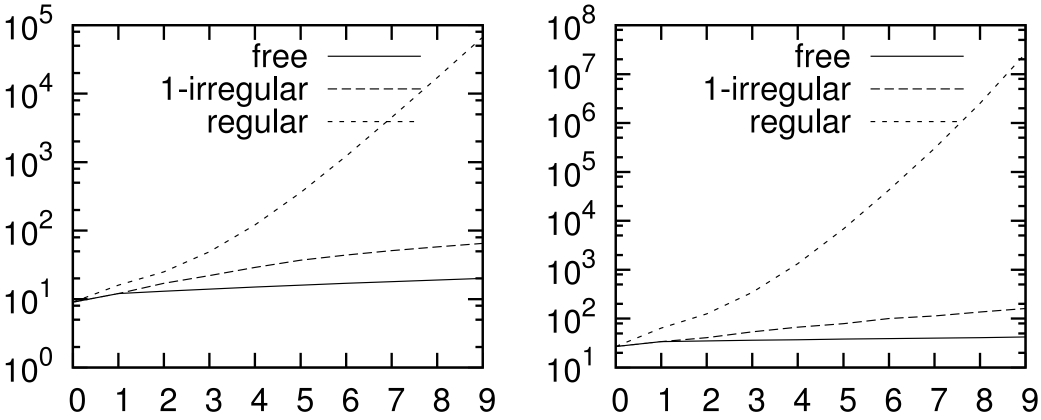


Figure 2: The number of DOFs vs. the number of successive refinement steps for the 2D case in a square (left) and for the 3D case in a cube (right).

Figure 1 for 2D case, for a 3D case the construction is similar, but figure would be difficult to draw. In Figure 2 we can see comparison of number of degrees of freedom in meshes obtained by successive refinement towards singularity as shown in Figure 1 and similar construction in 3D. Even though this construction is slightly artificial, we can see, that mesh with arbitrary-level hanging nodes has much less degrees of freedom than two others. It is caused by the fact, that no unnecessary refinements are performed.

Forced refinements slow down the convergence, worsen the conditioning of stiffness matrices, and their algorithmic treatment is problematic, because they can “spread” through the mesh in a recursive nature. Most existing adaptivity algorithms in both low- and higher-order FEM suffer from these drawbacks.

3 Construction of basis functions

In the finite element method, solution of the problem is sought as a combination of basis functions. In the concept of hierarchical basis, each basis function is related to an entity in the mesh, which in the case of three dimensional mesh can be vertex, edge, face or element interior.

Let us address space H^1, which is used for discretization of elliptic problems. Conformity requirement of this space is continuity. Therefore, all basis functions has to be created in such way, that they are continuous in all vertices, edges and faces. In the presentation, a rather technical description of construction is shown. The idea is following. In the regular mesh, basis functions are constructed simply by ”gluing“ pieces together, as shown in Figure 3 for a vertex function. The process is similar for edge and face functions, even though here the situation is complicated by a necessity of proper orientation handling. But still, when dealing with regular mesh, one has to consider only elements adjacent to given vertex, edge or face. On all other elements the basis function equals zero. Bubble (or interior) functions are simple, they are local to one element and zero elsewhere and therefore their continuity is clear.

For the case of meshes with hanging nodes, new problems arise. Here much more elements may be involved and great effort has to be made to keep basis functions conforming. A rather sophisticated algorithm has been described in [4] for two dimensional case. We used the idea, but in the 3D setting everything is much more complicated. In Figure 4, an element after several refinements is shown and we can see, that much more elements are involved in construction of

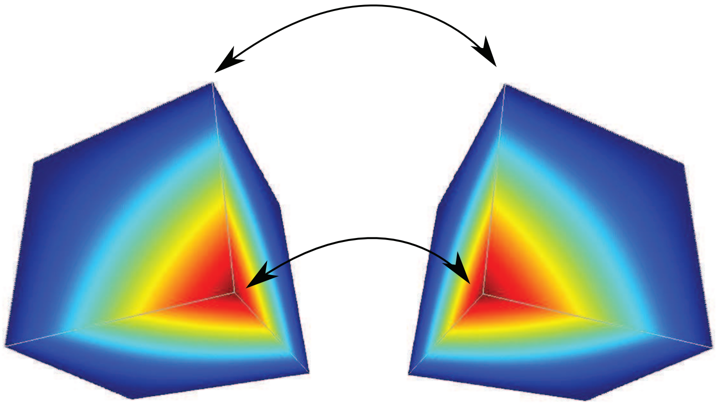


Figure 3: Two elements with images of local basis vertex function being “glued“ together to form part of a global vertex function.

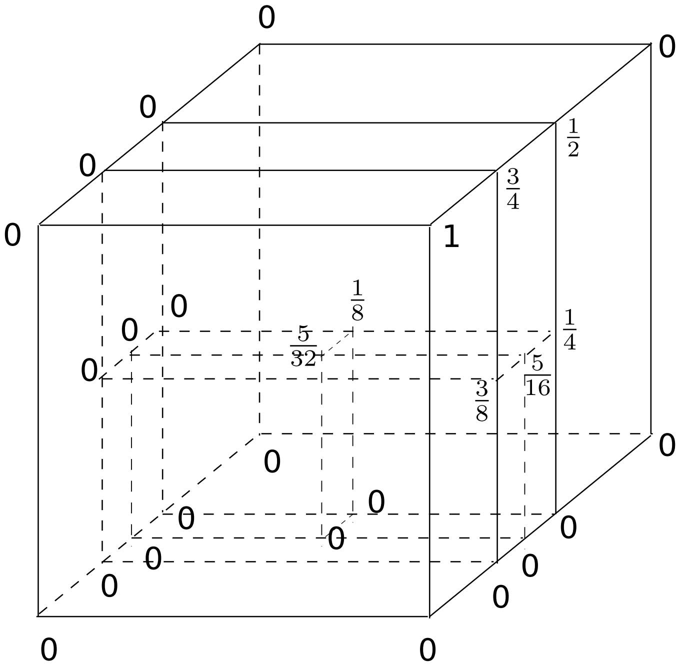


Figure 4: Example of one element of the coarse mesh with many refinements. Numbers assigned to vertices represent coefficients of contributing local basis functions, when constructing vertex basis function (associated to a vertex with number 1).

a vertex basis function. For edges and faces the situation is even more complicated, because, for example, values on face may constrain values in many other faces, edges and vertices in the mesh. A detailed algorithm which determines what local basis functions and with which coefficients should be included to form global basis function will be presented.

4 Conclusion

We present algorithm of construction of conforming basis functions of higher order in meshes with arbitrary-level hanging nodes. It is part of more complex work related to development of *hp*-FEM software for 3D elliptic, electromagnetic and other problems.

In the future we want to focus on solving difficult coupled problems arising in engineering practice. Such problems in 3D may lead to necessity of solving huge linear systems. Experiments in two spatial dimensions suggest, that when using *hp*-adaptive algorithms, such systems may become significantly smaller and therefore solvable in reasonable time.

Acknowledgement: This work has been supported by the grant GAAVČR IAA100760702.

References

[1] L. Demkowicz, J. Kurtz D. Pardo, M. Paszynski, W. Rachowicz,, A. Zdunek: *Computing with hp-adaptive finite elements, Volume 2.* Chapman & Hall/CRC Press 2008.

[2] P. Kůs, P. Šolín, I. Doležel: *Solution of 3D singular electrostatics problems using adaptive hp-fem.* COMPEL, 27(4), 2008, 939–945.

[3] P. Šolín, K. Segeth, I. Doležel: *Higher-order finite element methods.* Chapman & Hall/CRC Press, 2004.

[4] P. Šolín, J. Červený, I. Doležel: *Arbitrary-Level Hanging Nodes and Automatic Adaptivity in the hp-FEM.* Math. Comput. Simulation, 2007.

Geosynthetic tubes filled with liquids with different densities

J. Malík

Institute of Geonics AS CR, Ostrava

1 Introduction

Geosynthetic tubes have found applications in many branches of engineering. The reader can find a description of these applications, for instance, in the monograph [7]. Geosynthetic tubes have been studied in many papers, but only the problems related to the tubes filled with a single liquid have been analyzed.

The models of geosynthetic tubes on a rigid horizontal foundation are presented, for instance, in [3, 4, 6, 9]. The mathematical models of geosynthetic tubes filled with both liquid and air are investigated in [1].

Application of stacked geosynthetic tubes attracts more and more attention. Such problems are solved in [8], where the behavior of stacked tubes is analyzed both on a rigid foundation as well as on a deformable one. Mathematical problems connected with existence, stability, and uniqueness are analyzed in [1, 5]. The existing numerical methods are reviewed and compared in [2].

2 Formulation of the problem

In this section we formulate the basic hypotheses and the differential equations of equilibrium for a geosynthetic tube filled with several liquids sitting on the rigid horizontal foundation.

The cross-section of the tube is depicted in Figure 1. Notice that the shape of the cross-section is symmetric with respect to the y - axis.

Our aim is to describe the shape of the cross-section and to find the corresponding tension t, the pressures $p_0, p_1, \ldots, p_n$ with respect to the given perimeter l, the areas $v_1, \ldots, v_n$ and the densities $\rho_1, \ldots, \rho_n$. Notice that the given data cannot be independent. Concretely, hypothesis (v) formulated above yield the inequalities

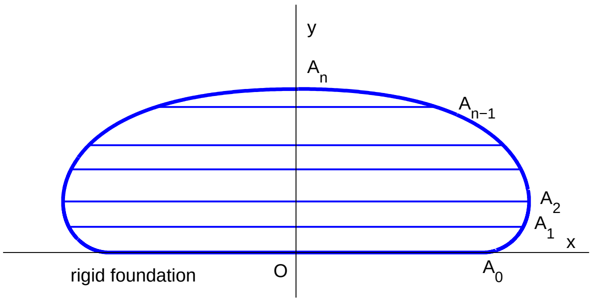


Figure 1: Scheme of the cross-section of the tube.

$$\rho_1 > \rho_2 > \ldots > \rho_n \,.$$

Moreover, the maximal area of the cross-section related to the fix perimeter l corresponds to the area of the circle. Thus the inequality

$$\sum_{i=1}^{n} v_i < \frac{l^2}{4\pi}\,, \tag{1}$$

must hold. With respect to the theoretical results in [5], we can expect that this inequality also ensures the solvability of the problem.

Since

$$p_i = p_{i-1} - g\rho_i(y_i - y_{i-1}), \quad i = 1, \ldots, n, \tag{2}$$

the pressures fulfill the inequalities

$$p_0 > p_1 > \ldots > p_n,$$

where p_0 is the pressure on the bottom, p_n is the pressure on the top. Due to hypothesis (vi), the pressure in the liquids acts in the perpendicular direction to the synthetic fabric. Moreover due to hypotheses (ii) and (vii), the friction between the tube and the foundation does not influence the shape of the cross-section. Thus there is no force in the tangential direction, which results in a constant tension force in the fabric.

First of all, we will formulate the problem with respect to the parameter s. So we consider the continuous functions $x(s)$, $y(s)$, $\theta(s)$ to describe the shape of the cross-section curve. The equations of equilibrium for the geosynthetic tube filled with n liquids read

$$\begin{aligned} \frac{dx}{ds} &= \cos\theta(s)\,, \\ \frac{dy}{ds} &= \sin\theta(s)\,, \\ t\,\frac{d\theta}{ds} &= p_i - g\rho_{i+1}(y(s) - y_i)\,, \quad i = 0, 1, \ldots, n-1\,, \end{aligned} \tag{3}$$

where $s \in (s_i, s_{i+1})$. The equations (3) describe the shape of the part of the cross-sectional curve in the layer occupied by the liquid with the density ρ_{i+1}. Moreover, the following conditions

$$x_n \equiv x(s_n) = 0\,,\ y_0 \equiv y(s_0) = 0\,,\ \theta_0 \equiv \theta(s_0) = 0\,,\ \theta_n \equiv \theta(s_n) = \pi \tag{4}$$

are satisfied, which is evident from Figure 1. With respect to the prescribed values of the perimeter l and the areas $v_1, \ldots, v_n$, it holds the following equalities:

$$s_n = l/2 \tag{5}$$

and

$$\int\limits_{s_{i-1}}^{s_i} x\,\frac{dy}{ds}\,ds = v_i\,, \quad i = 1, \ldots n\,. \tag{6}$$

To find the solution to our problem, we have to determine the parameters t, s_i, p_i, $i = 0, 1, \ldots, n$, and the continuous functions $x(s)$, $y(s)$, $\theta(s)$ on the interval (s_0, s_n) so that the differential equations (3), the conditions (4) and the relations (2), (5) and (6) are fulfilled.

3 Numerical model problems

In this section we use the numerical algorithms described in the previous section to solve a few numerical model problems. We analyze a geosynthetic tube filled with two, three, and four liquids with various densities. We use the perimeter 10 m in all the investigated examples. We start with a tube filled with two liquids with mass densities 1000 kg/m^3 and 1300 kg/m^3. Let us consider that the volumes of the liquids are divided in the proportion 1 : 1. Now we are looking for the mutual dependence between the whole area of the cross-section and such quantities as the length of the contact zone, the height of the tube, the pressure on the bottom and top of the tube and the tension in the geosynthetic fabric. All these quantities are compared with the same quantities for the geosynthetic tube filled with the single liquid with the average density 1150 kg/m^3.

The graph in Figure 2 describes the dependence of the tube height filled with two liquids on the cross-sectional area. Notice that the limit heights are 0 m and $10/\pi$ m which correspond to the height of an empty tube and the diameter of the circle cross-section of the tube, respectively.

The difference between the tube heights for two liquids and for the single liquid with the average density is depicted in Figure 3. The graph in Figure 3 shows that the tube height filled with two liquids is greater than the height of the tube filled with the single liquid for all the values of the cross-sectional area. The maximal difference is approximately achieved for the same value of the cross-sectional area as in the case of the contact zones.

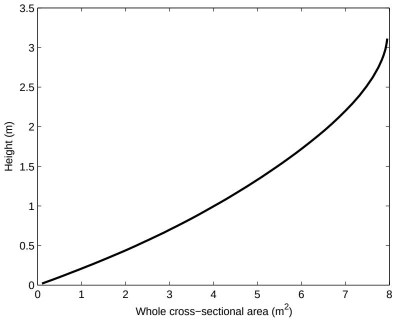


Figure 2: The height of the tube filled with two liquids.

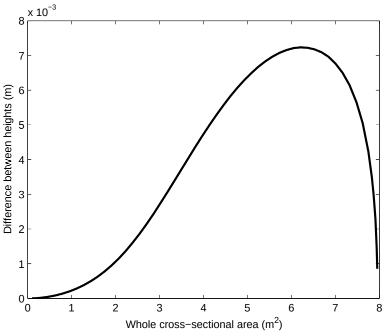


Figure 3: The difference between the tube heights for two liquids and for a single liquid with the average density.

The shape of the cross-section of the tube filled with two liquids (full line) and a modified shape of the cross-section of the tube filled a single liquid (dotted line) is depicted in Figure 4. The shape for the single liquid is modified so that the difference between the shapes is enlarged fifteen times. The cross-sectional area is 3.0 m^2.

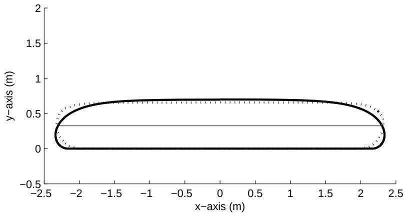


Figure 4: The shape of the cross-section of the tube filled with two liquids (full line) and the modified shape of the cross-section of the tube filled with a single liquid (dotted line). The cross-sectional area is 3.0 m^2.

Acknowledgment: The authors would like to thank the anonymous reviewers for their valuable comments. The research was supported by grant GA CR 103/08/1700.

References

[1] S.S. Antman, M. Schagerl: *Slumping instabilities of elastic membranes holding liquids and gases.* International Journal of Non–Linear Mechanics 40, 2005, 1112–1138.

[2] S. Cantré, *Geotextile tubes – analytical design aspect.* Geotextiles and Geomembranes 20, 2002, 305–319.

[3] K.K. Kazimierowicz: *Simple analysis of deformation of sand – sausages.* Fifth International Conference on Geotextiles, Geomembranes and Related Product, Vol. 2, Hydraulic Applications and Related Research,Singapore, 1994, 775–778.

[4] D. Leshchinsky, O. Leshchinsky, H.J. Ling, P.A. Gilbert: *Geosynthetic tubes for confining pressurized slurry: some design aspects.* Journal of Geotechnical Engineering 122, 1996, 682–90.

[5] Malík, J.: *Some problems connected with 2D – modelling of geosynthetic tubes.* Nonlinear Analysis: Real World Applications 10, 2009, 810–823.

[6] V. Namias: *Load – supporting fluid–filled cylindrical membranes.* Journal of Applied Mechanics 52, 1985, 913–918.

[7] K.W. Pilarczyk: *Geosynthetic and Geosystems in Hydraulic and Coastal Engineering.* Taylor & Francis, 2007.

[8] R.H. Plaut, C.R. Klusman: *Two–dimensional analysis of stacked geosynthetic tubes on deformable foundations.* Thin–Walled Structures 34, 1999, 179–194.

[9] R.H. Plaut, S.I. Liapis, D.P. Telionis: *Wen the levee inflates.* Civil Engineering (ASCE) 68 (1), 1998, 62–64.

Some mathematical problems around the GOOGLE search engine

I. Marek

Czech Technical University in Prague

1 Introduction

It is known that the Google search engine opened unusual interest for its fundamental principles in many areas of research. Our contribution is concerned with the celebrated Google matrix whose importance in computing the PageRank is undisputable. A worldwide discussion concerning many aspects of search engines resulted in many journal publications as well as a monograph [6]. The above mentioned problem how to compute the PageRank efficiently led to an elementary but very interesting result in Linear Algebra, to the so called Google lemma. Within short period many proofs and generalizations of this lemma have been proposed and with large probability some more will appear. An increasing interest to some specific disciplines of Mathematics and Computer Science as well as many other areas of research directions should be welcome.

2 Generalities

All matrice appearing in the next sections are $N \times N$ matrices possibly expressed using their block structure. As standard, we denote by $\rho(C)$ the spectral radius of square matrix C, i.e.

$$\rho(C) = \max \{|\lambda| : \lambda \in \sigma(C)\},$$

where $\sigma(C)$ denotes the spectrum of C. We call

$$\gamma(C) = \sup \{|\lambda| : \lambda \in \sigma(C), \lambda \neq \rho(C)\}.$$

the convergence factor of C. We define quantity $\tau(C)$ by setting

$$\tau(C) = \max \{|\lambda| : \lambda \in \sigma(C), |\lambda| < \rho(C)\}$$

and call it *subspectral radius of* C.

2.1. Remark Let C be any $N \times N$ matrix. Then obviously,

$$\rho(C) \geq \gamma(C) \geq \tau(C).$$

2.2. Remark *Let T be a matrix whose elements are nonnegative real numbers. It is well known that*
1*)*

$$\lim_{k\to\infty} T^k = 0 \Longleftrightarrow \rho(T) < 1;$$

2*)*

$$\lim_{k\to\infty} \left(\frac{1}{\rho(T)}T\right)^k = T_\infty \neq 0 \Longrightarrow \gamma\left(\frac{1}{\rho(T)}T\right) < 1;$$

3 A short proof of the Google lemma

We are going to examine the following system of problems parameterized by parameter $\alpha \in (\frac{1}{2}, 1)$:

$$G(\alpha) = \alpha G^{(1)} + (1-\alpha) G^{(2)},$$

where $G^{(1)}$ is a (column) stochastic matrix and $G^{(2)}$ a suitable (low rank) irreducible stochastic matrix.

We establish the following result and present it as

3.1. Lemma *Suppose $G^{(2)} = ve^T$, where $v = (v_1, ..., v_N)^T$ is a vector whose all components are nonnegative reals and $e^T = (1, ..., 1)$, $e^T v = 1$, i.e. $G^{(2)}$ represents a rank-one stochastic matrix. Then the convergence factor can be bounded as follows*

$$\gamma(G(\alpha)) \leq \alpha.$$

Proof Let $\hat{x}(\alpha)$ denote the Perron eigenvector. It is easy to see that vector $\hat{x}(\alpha)$ has all its components nonnegative and it can be normalized by setting $e^T \hat{x}(\alpha) = 1$. It follows that $\hat{x}(\alpha) = G(\alpha)\hat{x}(\alpha) = \alpha G^{(1)} \hat{x}(\alpha) + (1-\alpha)v$ and hence

$$\hat{x}(\alpha) = \left[\frac{1}{1-\alpha} (I - \alpha G^{(1)}) \right]^{-1} v.$$

Thus, the Perron projection of $G(\alpha)$ reads $Q(\alpha) = \hat{x}(\alpha) e^T$. We check easily that $Q(\alpha)G(\alpha)Q(\alpha) = G(\alpha)Q(\alpha) = Q(\alpha)$ and

$$(I - Q(\alpha))\, G^{(2)} \,(I - Q(\alpha)) = \left(G^{(2)} - Q(\alpha) \right) (I - Q(\alpha)) = G^{(2)} \,(I - Q(\alpha)) = 0. \tag{1}$$

The validity of the statement of Lemma 3.1 follows from the relation representing the unique spectral decomposition of matrix $G(\alpha) = Q(\alpha) + (I - Q(\alpha))\, \alpha G^{(1)} \,(I - Q(\alpha))$. The proof is complete.

The above proof opens a way to generalizations. A crucial point in the above proof is a special kind of relationship between the original transition matrix $G^{(1)}$ and the perturbation $G^{(2)}$ consisting of relations (1).

4 A generalization of the GOOGLE lemma

A speciality of our proof of the GOOGLE lemma demonstrated in the previous section consists of showing that the perturbation vector is fully absorbed by the Perron projection of the convex combination. An application of this fact to more general situation would be possible if we find another type of perturbation with the absorbtion property and a method offering a convergent procedure to compute a corresponding stationary probability vector. We show that such a pair appears quite frequently.

Let $p \geq 2$ be a positive integer and

$$\begin{gathered} G^{(2)} = \textstyle\sum_{k=1}^{p} \lambda^{k-1} Q_k, \ \lambda = \exp\left\{ \frac{2\pi i}{p} \right\}, i^2 = -1, \\ Q_1^{(2)} = \hat{x}_2 e^T, Q_k^{(2)} Q_j^{(2)} = Q_j^{(2)} Q_k^{(2)} = \delta_{jk}, j, k = 1, ..., p. \end{gathered} \tag{1}$$

Assume that $G^{(2)}$ is an irreducible blockwise cyclic stochastic matrix of order p and (1) its spectral decomposition. We immediately see that both the block index of cyclicity as well as rank of $G^{(2)}$ equal p.

4.1. Theorem *Assume $G^{(1)}$ is a stochastic matrix, $G^{(2)}$ is defined in (1), both of order $N \times N$, and $G(\alpha) = \alpha G^{(1)} + (1-\alpha)G^{(2)}, \alpha \in (\frac{1}{2}, 1)$. If also $G(\alpha)$ is p-cyclic, then*

$$\tau\left(G(\alpha)\right) = \alpha. \tag{2}$$

Proof Since obviously

$$G^{(3)} = Q_1^{(2)} e^T$$

is an irreducible rank-one stochastic matrix the GOOGLE lemma 4.1 implies that a unique Perron projection of matrix $\alpha G^{(1)} + (1-\alpha)G^{3)}$ reads as follows

$$Q^{(1)}(\alpha) = \left(\frac{1}{1-\alpha}\left(I - \alpha G^{(1)}\right)\right)^{-1} \hat{x}_2.$$

p-Cyclicity of matrix $G(\alpha)$ then implies that its peripheral part possesses the following spectral decomposition (see [1])

$$\hat{x}(\alpha)e^T + \sum_{k=2}^{p} \lambda^{j-1} Q_j(\alpha),\ Q_j(\alpha) = y_j f^T,$$

where $\hat{x}(\alpha)^T = \left(\hat{x}_{(1)}^T, ..., \hat{x}_{(p)}^T\right)$, and

$$\begin{gathered} y_j^T = \left(\lambda^{j-1}\hat{x}_{(1)}^T, ..., \lambda^{(j-1)p}\hat{x}_{(p)}^T\right), \\ f_j^T = \left(\overline{\lambda}^{j-1} e(n_1)_{(1)}^T, ..., \overline{\lambda}^{(j-1)p} e(n_p)_{(p)}^T\right), \\ e(n_j) = (1, ..., 1)^T \in \mathcal{R}^{n_j}, j = 2, ..., p, \textstyle\sum_{k=1}^{p} n_k = N \\ \overline{\xi} = x_1 - ix_2,\ \text{for } \xi = x_1 + ix_2, x_1, x_2 \in \mathcal{R}^1. \end{gathered}$$

The conclusion of Theorem 4.1 follows from the fact that [9]

$$Q_j(\alpha) = \lim_{m\to\infty} \frac{1}{m} \sum_{k=1}^{m} \left(\frac{1}{\lambda^{j-1}} G(\alpha)\right)^k,\ \ \lambda = \exp\{2\pi i/p\},\ j = 1, ..., p.$$

5 An application

In this section we present an application of the generalized GOOGLE lemma. It consists of convergence of a two-level computation method for a problem with data of restricted precision.

5.1. Theorem *Assume B is an irreducible stochastic matrix being cyclic of index p. Further we assume that the elements of B are known exactly but with some error, say $B = B^{(1)} + C$ with some stochastic $B^{(1)}$ and an error matrix $\|C\| \leq \eta$ with η fixed. To compute the appropriate stationary probability vector of $B^{(1)}$ we utilize Algorthm 4.1. $SPV(B(\alpha); T; t, s = 1; y^{(0)}; \varepsilon)$, where $B(\alpha) = \alpha B^{(1)} + (1-\alpha)B^{(2)}, I - B(\alpha) = M(I - T), T = M^{-1}W, (1/2) < \alpha < 1$, as formulated in[10]. Here $B^{(2)} = \sum_{j=1}^{p} \lambda^{j-1} Q_j^{(2)}, \lambda = \exp\{2\pi i/p\}$. Then Algorithm 4.1 returns a sequence of iterants $\{y^{(k)}\}$ such that*

$$\left\|y^{(k)} - \hat{y}\right\| \leq \kappa\left(\tau(T)\right)^k,\ k = 0, 1, ...$$

where $\hat{y} = B(\alpha)\hat{y}, \hat{y}e^T = 1, e = (1, ..., 1)^T$ and κ is independent of k.

5.2. Remark *We see that the data i.e $B^{(1)}$ is perturbed by a term proportional to $C = \sum_{j=1}^{p} Q_j^{(2)}$ and we insist relation $\|(1-\alpha)C\| \leq \eta$ with $0 < \eta$ to hold.*

Acknowledgement: This work has been supported by grant Nr. 201/09/1544 of the Grant Agency of the Czech Republic and by grant MSM 6840770010 of the Ministry of Education, Sports and Youth of the Czech Republic.

References

[1] P.J. Courtois, P. Semal: *Block iterative algorithms for stochastic matrices.* Linear Algebra and Its Applications 76, 2006, 59–80.

[2] L. Eldén: *The eigenvalues of the Google matrix.* Technical Report LiTH-MAR-R-04-01, Department of Mathematics Linkőping University, Linkőping, Sweden, 2004.

[3] T.H. Haveliwala, S.D. Kamvar: *The second eigenvalue of the Google matrix.* Technical Report, Computer Science Department, Stanford University, Palo Alto, 2003.

[4] I. Ipsen, T. Selee: *PageRank computation, with special attention to dangling nodes.* SIAM J. Matrix Anal. Appl. 29, 4, 2007, 1281–1296.

[5] S.D. Kamvar, T.H. Hawelivala, G.H. Golub: *Extrapolation methods for accelerating PageRank compoutations.* In: Proceedings of the Twelfth Internationl World wide Web Conference (WWW03), Toronto, ACM Press, New York 2003, 261–273.

[6] A.N. Langville, C.D. Meyer: *A reordering for the PageRank problem.* SIAM J. Sci. Comput. 27, 2006, 2112–2120.

[7] A.N. Langville, C.D. Meyer: *Google's PageRank and beyond. The Science of Search Engine Rankings.* Princeton University Press 2006.

[8] C.P. Lee, G.H. Golub and S.A. Zenios: *A two-stage algorithm for computiong PageRank and multi-stage generalizations.* Internet Math. 4, 4, 2007, 299–328.

[9] I. Marek: *C-convergence of iterations of bounded linear operators.* Comment. Math. Univ. Carol. 2, 4, 1961 22–24.

[10] I. Marek, P. Mayer: *Convergence theory of a class of aggregation/ disaggregation iterative methods for computing stationary probability vectors of stochastic matrices.* Linear Algebra Appl. 363, 2002, 177–200.

[11] I. Pultarová: *Local convergence analysis of aggregation/disaggregation methods with polynomial correction.* Linear Algebra Appl. 421, 2007, 122–137.

[12] I. Pultarová: *Necessary and sufficiient local convergence condition of one class of aggregation-disaggregation methods.* Numerical Linear Algebra with Applications 15, 2008, 339–354.

[13] S. Serra-Capizzano: *Jordan canonical form of the Google matrix.* SIAM J. Matrix Appl. 27, 2005, 305–312.

[14] W.J. Stewart: *Introduction to the numerical solution of Markov chains.* Princeton University Press, Princeton, NJ., 1994.

Fast solver based on Fourier transform for linear elasticity problem

L. Mocek

VŠB - Technical University of Ostrava

1 Introduction

The main goal of this paper is to briefly show how to solve elliptic boundary value problems for linear elasticity using fictitious domain method and efficient solvers based on discrete Fourier transform and the Schur complement reduction using orthogonal projectors. We start from the fictitious domain formulation of a given problem. We briefly mention the main ideas and we also mention the new fictitious domain approach based on definition of new auxiliary boundary, which is used to get smoother solution on origin domain. Using mixed finite element discretization we get the discrete algebraic saddle-point system, which can be solved effectively by combination of the Schur complement reduction and the Fourier transform. For evaluation of the stiffness matrix we use spectral decomposition of the stiffness matrix by the Discrete Fourier transform and for its product with a vector which is used later for finding the solution, we use Fast Fourier transform. For this evaluation it is not necessary to store the whole stiffness matrix which is big advantage, because the order of stiffness matrix is usually large. For solving of whole algebraic saddle-point system we use the Schur complement reduction. Because the stiffness matrix is singular, the algebraic system is going to be reduced to the other one and afterwards we combine method based on the Schur complement reduction with using of orthogonal projectors. Finally the proposed method is illustrated on numerical examples.

2 Fictitious domain method

Before we formulate linear elasticity problem we briefly explain the basics of fictitious domain method. Let ω be bounded domain in R^2 with the Lipchitz boundary $\partial\omega$. On this domain we define an elliptic boundary value problem. The main idea is to embed the real domain of our original problem with possibly complicated geometry ω to a new simple shaped domain Ω (for example rectangle) called fictitious domain, see Fig. 1. The original problem is reformulated to a new one defined in the fictitious domain Ω. The advantage of this method is that we can use special partition on Ω, which enable us to apply effective solvers for evaluation of resulting algebraic system. We can consider the original boundary conditions as a constraint. In elastic approach, we enforce this constraint by the Lagrange multipliers defined on the boundary γ of the original domain ω. Therefore the fictitious domain solution has a singularity on γ that can result in an intrinsic error of the computed solution.

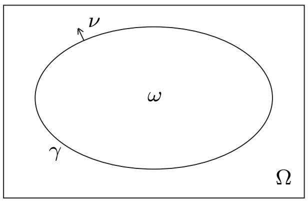


Figure 1: FDM

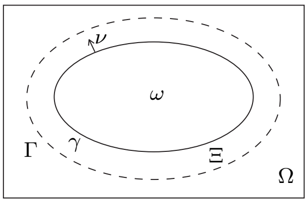


Figure 2: Modified FDM

To remove the above problem we propose a new approach [3], in which we move singularity away from boundary γ. This modification is based on introduction a new control variable instead of the Lagrange multiplier defined on the other auxiliary boundary Γ located outside of the domain $\overline{\omega}$, see Fig. 2. The boundary Γ satisfies the condition $\delta = \mathrm{dist}(\Gamma, \gamma) > 0$. This new control variable enforces the original boundary condition on γ. Because the singularity is moved from $\overline{\omega}$, the solution is smoother in ω.

3 Formulation of the linear elasticity problem

We consider elastic body which is represented by domain $\omega \subset R^2$ with smooth boundary $\gamma = \overline{\gamma_u} \cup \overline{\gamma_p}$, divided into two disjoint parts. The zero displacement is imposed on γ_u while surface tractions of density $p \in (L^2(\gamma_p))^2$ on γ_p. Let us formulate linear elasticity problem:

$$\left.\begin{array}{rcll} -div\,\sigma(u) & = & f & \text{in} \quad \omega, \\ u & = & 0 & \text{on} \quad \gamma_u, \\ \sigma(u)\nu & = & p & \text{on} \quad \gamma_p, \end{array}\right\} \tag{1}$$

where $\sigma(u)$ is the stress tensor in ω, $\nu = (\nu_1, \nu_2)$ is the unit outward normal vector to γ, $u = (u_1, u_2)$ and we prescribe forces of density $f|_\omega \in (L^2_{loc}(R^2))^2$ in ω. The stress tensor is related to the linearized strain tensor $\varepsilon(u) := 1/2(\nabla u + \nabla^T u)$ by Hooke's law for linear isotropic materials:

$$\sigma(u) := \lambda\, tr(\varepsilon(u))I + 2\mu\,\varepsilon(u) \quad in \quad \omega,$$

where "tr" denotes the trace of matrices, $I \in R^{2\times 2}$ is the identity matrix and $\lambda, \mu > 0$ are the Lamè constants.

We define operator $div\,\sigma(u)$ as

$$div\,\sigma(u) = \left(\begin{array}{c|c} (\lambda+2\mu)\dfrac{\partial^2 u_1}{\partial x_1^2} + \mu\dfrac{\partial^2 u_1}{\partial x_2^2} & (\lambda+\mu)\dfrac{\partial^2 u_2}{\partial x_1 \partial x_2} \\ \hline (\lambda+\mu)\dfrac{\partial^2 u_1}{\partial x_1 \partial x_2} & \mu\dfrac{\partial^2 u_2}{\partial x_1^2} + (\lambda+2\mu)\dfrac{\partial^2 u_2}{\partial x_2^2} \end{array}\right), \tag{2}$$

and the space

$$V(\Omega) = (H^1_{per}(\Omega))^2, \quad H^1_{per}(\Omega) = \{v \in H^1(\Omega) | v \,\text{is periodic on}\, \partial\Omega\}.$$

The modified fictitious domain formulation of (1) is following:

$$\left.\begin{array}{l} Find\;(\hat{u}, \lambda) \in V(\Omega) \times \Lambda(\Gamma)\; such\; that \\ a_\Omega(\hat{u}, v) + \langle v, \lambda\rangle_\Gamma = \displaystyle\int_\Omega f v \,\mathrm{d}x \quad \forall v \in V(\Omega), \\ \langle \mu_u, \hat{u}\rangle_{\gamma_u} = 0 \quad \forall \mu_u \in \Lambda(\gamma_u), \\ \langle \mu_p, \sigma(\hat{u})\nu\rangle_{\gamma_p} = \langle \mu_p, p\rangle_{\gamma_p} \quad \forall \mu_p \in \Lambda(\gamma_p), \end{array}\right\} \tag{3}$$

where $\Lambda(\Gamma) = (H^{-1/2}(\Gamma))^2$, $\Lambda(\gamma_u) = (H^{-1/2}(\gamma_u))^2$, $\Lambda(\gamma_p) = (H^{-1/2}(\gamma_p))^2$, and $\langle\,,\rangle_\Gamma$, $\langle\,,\rangle_{\gamma_u}$, and $\langle\,,\rangle_{\gamma_p}$ stand for the duality pairings between $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$, $H^{1/2}(\gamma_u)$ and $H^{-1/2}(\gamma_u)$, $H^{1/2}(\gamma_p)$ and $H^{-1/2}(\gamma_p)$ respectively. Finally $a_\Omega : V(\Omega) \times V(\Omega) \to R$ and $\langle v, \lambda\rangle_\Gamma : V(\Omega) \times \Lambda(\Gamma) \to R$ are two bounded bilinear forms.

The dicretization of (3) using finite element method [1] leads to the following algebraic saddle point system:

$$\left(\begin{array}{c|c} A & B_\Gamma^T \\ \hline B_\gamma & 0 \end{array}\right)\left(\begin{array}{c} u \\ \hline \lambda \end{array}\right) = \left(\begin{array}{c} f \\ \hline g \end{array}\right), \tag{4}$$

where $A \in R^{2n \times 2n}$ is the stiffness matrix, the matrices $B_\Gamma \in R^{2m \times 2n}$ and $B_\gamma = (B_{\gamma_u}, C_{\gamma_p})^T \in R^{2m \times 2n}$ are determined by geometries of Γ and γ, respectively, and by the imposed boundary conditions, they have full row-ranks and also they are highly sparse. The vectors f and g are given as $f \in R^{2n}$, $g = (0, p)^T \in R^{2m}$, respectively. We solve this algebraic system with the method based on Schur complement reduction.

Due to the choice of the space with periodic boundary condition on $\partial\Omega$, the matrix A is singular but the advantage is that A has a block circulant structure which allows to use the highly efficient solver based on the Fourier transform. For this reason we can use Discrete Fourier Transform for spectral decomposition of stiffness matrix A and after that easily evaluate $A^\dagger y$ by Fast Fourier Transform without storing A and it is big advantage against other algebraic solvers. We denote $A^\dagger$ as generalized inverse of A and $y \in R^{2n}$. This product appears in multiplying procedure of Shur complement reduction which is used to solve this problem.

4 Solver for linear elasticity problem based on DFT

Let us describe this multiplying procedure in more details. We solve our problem in fictitious domain Ω. On the sides of $\Omega = (0, L_x) \times (0, L_y)$ we consider equidistant partitions into n_x and n_y segments with stepsizes $h_x = L_x/n_x$ and $h_y = L_y/n_y$, respectively. Domain Ω is decomposed into $n = n_x n_y$ partitions. On this rectangulation we introduce the fine element subspace V_h, which is formed by piecewise bilinear functions. Then the stiffness matrix A reads as follows:

$$A = \left(\begin{array}{c|c} (\lambda + 2\mu) A_x \otimes M_y + \mu M_x \otimes A_y & (\lambda + \mu) B_x \otimes B_y \\ \hline (\lambda + \mu) B_x \otimes B_y & \mu A_x \otimes M_y + (\lambda + 2\mu) M_x \otimes A_y \end{array} \right), \tag{5}$$

where symbol $\otimes$ stands for the Kronecker tensor product and A_k, M_k, $B_k \in R^{n_k \times n_k}$, $k = x, y$ are circulants with the first columns

$$\begin{aligned} a_k &= (1/h_k)(2, -1, 0, \ldots, 0, -1)^T \in R^{n_k}, \quad k = x, y, \\ m_k &= (h_k/6)(4, 1, 0, \ldots, 0, 1)^T \in R^{n_k}, \quad k = x, y, \\ b_k &= (1/2)(0, -1, 0, \ldots, 0, 1)^T \in R^{n_k}, \quad k = x, y, \end{aligned}$$

respectively. Eigenvalues of any circulant can be obtained by the DFT of its first column while eigenvectors are columns of the inverse to the DFT matrix. Based on this observation we can write:

$$A_k = X_k^{-1} D_{A_k} X_k, \quad M_k = X_k^{-1} D_{M_k} X_k, \quad B_k = X_k^{-1} D_{B_k} X_k, \quad k = x, y,$$

where D_{A_k}, D_{M_k}, D_{B_k}, $k = x, y$ are the respective diagonal matrices of eigenvalues and X_k, $k = x, y$ are DFT matrices. Substituing these expressions into (5) and using properties of the Kronecker tensor product, we obtain

$$A = \left(\begin{array}{c|c} X^{-1} & 0 \\ \hline 0 & X^{-1} \end{array} \right) \left(\begin{array}{c|c} D_{11} & D_{12} \\ \hline D_{21} & D_{22} \end{array} \right) \left(\begin{array}{c|c} X & 0 \\ \hline 0 & X \end{array} \right), \tag{6}$$

where $X = X_x \otimes X_y$, $D_{11} = (\lambda + 2\mu) D_{A_x} \otimes D_{M_y} + \mu D_{M_x} \otimes D_{A_y}$, $D_{22} = \mu D_{A_x} \otimes D_{M_y} + (\lambda + 2\mu)$ $D_{M_x} \otimes D_{A_y}$, $D_{12} = (\lambda + \mu) D_{B_x} \otimes D_{B_y}$, $D_{21} = D_{12}$. Let us denote D the second matrix on the right hand-side of (6). Then we can obtain generalized inverse of $A^\dagger$ replacing D by $D^\dagger$ in (6). We can rewrite D by the following factorization:

$$D = \left(\begin{array}{c|c} I & 0 \\ \hline D_{21} D_{11}^\dagger & I \end{array} \right) \left(\begin{array}{c|c} D_{11} & 0 \\ \hline 0 & D_{22} - D_{21} D_{11}^\dagger D_{12} \end{array} \right) \left(\begin{array}{c|c} I & D_{11}^\dagger D_{12} \\ \hline 0 & I \end{array} \right), \tag{7}$$

where $D_{11}^{\dagger} = \mathrm{diag}(d_1^{\dagger}, \cdots, d_n^{\dagger})$ with $d_i^{\dagger} = 1/d_i$, if $d_i \neq 0$, and $d_i^{\dagger} = 0$ if $d_i = 0$ and denote $D_{22m} := D_{22} - D_{21}D_{11}^{\dagger}D_{12}$, then we define

$$D^{\dagger} = \left(\begin{array}{c|c} I & D_{11}^{\dagger}D_{12} \\ \hline 0 & I \end{array}\right)^{-1} \left(\begin{array}{c|c} D_{11}^{\dagger} & 0 \\ \hline 0 & D_{22m}^{\dagger} \end{array}\right) \left(\begin{array}{c|c} I & 0 \\ \hline D_{21}D_{11}^{\dagger} & I \end{array}\right)^{-1}, \tag{8}$$

finally we get

$$A^{\dagger} = \left(\begin{array}{c|c} X^{-1} & 0 \\ \hline 0 & X^{-1} \end{array}\right) D^{\dagger} \left(\begin{array}{c|c} X & 0 \\ \hline 0 & X \end{array}\right). \tag{9}$$

We can obtain from (8) and (9) the product $A^{\dagger}y$, $y = (y_1, y_2)$.

5 Schur complement reduction

From the reason that the stiffness matrix A is singular, the first component u of (4) cannot be completely eliminated. It follows that the Schur complement reduction leads to another algebraic system with two unknowns. The first uknown λ from the previous saddle point system and new unknown α, which corresponds to the null-space of A. We can formulate this new algebraic system with unkowns (λ, α):

$$\begin{pmatrix} B_{\gamma}A^{\dagger}B_{\Gamma}^T & -B_{\gamma}N \\ -M^TB_{\Gamma}^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} B_{\gamma}A^{\dagger}f - g \\ -M^Tf \end{pmatrix}$$

and the first unknown u of the algebraic system (4) is given as $u = A^{\dagger}(f - B_{\Gamma}^T\lambda) + N\alpha$. We can simplify this algebraic system to the following reduced system

$$\begin{pmatrix} F & G_1^T \\ G_2 & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix}, \tag{10}$$

where

$$\begin{array}{l} F := B_{\gamma}A^{\dagger}B_{\Gamma}^T, \quad G_1 := -N^TB_{\gamma}^T, \quad G_2 := -M^TB_{\Gamma}^T, \\ d := B_{\gamma}A^{\dagger}f - g \quad e := -M^Tf. \end{array}$$

Now we define two orthogonal projectors P_1 and P_2 onto the null-spaces of G_1 and G_2. The first projector splits the saddle-point algebraic structure of the reduced system, the second projector decomposes the unknown $\lambda \in R^{2m}$ into two components $\lambda_{\mathbb{R}}$ and $\lambda_{\mathbb{N}}$ as

$$\lambda := \lambda_{\mathbb{R}} + \lambda_{\mathbb{N}},$$

where $\lambda_{\mathbb{R}}$ belongs to the range-space of G_2 ($\lambda_{\mathbb{R}} \in \mathbb{R}(G_2^T)$) and $\lambda_{\mathbb{N}}$ belongs to the null-space of G_2 ($\lambda_{\mathbb{N}} \in \mathbb{N}(G_2)$). Then λ is the first component of the solution to the algebraic system (10) if

$$\lambda_{\mathbb{R}} = G_2^T(G_2G_2^T)^{-1}e$$

and $\lambda_{\mathbb{N}}$ satisfies the following equation:

$$P_1F\lambda_{\mathbb{N}} = P_1(d - F\lambda_{\mathbb{R}}).$$

The component $\lambda_{\mathbb{N}}$ is solved by a projected Krylov subspace method for non-symmetric operators (see [3]). Finally the second component of algebraic system (10) is given by

$$\alpha = (G_1G_1^T)^{-1}G_1(d - F\lambda).$$

6 Numerical experiments

Let us show some numerical experiments. Let us define the domain ω as interior of the elipse

$$\omega = \{(x,y) \in R^2 | \frac{(x-0.5)^2}{0.4^2} + \frac{(y-0.5)^2}{0.2^2} < 1\},$$

which is embeded into the fictitious domain $\Omega = (0,1) \times (0,1)$ (see Fig. 3). The righthand sides of (1) are $f = -div\,\sigma(\hat{u})$ and $p = \sigma(\hat{u})\nu$, where $\hat{u}(x,y) = (0.1xy, 0.1xy)$, $(x,y) \in R^2$. The auxiliary boundary Γ is constructed by shifting γ in the direction of outward normal vector. In Fig. 4 we can see original and deformed geometries of ω and the difference between exact and computed solution is shown in Fig. 5. In Table 1 we can see the number of primal and control variables, number of iterations, computational time and relative errors of approximate solution $\hat{u}_h$ to exact solution in these norms:

$$E_{rel,(L_2(\omega))^2} = \frac{\|\hat{u}_h - \hat{u}\|_{(L_2(\omega))^2}}{\|\hat{u}\|_{(L_2(\omega))^2}}, \quad E_{rel,(H^1(\omega))^2} = \frac{\|\hat{u}_h - \hat{u}\|_{(H^1(\omega))^2}}{\|\hat{u}\|_{(H^1(\omega))^2}}.$$

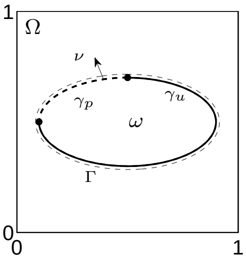


Figure 3: Geometry of ω.

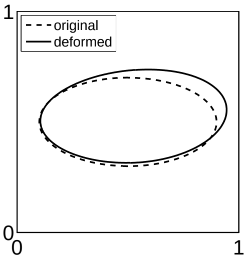


Figure 4: Original and deformed geometry.

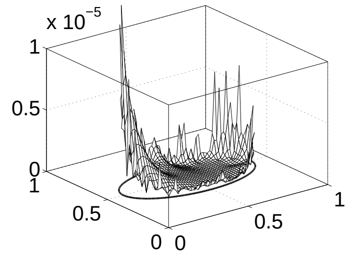


Figure 5: $|\hat{u}_h - \hat{u}|$ in ω.

Step h	prim/control	Iter	Time(s)	$E_{rel,(L_2(\omega))^2}$	$E_{rel,(H^1(\omega))^2}$
1/64	8450/44	43	0.312	4.1269e-003	1.8750e+000
1/128	33282/68	25	0.468	5.2323e-004	6.8257e-001
1/256	132098/112	37	2.215	1.0882e-004	3.1294e-001
1/512	526338/180	52	16.36	8.2582e-005	2.7259e-001

Table 1: Computational results.

References

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

[2] G.H. Golub, C.F. Van Loan: *Matrix computation.* 3rd ed. The Johns Hopkins University Press, Baltimore 1996.

[3] J. Haslinger, T. Kozubek, R. Kucera, G. Peichl: *Projected Schur complement method for solving non-symmetric systems arising from a smooth fictitious domain approach.* Lin. Algebra Appl. 14, 2007, 713–739.

[4] J. Haslinger, T. Kozubek, R. Kucera: *Fictitious domain method for linear elasticity.* SNA 2009.

On numerical behavior of the Arnoldi algorithm in finite precision arithmetic for matrices with close eigenvalues

G. Okša, M. Rozložník

Institute of Mathematics SAS, Bratislava
Institute of Computer Science AS CR, Prague

Let A be a symmetric matrix of order n. Our numerical example uses the Strakoš matrix of order $n = 30$, which is diagonal, positive definite. Its minimal eigenvalue is $\lambda_1 = 0.1$, maximal $\lambda_n = 100$, and $\lambda_i = \lambda_1 + (i-1)/(n-1)\, 0.9^{n-i}(\lambda_n - \lambda_1)$ for $2 \le i \le n-1$. The eigenvectors x_i are columns of the identity matrix of order n. Let us choose a small positive constant $\nu \ll 1$; our numerical example is for $\nu = 1.11 \times 10^{-12}$. Now modify λ_{n-1} as to get a very close pair with λ_n: $\lambda_{n-1} = \lambda_n - 2\nu$ (so that $\nu = (\lambda_n - \lambda_{n-1})/2$), and let $\mu \equiv (\lambda_n + \lambda_{n-1})/2$.

Let $v_1 = \sqrt{n}(1, 1, \ldots, 1)^T$ be the initial unit vector and compute (in finite precision arithmetic) two Krylov bases V_k and W_k by two implementations of the Arnoldi algorithm, whereby both of them ensure the orthogonality of computed basis vectors up to $O(\epsilon)$, where ϵ is the round-off unit ($\epsilon \approx 1.11 \times 10^{-16}$ in double precision arithmetic). We have used the Householder orthogonalization (HH) and the Iterated Modified Gram-Schmidt orthogonalization (IMGS). The bases were generated by following recurrences for $1 \le k \le n-1$:

$$\begin{aligned} AV_k &= V_{k+1}\, H^{(1)}_{k+1,k} + F^{(1)}_k, \quad \text{with} \quad \|F^{(1)}_k\| \le \|A\| O(k^{3/2}n)\epsilon, \\ AW_k &= W_{k+1}\, H^{(2)}_{k+1,k} + F^{(2)}_k, \quad \text{with} \quad \|F^{(2)}_k\| \le \|A\| O(k^{3/2}n)\epsilon, \end{aligned}$$

where $H^{(i)}_{k+1,k}$, $i = 1, 2$, are computed upper Hessenberg matrices of order $(k+1) \times k$.

When looking at the correlation coefficient $c_i = |w_i^T v_i|$, $1 \le i \le n$, one can observe *the loss and recapture of correlation* between iterations 17–24 (see Fig. 1). This surprising observation is closely related to the convergence behavior of two maximal Ritz values (see Fig. 2). First, the maximal Ritz value θ_k^k converges to μ and remains in its vicinity for iterations 14–25. Second,

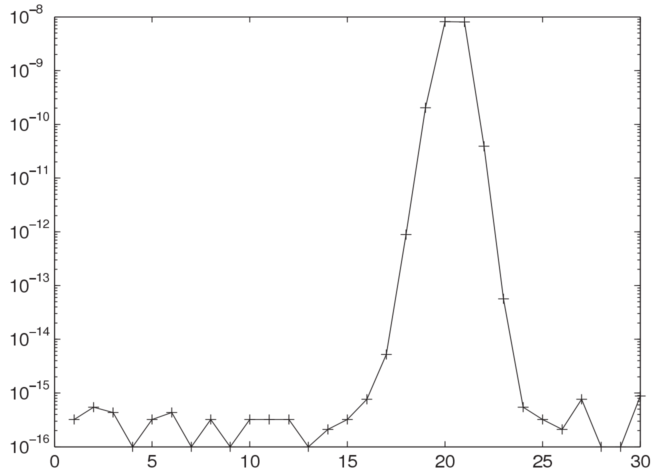


Figure 1: Loss and recapture of correlation: $|1 - c_i|$.

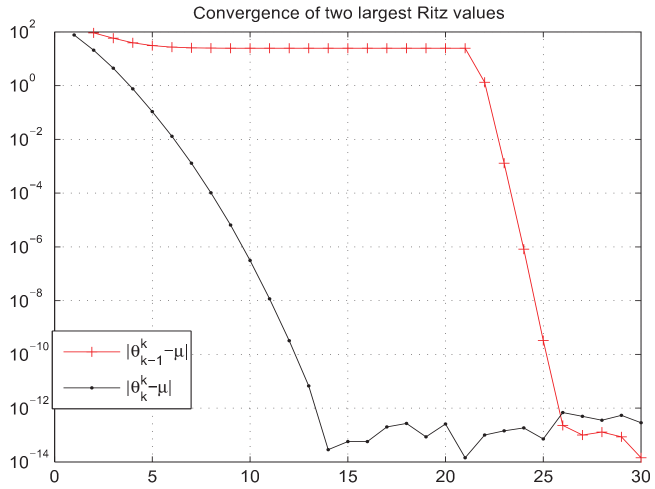


Figure 2: Convergence of two largest Ritz values.

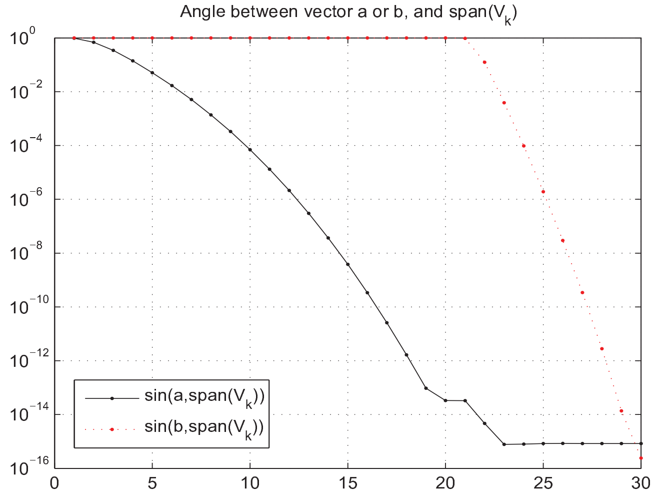


Figure 3: Angles between the vectors a, b, and the subspace span(V_k).

the next-to-maximal Ritz value θ^k_{k-1} approximates λ_{n-2} up to the iteration $k = 21$ and only after that it starts to move towards λ_{n-1}. When both eigenvalues are well approximated by their corresponding Ritz values, the correlation is fully recaptured after the iteration $k = 25$.

Perhaps more insight can be gained by answering the question of how the two-dimensional eigenspace $\mathcal{X}_2 \equiv \text{span}(x_{n-1}, x_n)$ is approximated during the computation. Define two mutually orthogonal vectors: $a \equiv (x_{n-1}+x_n)/\sqrt{2}$, $b \equiv (x_{n-1}-x_n)/\sqrt{2}$, so that $\mathcal{X}_2 = \text{span}(a, b)$, i.e., (a, b) is another orthonormal basis of $\mathcal{X}_2$. Notice that a is the unit orthogonal projection of the starting vector v_1 into $\mathcal{X}_2$, but $b^T v_1 = 0$. In other words, at the beginning of computation the Krylov space contains only information w.r.t. one dimension of $\mathcal{X}_2$ (along a) and the other dimension (along b) has to be built up starting from zero.

Angles between span(V_k) and the vectors a and b are depicted in Fig. 3, while the components $|a^T v_k|$ and $|b^T v_k|$ are depicted in Fig. 4. Starting with $|b^T v_1| = 0$, the b-component increases up to the iteration $k = 22$. At the same time, $|b^T v_k|$ differs from $|b^T w_k|$ more and more, so that

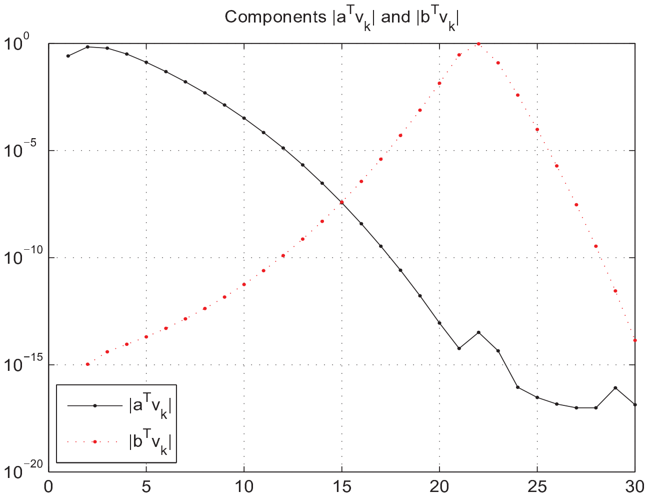


Figure 4: Components $|a^T v_k|$ and $|b^T v_k|$.

when $|b^T v_k| > \sqrt{\epsilon} \approx 10^{-8}$ the correlation starts to deteriorate significantly. Recall that at the iteration $k = 22$ the second largest Ritz pair appears with $\theta^k_{k-1} > \lambda_{n-2}$ so that the approximation of the whole $\mathcal{X}_2$ finally begins. Notice that $|a^T v_k|$ reaches its maximum at $k = 2$ and then almost steadily decreases.

It turns out that it is the b-component of basis vectors, which is sensitive in both implementations of the Arnoldi method. To understand this, we analyze two steps of the Arnoldi process at iteration $k + 1$ in exact arithmetic regardless to its computer implementation:

$$\begin{aligned} &1. \quad y_k = Av_k, \\ &2. \quad \beta_{k+1}v_{k+1} = (I - V_kV_k^T)y_k. \end{aligned} \tag{1}$$

Working with the orthonormal basis $(x_1, x_2, \dots, x_{n-2}, a, b)$, where x_i, $1 \leq i \leq n-2$, are the eigenvectors of A, one can express v_k as

$$v_k = \sum_{i=1}^{n-2}(x_i^T v_k)x_i + (a^T v_k)a + (b^T v_k)b,$$

so that

$$Av_k = \sum_{i=1}^{n-2}\lambda_i(x_i^T v_k)x_i + [\mu(a^T v_k) + \nu(b^T v_k)]a + [\mu(b^T v_k) + \nu(a^T v_k)]b.$$

We see immediately, that because the vectors a and b are *not* the eigenvectors of A, Av_1 has a (small) b-component $\nu(a^T v_1)$ even when $b^T v_1 = 0$! When $\nu \ll 1$ and computations are made in finite precision arithmetic, the b-component of Av_1 can be severely affected by rounding errors.

The second step from (1) can be written as follows:

$$\begin{aligned} \beta_{k+1}v_{k+1} = &\sum_{i=1}^{n-2}\lambda_i(x_i^T v_k)(I - V_kV_k^T)x_i \\ &+ [\mu(a^T v_k) + \nu(b^T v_k)](I - V_kV_k^T)a \\ &+ [\mu(b^T v_k) + \nu(a^T v_k)](I - V_kV_k^T)b. \end{aligned} \tag{2}$$

Let us define the subspace $\mathcal{V}_k = \operatorname{span}(V_k)$ and its orthogonal complement $\mathcal{V}_k^{\perp}$. Then:

$$\begin{aligned}(I - V_k V_k^T)x_i &= \sin\angle(x_i, \mathcal{V}_k)\, n_i^{(k)}, \quad \text{where} \quad n_i^{(k)} \in \mathcal{V}_k^{\perp}, \|n_i^{(k)}\| = 1,\\ (I - V_k V_k^T)a &= \sin\angle(a, \mathcal{V}_k)\, n_a^{(k)}, \quad \text{where} \quad n_a^{(k)} \in \mathcal{V}_k^{\perp}, \|n_a^{(k)}\| = 1,\\ (I - V_k V_k^T)b &= \sin\angle(b, \mathcal{V}_k)\, n_b^{(k)}, \quad \text{where} \quad n_b^{(k)} \in \mathcal{V}_k^{\perp}, \|n_b^{(k)}\| = 1.\end{aligned} \tag{3}$$

The set of equations in (3) defines the normal vectors $n_i^{(k)}$, $n_a^{(k)}$, $n_b^{(k)}$ that can be again decomposed in our orthonormal basis. Now we can use this decomposition together with (3) in (2), but we will write the expression only for the b-component:

$$\begin{aligned}\beta_{k+1}(b^T v_{k+1}) = {} & [\mu \sin\angle(b, \mathcal{V}_k)(b^T n_b^{(k)}) + \nu \sin\angle(a, \mathcal{V}_k)(b^T n_a^{(k)})]\,(b^T v_k)\\ & + [\mu \sin\angle(a, \mathcal{V}_k)(b^T n_a^{(k)}) + \nu \sin\angle(b, \mathcal{V}_k)(b^T n_b^{(k)})]\,(a^T v_k)\\ & + \sum_{i=1}^{n-2} \lambda_i \sin\angle(x_i, \mathcal{V}_k)(x_i^T v_k)(b^T n_i^{(k)}).\end{aligned} \tag{4}$$

To analyze (4) in general seems to be difficult. However, when b *remains perpendicular to* $\mathcal{V}_k$, i.e., $b \in \mathcal{V}_k^{\perp}$ (see Fig. 3 for all iterations ≤ 21), one gets:

$$b^T v_{k+1} = \frac{\mu}{\beta_{k+1}}\,(b^T v_k) + \frac{\nu}{\beta_{k+1}}\,(a^T v_k). \tag{5}$$

When $\mu/\beta_{k+1} > 1$, (5) suggests an *amplification* of previous b-component and its subsequent slight *modification* (since ν is very small).

In finite precision arithmetic, $b^T v_1 = 0$ and $b^T v_2$ is very small (regardless to the implementation) so that it is prone to rounding errors (which *depend* on implementation). This small difference in b-component of v_2 between two implementations is *amplified* according to (5), when the b-component increases. Hence, the loss of correlation between two bases starts right from the beginning of computation and becomes evident when $|b^T v_k| \approx \sqrt{\epsilon} \approx 10^{-8}$. On the other hand, the recapture of correlation is possible only when the whole eigenspace $\mathcal{X}_2$ is well approximated by the last two Ritz vectors. This is equivalent to the fast decrease of $|b^T v_k|$ after the iteration $k = 22$ and to the tight approximation of both λ_{n-1} and λ_n by two largest Ritz values.

When A has an exactly double maximal eigenvalue, the last Ritz vector converges again to a, i.e., to the orthogonal projection of v_1 into $\mathcal{X}_2$. However, since now any linear combination of vectors a and b *is* an eigenvector, there arises no 'spurious' component along b in the matrix-vector multiplication. Therefore, the whole eigenspace $\mathcal{X}_2$ is approximated only in the last iteration $k = n$ and there is no loss of correlation between two computed Arnoldi bases.

Acknowledgment: G. Okša was supported by the VEGA grant no. 2/0003/11. The work of M. Rozložník was supported by Grant Agency of the Academy of Sciences of the Czech Republic under the project IAA100300802 and by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Application".

Parameter estimation of reaction-diffusion model based on spatio-temporal FRAP images of thylakoid membranes

Š. Papáček, D. Štys, R. Kaňa, C. Matonoha

[1,2] Institute of Physical Biology, University of South Bohemia, Nové Hrady
[3] Institute of Microbiology AS CR, Třeboň
[4] Institute of Computer Science AS CR, Prague

1 Introduction

The determination of phycobilins diffusivity in thylakoid lumen from fluorescence recovery after photobleaching (FRAP) experiments was usually done by analytical models [5, 3]. However, the analytical models need some unrealistic conditions to be supposed. This study describes the development and validation of a method based on finite difference simulation of diffusion process governing by the Fickian diffusion equation and on the minimizing of an objective function representing the disparity between the experimental and simulated time-varying concentration profiles.

2 Model development

2.1 Theory

During a FRAP experiment, a sample either containing a fluorescent solute or having the natural capacity for fluorescent signal emission, is briefly exposed to intense laser illumination to bleach a target region of a specified geometry (in our case, the computational domain is an Euclidian 2D rectangular domain). For an arbitrary bleach spot and assuming (i) local homogeneity (assuring that the concentration profile is smooth), (ii) isotropy (diffusion coefficient is space-invariant), (iii) an unrestricted supply of unbleached particles outside of the target region, and (iv) negligible out-of-domain concentration gradients, the recovery of unbleached particle concentration C as a function of spatial coordinate $\vec{r}$ and time t is modelled with a following diffusion-reaction equation on two-dimensional domain Ω:

$$\frac{\partial C}{\partial t} - \nabla \cdot (D \nabla C) = R(C) \ , \tag{1}$$

where D is the fluorescent particle diffusivity in domain Ω (i.e. in some selected part of thylakoid lumen), and $R(C)$ is a reaction term modelling the binding of particles.

The initial condition, and time varying Dirichlet boundary conditions are:

$$C_0 = C(t_0) \ \text{ on } \Omega, \quad C(t) = g(\vec{r}, t) \text{ on } \partial\Omega. \tag{2}$$

The reaction term $R(C)$ is often viewed as negligible under assumptions that the fluorescent molecules do not bind to the medium and that photobleaching of these molecules during recovery is negligible. Consequently, if $R(C)$ is neglected, (1) becomes the Fickian diffusion equation. In contrast, under continual photobleaching during image acquisition, this reaction term could be described as a first order reaction:

$$R(C) = -k_S\ C\ , \tag{3}$$

where k_S is a rate constant describing bleaching during scanning [2].

Another source of error, often negligible, is the time dependence of the fluorescent signal ϕ emitted by fluorescent particles. Although within (1) and within objective function J, cf. (8), we use the concentrations C, in fact we measure the fluorescence level and not directly C. If the following relation holds: $C = k_F\phi$, where k_F is a constant, than we can work with the measured signal without necessity of any recalculation (e.g. by a normalization of the overall signal). On the contrary, if k_F is time dependent, then we should design an experiment and estimate this dependence, in order to have a correct form of (1).

2.2 One dimensional model

For a linear bleach spot perpendicular to a longer axis (let this axis be denoted as r) and assuming local homogeneity and isotropy, an unrestricted supply of unbleached solute outside of the target region and negligible out-of-domain concentration gradients, recovery of unbleached particle concentration as a function of spatial coordinate and time t is modeled with a linear, diffusion-reaction equation:

$$\frac{\partial C}{\partial t} - D\frac{\partial^2 C}{\partial r^2} = R(C)\ , \tag{4}$$

Furthermore, adopting the form of reaction term according to (3), and after introducing the dimensionless spatial coordinate x, the dimensionless diffusion coefficient p, the dimensionless time τ and the dimensionless concentration y by

$$r := xL\ ,\ D := p\ D_0\ ,\ t := \tau\frac{L^2}{D_0}\ \ ,\ y := \frac{C}{c_m}\ , \tag{5}$$

where L is the length of our specimen in direction perpendicular to bleach spot, D_0 is a constant with some characteristic value (unit: $\mathrm{m}^2\mathrm{s}^{-1}$), and c_m is a characteristic (e.g. maximal) concentration of C, we finally have the following form of dimensionless diffusion-reaction equation on one-dimensional domain, i.e. for $x \in [0,1]$

$$\frac{\partial y}{\partial \tau} - p\frac{\partial^2 y}{\partial x^2} = -\frac{k_S L^2}{D_0} y\ . \tag{6}$$

The initial condition, and time varying Dirichlet boundary conditions are:

$$y_0 = y(x,\tau_0)\ \text{ for } x \in [0,1],\quad y(0,\tau) = g_0(\tau),\quad y(1,\tau) = g_1(\tau). \tag{7}$$

2.3 Experimental data

Based on FRAP experiments, see Fig. 1, we have not a smooth function for the initial condition, but a vector of values $y_{exp}(x_i, t_0)$, $i = 1,...N$. Similarly, for the boundary conditions we have two vectors, each one composed from M values, M is the number of time points in the time axis, where the measurements were taken: $y_{exp}(0, t_j)$, $j = 1,...M$, on the left, and $y_{exp}(1, t_j)$, $j = 1,...M$, on the right edge of interval [0,1]. The resting experimental data, in fact characterizing the diffusion process, form a 2D matrix of dimension (N,M), which can be read by columns as the concentration profiles (along x axis) in M discrete time points. The forthcoming task is the analysis of measurement noise and its correct filtering.

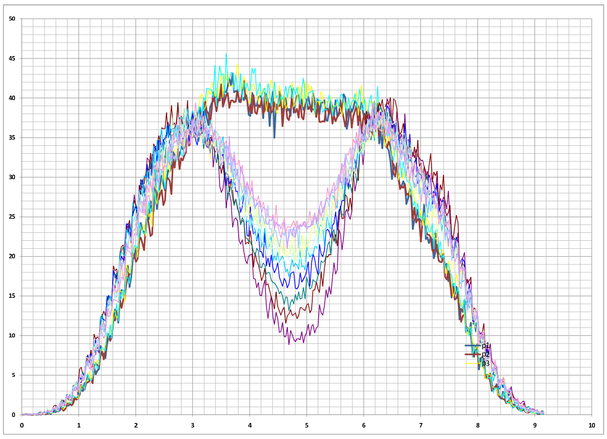


Figure 1: An example of time series of FRAP measurements with photosynthetic proteins.

2.4 Determination of diffusivity as a single parameter estimation problem

The problem of phycobilins diffusivity determination based on time series of experimental data will be further formulated as a single parameter estimation problem. We construct an objective function J representing the disparity between the experimental and simulated time-varying concentration profiles, and then within a suitable method we look for such a value p minimizing J. The usual form of an objective function is the sum of squared differences between the experimentally measured and numerically simulated time-varying concentration profiles:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{M} \left[y_{exp}(x_i, \tau_j) - y_{sim}(x_i, \tau_j) \right]^2 \ , \tag{8}$$

where $y_{sim}(x_i, \tau_j)$ are the simulated values resulting from the solution of PDE (6) with the initial and boundary conditions (7). The implementation of both direct problem, i.e. the solution of PDE (6) with the initial and boundary conditions (7) for the known parameter p, and a single parameter estimation problem is describe in the following section.

3 Implementation

Firstly we started neglecting the reaction term (i.e. we put $k_S = 0$). Hence, we are minimizing J with respect to p, which represents a one-dimensional optimization problem. We have used a suitable optimization method from the UFO system which generates a sequence of iterates $\{p_k,\ k > 0\}$ leading to a value which minimizes J (see [4]). In order to compute a function value of J_k in (8) for a given p_k in the k-th iteration, we need to know both the values of $y_{exp}(x_i, \tau_j)$, $i = 1, ..., N$, $j = 1, ..., M$, and the simulated values $y_{sim}(x_i, \tau_j)$, $i = 1, ..., N$, $j = 1, ..., M$, for a given p_k as well. It means that in each iteration we need to solve the problem (6)-(7) for the initial and boundary conditions defined by the current value of p_k and the experimental data: $y_0 = y_{exp}(x, \tau_0)$ for $x \in [0, 1]$, $y(0, \tau) = y_{exp}(0, \tau)$, $y(1, \tau) = y_{exp}(1, \tau)$.

This 'direct' problem was solved numerically using the finite difference scheme for uniformly distributed nodes with the space steplength Δh and time steplength $\Delta\tau$. We have used an explicit scheme, cf. [1], which can be generally written in the form

$$y(x_i, \tau_j + \Delta\tau) = \beta y(x_i - \Delta x, \tau_j) + (1 - 2\beta) y(x_i, \tau_j) + \beta y(x_i + \Delta x, \tau_j),$$

where (x_i, τ_j) is an inner node of the difference scheme and $\beta = p_k \frac{\Delta\tau}{\Delta h^2}$ (p_k is the value in the k-th iteration). It is known that in this case the condition $\beta \leq 1/2$ has to be satisfied.

Taking into account the biological reality residing in possible time dependence of phycobilins diffusivity, we further consider two cases. First, we can take both sums for i and j in (8) together. In this case, the scalar p is a result of minimization problem for J. Secondly, we can consider each i-th space row separately. In this case, the N solutions $p^{(1)}, ..., p^{(N)}$ correspond to each minimization problem for fixed i in the sum (8) and we have a 'dynamics' of diffusivity p evolution.

Our program is actually under testing, however, for the previously known diffusion coefficient and the data simulated by the random walk model it computes correct results. Afterward, we determined the diffusivities for the real data of FRAP measurements (with the red algae *Porphyridium cruentum*). The range of result 10^{-14} $\mathrm{m^2s^{-1}}$ is in agreement with reference values.

4 Conclusion

Our method for diffusion parameter estimation from FRAP data improves on other models by accounting for experimentally measured post-bleaching fluorescence profiles and time-varying boundary conditions, and can includes a reaction term to account for the time varying fluorescence signal (maybe due to the detrimental effects of low level photobleaching produced by image acquisition during recovery). Analysis of simulated FRAP data demonstrate the advantages of this method over common analytical approaches, including a low sensitivity to variations in the spot radius and to the effects of photobleaching during scanning.

Acknowledgement: This work was supported by the grant MŠMT MSM 600 766 58 08 of the Ministry of Education, Youth and Sports of the Czech Republic, by the Grant Agency of the Czech Republic through the research grant No. 102/05/0011, the institutional research plan No. AV0Z10300504, project: „Jihočeské výzkumné centrum akvakultury a biodiverzity hydrocenóz“ (CENAKVA), OP VaVpI, and by the South Bohemia University grant GAJU 152/2010/Z.

References

[1] I. Babuška, M. Práger, E. Vitásek: *Numerical processes in differential equations.* John Wiley & Sons, London, 1966.

[2] O.N. Irrechukwu, M.E. Levenston: *Improved estimation of solute diffusivity through numerical analysis of FRAP experiments.* Cellular and Molecular Bioengineering 2(1), 2009, 104–117.

[3] R. Kaňa , O. Prášil, C.W. Mullineaux: *Immobility of phycobilins in the thylakoid lumen of a cryptophyte suggests that protein diffusion in the lumen is very restricted.* FEBS letters 583(4), 2009, 670–674.

[4] L. Lukšan, M. Tuma, J. Vlček, N. Ramešová, M. Šiška, J. Hartman and C. Matonoha: *UFO 2008 - Interactive system for universal functional optimization.* Technical Report V-1040, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague 2008.

[5] C.W. Mullineaux, M.J. Tobin, G.R. Jones: *Mobility of photosynthetic complexes in thylakoid membranes.* Nature 390, 1997, 421-–424.

Error estimates and domain decomposition methods

I. Pultarová

Faculty of Civil Engineering, Czech Technical University in Prague

1 Introduction

During a process of numerical solution of partial differential equations using domain decomposition methods, a good error indicator could help us to decide whether the error of a current approximation is sufficiently low or not on a particular subdomain. If we use the domain decomposition method balanced by constraints (BDDC) [2], we can decrease or increase the number of coarse degrees of freedom (DOF) on such subdomains. We derive our further considerations from the equilibrated residual strategy which is described in [1] and developed e.g. in [3]. The a posteriori error estimation techniques can be used though the current solution is not the exact solution of the underlying linear system. In this contribution we discuss how the estimates can be applied to BDDC methods without much additional effort.

Let us suppose a second order elliptic partial differential equation in a two-dimensional domain Ω with homogeneous Dirichlet boundary conditions on the boundary $\partial\Omega$. Let the weak formulation be to find $u_W \in W$ such that

$$B(u_W, v) = (f, v),$$

$v \in W$, where W is an appropriate function space. Ordinary and energy scalar products (u, v) and $B(u, v)$ are defined as usual. Let V be a space of finite element (FE) linear or bilinear functions on triangular or quadrilateral mesh satisfying the boundary conditions. Let us denote by u_V the solution in V

$$B(u_V, v) = (f, v)$$

for all $v \in V$. This discretized problem can be represented by a system of linear equations

$$Ku = b.$$

Partition Ω into subdomains Ω_m, $m = 1, \dots, n$, yields n separate problems, some of them indefinite. Let the subscript o denote DOFs belonging to internal nodes of all subdomains and let the DOFs of nodes on internal boundaries of all subdomains have subscript r. After reordering the nodes and after assembling the blocks by integrating only over individual subdomains, we get a new matrix of the system of algebraic equations

$$\begin{pmatrix} K_o & K_{or} \\ K_{or}^T & K_r \end{pmatrix} \begin{pmatrix} u_o \\ u_r \end{pmatrix} = \begin{pmatrix} f_o \\ f_r \end{pmatrix}. \tag{1}$$

Submatrix K_o is block diagonal and positive definite, its dimension equals to the number of all internal nodes. Matrix K_r is positive semidefinite and its dimension is larger than the number of nodes on interfaces because each of the interface DOFs belongs to more than one subdomain. After elimination of K_{or}^T we get a Schur complement formulation for the interface unknowns u_r

$$Su_r = f_S, \tag{2}$$

where

$$S = K_r - K_{or}^T K_o^{-1} K_{or}, \qquad f_S = f_r - K_{or}^T K_o^{-1} f_o.$$

In the BDDC methods, a coarse problem is built and solved of a dimension much lower than that of S in order to transfer the information among the subdomains and to provide the subproblems with the Dirichlet boundary condition.

2 Equilibrated residual method for subdomains

The equilibrated residual method for a posteriori error estimates is described in [1]. Fluxes over element edges are calculated and smoothed on every patch of elements which share a single vertex. Then the energy norm of the error is computed from the solution of Neumann problems on all elements. In our approach, we exploit this basic idea, but there are two main differences. First, instead of patches of elements we use subdomains and moreover, only the interface unknowns are calculated with. Second, we can compute the estimates in every BDDC iteration, it means that we do not need the exact solution of the linear systems (1) or (2).

For the error of an approximate solution u_i in step i, we have $e = u_i - u_W \in W$. Then the energy norm of the error $|||e|||$ is

$$|||e||| = \sup_{v\in W,\,|||v|||=1} B(e,v) = \sup_{v\in W,\,|||v|||=1} B(u_i - u_W, v) = \sup_{v\in W,\,|||v|||=1} (B(u_i,v) - (f,v)).$$

The involved scalar products can be computed over the individual subdomains. Let us consider a set of functions g defined on boundaries of subdomains inside Ω such that

$$\sum_m \int_{\partial\Omega_m} gv \,\mathrm{d}s = 0.$$

Then we have

$$|||e||| = \sup_{v\in W,\,|||v|||=1} \sum_m \left(B(u_{im}, v) - (f,v) + \int_{\partial\Omega_m} gv \,\mathrm{d}s \right), \tag{3}$$

where u_{im} is u_i restricted to Ω_m. The right hand side of (3) can be substituted by

$$|||e||| = \sup_{v\in W,\,|||v|||=1} \sum_m B(\phi_m, v),$$

where $\phi_m \in W_m$ is a solution of

$$B(\phi_m, v) = B(u_{im}, v) - (f,v) + \int_{\partial\Omega_m} gv \,\mathrm{d}s \tag{4}$$

on Ω_m, $v \in W_m$, where W_m is an appropriate function space on Ω_m, $m = 1, 2, \dots, n$. If some domain Ω_m does not coincide with $\partial\Omega$, then the associated problem has only Neumann boundary conditions given by g on $\partial\Omega_m$. When g are the outer normal derivatives of the exact solution on $\partial\Omega_m$, we obtain the exact error e_m on Ω_m. In any case we have

$$|||e||| \le \sum_m |||\phi_m|||. \tag{5}$$

After discretization of (4), we have

$$B(\phi_m, v) = B(u_{im}, v) - (f,v) + \int_{\partial\Omega_m} gv \,\mathrm{d}s, \tag{6}$$

where ϕ_m and v are from FE function spaces on Ω_m, $m = 1, 2, \dots, n$. Then of course instead of (5) we obtain only an error indicator.

Let us stress that there are only two conditions that must be fulfilled: a) the sum of the chosen fluxes g have to be zero, b) the problems on interior subdomains must be solvable.

Matrix representation of the introduced considerations can be as follows. Let the systems

$$K_m u = r_m \tag{7}$$

represent the discretized equations (6) and let

$$S_m u = r_{Sm} \tag{8}$$

are the associated Schur complement representations. Adding fluxes g on subdomains means adding vectors $\tilde{r}_m$ to right hand sides of (7) or equivalently $\tilde{r}_{Sm}$ to (8) to the positions of the interface unknowns.

Condition a) is fulfilled for example whenever the fluxes have zero sums on every interface of a pair of subdomains. Condition b) is fulfilled if for interior subdomains the equations (7) or equivalently (8) are solvable. We can calculate the fluxes for patches of subdomains, but we can also equilibrate the residuals at the same time for all edges by solving one system of equations. Of course, such set of fluxes g or equivalently of vectors $\tilde{r}_{Sm}$ (or $\tilde{r}_m$) is not unique. In our experiments we choose vectors $\tilde{r}_{Sm}$ like multiples of residuals r_{Sm} on each interface edge. We follow the idea of [1] and minimize the distances of the resulting right hand sides $r_{Sm} + \tilde{r}_{Sm}$ from averages of residuals which belong to opposite sides of an interface shared by any two subdomains. We can simplify the equilibrating of residuals in such manner that only the sums of fluxes over whole interfaces are balanced and not over the individual elements. Then the dimension of this problem is equal to the number of interfaces between subdomains.

Instead of a posteriori error estimates, this method rather yields suggestions of residual partitioning for the BDDC method. The estimate is an indicator of $|||u_i - u_V|||$, where u_V is the exact FE solution of the problem.

3 Numerical example

Let us solve the equation

$$\frac{\partial^2 u}{\partial x^2} + 10^{-3}\frac{\partial^2 u}{\partial y^2} = 1$$

in $\Omega = (0,1) \times (0,1)$ with $u = 0$ on $\partial\Omega$. Let Ω be partitioned into 3×3 rectangular subdomains. We solve this problem by the conjugate gradient method which is preconditioned by BDDC method and use bilinear FEs on rectangular elements.

Error estimates in energy norm after the forth step of the conjugate gradient method are displayed in Figure 1 for different choices of the mesh resolution and compared with the exact error and with the error computed from residuals r_{Sm} on subdomains

$$|||e_r||| = r_{Sm}^T S_m^{\#} r_{Sm},$$

where $S_m^{\#}$ is the Moore-Penrose pseudoinverse of S_m. The estimates for the overall errors are presented on the left, while on the right, the estimates are shown only for the central subdomain which does not coincide with the boundary $\partial\Omega$. The mesh resolutions are from 5 to 15 nodes in every subdomain in each direction. In this example, the BDDC method uses all corner nodes and one average on each interface edge as coarse DOFs.

Acknowledgement: This work has been supported by the project CEZ MSM 6840770001 and by the Grant Agency of The Czech Republic under the contract No. 201/09/P500 and No. 201/09/1544.

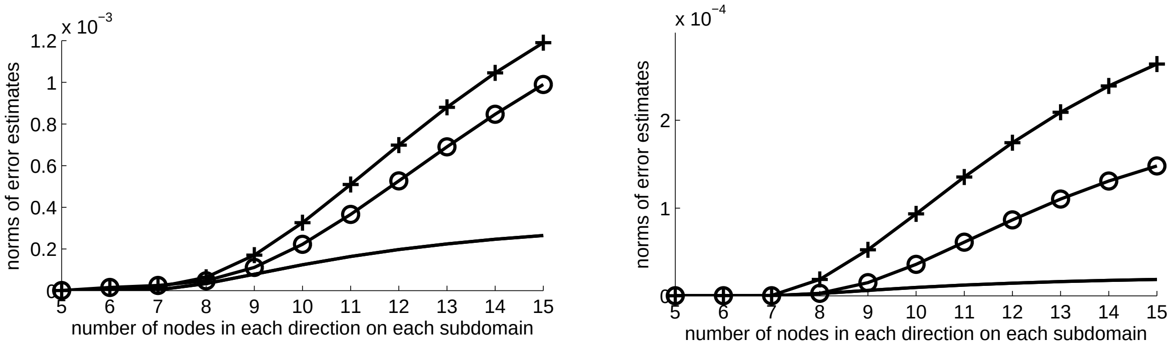


Figure 1: Error estimates in energy norm for different meshes after the forth step of the conjugate gradient method preconditioned by BDDC. True error (simple line), residual estimate (croses) and equilibrated residual based estimate (circles). Error estimates on Ω (left) and on the central subdomain (right).

References

[1] M. Ainsworth, J.T. Oden: *A posteriori error estimation in finite element analysis.* John Wiley & Sons, Inc., 2000.

[2] J. Mandel, B. Sousedík, C.R. Dohrmann: *Multispace and multilevel BDDC.* In: Computing 83, 2008, 55–85.

[3] T. Vejchodský: *Guaranteed and locally computable a posteriori error estimate.* In: IMA Journal of Numerical Analysis 26, 2006, 525–540

On a posteriori error estimates for biharmonic problems

K. Segeth

Technical University of Liberec

1 Introduction

In this survey contribution, we present and compare, from the viewpoint of adaptive computation, several recently published error estimation procedures for the numerical solution of biharmonic and some further fourth order problems mostly in 2D, including computational error estimates.

In the *hp*-adaptive finite element method, there are two possibilities to assess the error of the computed solution a posteriori: to construct a classical *analytical error estimate* (see their classification in [8]) or to obtain, by the same procedure as the approximate solution, a *computational error estimate*. In the latter case, a *reference solution* is computed on a systematically refined mesh and, at the same time, with the polynomial degree of all elements increased by 1.

We use common notation based primarily on the book [3]. For the lack of space, we sometimes only refer to the notation introduced in the papers quoted. The complete hypotheses of the theorems presented should be also looked for there.

2 Dirichlet and second problems for biharmonic equation

2.1. Dirichlet problem. Let $\Omega \subset R^2$ have a polygonal boundary Γ. We consider the 2D biharmonic problem

$$\Delta^2 u = f \quad \text{in} \quad \Omega, \tag{1}$$

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \Gamma \tag{2}$$

with $f \in L_2(\Omega)$ that models, e.g., the vertical displacement of the mid-surface of a clamped plate subject to bending.

We use the standard formulation of the weak solution $u \in X = H_0^2(\Omega)$ and approximate solution $u_h \in X_h$ written in the form $\langle F(u), v\rangle = 0$ and $\langle F_h(u_h), v_h\rangle = 0$. Denote by k, $k \geq 1$, the maximum degree of polynomials in X_h. Further, put $f_h = \sum_{T\in\mathcal{T}_h} \pi_{l,T} f$, where T is a triangle of the triangulation $\mathcal{T}_h$, $\mathcal{E}_h$ is the set of all its edges, P_l, $l \geq 0$ fixed, is the space of polynomials of degree at most l and $\pi_{l,S}$, $S \in \mathcal{T}_h \cup \mathcal{E}_h$, is the L_2 orthogonal projection of $L_1(S)$ onto $P_l(S)$. Put $\varepsilon_T = \|f - f_h\|_{0;T}$. Let h_T be the diameter of the triangle T. Defining the *local residual a posteriori error estimator* $\eta_{\mathrm{V},T}$ for all $T \in \mathcal{T}_h$, we have the following theorem [8].

Theorem 2.1. *Let $u \in X$ be the unique weak solution of the problem* (1), (2) *and let $u_h \in X_h$ be an approximate solution of the corresponding discrete problem. Then we have the a posteriori estimates*

$$\|u - u_h\|_2 \leq c_1 \left(\sum_{T\in\mathcal{T}_h} \eta^2_{\mathrm{V},T}\right)^{1/2} + c_2 \left(\sum_{T\in\mathcal{T}_h} h_T^4 \varepsilon_T^2\right)^{1/2} + c_3\|F(u_h) - F_h(u_h)\| + c_4\|F_h(u_h)\|$$

and

$$\eta_{V,T} \leq c_5\|u - u_h\|_{2;\omega_T} + c_6 \left(\sum_{T' \subset \omega_T} h_{T'}^4 \varepsilon_{T'}^2 \right)^{1/2}$$

for all $T \in \mathcal{T}_h$. The quantities $c_1, \ldots, c_6$ depend only on h_T/ρ_T, and the integers k and l. Here ω_T is the set of all neighbors of the triangle T and ρ_T the diameter of the circle inscribed to T.

The proof is given in [8].

The same problem is treated in, e.g., [9] with a residual error estimator giving similar results.

2.2. Dirichlet problem in mixed formulation. Let $\Omega \subset R^2$ be a convex polygon with boundary Γ. Again, we consider the biharmonic problem (1), (2) with $f \in H^{-1}(\Omega)$. The problem is concerned in practice with both linear plate analysis and incompressible flow simulation.

We employ the Ciarlet-Raviart weak formulation of the problem (1) and (2) for the solution $\{w = \Delta u, u\}$ and the corresponding conforming second order approximate solution $\{w_h, u_h\}$. Let us put $f_h = \pi_{0,T} f$ on $T \in \mathcal{T}_h$.

The local residuals $\mathcal{P}_T$, $\mathcal{R}_T$, $\mathcal{P}_E$, and $\mathcal{R}_E$ are defined in [2]. We introduce the *local residual a posteriori error estimators* $\eta_{C,T}$ and $\widetilde{\eta}_{C,T}$ computed from the local residuals. We put $e_h(u) = u - u_h$ and $e_h(w) = w - w_h$. Then the following theorem holds [2].

Theorem 2.2. *Let $\{w, u\}$ be the unique mixed weak solution of the problem* (1) *and* (2)*, and let $\{w_h, u_h\}$ be an approximate solution of the corresponding discrete problem. For $T \in \mathcal{T}_h$ we then have the a posteriori estimates*

$$\|e_h(u)\|_1 + h\|e_h(w)\|_0 \leq C_1 \left(\left(\sum_{T \in \mathcal{T}_h} \eta_{C,T}^2 \right)^{1/2} + h^2 \left(\sum_{T \in \mathcal{T}_h} \widetilde{\eta}_{C,T}^2 \right)^{1/2} \right),$$

$$\eta_{C,T} + h^2 \widetilde{\eta}_{C,T} \leq C_2 \left(|e_h(u)|_{1;\omega_T} + h_T\|e_h(w)\|_{0;\omega_T} + h_T^3 \sum_{T' \subset \omega_T} \varepsilon_{T'} \right)$$

with some positive constants C_1 and C_2 independent of $h = \max_{T \in \mathcal{T}_h} h_T$.

The proof is given in [2].

The second problem for the biharmonic equation is treated in mixed formulation in [5] with a gradient recovery error estimator.

2.3. Kirchhoff plate bending problem. A similar problem describing the bending of an isotropic linearly elastic plate is studied in [1]. The nonconforming finite element approximation of the problem is constructed in the discrete Morley space and the residual error estimator is used.

3 Dirichlet problem for fourth order elliptic equation

3.1. Dirichlet problem in 1D. Put $\Omega = (0, 1) \subset R^1$. Let all the functions concerned be scalar-valued functions of a scalar variable. We consider the one dimensional boundary value problem for the ordinary fourth order equation

$$(au'')'' = f \quad \text{in} \quad \Omega \tag{3}$$

with the boundary conditions

$$u(0) = u'(0) = 0, \quad u(1) = u'(1) = 0. \tag{4}$$

This is a model for the vertical displacement of a beam clamped on both ends and subject to bending. In the model, $a(x) = E(x)I(x)$ is a positive, bounded, and Lipschitz continuous function in Ω, where E is Young's modulus of elasticity and I the moment of inertia. The distributed transverse load is denoted by $f \in L_2(\Omega)$.

We use the standard formulation of the weak solution $u \in X = H_0^2(\Omega)$ and $u_h \in X_h$, i.e., $a(u,v) = \int_\Omega fv$ and $a(u_h, v_h) = \int_\Omega f v_h$, X_h being the space of piecewise cubic Hermite polynomials. Moreover, we use the corresponding energy norm $|\!|\!| v |\!|\!|^2 = a(v,v)$.

In [6], a *recovery operator* Gv_h for the second derivative of $v_h \in X_h$ is introduced. Now we can define the *local recovery a posteriori error estimator* $\eta_{\mathrm{P},T}$ for all triangles T of the triangulation $\mathcal{T}_h$ and have the following theorem [6].

Theorem 3.1. *Let $u \in H_0^2(\Omega)$ be the unique weak solution of the problem* (3), (4) *and let $u_h \in X_h$ be an approximate solution of the corresponding discrete problem. Then we have the global a posteriori estimate*

$$\left| \left(\sum_{T \in \mathcal{T}_h} \eta_{\mathrm{P},T}^2 \right)^{1/2} - |\!|\!| u - u_h |\!|\!| \right| \le \left\| a^{1/2} \left(G_h u_h - \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} \right) \right\|_0 \le Ch^3$$

for the difference of the global error estimator and the energy norm of the true error. C is a constant that may depend on u. The global error estimator is asymptotically exact.

The proof is given in [6].

3.2. Dirichlet problem. Let $\Omega \in R^n$ be a bounded connected domain and Γ its Lipschitz continuous boundary. We consider the 4th order elliptic problem for a scalar-valued function u,

$$\operatorname{div} \operatorname{Div}(\gamma \nabla\nabla u) = f \quad \text{in} \quad \Omega, \tag{5}$$

with the boundary condition (2) and $f \in L_2(\Omega)$, $\gamma = [\gamma_{ijkl}]_{i,j,k,l=1}^n$ and $\gamma_{ijkl} = \gamma_{jikl} = \gamma_{klij} \in L_\infty(\Omega)$.

We define the energy norm $|\!|\!| \Phi |\!|\!|$ in $L_2(\Omega, R^{n\times n})$ and the *global a posteriori error estimator* $\eta_{\mathrm{R}}(\beta, \Phi, \bar{u})$ like in [7], where β is an arbitrary positive real number and Φ an arbitrary smooth matrix-valued function. The estimator depends on the constant from the Friedrichs inequality for $\nabla\nabla$ on $H_0^2(\Omega)$. We then have the following theorem [7].

Theorem 3.2. *Let $u \in H_0^2(\Omega)$ be the weak solution of the problem* (5), (2) *and $\bar{u} \in H_0^2(\Omega)$ an arbitrary function. Then*

$$|\!|\!| \nabla\nabla(\bar{u} - u) |\!|\!|^2 \le \eta_{\mathrm{R}}(\beta, \Phi, \bar{u}) \tag{6}$$

for any positive number β and any matrix-valued function $\Phi \in H(\operatorname{div}\operatorname{Div}, \Omega)$.

The proof of the theorem is based on a more general statement proven in [7]. There is an interesting question of optimizing the inequality (6) with respect to β and Φ.

A similar 2D nonlinear Dirichlet problem is solved in [4]. A global error estimator is introduced and similar results are obtained there.

4 Conclusion

The quantitative properties of the estimators cannot be easily assessed and compared analytically. There are, however, global analytical error estimates for some classes of problems (see, e.g., [4], [7]) that require as few unknown constants as possible. The a posteriori estimates with unknown constants, however, are not optimal for the practical computation. They can be efficient if they are asymptotically exact.

The computation of the reference solution is rather time-consuming but the refence solution is obtained by the same software that is used to compute the approximate solution. We use reference solutions as robust error estimators with no unknown constants to control the adaptive strategies in the most complex finite element computations.

Acknowledgement: This work has been carried out under the state subsidy to the research and development project Advanced Remediation Technologies and Processes Center 1M0554 of the Programme of Research Centers supported by the Ministry of Education, Youth, and Sports of the Czech Republic.

References

[1] L. Beirão da Veiga, J. Niiranen, R. Stenberg: *A posteriori error estimates for the Morley plate bending element.* Numer. Math. 106, 2007, 165–179.

[2] A. Charbonneau, K. Dossou, R. Pierre: *A residual-based a posteriori error estimator for the Ciarlet-Raviart formulation of the first biharmonic problem.* Numer. Methods Partial Differential Equations 13, 1997, 93–111.

[3] P.G. Ciarlet: *The finite element method for elliptic problems.* North Holland, Amsterdam, 1978.

[4] J. Karátson, S. Korotov: *Sharp upper global a posteriori error estimates for nonlinear elliptic variational problems.* Appl. Math. 54 (2009), pp. 297–336.

[5] K. Liu, X. Qin: *A gradient recovery-based a posteriori error estimators for the Ciarlet-Raviart formulation of the second biharmonic equations.* Appl. Math. Sci. 1, 2007, 997–1007.

[6] S.B. Pomeranz: *A posteriori finite element method error estimates for fourth-order problems.* Comm. Numer. Methods Engrg. 11, 1995, 213–226.

[7] S. Repin: *A posteriori estimates for partial differential equations.* Walter de Gruyter, Berlin, 2008.

[8] R. Verfürth: *A review of a posteriori error estimation and adaptive mesh refinement techniques.* John Wiley & Sons, Chichester, and B.G. Teubner, Stuttgart, 1996.

[9] M. Wang, W. Zhang: *Local a priori and a posteriori error estimate of TQC9 element for the biharmonic equation.* J. Comput. Math. 26, 2008, 196–208.

Experimental grid for numerical linear algebra

I. Šimeček, J. Hladík, J. Krupka, M. Hovorka

Faculty of Information Technologies, Czech Technical University in Prague

1 Introduction

Time is very often the limiting factor in scientific codes. These codes can be accelerated by parallel executing on special distributed systems (grids). This is usual but very difficult solution. In this paper, we describe a design of the new heterogenous grid for the numerical linear algebra with maximal ratio between prize and computational power. Contributions of this paper is twofold: 1) a design of new parallel routines 2) an approach for parallelization of scientific codes by converting local numerical library calls into remote grid calls.

1.1 GPU computing

Nowadays, there is a new trend in the high-performance computing to accelerate computations by means of Graphics Processing Units (GPU). This trend recently emerged into a new research area called General-Purpose Computing on Graphics Processing Units (shortly GPGPU). This is a consequence of the fact that the GPUs of modern graphic cards overcome modern CPUs in the memory bandwidth, the number of computational units, and possibilities of the vector execution. The GPGPU programming is simplified by several existing APIs (Application Programming Interfaces), the most popular and well-established ones are CUDA [1, 2] and OpenCL [3]. Thanks these APIs the GPGPU computations are widespread and used in many scientific projects.

The computational abilities of single GPU are very impressive, but some problems, especially with large memory requirements, are still hard to solve. Although the amount of memory on GPUs is increasing rapidly, it is still much less than we need and this leads to the limited application of GPGPU in many scientific problems. Possible solution to that problem could be to connect graphic cards into a GPGPU cluster to distribute computing and memory demands across all available GPU. The benefit of this approach is that it allows us to interconnect GPUs from various vendors but naturally there arise a new problem known as load balancing of GPUs that we have to face to retain high computational performance.

1.2 Sparse matrix storage formats

The sparse matrix storage scheme (format) have great impact on performance and scalability of the sparse matrix-vector multiplication operation and other iterative algorithms for sparse matrix computations. Ideal format ensures minimal memory storage requirements, maximum utilization of floating point vector units, maximum utilization of cache memories, and enables load balanced parallelization of the algorithms on massively parallel systems.

Several sparse matrix formats have been proposed and some are due to their simplicity widely used, such as Compressed Sparse Row/Column (CSR/CSC) or Jagged Diagonal Storage (JDS) formats. The feasibility of particular format is given mainly by the sparsity pattern of a matrix. Sparse matrices often contain dense submatrices (blocks). Therefore, some formats use blocking techniques which exploit knowledge about clustering of matrix non-zero entries. These blocking formats like SPARSITY, CARB, or M-CARB, may give significantly better performance of the algorithms on sparse matrices than allows the CSR format, due to eliminating memory read stalls, consuming less memory, allowing a better use of registers, and improving vector unit utilization.

But these specialized and efficient formats have also some drawbacks. They suffer from a large transformation overhead, are designed only for a limited set of matrix operations, or do not support fast adding or removing nonzero elements.

2 Goals of project

The schedule of this project consist of these steps

- Initial installation of HW and SW,
- Parallel GPU routines using sparse matrix storage formats
- Implementation of remote grid calls.

that are discussed in details later.

2.1 Initial installation of HW and SW

There are a lot of grids differ in their sizes, capabilities and purposes. We want to design the grid with the maximal ratio between prize and computational power. To achieve this goal with limited budget, we must maximize GPU usage for computation.

2.1.1 Grid architecture

We assume that system (grid) is divided into cluster of computers (nodes) with graphic cards (not necessarily of the same type) connected by Internet network. For the communication among the nodes inside one cluster we will assume a MPI (Message Passing Interface) library. Each cluster has exactly one server of service. Server will manage other (slave) parts (CPUs and GPUs) and monitor their workload.

2.1.2 Current HW configuration

Current HW configuration includes: five Geforce 470, one Tesla C2050, two Tesla C1060, one GeForce 280. All GPUs are borrowed by Prague CUDA Teaching Centre (PCTC). In our grid "new" and "old" GPUs are mixed, this requires good load-balancing strategy.

2.1.3 Current SW configuration

We also install third-party routines for shared memory or distributed CPU computing: ScaLAPACK (library of high-performance linear algebra routines for distributed-memory message-passing MIMD computers), PARDISO, SuperLU, and so on.

2.2 Parallel GPU routines using sparse matrix storage formats

Currently, several vendor supported libraries in CUDA that efficiently implement Basic Linear Algebra Subroutines (BLAS) and Fast Fourier Transformation (FFT) are available, these are CUBLAS and CUFFT. Many existing linear algebra libraries focus on efficient implementation of basic vector and matrix operations while the support for the sparse matrix computations is not included. We will overcome this limitation by implementation of new variant of these routines. The project's goal is to overcome this limitation and design sparse matrix operations with data formats suitable for GPU architecture and for GPU cluster. This work will extend the ideas of ScaLAPACK. We will concentrate on these operations (for dense or sparse formats):

- matrix-matrix multiplication,
- Cholesky and LU factorization,
- eigensolvers.

2.3 Implementation of remote grid calls

2.3.1 Idea

Usually, only special variants of codes are executed on the grid. This approach has serious drawback that code must be modified for grid computing. We want to overcome this limitation and extend the utilization of the grid. To do this, we rewrite interface for some routines for numerical linear algebra (shortly NLA, like BLAS or LAPACK). So, most of codes without any modifications can used the computational power of the grid.

The difference will occur when client (computer outside the grid) want to proceed any NLA routine. A heuristic on client side firstly estimate if it will be faster to compute this routine locally or send it to the grid for execution.

If the condition is true, the client do a remote call of this NLA routine by sending a demand to any server of grid. The server consider this demand and choose one of following actions:

- Compute this demand by itself (one node of cluster is used)
- Compute this demand by its cluster (all nodes of cluster are used)
- Re-send this demand to other server (nodes of different cluster are used)
- Refuse this demand (grid is full). Client is forced to do the local computation.

After the remote grid call is executed, results are send back to the client.

2.3.2 Discussion

- Advantages of remote grid calls
  1. Time: program can be faster executed because most time-consuming parts are moved to more powerful computer than user's one.
  2. Implementation: some parts of program can be executed in parallel without any additional modifications

3. Administration: all mathematical libraries can be installed on the server of service.
4. Economical: the proposed grid is not very expensive, but it provides very good performance.

- Drawbacks

1. The server of service must have a good connectivity and fast and reliable connections to other servers of grid are also required.
2. Network latency and bandwidth must be taken in account.
3. The service is suitable only from some algorithms (most time-consuming parts are NLA calls, without GUI, input parameters can be given command line.)
4. Algorithms must have computational demands greater than the communication overhead (matrix-matrix multiplication is a good example).

3 Conclusions

We propose the design of a the new distributed system for numerical linear algebra. The used grid and new approach (remote grid calls) allow the parallel execution of many of codes without any modifications.

4 Future works

- Non-blocking remote grid calls.
- Nodes can be dynamically connected or disconneted from the grid. This is great advantage because also classroom computer can join the grid.
- Support for another libraries like GMP, PETSc and so on.
- Compression of the comunication.
- Heuristic for a prediction of a workload and an execution time for some operations

Acknowledgement: This research has been supported by MŠMT under research program MSM6840770014, by CESNET Development Fund (project 390/2010), and by Prague CUDA Teaching Centre(PCTC).

References

[1] D.B. Kirk, W. mei W. Hwu: *Programming massively parallel processors: a hands-on approach.* Morgan Kaufmann, 2010.

[2] J. Sanders, E. Kandrot: *CUDA by example: an introduction to general-purpose GPU programming.* Addison-Wesley Professional, 2010.

[3] R. Tsuchiyama, T. Nakamura, T. Iizuka, A. Asahara, S. Miki: *The OpenCL programming book.* Fixstars Corporation, 2010.

Parallel implementation of three-level BDDC method

J. Šístek, P. Burda, M. Čertíková, J. Mandel, B. Sousedík

[1] Institute of Mathematics AS CR, Prague
[2,3] Czech Technical University in Prague
[4,5] University of Colorado, Denver

1 Introduction

The Balancing Domain Decomposition based on Constraints (BDDC) method by Dohrmann [2] is one of the most advanced methods of iterative substructuring for the solution of large systems of linear algebraic equations arising from discretization of boundary value problems.

In the case of many substructures, solving the coarse problem exactly becomes a bottleneck. This has been observed also for the FETI-DP method (e.g. in [3]), which is closely related to BDDC. For this reason, recent research in the area is directed towards inexact solutions of the coarse problem. Klawonn and Rheinbach in [3] use algebraic multigrid to obtain an approximate coarse correction within FETI-DP method and achieve excellent scalability with the resulting implementation.

We follow a different approach. As was mentioned already in [2], for BDDC method, it is straightforward to substitute the exact solution of the coarse problem by another step of BDDC method with subdomains playing the role of elements. In this way, the algorithm of three-level BDDC method is obtained (studied e.g. in [6]). One may try even recursive applications of the method called *Multilevel BDDC* [4]. Unlike for other methods, such extension is natural for BDDC, since the coarse problem has the same structure as the original problem.

It is our long-term goal to develop an efficient parallel implementation of the Multilevel BDDC method and make it publicly available. In this paper, we present results of the recently developed parallel implementation of the three-level BDDC method, and its comparison with standard (two-level) BDDC method. Even these preliminary results suggest which drawbacks of the two-level implementation might be overcome by the extension to more levels.

2 BDDC algorithm with two and three levels

The BDDC method provides a preconditioner to the *reduced interface problem* $\widehat{\mathbf{S}}\,\widehat{\mathbf{u}} = \widehat{\mathbf{g}}$, where $\widehat{\mathbf{S}}$ is a *Schur complement* with respect to interface and $\widehat{\mathbf{g}}$ is sometimes called *condensed right hand side*. This problem is solved by the preconditioned conjugate gradients (PCG) method by means of *iterative substructuring* (details may be found e.g. in [5]).

Let us begin with description of the standard (two-level) BDDC method. Let $\mathbf{K}_i$ be the local subdomain matrix, obtained by the sub-assembling of element matrices of elements contained in i-th subdomain. We introduce the *coarse space basis functions* on each subdomain represented by columns of matrix $\boldsymbol{\Psi}_i$, which is the solution to the saddle point problem with multiple right hand sides

$$\begin{bmatrix} \mathbf{K}_i & \mathbf{C}_i^T \\ \mathbf{C}_i & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_i \\ \boldsymbol{\Lambda}_i \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}. \tag{1}$$

Matrix $\mathbf{C}_i$ represents constraints on functions $\boldsymbol{\Psi}_i$, one row per each. These constraints enforce continuity of approximate solution at *corners* and of averages over some subsets of interface (*edges* or *faces*) between adjacent subdomains. The *local coarse matrix* $\mathbf{K}_{Ci} = \boldsymbol{\Psi}_i^T \mathbf{K}_i \boldsymbol{\Psi}_i = -\boldsymbol{\Lambda}_i$ is constructed for each subdomain. Let $\mathbf{R}_{Ci}$ realize the restriction of global coarse degrees of freedom to local coarse degrees of freedom. Using this matrix, we can construct the global *coarse matrix* by the assembly procedure, formally written as $\mathbf{K}_C = \sum_{i=1}^{N} \mathbf{R}_{Ci}^T \mathbf{K}_{Ci} \mathbf{R}_{Ci}$.

Suppose $\widehat{\mathbf{r}} = \widehat{\mathbf{g}} - \widehat{\mathbf{S}}\, \widehat{\mathbf{u}}$ is a residual within the PCG method. The residual assigned to i-th subdomain is computed as $\mathbf{r}_i = \mathbf{E}_i^T \widehat{\mathbf{r}}$, where matrices $\mathbf{E}_i^T$ distribute $\widehat{\mathbf{r}}$ to subdomains (see [5] for details). The subdomain correction is now defined as the solution to system

$$\begin{bmatrix} \mathbf{K}_i & \mathbf{C}_i^T \\ \mathbf{C}_i & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z}_i \\ \lambda_i \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{0} \end{bmatrix}. \tag{2}$$

The residual for the coarse problem is constructed using the coarse basis functions subdomain by subdomain and assembling the contribution as $\mathbf{r}_C = \sum_{i=1}^{N} \mathbf{R}_{Ci}^T \boldsymbol{\Psi}_i^T \mathbf{E}_i^T \widehat{\mathbf{r}}$. The coarse correction is defined as the solution to problem $\mathbf{K}_C\, \mathbf{z}_C = \mathbf{r}_C$. Both corrections are then added together and averaged on the interface by matrices $\mathbf{E}_i$ to produce the preconditioned residual $\widehat{\mathbf{z}} = \sum_{i=1}^{N} \mathbf{E}_i \left(\boldsymbol{\Psi}_i \mathbf{R}_{Ci} \mathbf{z}_C + \mathbf{z}_i \right)$.

In the *Three-level BDDC* method, the matrix $\mathbf{K}_C$ is not constructed on the second level. Instead, subdomains of the basic (first) level are grouped into subdomains of the second level in the same way as elements of the original mesh are grouped into subdomains of the first level. The whole procedure described in this section is now repeated for the second level and thus the final coarse problem represents the third level. The only difference between the first and the second level is the *interior pre-correction* and *post-correction* applied on the second level. These corrections were used also for the two-level method in the original paper [2], in which BDDC was formulated for global (i.e. not reduced to interface) problem. Details of the three-level BDDC algorithm (as a special case of the Multilevel BDDC algorithm) can be found in [4].

3 Parallel implementation

Our implementation of the two- and three- level BDDC methods is written in Fortran 95 programming language using MPI library. It relies heavily on the sparse direct solver MUMPS: a sequential instance of MUMPS is used for solving each subdomain problem, another sequential instance is used to solve interior problems (called *discrete Dirichlet problems* [5]) at each subdomain, and finally a parallel instance of MUMPS is used to solve the resulting coarse problem at the highest level. The program passes the matrix of the coarse problem to MUMPS in the distributed assembled form, i.e. the local coarse matrices $\mathbf{K}_{Ci}$ reside at the processor where they are created.

Since division into subdomains has a significant impact on the efficiency of the method, it is useful to create divisions independently of number of available processors. Thus, the solver supports assignment of several subdomains to each processor.

The implementation uses ParMETIS package to generate division of elements into subdomains on the first level and the METIS package to generate the division on the second level.

In Figure 1, simplified schemes of the hierarchy in the implementation of the preconditioner are given for two and three levels, respectively.

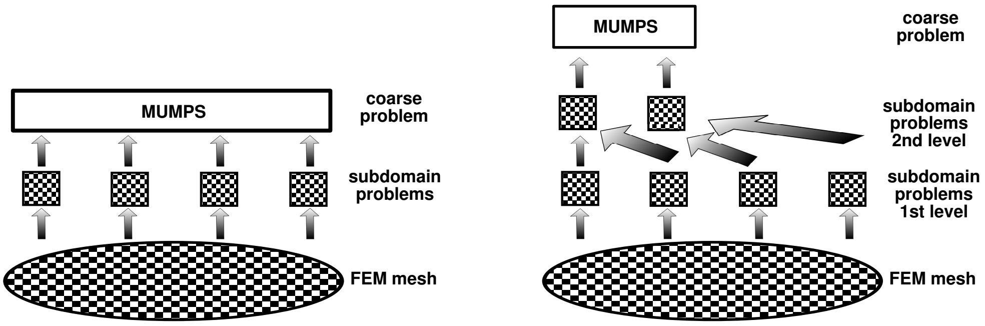


Figure 1: Schemes of parallel implementation of standard (two-level) BDDC (left) and three-level BDDC (right).

4 Numerical results

The implementation has been tested on a large 3D problem of linear elasticity. This problem represents mechanics of a geocomposite and was analysed in [1]. The problem is discretized using unstructured grid of about 12 million linear tetrahedral elements, resulting in approximately 6 million unknowns.

The mesh was divided into 1,024 subdomains on the first level and 128 subdomains on the second level in the three-level version. Resulting coarse problems (using corners and averages on all edges and faces) contain 86,094 unknowns on the first level and 11,265 on the second level.

Table 1 contains strong scaling test with implementation using two and three levels. The iterations of PCG were stopped when the relative residual $\|\widehat{\mathbf{r}}\|/\|\widehat{\mathbf{g}}\|$ decreased bellow 10^{-6}. All these computations were performed on the IBM SP6 computer at CINECA Supercomputing centre, Bologna.

#proc	64	128	256	512	1,024
2 levels (1,024+1), 46 PCG iter, cond. est. 50.3					
set-up (sec)	61.0	37.7	25.7	23.2	39.5
iter (sec)	22.3	19.9	27.8	44.9	97.5
total (with I/O) (sec)	723.7	473.1	317.1	220.2	240.5
3 levels (1,024+128+1), 56 PCG iter, cond. est. 78.6					
set-up (sec)	49.5	29.0	18.4	12.6	11.0
iter (sec)	28.5	22.6	16.7	14.7	13.2
total (with I/O) (sec)	779.2	442.3	278.2	182.1	132.7

Table 1: Strong scaling using two and three levels.

It has been confirmed by our experiment, that the coarse problem solution causes problems with scalability in both two-level and three-level cases. While most parts of the implementation scale very well, the coarse problem presents a bottleneck for scalability not only in the set-up phase, but mainly in the part of iterations. In other words, it becomes costly (with respect to each iteration) to solve the coarse problem, which is not extensive in size, on too many processors and broadcast its solution to them. Slightly surprisingly, it appears more feasible for this implementation to leave some processors idle and solve the problem on a smaller subset of processors, precisely as it happens in the three-level implementation. One should note, that idle processors appear in the three-level case on the second and the third level when more than 128 processors are used.

5 Conclusion

We have presented a parallel implementation of the three-level BDDC preconditioner and compared it to the two-level version. Since the implementation uses an efficient parallel sparse direct solver (MUMPS), the coarse problem does not present a severe bottleneck for factorization in the set-up phase for the presented problem. However, its solution slows down the computation in the phase of iterations.

From our first experiments, it appears that the three-level preconditioner tends to scale better in both parts - set-up and PCG. The worse approximation properties of the three-level method, which are theoretically analysed in [4] and demonstrated here by higher number of PCG iterations (Table 1), seem to be compensated by faster solution of the coarse problem in each iteration.

We expect, that these advantages of the three-level BDDC method would pronounce further for larger problems, where the bottleneck presented by the coarse problem would be encountered also during factorization. Such problems as well as the extension to multiple levels will be the subject of our further research.

Acknowledgement: We are grateful to Prof. Blaheta and Dr. Starý from the Institute of Geonics AS CR for providing the large-scale linear elasticity problem. We acknowledge the support of this research by the Grant Agency of the Academy of Sciences under grant IAA100760702, by the Czech Science Foundation under grant GA CR 106/08/0403, and by Institutional Research Plan AV0Z 10190503 of the AS CR. This research was also supported by NSF grant DMS-0713876. Some parts of this work have been performed under the HPC-Europa2 project with the support of the European Commission.

References

[1] R. Blaheta, O. Jakl, J. Starý, K. Krečmer: *The Schwarz domain decomposition method for analysis of geocomposites.* In: Proceedings of the Twelfth International Conference on Civil, Structural and Environmental Engineering Computing, Stirlingshire, Scotland, 2009, B. Topping, L. C. Neves, and R. Barros, (Eds.), Civil-Comp Press.

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

[3] A. Klawonn, O. Rheinbach: *Highly scalable parallel domain decomposition methods with an application to biomechanics.* ZAMM Z. Angew. Math. Mech. 90, 1, 2010, 5–32.

[4] J. Mandel, B. Sousedík, C.R. Dohrmann: *Multispace and multilevel BDDC.* Computing 83, 2-3, 2008, 55–85.

[5] J. Šístek, J. Novotný, J. Mandel, M. Čertíková, P. Burda: *BDDC by a frontal solver and stress computation in a hip joint replacement.* Math. Comput. Simulation 80, 6, 2010, 1310–1323.

[6] X. Tu: *Three-level BDDC in three dimensions.* SIAM J. Sci. Comput. 29, 4, 2007, 1759–1780.

The problem of moments and its connections

M. Tůma

Faculty of Electrical Engineering and Communication, Brno University of Technology

1 Introduction

This contribution is about the problem of moments. During the last 150 years many books and papers have been published about this problem. Many mathematicians studied it from many different points of view. It is very interesting how many connections between the different parts of mathematics has been found in these works. One can see the classical references [9] and [2]. An interesting historical review about the birth of the problem of moments can be found in [6]. As the time went on, the problem of moments was used in order to solve various questions in mathematical statistics, theory of probability and mathematical analysis.

2 Formulation of the problem

Given the sequence of real numbers $\{\xi_k\}_{k=0}^{\infty}$. The problem is to find the following positive measure μ such that

$$\xi_k = \int_I x^k d\mu(x), \qquad k = 0, 1, ... \tag{1}$$

In the case when $I = [0, \infty)$ we talk about the Stieltjes moment problem. The case when $I = \mathbb{R}$ is called the Hamburger moment problem. The real numbers $\{\xi_k\}_{k=0}^{\infty}$ are then called the moments. The terminology was taken from mechanics. If the measure μ represents the distribution of the mass over the real semi-axis, then the integrals

$$\int_0^{\infty} x d\mu(x), \int_0^{\infty} x^2 d\mu(x)$$

represent the first (statical) moment and the second moment (moment of inertia).

One can ask the following questions:

- Does the measure μ exist for the sequence of the moments $\{\xi_k\}_{k=0}^{\infty}$?
- If the measure μ exists, is it determined uniquely?

Now lets take a look on the similar problem. Given the same sequence of the moments $\{\xi_k\}_{k=0}^{\infty}$. The problem is to find the following positive measure μ_n such that the first $2n$ moments are matched, i.e.,

$$\xi_k = \int_I x^k d\mu_n(x), \qquad k = 0, 1, ...2n - 1. \tag{2}$$

The formulation above is often called the truncated problem of moments, one can see e.g. [1]. Searching for this measure μ_n is closely connected with many different methods in the mathematics. The aim of this contribution is to give an overview of many connections one could find. It will be shown how can the knowledge of these connections lead to the new results.

3 Connections

It is known for a long time that the finding of the μ_n instead of μ is closely connected with the Gauss-Christoffel quadrature, see e.g. [11], [7]. Under certain settings the problem of moments can be seen as the theoretical background for the Lanczos method and the CG method. The connection with the CG and with the the Gauss-Christoffel quadrature is known since the introduction of the CG and it was well described by M. R. Hestenes and E. Stiefel in their joint paper [4]. In [8] the results about the sensitivity of the Gauss-Christoffel quadrature with respect to the small perturbations of the measure are given. Obtaining of these results would not be possible without the deep knowledge of the connection with the problem of moments.

Russian mathematician Yu V. Vorobyev presented the general problem of moments in the Hilbert space in [12]. Let $z_0, z_1, ..., z_n$ be $n+1$ prescribed linearly independent elements of the Hilbert space H. Consider the n-dimensional subspace H_n

$$H_n = \text{span}\{z_0, z_1, ..., z_{n-1}\}.$$

The linear operator A_n defined on the subspace H_n is constructed in the following way

$$\begin{aligned} A_n z_0 &= z_1, \\ A_n^2 z_0 &= z_2, \\ &\dots \\ A_n^{n-1} z_0 &= z_{n-1}, \\ A_n^n z_0 &= E_n z_n, \end{aligned} \tag{3}$$

where $E_n z_n$ is the projection of z_n on H_n.

Vorobyev applied his work about the moments on solving differential, integral and finite difference equations and also on resolving spectrum of bounded operators in the Hilbert space. In the case of the self-adjoint operators Vorobyev pointed out the connection of his work with the CG method. The Vorobyev problem of moments was used by Z. Strakoš and P. Tichý in their approach of approximating the scattering amplitude, see [10].

The problem of moments is closely connected with the Sturm-Liouville problem. In [3] the connections between the singular Sturm-Liouville problem, Jacobi matrices and Hamburger moment problem are described in an elegant way. The nature of the solutions of the singular Sturm-Liouville problem is connected with the determinacy of the associated Hamburger moment problem.

There is also the relation between the model reduction in the linear dynamical systems

$$\begin{aligned} z^{'}(t) &= Az(t) + bu(t), \\ y(t) &= b^* z(t) \end{aligned} \tag{4}$$

and the problem of moments. In [7, pp. 101-108] an elegant description of the connection between the model reduction of the above system and the problem of moments is given. Consider the expansion of the transfer function $T(\lambda)$ which is connected to the dynamical system (4)

$$\begin{aligned} -T(\lambda) &= \lambda^{-1} b^* (I - \lambda^{-1} A)^{-1} b = \\ &= \lambda^{-1}(b^* b) + \lambda^{-2}(b^* A b) + ... + \lambda^{-2n}(b^* A^{2n-1} b) + ... \end{aligned} \tag{5}$$

A reduced model of order n which matches the first $2n$ terms in the above expansion is known as the minimal partial realization. The concept of the minimal partial realization was introduced in the control theory literature by R. E. Kalman in 1979, see [5]. The idea to find the reduced model is again nothing else than the problem of moments such that the first $2n$ moments are matched, see (2).

References

[1] V.M. Adamyan, I.M. Tkachenko, M. Urrea: *Solution of the stieltjes truncated moment problem.* Journal of Applied Analysis 9, 2003, 57–74.

[2] N.I. Akhiezer: *The classical moment problem and some related questions in analysis.* Translated by N. Kemmer. Hafner Publishing Co., New York, 1965.

[3] A.G. García, M.A. Hernández-Medina: *Discrete Sturm-Liouville problems, Jacobi matrices and Lagrange interpolation series.* J. Math. Anal. Appl. 280 (2), 2003, 221–231.

[4] M.R. Hestenes, E. Stiefel: *Methods of conjugate gradients for solving linear systems.* J. Research Nat. Bur. Standards 49, (1953), 1952, 409–436.

[5] R.E. Kalman: *On partial realizations, transfer functions, and canonical forms.* Acta Polytech. Scand. Math. Comput. Sci. Ser. 31, 1979, 9–32.

[6] T.H. Kjeldsen: *The early history of the moment problem.* Historia Mathematica 20, 1993, 19–44.

[7] J. Liesen, Z. Strakoš: *Principles and analysis of Krylov subspace methods.* In preparation, 2010.

[8] D.P. O'Leary, Z. Strakoš, P. Tichý: *On sensitivity of Gauss-Christoffel quadrature.* Springer-Verlag, 2007.

[9] J.A. Shohat, J.D. Tamarkin: *The problem of moments.* American Mathematical Society Mathematical surveys, vol. II. American Mathematical Society, New York, 1943.

[10] Z. Strakoš, P. Tichý: *On efficient numerical approximation of the bilinear form $c^*a^{-1}b$.* Submitted for publication in SIAM Journal on Scientific Computing, 2010.

[11] Z. Strakoš: *Model reduction using the Vorobyev moment problem.* Numer. Algorithms 51 (3), 2009, 363–379.

[12] Yu V. Vorobyev: *Method of moments in applied mathematics.* Gordon and Breach Science Publishers, New York, 1965.

Agresivní zhrubování v metodě zhlazených agregací

P. Vaněk

Západočeská univerzita v Plzni

Důležitou partií výsledků o metodě zhlazených agregací tvoří výsledky týkající se agresivního zhrubování. Zde, hrubý prostor je podstatně menší než jemný prostor a tato skutečnost je kompenzována mocným hladičem. Metoda zhlazených agregací je mimořádně vhodná pro agresivní zhrubování v kombinaci s polynomiálním hladičem odvozeným od prolongátorového hladiče. Zde hlazení prolongátoru pracuje jako preconditioner. Tato skutečnost bude demonstrována na jednoduché dvojúrovňové metodě. Bude prezentován klíčový dvojúrovňový výsledek ve dvou variantách a obecný víceúrovňový výsledek.

Discrete Green's function – a closer look

T. Vejchodský

Institute of Mathematics AS CR, Prague

1 Introduction

For linear elliptic problems the Green's function provides a solution operator. Similarly, in the context of the finite element method the discrete Green's function (DGF) provides the solution operator for the discrete problem. Therefore, certain properties of the finite element solution can be deduced form the properties of the DGF.

Typical example of such a property is the discrete maximum principle. It is satisfied if and only if the corresponding DGF is nonnegative. In the lowest-order finite element methods the DGF can be equivalently replaced by the inverse of the stiffness matrix. However, in the higher-order methods this replacement cannot be done and the DGF plays the crucial role there.

We choose as a model problem the Possion equation with homogeneous Dirichlet boundary conditions, discretize it by the finite element method of certain order and study the nonnegativity of the corresponding DGF. Numerical experiments published recently in [2] indicate that for higher-order approximations the DGF is nonnegative everywhere in the computational domain in exceptional cases only. In this short contribution we propose to study the nonnegativity in an interior region of the computational domain only. We present additional numerical experiments trying to identify triangulations yielding this interior nonnegativity.

2 Model problem, DGF, and discrete maximum principle

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain. We consider the Poisson equation in Ω and the homogeneous Dirichlet boundary conditions on $\partial\Omega$:

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

This problem is discretized by the finite element method of order p. Thus, we consider a triangulation $\mathcal{T}_h$ of Ω and introduce a space V_h of piecewise polynomial and globally continuous functions:

$$V_h = \{v_h \in C_0(\overline{\Omega}) : v_h|_K \in \mathbb{P}^p(K) \quad \forall K \in \mathcal{T}_h\},$$

where $C_0(\overline{\Omega})$ stands for the space of continuous functions on $\overline{\Omega}$ whose values on $\partial\Omega$ vanish and $\mathbb{P}^p(K)$ denotes the space of polynomials of degree at most p in the triangle $K \in \mathcal{T}_h$.

The finite element formulation of problem (1) reads as follows: find $u_h \in V_h$ such that

$$a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h. \tag{2}$$

As usual, $a(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h \,\mathrm{d}x$ stands the energetic bilinear form and $(f, v_h) = \int_\Omega f v_h \,\mathrm{d}x$ denotes the $L^2(\Omega)$ inner product.

The DGF is defined as the approximate solution of the adjoint problem: for any $y \in \Omega$ we define $G_{h,y} \in V_h$ as the unique solution of the Galerkin problem

$$a(v_h, G_{h,y}) = v_h(y) \quad \forall v_h \in V_h. \tag{3}$$

Instead of $G_{h,y}(x)$ we will use the standard notation $G_h(x,y) = G_{h,y}(x)$. It can be easily shown (see e.g. [1] or Lemma 1 below) that G_h is symmetric in the sense that $G_h(x,y) = G_h(y,x)$ for all $(x,y) \in \Omega^2$. In addition, from the definition of the DGF (3) and from the definition of the finite element solution (2), we immediately infer the well known representation formula

$$u_h(y) = \int_\Omega G_h(x,y) f(x)\,\mathrm{d}x. \tag{4}$$

Furthermore, the DGF $G_h(x,y)$ can be easily expressed in terms of any basis in V_h (see e.g. [1]):

Lemma 1. *Let φ_1, φ_2, ..., φ_n be a basis of V_h. Let $A \in \mathbb{R}^{n\times n}$ be the corresponding stiffness matrix, i.e. $A_{ij} = a(\varphi_j, \varphi_i)$, $i,j = 1,2,\dots,n$. Then*

$$G_h(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} \varphi_i(y)(A^{-1})_{ij}\varphi_j(x), \quad \forall (x,y) \in \Omega^2. \tag{5}$$

In the experiments below, we use expression (5) to study the nonnegativity of the DGF G_h in Ω^2. The interest in the nonnegativity of G_h is motivated by the direct connection with the discrete maximum principle. Given a fixed triangulation and the corresponding space V_h, we say that problem (2) satisfies the *discrete maximum principle* (DMP) if

$$f \ge 0 \text{ a.e. in } \Omega \quad \Rightarrow \quad u_h \ge 0 \text{ in } \Omega. \tag{6}$$

The representation formula (4) immediately proves the fact that problem (2) satisfies the DMP if and only if the corresponding DGF G_h is nonnegative in Ω^2.

Numerical experiments presented in [2] indicate that for higher-order finite elements the DGF G_h is nonnegative in an exceptional case only. Namely, for $p = 2$ and for all elements in the triangulation being close to the equilateral triangle. These experiments also indicate that the negative values of the DGF are usually close to the boundary. Therefore, we define certain layer $\mathcal{B} \subset \Omega$ of points close to the boundary $\partial\Omega$. We denote the complement of $\mathcal{B}$ in Ω as $\mathcal{I} = \Omega\setminus\mathcal{B}$ and we call $\mathcal{B}$ and $\mathcal{I}$ the boundary and the interior region, respectively. Since the requirement (6) is too strong to be satisfied by the higher-order elements we can naturally ask if one of the following weaker requirements is satisfied:

$$f \ge 0 \text{ a.e. in } \Omega \quad \Rightarrow \quad u_h \ge 0 \text{ in } \mathcal{I}, \tag{7}$$

$$f \ge 0 \text{ a.e. in } \mathcal{I} \text{ and } f = 0 \text{ a.e. in } \mathcal{B} \quad \Rightarrow \quad u_h \ge 0 \text{ in } \mathcal{I}. \tag{8}$$

From the representation formula (4) we easily see that requirement (8) is satisfied if and only if $G_h(x,y) \ge 0$ for all $(x,y) \in \mathcal{I}^2$. Similarly, requirement (7) is satisfied if and only if $G_h(x,y) \ge 0$ for all $(x,y) \in \Omega \times \mathcal{I}$. Due to the symmetry, the nonnegativity of G_h in $\Omega \times \mathcal{I}$ is equivalent to the nonnegativity in $\Omega^2 \setminus \mathcal{B}^2$.

3 Numerical experiments

In the presented experiments we try to justify the meaningfulness of properties (7) and (8) for higher-order finite elements. We consider Poisson problem (1) discretized on uniform triangulations of Ω and we test the nonnegativity of the DGF in Ω^2, in $\Omega^2 \setminus \mathcal{B}^2$, and in $\mathcal{I}^2$. We study how this nonnegativity depends on the angles in the triangulations. Since the triangulations are uniform, there are just two independent angles α and β (the third angle is $\gamma = \pi - \alpha - \beta$). We systematically test many pairs of angles α and β and display the results in a panel, where a point with coordinates (α, β) is colored according to the nonnegativity of the DGF in the tested regions. See Figure 2.

In Experiment A, the domain Ω is a triangle. The corresponding finite element mesh consists of 64 congruent triangles – see Figure 1 (left). The elements are enumerated in a spiral ways. Thus, the elements adjacent to the boundary have indices $1, 2, \ldots, 39$ and they form the boundary region $\mathcal{B}$. The interior elements with indices $40, 41, \ldots, 64$ form the interior region $\mathcal{I}$. Finally, we stress that the shape of the triangle Ω (as well as the shape of any triangle in the mesh) is determined by the two angles α and β.

The panels in Figure 2 show the results for polynomial degrees $p = 2, 3, 4$. Each point in these panels correspond to a pair of angles α and β. We construct the triangle Ω with these two angles, we create the uniform mesh in Ω, and we compute the corresponding DGF G_h. If $G_h(x, y) \geq 0$ for all $(x, y) \in \Omega^2$ then the color of point (α, β) is black. Otherwise, if $G_h(x, y) \geq 0$ for all $(x, y) \in \Omega^2 \setminus \mathcal{B}^2$ then the color is darker gray. Otherwise, if $G_h(x, y) \geq 0$ for all $(x, y) \in \mathcal{I}^2$ then the color is lighter gray. Otherwise, the DGF G_h has certain negative values in all tested areas and the corresponding color is almost white. Of course, checking nonnegativity of a polynomial is a difficult task. Therefore, we introduce in each element 153 sample points – see Figure 1 (right) – and test the nonnegativity in these sample points only.

We observe that the DGF G_h is nonnegative everywhere in Ω^2 for $p = 2$ and for triangles close to the equilateral one only. Nevertheless, the darker and the lighter gray regions corresponding to the properties (7) and (8), respectively, are substantial in all cases. In addition, numerical experiments for polynomial degrees up to $p = 10$ indicate that these areas corresponding to the validity of properties (7) and (8) increase with growing p. However, this increase is not monotone.

Examining the DGF G_h in more details we find out that many negative values of G_h are caused by the presence of three edges lying inside Ω and having both their end-points on $\partial\Omega$ (e.g. the edge between elements 1 and 22). Therefore, we remove the three corner elements (the one with indices 1, 8, and 15) – see Figure 1 (middle) – and perform the same tests as above. This is Experiment B. Its results are presented in the second row of panels in Figure 2. In comparison with Experiment A, we observe substantial changes of the dark gray regions corresponding to property (7). On the other hand, there is practically no influence on the light gray region corresponding to property (8).

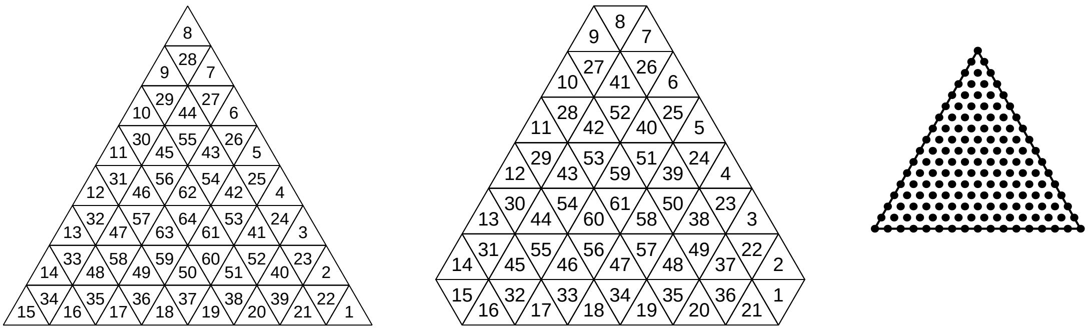


Figure 1: Uniform triangulations of the triangle (left) and of the triangle without corners (middle). Right panel shows the distribution of sample points in an element.

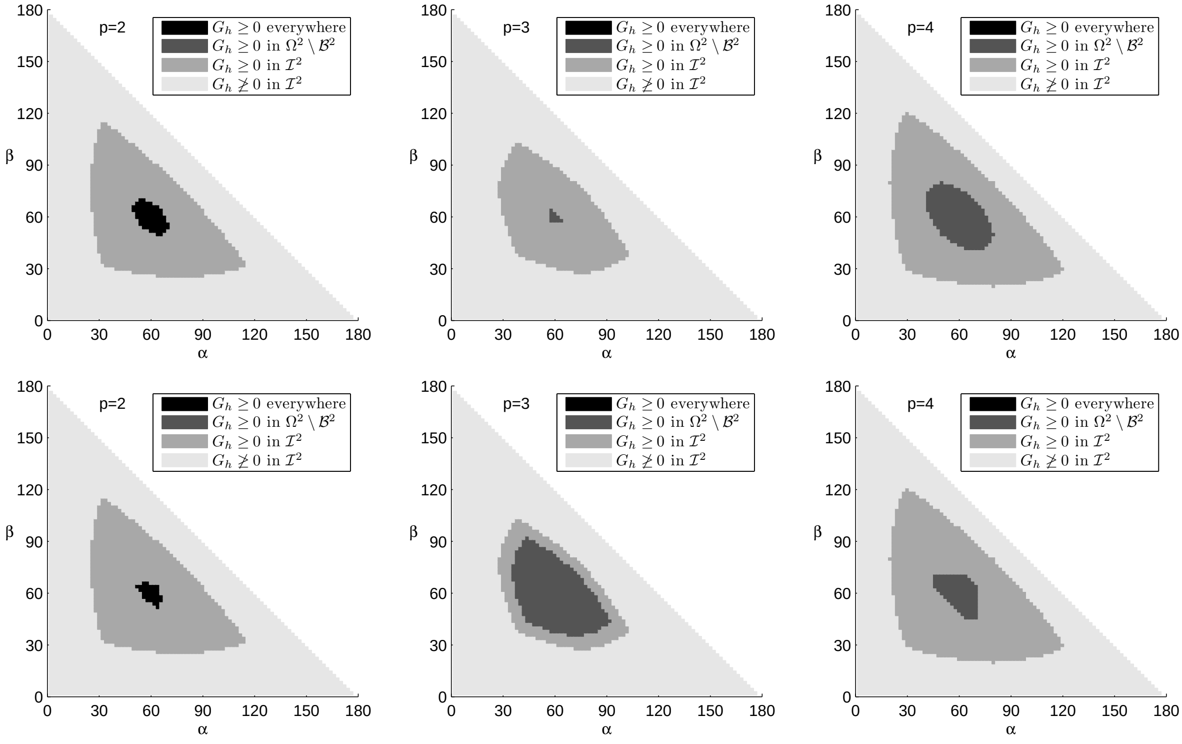


Figure 2: Results of Experiment A (first row) and of Experiment B (second row).

4 Conclusions

The performed experiments indicate that the higher-order DGF is negative mostly in the boundary region. It seems that if the triangular elements have angles close to 60° then the higher-order approximate solution u_h is automatically nonnegative everywhere in the interior elements provided the corresponding right-hand side f is nonnegative. Further, it seems that for triangles with the minimal angle above roughly 30° and the maximal angle below roughly 120° the property (8) is satisfied, i.e. if f vanishes in elements adjacent to the boundary and if it is nonnegative elsewhere then the finite element solution u_h is nonnegative everywhere in the interior elements.

Acknowledgement: The author gratefully acknowledges the support of grant IAA100760702 of the Grant Agency of the Academy of Sciences of the Czech Republic, and of the institutional research plan no. AV0Z10190503 of the Academy of Sciences of the Czech Republic.

References

[1] T. Vejchodský, P. Šolín: *Discrete Green's function and maximum principles.* In: J. Chleboun, K. Segeth, T. Vejchodský (Eds.): Programs and Algorithms of Numerical Mathematics 13, Institute of Mathematics, Academy of Sciences, Czech Republic, Prague, 2006, 247–252.

[2] T. Vejchodský: *Angle conditions for discrete maximum principles in higher-order FEM.* In: G. Kreiss, P. Lötstedt, A. Målqvist, M. Neytcheva (Eds.): Numerical Mathematics and Advanced Applications ENUMATH 2009 Springer, Berlin, 2010, 901–909.

Fast Fourier transform based method for modelling of heterogeneous materials

J. Vondřejc, J. Zeman, I. Marek

Faculty of Civil Engineering, Czech Technical University in Prague

Problem setting

We consider a composite material represented by a periodic unit cell $\mathcal{Y} = \prod_{\alpha=1}^{d}(-Y_\alpha, Y_\alpha) \subset \mathbb{R}^d$. In the context of linear electrostatics, the associated unit cell problem reads as

$$\boldsymbol{\nabla} \times \boldsymbol{e}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{\nabla} \cdot \boldsymbol{e}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{j}(\boldsymbol{x}) = \boldsymbol{L}(\boldsymbol{x}) \cdot \boldsymbol{e}(\boldsymbol{x}),\ \boldsymbol{x} \in \mathcal{Y} \tag{1}$$

where $\boldsymbol{e}$ is a $\mathcal{Y}$-periodic vectorial electric field, $\boldsymbol{j}$ denotes the corresponding vector of electric current and $\boldsymbol{L}$ is a second-order positive-definite tensor of electric conductivity. In addition, the field $\boldsymbol{e}$ is subject to a constraint $\boldsymbol{e}^0 = \frac{1}{|\mathcal{Y}|}\int_{\mathcal{Y}} \boldsymbol{e}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, where $\boldsymbol{e}^0$ denotes a prescribed macroscopic electric field and $|\mathcal{Y}|$ represents the d-dimensional measure of $\mathcal{Y}$.

The original problem is equivalent to the periodic Lippmann-Schwinger integral equation, formally written as

$$\boldsymbol{e}(\boldsymbol{x}) + \int_{\mathcal{Y}} \boldsymbol{\Gamma}^0(\boldsymbol{x}-\boldsymbol{y}) \cdot \left(\boldsymbol{L}(\boldsymbol{y}) - \boldsymbol{L}^0\right) \cdot \boldsymbol{e}(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = \boldsymbol{e}^0, \qquad \boldsymbol{x} \in \mathcal{Y}, \tag{2}$$

where the $\boldsymbol{\Gamma}^0$ operator is derived from the Green's function of the initial problem with $\boldsymbol{L}(\boldsymbol{x}) = \boldsymbol{L}^0$ and $\boldsymbol{e}^0 = \boldsymbol{0}$ and can be expressed in Fourier space as

$$\hat{\boldsymbol{\Gamma}}^0(\boldsymbol{k}) = \begin{cases} \boldsymbol{0}, & \boldsymbol{k} = 0 \\ \frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{\boldsymbol{\xi}\cdot\boldsymbol{L}^0\cdot\boldsymbol{\xi}}, & \boldsymbol{k} = (k_\alpha)_{\alpha=1}^d, \boldsymbol{\xi} = (\xi_\alpha)_{\alpha=1}^d, \xi_\alpha = \frac{k_\alpha}{Y_\alpha}, \boldsymbol{k} \in \overline{\mathbb{Z}}^{\boldsymbol{N}} \end{cases} \tag{3}$$

Discretization of integral equation

Numerical solution of the Lippmann-Schwinger equation is based on a discretization of a unit cell $\mathcal{Y}$ into a regular periodic grid with $N_1 \times \cdots \times N_d$ nodal points and grid spacings $\boldsymbol{h} = (2Y_1/N_1, \ldots, 2Y_d/N_d)$. The searched field $\boldsymbol{e}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{Y}$, in (2) is approximated by a trigonometric polynomial $\boldsymbol{e}^{\boldsymbol{N}}$ in the form (cf. [2])

$$\boldsymbol{e}(\boldsymbol{x}) \approx \boldsymbol{e}^{\boldsymbol{N}}(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\overline{\mathbb{Z}}^{\boldsymbol{N}}} \widehat{\boldsymbol{e}}(\boldsymbol{k})\varphi_{\boldsymbol{k}}(\boldsymbol{x}), \quad \overline{\mathbb{Z}}^{\boldsymbol{N}} = \left\{\boldsymbol{k} \in \mathbb{Z}^d : -\frac{N_\alpha}{2} < k_\alpha \leq \frac{N_\alpha}{2}, \alpha = 1, \ldots, d\right\}$$

where $\boldsymbol{N} = (N_1, \ldots, N_d)$, $\widehat{\boldsymbol{e}}$ designates the Fourier coefficients and $\varphi_{\boldsymbol{k}} = \exp\left(\mathrm{i}\pi \sum_{i=1}^d x_i\xi_i\right)$ with $\xi_i = \frac{k_i}{Y_i}$ are basis functions.

The trigonometric collocation method (e.g. [2]) is based on the projection of the Lippmann-Schwinger equation (2) to the space of the trigonometric polynomials $\left\{\sum_{\boldsymbol{k}\in\overline{\mathbb{Z}}^{\boldsymbol{N}}} c_{\boldsymbol{k}}\varphi_{\boldsymbol{k}}, c_{\boldsymbol{k}} \in \mathbb{C}\right\}$ leading to linear system of equations

$$\mathsf{A}\mathbf{e} = \mathbf{e}^0, \tag{4}$$

where $\mathsf{e} = \left(\mathsf{e}_\alpha^{\boldsymbol{k}}\right)_{\alpha=1,\ldots,d}^{\boldsymbol{k}\in\mathbb{Z}^{\boldsymbol{N}}} \in \mathbb{R}^{d\times\boldsymbol{N}}$ and $\mathsf{e}^0 = \left((\mathsf{e}^0)_\alpha^{\boldsymbol{k}}\right)_{\alpha=1,\ldots,d}^{\boldsymbol{k}\in\mathbb{Z}^{\boldsymbol{N}}} \in \mathbb{R}^{d\times\boldsymbol{N}}$ store the corresponding solution and of the macroscopic field, respectively. The action of the linear operator (block matrix) $\mathsf{A} = [\mathsf{A}_{\alpha\beta}^{\boldsymbol{km}}]_{\alpha,\beta=1,\ldots,d}^{\boldsymbol{k},\boldsymbol{l}\in\mathbb{Z}^{\boldsymbol{N}}}$ on vector e produces vector $\mathsf{Ae} \in \mathbb{R}^{d\times\boldsymbol{N}}$ with components

$$(\mathsf{Ae})_\alpha^{\boldsymbol{k}} = \sum_{\beta=1}^{d} \sum_{\boldsymbol{m}\in\mathbb{Z}^{\boldsymbol{N}}} \mathsf{A}_{\alpha\beta}^{\boldsymbol{km}} \mathsf{e}_\beta^{\boldsymbol{m}} \tag{5}$$

Furthermore, the non-symmetric matrix A can be expressed as

$$\mathsf{A} = \mathsf{I} + \mathsf{B} = \mathsf{I} + \mathsf{F}^{-1}\hat{\mathsf{\Gamma}}\mathsf{F}(\mathsf{L} - \mathsf{L}^0) \tag{6}$$

where I is the unit matrix of size $d \times d \times \boldsymbol{N} \times \boldsymbol{N}$, the explicit forms of the individual terms can be found in [3].

Solution using conjugate gradients

The original Fast Fourier Transform-based Homogenization (FFTH) scheme formulated by Moulinec and Suquet in [1] is based on the Neumann expansion of the matrix inverse $(\mathsf{I} + \mathsf{B})^{-1}$, so as to yield the m-th iterate in the form

$$\mathsf{e}^{(m)} = \sum_{j=0}^{m} (-\mathsf{B})^j \, \mathsf{e}^{\mathbf{0}}. \tag{7}$$

We have proposed in [3] to solve the non-symmetric linear system using Conjugate gradients and presented numerical experiments, which suggest convergence of CG algorithm.

In this contribution, we outline basic ideas of the convergence proof. Without a loss of generality, we consider the special form of reference conductivity $\boldsymbol{L}^0 = \rho\boldsymbol{I}$ with $\rho > 0$ and reformulate (4) in the form:

$$\mathsf{P}_{\boldsymbol{\mathcal{E}}}\mathsf{L}\mathsf{e}_{\boldsymbol{\mathcal{E}}} = \mathsf{e}^0 \tag{8}$$

where $\mathsf{P}_{\boldsymbol{\mathcal{E}}} = \mathsf{F}^{-1}\hat{\mathsf{\Gamma}}^0\mathsf{F}\left(\mathsf{L}^0\right)^{-1}$ is a projection matrix on a subspace $\boldsymbol{\mathcal{E}} = \{\mathsf{P}_{\boldsymbol{\mathcal{E}}}\mathsf{x} | \mathsf{x} \in \mathbb{R}^{d\times\boldsymbol{N}}\} \subset \mathbb{R}^{d\times\boldsymbol{N}}$ and the solution $\mathsf{e}_{\boldsymbol{\mathcal{E}}} \in \boldsymbol{\mathcal{E}}$. The linear system (8) can be alternatively reformulated as a minimization problem

$$\mathsf{e} = \mathsf{e}^0 + \mathrm{argmin}_{\mathsf{e}_{\boldsymbol{\mathcal{E}}}\in\boldsymbol{\mathcal{E}}}\, \phi(\mathsf{e}_{\boldsymbol{\mathcal{E}}})$$

where $\phi(\mathsf{e}_{\boldsymbol{\mathcal{E}}})$ is a linear functional defined as

$$\phi(\mathsf{e}_{\boldsymbol{\mathcal{E}}}) = \frac{1}{2}\left(\mathsf{L}\mathsf{e}_{\boldsymbol{\mathcal{E}}}, \mathsf{e}_{\boldsymbol{\mathcal{E}}}\right) + \left(\mathsf{L}\mathsf{e}^0, \mathsf{e}_{\boldsymbol{\mathcal{E}}}\right)$$

where $\left(\cdot, \cdot\right)$ denotes scalar product on $\mathbb{R}^{d\times\boldsymbol{N}}$, i.e.

$$\left(\mathsf{u}, \mathsf{v}\right) = \sum_{\alpha=1}^{d} \sum_{\boldsymbol{k}\in\mathbb{Z}^{\boldsymbol{N}}} \mathsf{v}_\alpha^{\boldsymbol{k}}\mathsf{v}_\alpha^{\boldsymbol{k}}.$$

The convergence of the conjugate gradient method then follows from projection properties of $\mathsf{P}_{\boldsymbol{\mathcal{E}}}$, which implies symmetry of linear system (4) in subspace $\boldsymbol{\mathcal{E}}$.

Acknowledgement: This research was supported by the Czech Science Foundation, through projects No. GAČR 103/09/1748, No. GAČR 103/09/P490, and by the Grant Agency of the Czech Technical University in Prague through project No. SGS10/124/OHK1/2T/11.

References

[1] H. Moulinec, and P. Suquet: *A fast numerical method for computing the linear and nonlinear mechanical properties of composites.* Comptes rendus de lAcadmie des sciences. Série II, Mécanique, physique, chimie, astronomie 318, 11, 1994, 1417–1423.

[2] G. Vainikko: *Fast solvers of the Lippmann-Schwinger equation.* Direct and Inverse Problems of Mathematical Physics, Gilbert, R.P., Kajiwara, J., and S. Xu, Y. (Eds.), International Society for Analysis, Applications and Computation, vol. 5, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, 423–440.

[3] J. Zeman, J. Vondřejc, J. Novák, and I. Marek: *Accelerating a fft-based solver for numerical homogenization of periodic media by conjugate gradients.* Journal of Computational Physics 229, 21, 2010, 1865–1871.

Winter school lectures

O. Axelsson
Operator splittings for solving nonlinear coupled multiphysics problems with an application for interface modeling

Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, A. Markopoulos
Scalable FETI based algorithms for contact problems: theory, implementation, and numerical experiments

I. Hnětynková, M. Plešinger, Z. Strakoš
Ill-posed inverse problems in image processing: introduction, structured matrices, spectral filtering, regularization, noise revealing

P. Vaněk:
Základy algebraického multigridu založeného na zhlazených agregacích

J. Zeman, A. Mielke, T. Roubíček
Analysis of a rate-independent model of non-local damage and its numerical approximation

Operator splittings for solving nonlinear coupled multiphysics problems with an application for interface modeling

O. Axelsson

Institute of Geonics AS CR, Ostrava

The solution of multiphysics problems can be very demanding on computer time. A possible remedy for evolutionary problems is to use operator splittings. Some such methods are described and analyzed. To handle stiff problems an implicit and stable time-stepping method of second order of accuracy is used. This allows bigger time-steps for the control of the operator splitting errors. For nonlinear problems, a Newton solution method is used for each separate equation, and after completion of some steps of the method the equations are updated, in this way preparing for the start of additional iterations or of a new time-step.

An application for a nonlinear interface modeling problem arising in a moving fluid is described. Hereby an inner-outer iteration method is used to solve the arising linearized algebraic equations. There is no need to update the preconditioners used.

Scalable FETI based algorithms for contact problems: theory, implementation, and numerical experiments

Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, A, Markopoulos

VŠB - Technical University of Ostrava

We report the results of our research in development of the algorithms with both numerical and parallel scalability for the solution of contact problems of elasticity. Our talk covers 2D and 3D problems discretized by the finite element or boundary element method, possibly with “floating” bodies, including the multibody frictionless problems, both static and dynamic, and the problems with a given (Tresca) friction. A common feature of all the problems considered in our talk is a strong nonlinearity due to the interface conditions. Since even the algorithms for the solution of linear problems have the linear complexity at least, it follows that a scalable algorithm for contact problems has to treat the nonlinearity in a sense for free.

After introducing the variational inequalities that describe the equilibrium of a system of elastic bodies in mutual contact under the interface conditions considered in our talk, we briefly review the TFETI (total finite element tearing and interconnecting) based domain decomposition methodology adapted to the solution of contact problems of elasticity, including optimal estimates. Recall that TFETI differs from the classical FETI or FETI2 as introduced by Farhat and Roux by imposing the prescribed displacements by the Lagrange multipliers and treating all subdomains as “floating”.

Then we present our in a sense optimal algorithms for the solution of the resulting quadratic programming and QPQC (quadratic programming - quadratic constraints) problems. A unique feature of these algorithms is their capability to solve the class of such problems with homogeneous equality constraints and separable inequality constraints in $O(1)$ matrix–vector multiplications provided the spectrum of the Hessian of the cost function is in a given positive interval [1], [2].

Finally we put together the above results to develop scalable algorithms for the solution of the above problems [3], [4],[5], [6], [7]. A special attention is paid to the construction of an initial approximation which is not far from the solution, so that the above results guarantee that the cost of the solution increases nearly proportionally with the dimension of the discretized problem and to effective implementation of generalized inverse matrices of floating subdomains. We illustrate the results by numerical experiments and by the solution of difficult real world problems, such as analysis the roller bearings in Figure 1 with 73 bodies under nonsymmetric loading. We conclude by a brief discussion of other results [8] and current research.

References

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

[2] Z. Dostál and T. Kozubek, *An optimal algorithm with superrelaxation for minimization of a quadratic function subject to separable spherical constraints with applications*, submitted.

[3] Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, A. Markopoulos, *Scalable TFETI algorithm for the solution of multibody contact problems of elasticity*. International Journal for Numerical Methods in Engineering, 82, No. 11, 1384-1405 (2010).

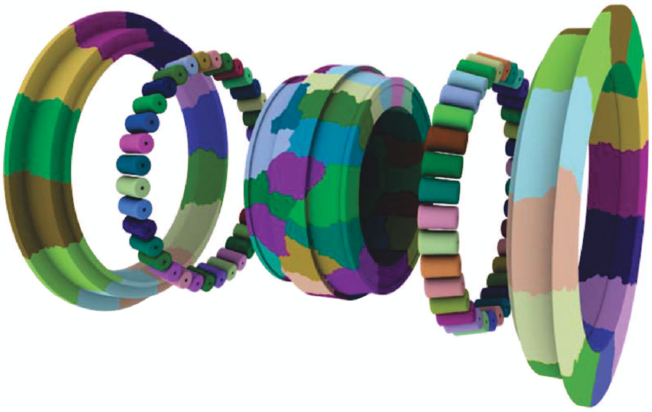


Figure 1: Roller bearings of wind generator

[4] J. Bouchala, Z. Dostál, M. Sadowská, *Scalable Total BETI based algorithm for 3D coercive contact problems of linear elastostatics*, Computing, 85(2009) 189-217. IF 0.881

[5] M. Sadowská, Z. Dostál, T. Kozubek, J. Bouchala, and A. Markopoulos, *Scalable Total BETI based solver for 3D multibody frictionless contact problems in mechanical engineering*. Submitted.

[6] Z. Dostál, T. Kozubek, A. Markopoulos, T. Brzobohatý, V. Vondrák, P. Horyl, *Scalable TFETI algorithm for two dimensional multibody contact problems with friction*, accepted in Journal of Computational and Applied Mathematics.

[7] Z. Dostál, T. Kozubek, A. Markopoulos, T. Brzobohatý, V. Vondrák, P. Horyl, *Theoretically supported scalable TFETI algorithm for the solution of multibody 3D contact problems with friction*, submitted.

[8] V. Vondrák, T. Kozubek, Z. Dostál, *Parallel solution of contact shape optimization problems based on Total FETI domain decomposition method*, Engineering Optimization, accepted.

Ill–Posed Inverse Problems in Image Processing

Introduction, Structured matrices, Spectral filtering, Regularization, Noise revealing

I. Hnětynková[1], M. Plešinger[2], Z. Strakoš[3]

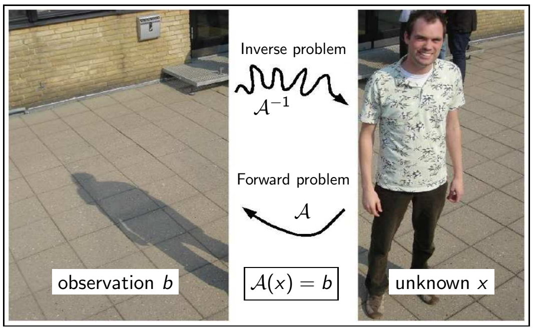
hnetynko@karlin.mff.cuni.cz, martin.plesinger@sam.math.ethz.ch, strakos@cs.cas.cz

[1,3]Faculty of Mathematics and Phycics, Charles University, Prague
[2]Seminar of Applied Mathematics, Dept. of Math., ETH Zürich
[1,2,3]Institute of Computer Science, Academy of Sciences of the Czech Republic

SNA '11, January 24—28

Motivation. A gentle start ...

What is it an inverse problem?



Inverse problem

$\mathcal{A}^{-1}$

Forward problem

$\mathcal{A}$

observation b

$\mathcal{A}(x) = b$

unknown x

[Kjøller: M.Sc. thesis, DTU Lyngby, 2007].

More realistic examples of ill-posed inverse problems

Computer tomography in medical sciences

Computer tomograph (CT) maps a 3D object of $M \times N \times K$ voxels by ℓ X-ray measurements on ℓ pictures with $m \times n$ pixels,

$$\mathcal{A}(\cdot) \equiv \quad : \mathbb{R}^{M\times N\times K} \longrightarrow \bigotimes_{j=1}^{\ell} \mathbb{R}^{m\times n}.$$

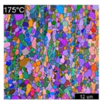
Simpler 2D tomography problem leads to the **Radon transform**. The inverse problem is ill-posed. (3D case is more complicated.)

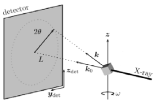
The mathematical problem is **extremely sensitive** to errors which are **always** present in the (measured) data: *discretization error* (finite ℓ, m, n); *rounding errors*; *physical sources of noise* (electronic noise in semiconductor PN-junctions in transistors, ...).

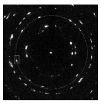
More realistic examples of ill-posed inverse problems

Transmision computer tomography in crystalographics

Reconstruction of an *unknown* orientation distribution function (ODF) of grains in a given sample of a polycrystalline matherial,







$$\mathcal{A}\left(\quad\right) \equiv \quad \longrightarrow \underbrace{\left(\quad, \ldots\right)}_{\text{observation} = \text{data} + \text{noise}}.$$

The *right-hand side* is a set of measured difractograms. [Hansen, Sørensen, Südkösd, Poulsen: SIIMS, 2009].

Further analogous applications also in geology, e.g.:

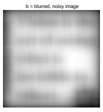
- Seismic tomography (cracks in tectonic plates),
- Gravimetry & magnetometry (ore mineralization).

More realistic examples of ill-posed inverse problems

Image deblurring—Our pilot application

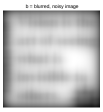
Our pilot application is the image deblurring problem

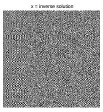
x = true image
Jonathan Swift
Vision is the
art of seeing
what is
invisible to
others.



$$\mathcal{A}\left(\quad\right) \longrightarrow \quad = \text{data} + \text{noise}.$$

It leads to a linear system $Ax = b$ with square nonsingular matrix. Let us motivate our tutorial by a "naive solution" of this system





$$\mathcal{A}^{-1}\left(\quad\right) = \quad .$$

[Nagy: Emory University].

More realistic examples of ill-posed inverse problems

General framework

In general we deal with a linear problem

$$Ax = b$$

which typically arose as a discretization of a

Fredholm integral equation of the 1st kind

$$y(\mathbf{s}) = \int K(\mathbf{s}, \mathbf{t}) x(\mathbf{t}) d\mathbf{t}.$$

The observation vector (right-hand side) is contaminated by noise

$$b = b^{\text{exact}} + b^{\text{noise}}, \quad \text{where} \quad \|b^{\text{exact}}\| \gg \|b^{\text{noise}}\|.$$

More realistic examples of ill-posed inverse problems

General framework

We want to compute (approximate)

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

Unfortunatelly, because the problem is inverse and ill-posed

$$\|A^{-1} b^{\text{exact}}\| \ll \|A^{-1} b^{\text{noise}}\|,$$

the data we look for are in the naive solution covered by the inverted noise. The naive solution

$$x = A^{-1} b = A^{-1} b^{\text{exact}} + A^{-1} b^{\text{noise}}$$

typically has nothing to do with the wanted x^{exact}.

Outline of the tutorial

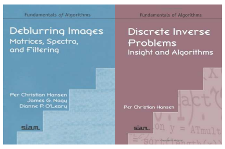
- **Lecture I—Problem formulation:**
  Mathematical model of blurring, System of linear algebraic equations, Properties of the problem, Impact of noise.
- **Lecture II—Regularization:**
  Basic regularization techniques (TSVD, Tikhonov), Criteria for choosing regularization parameters, Iterative regularization, Hybrid methods.
- **Lecture III—Noise revealing:**
  Golub-Kahan iteratie bidiagonalization and its properties, Propagation of noise, Determination of the noise level, Noise vector approximation, Open problems.

References

Textbooks + software

Textbooks:

- Hansen, Nagy, O'Leary: *Deblurring Images, Spectra, Matrices, and Filtering*, SIAM, FA03, 2006.
- Hansen: *Discrete Inverse Problems, Insight and Algorithms*, SIAM, FA07, 2010.



Sofwtare (MatLab toolboxes):

- HNO package,
- Regularization tools,
- AIRtools,
- ...

(software available on the homepage of P. C. Hansen).

Outline of Lecture I

- **1. Mathematical model of blurring:**
  Blurring as an operator on the vector space of matrices, Linear and spatial invariant operator, Point-spread-function, 2D convolution, Boundary conditions.
- **2. System of linear algebraic equations:**
  Gaußian blur, Exploiting the separability, 1D Gaußian blurring operator, Boundary conditions, 2D Gaußian blurring operator, Structured matrices.
- **3. Properties of the problem:**
  Smoothing properties, Singular vectors of A, Singular values of A, The right-hand side, Discrete Pickard condition (DPC), SVD and Image deblurring problem, Singular images.
- **4. Impact of noise:**
  Violation of DPC, Naive solution, Regularization and filtering.

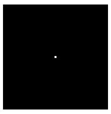
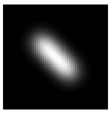
1. Mathematical model of blurring

1. Mathematical model of blurring

Blurring as an operator of the vector space of images

The **grayscale image** can be considered as a **matrix**, consider for convenience *black* $\equiv 0$ and *white* $\equiv 1$.

Consider a, so called, **single-pixel-image (SPI)** and a blurring operator as follows

$\mathcal{A}(X) = \mathcal{A}\Bigg($  $\Bigg) =$  $= B,$

where $X = [x_1, \ldots, x_k]$, $B = [b_1, \ldots, b_k] \in \mathbb{R}^{k \times k}$.

The image (matrix) B is called **point-spread-function (PSF)**.

(In Parts 1, 2, 3 we talk about the operator, the right-hand side is noise-free.)

1. Mathematical model of blurring

Linear and spatial invariant operator

Consider $\mathcal{A}$ to be:

1. linear (additive & homogenous),
2. spatial invariant.

Linearity of $\mathcal{A}$ allows to rewrite $\mathcal{A}(X) = B$ as a system of linear algebraic equations

$$Ax = b, \qquad A \in \mathbb{R}^{N\times N}, \quad x,\ b \in \mathbb{R}^{N}.$$

(We do not know how, yet.)

1. Mathematical model of blurring

Linear and spatial invariant operator

The matrix X containing the SPI has only one nonzero entry (moreover equal to one).

Therefore the unfolded X

$$x = \mathrm{vec}(X) = [x_1^T, \dots, x_k^T]^T \;=\; e_j$$

represents an Euclidean vector.

The unfolding of the corredponding B (containing the PSF) then represents jth column of A

$$A\, e_j = b = \mathrm{vec}(B) = [b_1^T, \dots, b_k^T]^T.$$

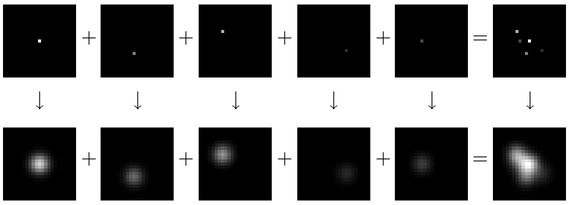
The matrix A is composed columnwise by unfolded PSFs corresponding to SPIs with different positions of the nonzero pixel.

1. Mathematical model of blurring

Linear and spatial invariant operator

Spatial invariance of $\mathcal{A}$ $\;\equiv\;$ The PSF is the same for all positions of the nonzero pixel in SPI. (What about pixels close to the border?)

Linearity + spatial invariance:



First row: Original (SPI) images (matrices X).
Second row: Blurred (PSF) images (matrices $B = \mathcal{A}(X)$).

1. Mathematical model of blurring

Point—spread—function (PSF)

Linear and spatially invariant blurring operator $\mathcal{A}$ is **fully described** by its action on one SPI, i.e. **by one PSF**. (Which one?)

Recall: Up to now the *width* and *height* of both the SPI and PSF images have been equal to some k, called the **window size**.

For correctness the window size must be properly chosen, namely:

- the window size must be sufficiently large
  (increase of k leads to extension of PSF image by black),
- the window is typically square (for simplicity),
- we use window of odd size (for simplicity), i.e.

$$k = 2\ell + 1.$$

1. Mathematical model of blurring

Point—spread—function (PSF)

The square window with sufficiently large odd size $k = 2\ell + 1$ allows to consider SPI image given by the matrix

$$SPI = e_{\ell+1} e_{\ell+1}^T \in \mathbb{R}^{k\times k}$$

(the only nonzero pixel is in the middle of SPI).

The corresponding PSF image given by the matrix

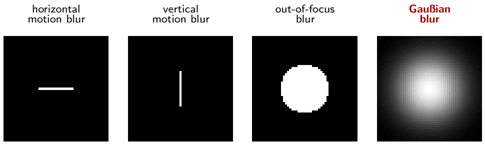
$$PSF_{\mathcal{A}} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,k} \\ \vdots & \ddots & \vdots \\ p_{k,1} & \cdots & p_{k,k} \end{bmatrix} = \begin{bmatrix} \bar{p}_{-\ell,-\ell} & \cdots & \bar{p}_{-\ell,+\ell} \\ \vdots & \ddots & \vdots \\ \bar{p}_{+\ell,-\ell} & \cdots & \bar{p}_{+\ell,+\ell} \end{bmatrix} \in \mathbb{R}^{k\times k}$$

will be further used for the description of the operator $\mathcal{A}$.

1. Mathematical model of blurring

Point—spread—function (PSF)

Examples of $PSF_{\mathcal{A}}$:



1. Mathematical model of blurring

2D convolution

We have the linear, spatial invariant $\mathcal{A}$ given by $PSF_{\mathcal{A}} \in \mathbb{R}^{k \times k}$.
Consider a general grayscale image given by a matrix $X \in \mathbb{R}^{m \times n}$.
How to realize the action of $\mathcal{A}$ on X, i.e. $B = \mathcal{A}(X)$, using $PSF_{\mathcal{A}}$?

Entrywise application of PSF:

1. $X = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{i,j}$, where $X_{i,j} \equiv x_{i,j}(e_i e_j^T) \in \mathbb{R}^{m \times n}$;
2. realize the action of $\mathcal{A}$ on the single-pixel-image $X_{i,j}$

$$X_{i,j} = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & x_{i,j} SPI & 0 \\ \hline 0 & 0 & 0 \end{array}\right] \longrightarrow B_{i,j} \equiv \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & x_{i,j} PSF_{\mathcal{A}} & 0 \\ \hline 0 & 0 & 0 \end{array}\right];$$

3. $B = \sum_{i=1}^{m} \sum_{j=1}^{n} B_{i,j}$ due to the linearity of $\mathcal{A}$.

1. Mathematical model of blurring

2D convolution

Example: $B = \sum_{i=1}^{m} \sum_{j=1}^{n} B_{i,j} = \ \ldots$

$$+x_{2,2}\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{2,3}\begin{pmatrix} 0 & p_{1,1} & p_{1,2} & p_{1,3} & 0 \\ 0 & p_{2,1} & p_{2,2} & p_{2,3} & 0 \\ 0 & p_{3,1} & p_{3,2} & p_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{2,4}\begin{pmatrix} 0 & 0 & p_{1,1} & p_{1,2} & p_{1,3} \\ 0 & 0 & p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & p_{3,1} & p_{3,2} & p_{3,3} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+x_{3,2}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{3,3}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & p_{1,1} & p_{1,2} & p_{1,3} & 0 \\ 0 & p_{2,1} & p_{2,2} & p_{3,3} & 0 \\ 0 & p_{3,1} & p_{3,2} & p_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{3,4}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{1,1} & p_{1,2} & p_{1,3} \\ 0 & 0 & p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & p_{3,1} & p_{3,2} & p_{3,3} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+x_{4,2}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} & 0 & 0 \end{pmatrix} + x_{4,3}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p_{1,1} & p_{1,2} & p_{1,3} & 0 \\ 0 & p_{2,1} & p_{2,2} & p_{2,3} & 0 \\ 0 & p_{3,1} & p_{3,2} & p_{3,3} & 0 \end{pmatrix} + x_{4,4}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{1,1} & p_{1,2} & p_{1,3} \\ 0 & 0 & p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}$$

$+ \ldots$, where

$$PSF_{\mathcal{A}} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}, \qquad \begin{aligned} b_{3,3} &= x_{2,2}\, p_{3,3} + x_{2,3}\, p_{3,2} + x_{2,4}\, p_{3,1} \\ &+ x_{3,2}\, p_{2,3} + x_{3,3}\, p_{2,2} + x_{3,4}\, p_{2,1} \\ &+ x_{4,2}\, p_{1,3} + x_{4,3}\, p_{1,2} + x_{4,4}\, p_{1,1} \end{aligned}.$$

1. Mathematical model of blurring

2D convolution

The entry $b_{i,j}$ of B is influenced by the entry $x_{i,j}$ and a few entries in its surroundings, depending on the support of $PSF_{\mathcal{A}}$.

In general:

$$b_{i,j} = \sum_{h=-\ell}^{\ell} \sum_{w=-\ell}^{\ell} x_{i-h,j-w} \bar{p}_{h,w}.$$

The blured image represented by matrix B is therefore the

2D convolution

of X with $PSF_{\mathcal{A}}$.

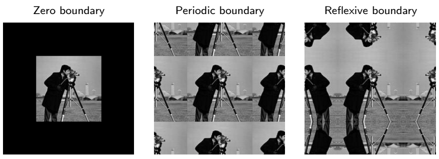
Boundary: Pixels $x_{\mu,\nu}$ for $\mu \in \mathbb{Z} \setminus [1, \ldots, m]$ or $\nu \in \mathbb{Z} \setminus [1, \ldots, n]$ ("outside" the original image X) are not given.

1. Mathematical model of blurring

Boundary conditions (BC)

Real-world blurred image B is involved by the information which is outside the scene X, i.e. by the boundary pixels $x_{\mu,\nu}$.
For the reconstruction of the real-world scene (deblurring) we do have to consider some **boundary condition**:

- Outside the scene is **nothing**, $x_{\mu,\nu} = 0$ (black), e.g., in astrononomical observations.
- The scene contains **periodic** patterns, e.g., in micro/nanoscale imaging of matherials.
- The scene can be prolongated by **reflecting**.



1. Mathematical model of blurring

Summary

Now we know "everything" about the simplest mathematical model of blurring:

- We consider linear, spatial invariant operator $\mathcal{A}$, which is represented by its point-spread-function $PSF_{\mathcal{A}}$.
- The 2D convolution of true scene with the point-spread-function represents the blurring.
- Convolution uses some information from the outside of the scene, therefore we need to consider some boundary conditions.

2. System of linear algebraic equations

2. System of linear algebraic equations

Basic concept

The problem $\mathcal{A}(X) = B$ can be rewritten (emploing the 2D convolution formula) as a system of linear algebraic equations

$$Ax = b, \qquad A \in \mathbb{R}^{mn \times mn}, \quad x = \mathrm{vec}(X), \ b = \mathrm{vec}(B) \in \mathbb{R}^{mn},$$

where the entries of A are the entries of the PSF, and

$$b_{i,j} = \sum_{h=-\ell}^{\ell} \sum_{w=-\ell}^{\ell} x_{i-h,j-w} \bar{p}_{h,w}.$$

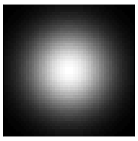
In general:

- PSF has small localized support,
- each pixel is influenced only by a few pixels in its close surroundings,
- therefore A is **sparse**.

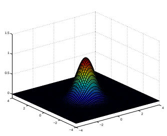
2. System of linear algebraic equations

Gaußian PSF / Gaußian blur

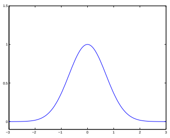
In the rest we consider **Gaußian blur**:



Gaußian PSF



$G_{\mathrm{2D}}(h, w)$



$G_{\mathrm{1D}}(\xi)$

where (in a continuous domain)

$$G_{\mathrm{2D}}(h, w) = e^{-(h^2+w^2)} = e^{-h^2} e^{-w^2}, \qquad G_{\mathrm{1D}}(\xi) = e^{-\xi^2}.$$

Gaußian blur is the simplest and in many cases sufficient model.
A big advantage is its **separability** $G_{\mathrm{2D}}(h, w) = G_{\mathrm{1D}}(h) G_{\mathrm{1D}}(w)$.

2. System of linear algebraic equations

Exploiting the separability

Consider the 2D convolution with Gaußian PSF in a continuous domain. Exploiting the separability, we get

$$\begin{aligned} B(i,j) &= \iint_{\mathbb{R}^2} X(i-h, j-w)\, e^{-(h^2+w^2)}\, dh\, dw \\ &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} X(i-h, j-w)\, e^{-h^2}\, dh \right) e^{-w^2}\, dw \\ &= \int_{-\infty}^{\infty} Y(i, j-w) e^{-w^2}\, dw, \end{aligned}$$

where $\qquad Y(i,j) = \int_{-\infty}^{\infty} X(i-h, j)\, e^{-h^2}\, dh.$

The blurring in the direction h (height) is **independent** on the blurring in the direction w (width).
In the discrete setting: The blurring of columns of X is **independent** on the blurring of rows of X.

2. System of linear algebraic equations

Exploiting the separability

As a direct consequence of the separability, the PSF matrix is a **rank one** matrix of the form

$$PSF_{\mathcal{A}} = c r^T, \qquad c, r \in \mathbb{R}^k.$$

The blurring of columns (rows) of X is realized by 1D (discrete) convolution with c (r), the discretized $G_{\mathrm{1D}}(\xi) = e^{-\xi^2}$.

Let A_C, A_R be matrices representing discete 1D Gaußian blurring operators, where

- A_C realizes blurring of columns of X,
- A_R^T realizes blurring of rows of X.

Then the problem $\mathcal{A}(X) = B$ can be rewritten as a **matrix equation**

$$A_C\, X\, A_R^T = B, \qquad A_C \in \mathbb{R}^{m \times m}, \quad A_R \in \mathbb{R}^{n \times n}.$$

2. System of linear algebraic equations

1D convolution

Consider the following example of an A_C related 1D convolution:

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array} \right] \left[\begin{array}{c} \xi_{-1} \\ \xi_0 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline \xi_7 \\ \xi_8 \end{array} \right],$$

where $b = [\beta_1, \ldots, \beta_6]^T$, $x = [\xi_1, \ldots, \xi_6]^T$,
and $c = [c_1, \ldots, c_5]^T$ is the 1D (Gaußian) point-spread-function.

2. System of linear algebraic equations

Boundary conditions

The vector $[x_{-1}, x_0 | x_1, \ldots, \xi_6 | \xi_7, \xi_8]^T$ represents the true scene. In the reconstruction we consider:

$$\begin{aligned} &[0, 0 | \xi_1, \ldots, \xi_6 | 0, 0]^T && \sim \text{ zero boundary condition,} \\ &[\xi_5, \xi_6 | \xi_1, \ldots, \xi_6 | \xi_1, \xi_2]^T && \sim \text{ periodic boundary condition, or} \\ &[\xi_2, \xi_1 | \xi_1, \ldots, \xi_6 | \xi_6, \xi_5]^T && \sim \text{ reflexive boundary condition.} \end{aligned}$$

In general $A_C = M + BC$, where

$$M = \begin{bmatrix} c_3 & c_2 & c_1 & & & \\ c_4 & c_3 & c_2 & c_1 & & \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & c_5 & c_4 & c_3 & c_2 \\ & & & c_5 & c_4 & c_3 \end{bmatrix},$$

and BC is a correction due to the boundary conditions.

2. System of linear algebraic equations

Boundary conditions

Zero boundary condition:

$$A_C x = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} 0 \\ 0 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline 0 \\ 0 \end{array}\right] = \begin{bmatrix} c_3 & c_2 & c_1 & & & \\ c_4 & c_3 & c_2 & c_1 & & \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & c_5 & c_4 & c_3 & c_2 \\ & & & c_5 & c_4 & c_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix},$$

i.e. here $BC = 0$ and $A_C = M$ is a **Toeplitz matrix**.

2. System of linear algebraic equations

Boundary conditions

Periodic boundary condition:

$$A_C x = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} \xi_5 \\ \xi_6 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline \xi_1 \\ \xi_2 \end{array}\right] = \begin{bmatrix} c_3 & c_2 & c_1 & & c_5 & c_4 \\ c_4 & c_3 & c_2 & c_1 & & c_5 \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ c_1 & & c_5 & c_4 & c_3 & c_2 \\ c_2 & c_1 & & c_5 & c_4 & c_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix},$$

$$\text{i.e. here} \quad BC = \begin{bmatrix} & & & c_5 & c_4 \\ & & & & c_5 \\ c_1 & & & & \\ c_2 & c_1 & & & \end{bmatrix}$$

and $A_C = M + BC$ is a **circulant matrix**.

2. System of linear algebraic equations

Boundary conditions

Reflexive boundary condition:

$$A_C x = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} \xi_2 \\ \xi_1 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline \xi_6 \\ \xi_5 \end{array}\right] = \begin{bmatrix} c_3+c_4 & c_2+c_5 & c_1 & & & \\ c_4+c_5 & c_3 & c_2 & c_1 & & \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & c_5 & c_4 & c_3 & c_2+c_1 \\ & & & c_5 & c_4+c_1 & c_3+c_2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix}$$

$$\text{i.e. here} \quad BC = \begin{bmatrix} c_4 & c_5 & & \\ c_5 & & & \\ & & & c_1 \\ & & c_1 & c_2 \end{bmatrix}$$

and $A_C = M + BC$ is a **Toeplitz-plus-Hankel matrix**.

2. System of linear algebraic equations

Boundary conditions—Summary

Three types of boundary conditions:

- zero boundary condition,
- periodic boundary condition,
- reflexive boundary condition,

correspond to the three types of matrices A_C and A_R:

- Toeplitz matrix,
- circulant matrix,
- Toeplitz-plus-Hankel matrix,

in the linear system of the form

$$A_C \, X \, A_R^T = B.$$

2. System of linear algebraic equations

2D Gaußian blurring operator—Kroneckerized product structure

Now we show how to rewrite the matrix equation $A_C \, X \, A_R^T = B$ as a system of linear algebraic equations in a usual form.

Consider $A_R = I_n$. The matrix equation

$$A_C \, X = B$$

can be rewritten as

$$(I_n \otimes A_C)\,\mathrm{vec}(X) = \begin{bmatrix} A_C & & \\ & \ddots & \\ & & A_C \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \mathrm{vec}(B),$$

where $X = [x_1, \dots, x_n]$, $B = [b_1, \dots, b_n]$, and $\otimes$ denotes the **Kronecker product**.

2. System of linear algebraic equations

2D Gaußian blurring operator—Kroneckerized product structure

Consider $A_C = I_m$. The matrix equation $X \, A_R^T = B$ can be rewritten as

$$(A_R \otimes I_m)\,\mathrm{vec}(X) = \begin{bmatrix} a_{1,1}^R I_m & \cdots & a_{1,n}^R I_m \\ \vdots & \ddots & \vdots \\ a_{n,1}^R I_m & \cdots & a_{n,n}^R I_m \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \mathrm{vec}(B).$$

Consequently $A_C \, X \, A_R^T = (A_C \, X) \, A_R^T$ gives

$$(A_R \otimes I_m)\mathrm{vec}(A_C \, X) = (A_R \otimes I_m)(I_n \otimes A_C)\mathrm{vec}(X).$$

Using properties of Kronecker product, this system is equivalent to

$$Ax = (A_R \otimes A_C)\,\mathrm{vec}(X) = \mathrm{vec}(B) = b,$$

where

$$A = \begin{bmatrix} a_{1,1}^R A_C & \cdots & a_{1,n}^R A_C \\ \vdots & \ddots & \vdots \\ a_{n,1}^R A_C & \cdots & a_{n,n}^R A_C \end{bmatrix} \in \mathbb{R}^{mn \times mn}.$$

2. System of linear algebraic equations

Structured matrices

We have

$$A = A_R \otimes A_C = \begin{bmatrix} a^R_{1,1}A_C & \cdots & a^R_{1,n}A_C \\ \vdots & \ddots & \vdots \\ a^R_{n,1}A_C & \cdots & a^R_{n,n}A_C \end{bmatrix} \in \mathbb{R}^{mn\times mn},$$

where A_C, A_R are Toeplitz, circulant, or Toeplitz-plus-Hankel.

If A_C is Toeplitz, then A is a matrix with Toeplitz blocks.
If A_R is Toeplitz, then A is a block-Toeplitz matrix.
If A_C and A_R are Toeplitz (zero BC), then A is

block—Toeplitz with Toeplitz blocks (BTTB).

Analogously, for periodic BC we get **BCCB** matrix, for reflexie BC we get a sum of four matrices **BTTB+BTHB+BHTB+BHHB**.

3. Properties of the problem

3. Properties of the problem

Smoothing properties

We have an inverse ill-posed problem $Ax = b$, a discretization of a Fredholm integral equation of the 1st kind

$$y(\mathbf{s}) = \int K(\mathbf{s},\mathbf{t})x(\mathbf{t})d\mathbf{t}.$$

The matrix A is a restriction of the integral kernel $K(\mathbf{s},\mathbf{t})$ (the convolution kernel in image deblurring)

- the kernel $K(\mathbf{s},\mathbf{t})$ has **smoothing property**,
- therefore the vector $y(\mathbf{s})$ is smooth,

and these properties are inherited by the discretized problem. Further analysis is based on the singular value decomposition

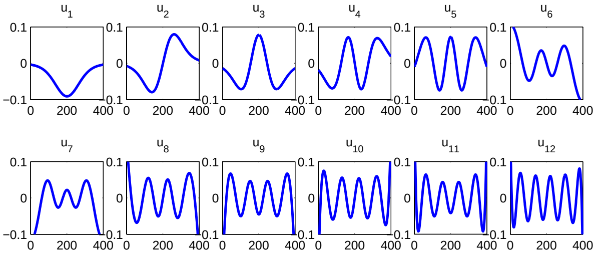
$$A = U\Sigma V^T, \qquad U \in \mathbb{R}^{N\times N}, \quad \Sigma \in \mathbb{R}^{N\times N}, \quad V \in \mathbb{R}^{N\times N},$$

(and $N = mn$ in image deblurring).

3. Properties of the problem

Singular vectors of A

Singular vectors of A represent bases with increasing frequencies:

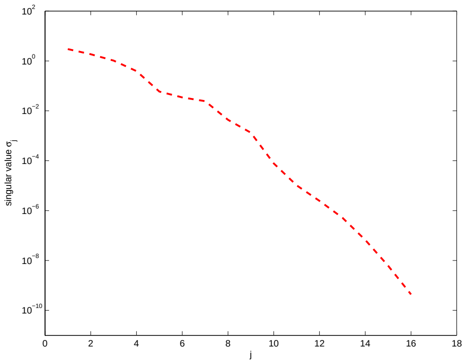


First 12 left singular vectors of 1D ill-posed problem **SHAW(400)** [Regularization Toolbox].

3. Properties of the problem

Singular values of A

Singular values decay without a noticeable gap (SHAW(400)):



3. Properties of the problem

The right-hand side

First recall that b is the discretized smooth $y(\mathbf{s})$, therefore

b is smooth, i.e. dominated by low frequencies.

Thus b has large components in directions of several first vectors u_j, and $|u_j^T b|$ **on average decay with** j.

3. Properties of the problem

The Discrete Pickard condition

Using the dyadic form of SVD

$$A = \sum_{j=1}^{N} u_j \sigma_j v_j^T, \quad N \text{ is the dimension of the discretized } K(\mathbf{s}, \mathbf{t}),$$

the solution of $Ax = b$ can be rewritten as a linear combination of right-singular vectors,

$$x = A^{-1} b = \sum_{j=1}^{N} \frac{u_j^T b}{\sigma_j} v_j.$$

Since x is a discretization of some real-world object $x(\mathbf{t})$ (e.g., an "true image") the sequence of these sums converges to $x(\mathbf{t})$ with $N \longrightarrow \infty$.

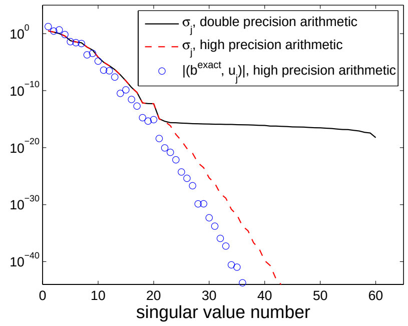
This is possible only if $|u_j^T b|$ are on average decay faster than σ_j.

This property is called the **(discrete) Pickard condition (DPC)**.

3. Properties of the problem

The Discrete Pickard condition

The discrete Pickard condition (SHAW(400)):



3. Properties of the problem

SVD and Image deblurring problem

Back to the image deblurring problem: We have

$$A_C \, X \, A_R^T = B \quad \Longleftrightarrow \quad (A_R \otimes A_C) \, \mathrm{vec}(X) = \mathrm{vec}(B).$$

Consider SVDs of both A_C and A_R

$$A_C = U_C \, \mathrm{diag}(s_C) V_C^T, \qquad A_R = U_R \, \mathrm{diag}(s_R) V_R^T,$$
$$s_C = [\sigma_1^C, \ldots, \sigma_m^C]^T \in \mathbb{R}^m, \qquad s_R = [\sigma_1^R, \ldots, \sigma_n^R]^T \in \mathbb{R}^n.$$

Using the basic properties of the Kronecker product

$$\begin{aligned} A = A_R \otimes A_C &= (U_R \, \mathrm{diag}(s_R) V_R^T) \otimes (U_C \, \mathrm{diag}(s_C) V_C^T) \\ &= (U_R \otimes U_C) \, \mathrm{diag}(s_R \otimes s_C)(V_R \otimes V_C)^T = U \Sigma V^T, \end{aligned}$$

we get SVD of A (up to the ordering of singular values).

3. Properties of the problem

SVD and Image deblurring problem

The solution of $A_C \, X \, A_R^T = B$ can be written directly as

$$X = V_C \underbrace{(\overbrace{(U_C^T \, B \, U_R)}^{\text{projections } u_j^T b} \oslash (s_C \, s_R^T))}_{\text{fractions } (u_j^T b)/\sigma_j} V_R^T,$$

where $K \oslash M$ denotes the Hadamard product of K with the componentwise inverse of M (using MatLab notation `K./M`).

Or using the dyadic expansion as

$$x = \sum_{j=1}^{N} \frac{u_j^T \, \mathrm{vec}(B)}{\sigma_j} v_j, \qquad X = \mathrm{mtx}(x), \qquad N = mn,$$

where $\mathrm{mtx}(\cdot)$ denotes an inverse mapping to $\mathrm{vec}(\cdot)$.

3. Properties of the problem

Singular images

The solution

$$x = \sum_{j=1}^{N} \underbrace{\frac{u_j^T \, \mathrm{vec}(B)}{\sigma_j}}_{\text{scalar}} v_j, \qquad X = \mathrm{mtx}(x), \qquad N = mn,$$

is a linear combination of right singular vectors v_j.

It can be further rewritten as

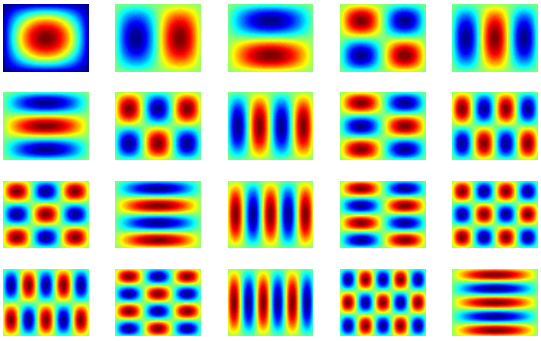
$$X = \sum_{j=1}^{N} \frac{u_j^T \, \mathrm{vec}(B)}{\sigma_j} V_j, \qquad V_j = \mathrm{mtx}(v_j) \in \mathbb{R}^{m \times n}$$

using **singular images** V_j (the reshaped right singular vectors).

3. Properties of the problem

Singular images

Singular images V_j (Gaußian blur, zero BC, artificial colors)



3. Properties of the problem

Note on computation of SVD

Recall that the matrices A_C, A_R are

- Toeplitz,
- circulant, or
- Toeplitz-plus-Hankel,

and often symmetric (depending on the symmetry of PSF).

Toeplitz matrix is fully determined by its first column and row,
circulant matrix by its first column (or row), and
Hankel matrix by the first column and the last row.

Eigenvalue decomposition (SVD) of such matrices can be efficiently computed using **discrete Fourier transform** (DFT/FFT algorithm), or **discrete cosine transform** (DCT algorithm).

4. Impact of noise

4. Impact of noise

Noise, Sources of noise

Consider a problem of the form

$$Ax = b, \quad b = b^{\mathrm{exact}} + b^{\mathrm{noise}}, \quad \|b^{\mathrm{exact}}\| \gg \|b^{\mathrm{noise}}\|,$$

where b^{noise} is unknown and represents, e.g.,

- rounding errors,
- discretization error,
- noise with physical sources (electronic noise on PN-junctions).

We want to approximate

$$x^{\mathrm{exact}} \equiv A^{-1} b^{\mathrm{exact}},$$

unfortunately

$$\|A^{-1} b^{\mathrm{exact}}\| \ll \|A^{-1} b^{\mathrm{noise}}\|.$$

4. Impact of noise

Violation of the discrete Pickard condition

The vector b^{noise} typically resebles **white noise**, i.e. it has flat frequency characteristics.

Recall that the singular vectors of A represent frequencies.

Thus the white noise components in left singular subspaces are about the same order of magnitude.
White noise

violates the discrete Pickard condition.

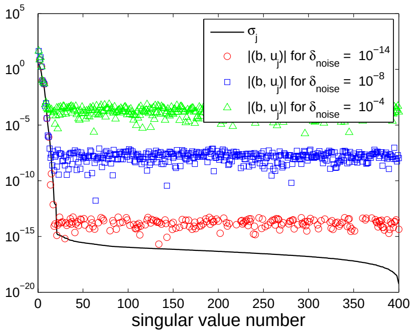
Summarizing:

- b^{exact} has some real pre-image x^{exact}, it satifies DPC
- b^{noise} does not have any real pre-image, it violates DPC.

4. Impact of noise

Violation of the discrete Pickard condition

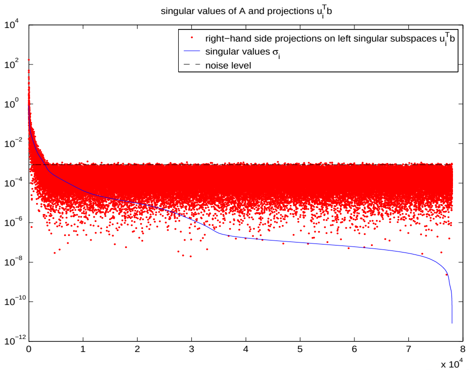
Violation of the discrete Pickard condition by noise (SHAW(400)):



4. Impact of noise

Violation of the discrete Pickard condition

Violation the dicrete Pickard condition by noise (Image deb. pb.):



4. Impact of noise

Violation of the discrete Pickard condition

Using $b = b^{\mathrm{exact}} + b^{\mathrm{noise}}$ we can write the expansion

$$x^{\mathrm{naive}} \equiv A^{-1} b = \sum\nolimits_{j=1}^{N} \frac{u_j^T b}{\sigma_j} v_j$$
$$= \underbrace{\sum\nolimits_{j=1}^{N} \frac{u_j^T b^{\mathrm{exact}}}{\sigma_j} v_j}_{x^{\mathrm{exact}}} + \underbrace{\sum\nolimits_{j=1}^{N} \frac{u_j^T b^{\mathrm{noise}}}{\sigma_j} v_j}_{\text{amplified noise}} .$$

Because σ_j decay and $|u_j^T b^{\mathrm{noise}}|$ are all about the same size, $|u_j^T b^{\mathrm{noise}}|/\sigma_j$ grow for large j. However, $|u_j^T b^{\mathrm{exact}}|/\sigma_j$ decay with j due to DPC. Thus the high-frequency noise covers all senseful information in x^{naive}.

Therefore x^{naive} is called the **naive solution**.

⟨MatLab demo⟩

4. Impact of noise

Regularization and filtering

To avoid the catastrophical impact of noise we employ regularization techniques.

In general the regularization can be understood as a filtering

$$x^{\mathrm{filtered}} \equiv \sum\nolimits_{j=1}^{N} \phi_j \frac{u_j^T b}{\sigma_j} v_j,$$

where the filter factors ϕ_j are given by some filter function $\phi_j = \phi(j, A, b, \ldots)$.

⟨Lecture II⟩

Summary

- We have an discrete inverse problem which is **ill-posed**. Our observation is often corrupted by (white) noise and we want to reconstruct the true pre-image of this observation.
- The whole concept was illustrated on the **image deblurring problem**, which was closely introduced and described.
- It was shown how the problem can be reformulated as a **system of linear algebraic equations**.
- We showed the typical **properties** of the corresponding matrix and the right-hand side, in particular the **discrete Pickard condition**.
- Finally, we illustrated the catastrophical **impact of the noise** on the reconstruction on an example.

Ill–Posed Inverse Problems in Image Processing

Introduction, Structured matrices, Spectral filtering, Regularization, Noise revealing

I. Hnětynková[1], M. Plešinger[2], Z. Strakoš[3]

hnetynko@karlin.mff.cuni.cz, martin.plesinger@sam.math.ethz.ch, strakos@cs.cas.cz

[1,3]Faculty of Mathematics and Phycics, Charles University, Prague
[2]Seminar of Applied Mathematics, Dept. of Math., ETH Zürich
[1,2,3]Institute of Computer Science, Academy of Sciences of the Czech Republic

SNA '11, January 24—28

Recapitulation of Lecture I

Linear system

Consider the problem

$$Ax = b, \quad b = b^{\text{exact}} + b^{\text{noise}}, \quad A \in \mathbb{R}^{N\times N}, \quad x, b \in \mathbb{R}^N,$$

where

- A is a discretization of a smoothing operator,
- singular values of A decay,
- singular vectors of A represent increasing frequencies,
- b^{exact} is smooth and satisfies the discrete Pickard condition,
- b^{noise} is unknown white noise,

$$\|b^{\text{exact}}\| \gg \|b^{\text{noise}}\|, \qquad \text{but} \qquad \|A^{-1}b^{\text{exact}}\| \ll \|A^{-1}b^{\text{noise}}\|.$$

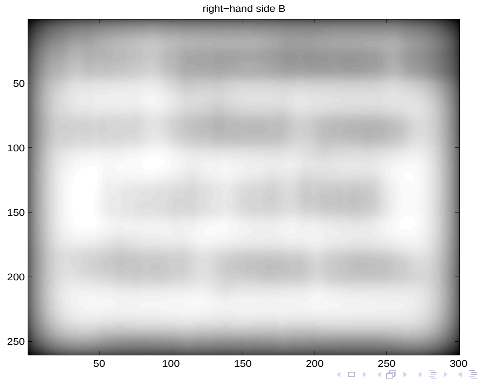
We want to approximate

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

Recapitulation of Lecture I

Right-hand side

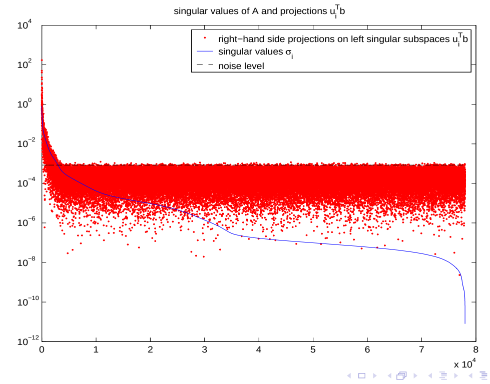
Smooth right-hand side (including noise):



Recapitulation of Lecture I

Violation of the discrete Pickard condition

Violation of the dicrete Pickard condition in the noisy b:



Recapitulation of Lecture I

Solution

Using SVD $A = U\Sigma V^T$ the **filtered** solution is

$$x^{\text{filtered}} = \sum_{j=1}^{N} \phi_j \frac{u_j^T b}{\sigma_j} v_j, \qquad x^{\text{filtered}} = V\Phi\Sigma^{-1}U^T b,$$

where $\Phi = \mathrm{diag}(\phi_1, \ldots, \phi_N)$. Particularly in the image deblurring problem

$$X^{\text{filtered}} = \sum_{j=1}^{N} \phi_j \frac{u_j^T \operatorname{vec}(B)}{\sigma_j} V_j, \qquad \text{where } V_j \text{ are singular images.}$$

The filter factors ϕ_j are given by some filter function

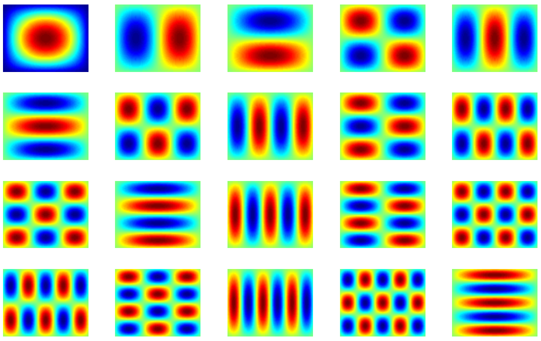
$$\phi_j = \phi(j, A, b, \ldots),$$

for $\phi_j = 1$, $j = 1, \ldots, N$, we get the **naive solution**.

Recapitulation of Lecture I

Singular images

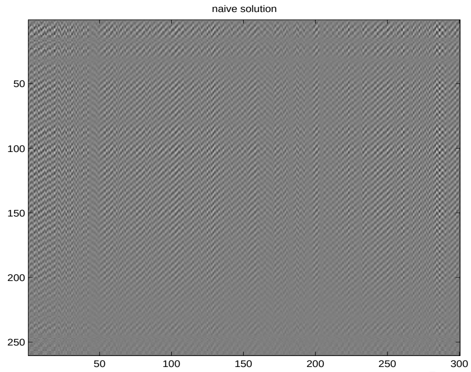
Singular images V_j (Gaußian blur, zero BC, artificial colors):



Recapitulation of Lecture I

Naive solution

The naive solution is dominated by high-frequency noise:



Outline of the tutorial

- **Lecture I—Problem formulation:**

  Mathematical model of blurring, System of linear algebraic equations, Properties of the problem, Impact of noise.
- **Lecture II—Regularization:**

  Basic regularization techniques (TSVD, Tikhonov), Criteria for choosing regularization parameters, Iterative regularization, Hybrid methods.
- **Lecture III—Noise revealing:**

  Golub-Kahan iteratie bidiagonalization and its properties, Propagation of noise, Determination of the noise level, Noise vector approximation, Open problems.

Outline of Lecture II

- **5. Basic regularization techniques:**

  Truncated SVD, Selective SVD, Tikhonov regularization.
- **6. Choosing regularization parameters:**

  Discrepancy principle, Generalized cross validation, L-curve, Normalized cumulative periodogram.
- **7. Iterative regularziation:**

  Landweber iteration, Cimmino iteration, Kaczmarz's method, Projection methods, Regularizing Krylov subspace iterations.
- **8. Hybrid methods:**

  Introduction, Projection methods with inner Tikhonov regularization.

5. Basic regularization techniques

5. Basic regularization techniques

Truncated SVD

The simplest regularization technique is the **truncated SVD (TSVD)**. Noise affects x^{naive} through the components corresponding to the smalest singular values,

$$x^{\text{naive}} = \underbrace{\sum_{j=1}^{k} \frac{u_j^T b}{\sigma_j} v_j}_{\text{data dominated}} + \underbrace{\sum_{j=k+1}^{N} \frac{u_j^T b}{\sigma_j} v_j}_{\text{noise dominated}} .$$

Idea: Omit the noise dominated part. Define

$$x^{\text{TSVD}(k)} \equiv \sum_{j=1}^{k} \frac{u_j^T b}{\sigma_j} v_j = \sum_{j=1}^{N} \phi_j \frac{u_j^T b}{\sigma_j} v_j,$$

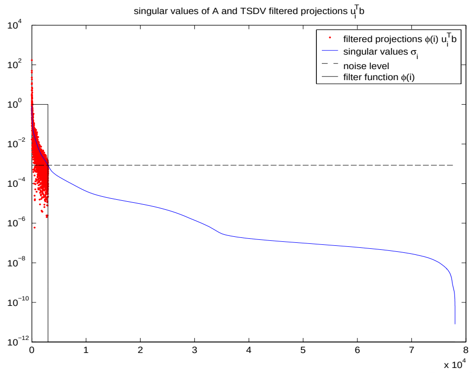
where

$$\phi_j = \begin{cases} 1 & \text{for} \quad j \le k \\ 0 & \text{for} \quad j > k \end{cases} .$$

5. Basic regularization techniques

Truncated SVD

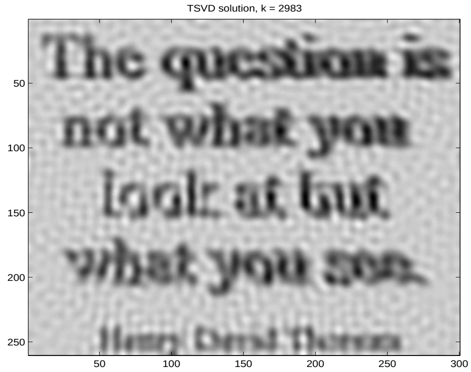
The TSVD filter function, $k = 2\,983$:



5. Basic regularization techniques

Truncated SVD

The TSVD solution, $k = 2\,983$:



5. Basic regularization techniques

Truncated SVD

Advantages:

- Simple idea, simple implementation, simple analysis,

$$A \quad \text{is replaced by} \quad U\Phi^{\dagger}\Sigma V^T, \quad \Phi = \mathrm{diag}(I_k, 0_{N-k}),$$

  i.e. the rank-k approximation of A.

Disadvantages:

- We have to compute the SVD of A (or the first k singular triplets).
- Choice of the **regularization parameter** k is usualy based on a knowledge of the norm of b^{noise} which is

  either revealed from the SVD analysis,

  or given explictly as an additional information.
- The noise dominated part still contains some information useful for reconstruction which is lost (step filter function).

5. Basic regularization techniques

Selective SVD

Similar approach to TSVD is the **selective SVD (SSVD)**. Consider $\|b^{\text{noise}}\|$ is known. Then

$$\|b^{\text{noise}}\| = \left(\sum\nolimits_{j=1}^{N} (u_j^T b^{\text{noise}})^2\right)^{1/2} \equiv \Delta^{\text{noise}}, \quad |u_j^T b^{\text{noise}}| \approx \varepsilon \equiv \frac{\Delta^{\text{noi}}}{N^{1/}}$$

because u_j represent frequencies and b^{noise} represents white noise.

We define

$$x^{\text{SSVD}(\varepsilon)} \equiv \sum\nolimits_{|u_j^T b| > \varepsilon} \frac{u_j^T b}{\sigma_j} v_j = \sum\nolimits_{j=1}^{N} \phi_j \frac{u_j^T b}{\sigma_j} v_j,$$

where

$$\phi_j = \begin{cases} 1 & \text{for} \quad |u_j^T b| > \varepsilon \\ 0 & \text{for} \quad |u_j^T b| \le \varepsilon \end{cases}.$$

5. Basic regularization techniques

Tikhonov approach

Classical **Tikhonov approach** is based on penalizing the norm of the solution

$$x^{\text{Tikhonov}(\lambda)} \equiv \arg\min_x \{\|b - Ax\| + \lambda\|Lx\|\},$$

where

- $\|b - Ax\|$ represents the residual norm,
- $\|Lx\|$ represents $(L^T L)$–(semi)norm of the solution,

  often $L = I_N$ (we restrict to this case),

  or it is a discretized 1st or 2nd order derivative operator,
- λ is the (positive) penalty parameter; clearly

$$\lim_{\lambda \longrightarrow 0} x^{\text{Tikhonov}(\lambda)} = x^{\text{naive}}.$$

5. Basic regularization techniques

Tikhonov approach

The Tikhonov minimization problem can be rewritten as

$$\begin{aligned} x^{\text{Tikhonov}(\lambda)} &= \arg\min_x \{\|b - Ax\| + \lambda\|Lx\|\} \\ &= \arg\min_x \{\|b - Ax\|^2 + \lambda^2\|Lx\|^2\} \\ &= \arg\min_x \left\{ \left\| \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ -\lambda L \end{bmatrix} x \right\|^2 \right\}, \end{aligned}$$

i.e. to get the Tikhonov solution we solve a **least squares (LS) problem**

$$\begin{bmatrix} A \\ -\lambda L \end{bmatrix} x = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

In particular, we do not have to compute the SVD of A.

5. Basic regularization techniques

Tikhonov approach

A solution of the Tikhonov LS problem

$$\begin{bmatrix} A \\ -\lambda L \end{bmatrix} x = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

can be analyzed through the system of normal equations

$$\begin{bmatrix} A \\ -\lambda L \end{bmatrix}^T \begin{bmatrix} A \\ -\lambda L \end{bmatrix} x = \begin{bmatrix} A \\ -\lambda L \end{bmatrix}^T \begin{bmatrix} b \\ 0 \end{bmatrix},$$

$$(A^T A + \lambda^2 L^T L)x = A^T b.$$

With the SVD of A, $A = U\Sigma V^T$, and $L = I_N = VV^T$ we get

$$(\Sigma^2 + \lambda^2 I_N)y = \Sigma U^T b,$$

where $y = V^T x$ and $x = Vy$.

5. Basic regularization techniques

Tikhonov approach

Thus

$$x^{\text{Tikhonov}(\lambda)} = V(\Sigma^2 + \lambda^2 I_N)^{-1}\Sigma U^T b,$$

which gives

$$\begin{aligned} x^{\text{Tikhonov}(\lambda)} &= \sum\nolimits_{j=1}^{N} \frac{\sigma_j}{\sigma_j^2 + \lambda^2}(u_j^T b) v_j \\ &= \sum\nolimits_{j=1}^{N} \frac{\sigma_j^2}{\sigma_j^2 + \lambda^2} \frac{u_j^T b}{\sigma_j} v_j = \sum\nolimits_{j=1}^{N} \phi_j \frac{u_j^T b}{\sigma_j} v_j, \end{aligned}$$

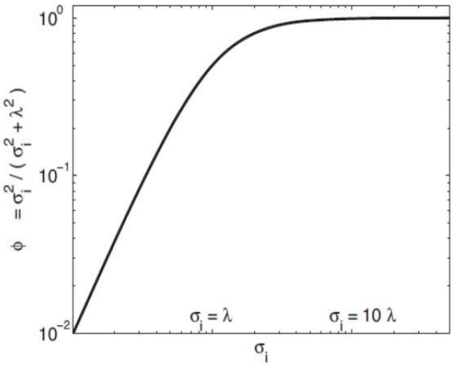
where

$$\phi_j = \frac{\sigma_j^2}{\sigma_j^2 + \lambda^2} \approx \begin{cases} 1 & \text{for} \quad \sigma_j \gg \lambda \\ \sigma_j^2/\lambda^2 & \text{for} \quad \sigma_j \ll \lambda \end{cases}, \qquad 0 < \phi_j < 1.$$

5. Basic regularization techniques

Tikhonov approach

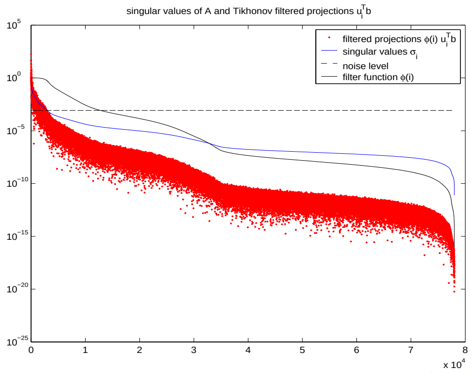
The behavior of the Tikhonov filter function:



5. Basic regularization techniques

Tikhonov approach

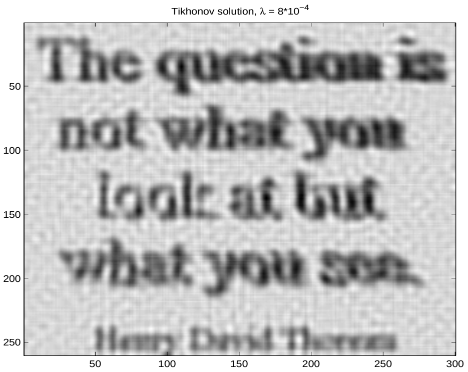
The Tikhonov filter function, $\lambda = 8 \times 10^{-4}$:



5. Basic regularization techniques

Tikhonov approach

The Tikhonov solution, $\lambda = 8 \times 10^{-4}$:



5. Basic regularization techniques

Tikhonov approach

Advantages:

- Simple idea, with $L = I_N$ simple analysis,

  $$A \quad \text{is replaced by} \quad U\Phi^{-1}\Sigma V^T, \quad \Phi = (\Sigma^2 + \lambda^2 I_N)^{-1}\Sigma^2.$$

- We do not have to compute SVD of A (compare with TSVD).
- The solution is given by some LS problem.
- The filter function is smooth (compare with TSVD).

Disadvantages:

- With $L \neq I_N$ the analysis is more complicated.
- We have to chose the **penalty parameter λ** (at this moment it is not clear how to do it).

5. Basic regularization techniques

Summary

We have two basic approaches:

- **Truncated SVD** (requires a part of the SVD of A)

  $$x^{\text{TSVD}(k)} = V\Phi\Sigma^{-1}U^T b, \quad \Phi = \text{diag}(I_k, 0_{N-k}),$$

  where k is a truncation (regularization) parameter.
- **Tikhonov regularization** (leads to a LS problem)

  $$x^{\text{Tikhonov}(\lambda)} = V\Phi\Sigma^{-1}U^T b, \quad \Phi = (\Sigma^2 + \lambda^2 I_n)^{-1}\Sigma^2,$$

  where λ is a penalty (regularization) parameter.

The question is:

How to choose the regularization parameters?

5. Basic regularization techniques

Note on monotonicity (TSVD)

The **norms of the TSVD solution and the residual**

$$\|x^{\mathrm{TSVD}(k)}\|, \quad \|b - Ax^{\mathrm{TSVD}(k)}\|$$

are **nondecreasing** and **nonincreasing**, respectively, with k.

Simply, using SVD,

$$\|x^{\mathrm{TSVD}(k)}\|^2 = \sum\nolimits_{j=1}^{k} \frac{(u_j^T b)^2}{\sigma_j^2}$$

is nondecreasing with k;

$$\|b - Ax^{\mathrm{TSVD}(k)}\|^2 = \|(I - \Phi)U^T b\|^2 = \sum\nolimits_{j=k+1}^{N} \frac{(u_j^T b)^2}{\sigma_j^2}$$

is nonincreasing with k (here $\Phi = \mathrm{diag}(I_k, 0_{N-k})$).

5. Basic regularization techniques

Note on monotonicity (Tikhonov)

Similarly the **norms of the Tikhonov solution and the residual**

$$\xi(\lambda) \equiv \|x^{\mathrm{Tikhonov}(\lambda)}\|^2 = \sum\nolimits_{j=1}^{N} \phi_j^2 \frac{(u_j^T b)^2}{\sigma_j^2},$$

$$\rho(\lambda) \equiv \|b - Ax^{\mathrm{Tikhonov}(\lambda)}\|^2 = \sum\nolimits_{j=1}^{N} (1 - \phi_j)^2 (u_j^T b)^2$$

are **increasing** and **decreasing**, respectively, with λ.

Recall that $0 < \phi_j < 1$,

$$\phi_j = \frac{\sigma_j^2}{\sigma_j^2 + \lambda^2}, \qquad \text{thus} \qquad (1 - \phi_j) = \frac{\lambda^2}{\sigma_j^2 + \lambda^2}.$$

Look at

$$\xi'(\lambda) = \frac{d\xi(\lambda)}{d\lambda}, \qquad \rho'(\lambda) = \frac{d\rho(\lambda)}{d\lambda}.$$

5. Basic regularization techniques

Note on monotonicity (Tikhonov)

First

$$\frac{d}{d\lambda}\phi_j^2 = -\frac{4}{\lambda}(1 - \phi_j)\phi_j^2, \quad \frac{d}{d\lambda}(1 - \phi_j)^2 = \frac{4}{\lambda}(1 - \phi_j)^2 \phi_j.$$

Then

$$\xi'(\lambda) = -\frac{4}{\lambda}\sum_{j=1}^{N}(1 - \phi_j)\phi_j^2 \frac{(u_j^T b)^2}{\sigma_j^2},$$

$\xi'(\lambda) < 0$ for $\lambda > 0$, i.e. $\xi(\lambda)$ is decreasing with λ.

Analogously

$$\rho'(\lambda) = \frac{4}{\lambda}\sum_{j=1}^{N}(1 - \phi_j)^2 \phi_j (u_j^T b)^2,$$

$\rho'(\lambda) > 0$ for $\lambda > 0$, i.e. $\rho(\lambda)$ is increasing with λ.

6. Choosing regularization parameters

6. Choosing regularization parameters

Spectral filtering, Error analysis

In general

$$\begin{aligned} x^{\mathrm{filtered}} &= V\Phi\Sigma^{-1}U^T b \\ &= V\Phi\Sigma^{-1}U^T b^{\mathrm{exact}} + V\Phi\Sigma^{-1}U^T b^{\mathrm{noise}} \\ &= V\Phi\Sigma^{-1}U^T A x^{\mathrm{exact}} + V\Phi\Sigma^{-1}U^T b^{\mathrm{noise}} \\ &= (V\Phi V^T)x^{\mathrm{exact}} + V\Phi\Sigma^{-1}U^T b^{\mathrm{noise}}, \end{aligned}$$

where $V\Phi V^T$ is called the **resolution matrix**.

The **absolute error** is

$$x^{\mathrm{exact}} - x^{\mathrm{filtered}} = \underbrace{(I_N - V\Phi V^T)x^{\mathrm{exact}}}_{\text{regularization error}} - \underbrace{V\Phi\Sigma^{-1}U^T b^{\mathrm{noise}}}_{\text{perturbation error}},$$

regularization error is caused by using filtered inverse,
perturbation error consists of the inverted and filtered noise.

6. Choosing regularization parameters

Spectral filtering, Over- and undersmoothing

There is **no universal approach** for chosing the regularization parameter (k or λ), the choice is always **problem dependent!**
In general:

- If $\Phi \approx I_N$ ($V\Phi V^T \approx I_N$), the regularization error is small, but the perturbation error (caused by noise) is large.

  The solution is **undersmoothed**.
- If $\Phi \approx 0_N$ ($V\Phi V^T$ is far from the identity), inverted noise is heavily damped, but we lose a lot of original data.

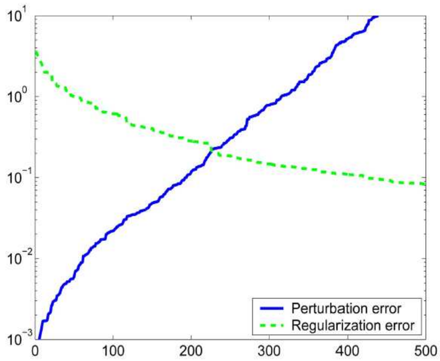
  The solution is **oversmoothed**.

A proper choice of the regularization parameter balances these two types of errors.

6. Choosing regularization parameters

Spectral filtering, A proper choice of the parameter

Regularization and perturbation error for TSVD method:



6. Choosing regularization parameters

Discrepancy principle

The **discrepancy principle**: Let

$$\|b^{\text{noise}}\| = \Delta^{\text{noise}}$$

be known either from the nature of the problem, or we have some **approximation** of it, see ⟨Lecture III⟩.

We look for a regularization parameter such that

$$\|b - Ax^{\text{filtered}}\| = \tau \Delta^{\text{noise}},$$

for some fixed τ.

Recall that for TSVD and Tikhonov regularization the norms of the residuals are **monotonic** in k and λ, respectively.

[Morozov: '66], [Morozov: '84].

6. Choosing regularization parameters

Generalized cross validation (GCV)

Using $x^{\text{filtered}} = V\Phi\Sigma^{-1}U^T b$ the residual satisfies

$$b - Ax^{\text{filtered}} = \left(I_N - AV\Phi\Sigma^{-1}U^T\right) b = \left(I_N - U\Phi U^T\right) b.$$

Defining the **generalized cross validation (GCV)** functional

$$G^{\text{filtered}}(\cdot) \equiv \frac{\|b - Ax^{\text{filtered}}\|^2}{\text{trace}(I_N - AV\Phi\Sigma^{-1}U^T)^2} = \frac{\|(I_N - \Phi)U^T b\|^2}{(N - \sum_{j=1}^N \phi_j)^2},$$

we look for its minimum.

(Note: The GCV functional depends on ordering of equations.)

[Chung, Nagy, O'Leary: '04], [Golub, Von Matt: '97], [Nguyen, Milanfar, Golub: '01].

6. Choosing regularization parameters

Generalized cross validation (GCV)

In particular for the TSVD and Tikhonov method we have

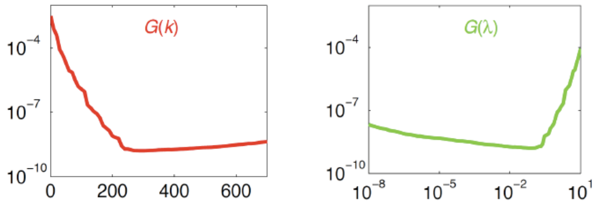
$$G^{\text{TSVD}}(k) = \frac{\sum_{j=k+1}^N (u_j^T b)^2}{(N-k)^2},$$

$$G^{\text{Tikhonov}}(\lambda) = \frac{\sum_{j=1}^N \left(\frac{u_j^T b}{\sigma_j^2 + \lambda^2}\right)^2}{\left(\sum_{j=1}^N \frac{1}{\sigma_j^2 + \lambda^2}\right)^2}.$$

6. Choosing regularization parameters

Generalized cross validation (GCV)

The GCV functional for TSVD (left) and Tikhonov (right) methods:



Note: The GCV functional is often flat close to the minimum.

6. Choosing regularization parameters

L-curve

Both norms

$$\|x^{\text{filtered}}\|, \qquad \|b - Ax^{\text{filtered}}\|$$

are monotonic with respect to the regularization parameter k, λ in TSVD and Tikhonov regularization, respectively.

We plot the norm of the regularized solution agains the norm of the residual. For emphasizing the point where both norms are **ballanced**, we use the **log-log** scale.

Criterion based on this approach is called the **L-curve**. The L-curve-optimal parameter then corresponds to the point with maximal curvature.

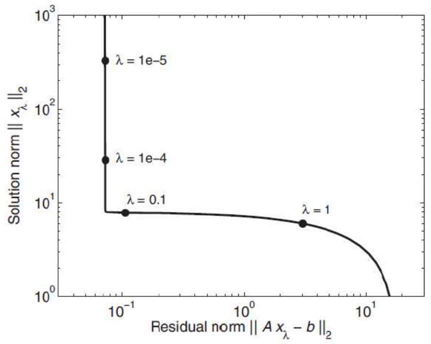
Note that for TSDV we have only discrete set of points (parameter k is discrete). The curvature is defined using an interpolation.

[Calvetti, Golub, Reichel: '99], [Calvetti, Morigi, Reichel, Sgallari: '00], [Calvetti, Reichel: '04].

6. Choosing regularization parameters

L-curve

Ideal L-curve for Tikhonov method (often the corner is not sharp). Here λ grows from the upper left to the bottom right corner along the curve:



6. Choosing regularization parameters

Normalized cumulative periodogram (NCP)

The last criterion is based on the assumption that the residual corresponding to the true solution

$$b^{\text{noise}} = b - Ax^{\text{exact}}$$

represents white noise. We try to choose a regularization parameter such that the residual

$$r^{\text{filtered}} = b - Ax^{\text{filtered}}$$

resembles white noise. See also ⟨Lecture III⟩.

The **normalized cumulative periodogram (NCP)** uses the statistical properties of Fourier spectrum of white noise.

[Rust: '98], [Rust: '00], [Rust, O'Leary: '08] (FFT-based), [Hansen, Kilmer, Kjeldsen: '06] (DCT-based).

6. Choosing regularization parameters

Normalized cumulative periodogram (NCP)

The NCP transforms the residual $r^{\text{filtered}} \in \mathbb{R}^N$ using the discrete Fourier transform (DFT/FFT algorithm) to get its spectrum

$$p^{\text{filtered}} = \mathcal{F}(r^{\text{filtered}}) = (p_1, p_2, \ldots, p_{\nu+1})^T, \qquad \nu = \lfloor N/2 \rfloor.$$

The periodogram is a vector C^{filtered} with coefficients

$$c_j = \frac{|p_2| + \ldots |p_{j+1}|}{|p_2| + \ldots |p_{\nu+1}|}, \qquad j = 1, \ldots, \nu.$$

If the residual consists only of white noise, then by the definiton of white noise the mean values satisfy

$$E[|p_2|] = E[|p_3|] = \ldots = E[|p_\nu|],$$

and by linearity of $E[\cdot]$, points $(j, E[c_j])$ would be on a straight line from $(0,0)$ to $(\nu, 1)$.

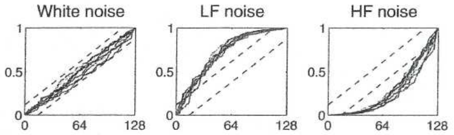
6. Choosing regularization parameters

Normalized cumulative periodogram (NCP)

Thus we look for the regularization parameter (k or λ) such that the coefficients of the periodogram c^{filtered} lie (moreorless) on a straight line,

$$\min_{k \text{ or } \lambda} \| C^{\text{filtered}} - C^{\text{white noise}} \|_2, \qquad C^{\text{white noise}} = \frac{1}{\nu}(1, 2, \ldots, \nu)^T.$$

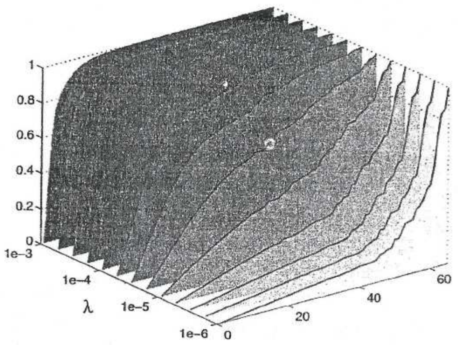
Note that the periodogram is normalized, i.e. $c_\nu = 1$.



6. Choosing regularization parameters

Normalized cumulative periodogram (NCP)

NCP for Tikhonov regularization:



[Hansen: SIAM, FA07, 2010].

6. Choosing regularization parameters

Further notes

Discrepancy principle: Converges as noise tends to zero, requires an explicite information about the norm of noise component of b, the solution tends to be oversmooth.

Generalized cross validation (GCV): No convergence when noise tends to zero, functional is flat close to the minimum, various adaptations for structured matrices (BCCB, etc.).

L-curve: No convergence when noise tends to zero, various adaptations (L-ribbon, etc.), well numericaly tracktable (it is sufficient to compute only a few points of the L-curve), troubles when using with TSVD because k is a discrete parameter.

Usually we need to solve one system with several values of the regularization parameter to choose the optimal one.

See also [Björk: '88], [Björk, Grimme, Van Dooren: '94].
For comparison see [Hansen: 98], [Kilmer, O'Leary: '01].

7. Iterative regularization

7. Iterative regularization

Introduction

Up to now we have considered **direct** regularization methods suitable for small problems (SVD-based methods, Tikhonov regularization leading to a LS problem which can be solved directly only in small dimensions).

For solving **large ill-posed problems**, it is advatagous to use **iterative regularization methods**. We briefly introduce several of them:

- stationary iterative methods (Landweber iteration, Cimmino iteration, Kaczmarz's method (ART)),
- projection methods (regularizing Krylov subspace iterations).

In all iterative methods the **number of iterations** plays the role of the **regularization parameter**.

7. Iterative regularization

Stationary iterative methods, Landweber iteration

Simultaneous iterative reconstruction techniques (SIRT) is a class of stationary iterative methods with a general scheme

$$x^{[\ell]} := x^{[\ell-1]} + \omega A^T M(b - Ax^{[\ell-1]}), \qquad \ell = 1, 2, \dots, k,$$

where M is a symmetric positive definite (SPD) matrix and ω is a relaxation parameter.

For example often used methods are:

- the **Landweber iteration** with $M = I_N$, and
- the **Cimminio iteration** with $M = D = \mathrm{diag}(d_1, \dots, d_N)$,

$$d_j = \frac{1}{N}\,\frac{1}{\|\underline{a}_j\|^2}\,,$$

where $\underline{a}_j$ is the (transposed) jth row of A (column vector).

7. Iterative regularization

Stationary iterative methods, Landweber iteration

The Landweber method

$$x^{[\ell]} := x^{[\ell-1]} + \omega A^T(b - Ax^{[\ell-1]}), \qquad \ell = 1, 2, \dots, k,$$

with $0 < \omega < 2\sigma_1^{-2}(A) = 2\|A^T A\|^{-1}$ gives the approximation

$$x^{[k]} = V\Phi^{[k]}\Sigma^{-1}U^T b, \qquad \Phi^{[k]} = I_N - (I_N - \omega\Sigma^2)^k,$$

i.e. $\phi_j^{[k]} = 1 - (1 - \omega\sigma_j^2)^k$.

Using the Taylor expansion for small σ_j's we get $\phi_j^{[k]} \approx k\omega\sigma_j^2$.

Thus the Landweber filters decay with the same rate as the Tikhonov filters ($\phi_j \approx \sigma_j^2\lambda^{-2}$).

7. Iterative regularization

Stationary iterative methods, Kaczmarz's method (ART)

Kaczmarz's method or **algebraic reconstruction technique (ART)** is an iterative algorithm given by the following scheme

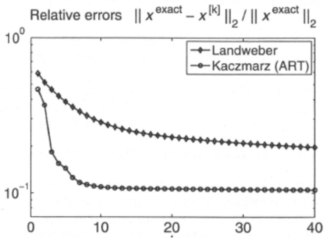
```
$x^{[\ell-1,0]} := x^{[\ell-1]}$,
for $j = 1, \dots, N$
    $x^{[\ell-1,j]} := x^{[\ell-1,j-1]} + \omega\,\underline{a}_j\,\frac{1}{\|\underline{a}_j\|^2}\,(b_j - \underline{a}_j^T x^{[\ell-1,j-1]})$,
end
$x^{[\ell]} := x^{[\ell-1,N]}, \qquad \ell = 1, 2, \dots, k.$
```

The ART method converges quite quickly in the first few iterations, after this the convergence may become very slow.

7. Iterative regularization

Stationary iterative methods, Kaczmarz's method (ART)

Comparison of relative error decay for Landweber and Kaczmarz's (ART) method:



[Hansen: SIAM, FA07, 2010].

7. Iterative regularization

Projection methods

In direct techniques we have looked for an approximation of x^{exact} which lies in a low dimensional subspace of $\mathbb{R}^N$ spanned by the first k right singular vectors of A (TSVD); or which is dominated by several first right singular vectors of A (Tikhonov).

Thus the approximation is always dominated by the low frequencies and the high frequecies are dumped.

We try to look for an approximation in an a-priori given low dimensional subspace $\mathcal{W}_k$ dominated by low frequencies.

7. Iterative regularization

Projection methods

Consider a subspace

$$\mathcal{W}_k = \mathrm{span}(w_1, \ldots, w_k) \subset \mathbb{R}^N, \qquad W_k = [w_1, \ldots, w_k] \in \mathbb{R}^{N \times k},$$

such that $W_k^T W_k = I_k$ and w_j are dominated by low frequecies. Then we solve

$$\min_{x \in \mathcal{W}_k} \|b - Ax\|.$$

This can be reformulated as a **projected problem**

$$\min_{y \in \mathbb{R}^k} \|b - (AW_k)y\|,$$

or, equivalently,

$$W_k^T (A^T A) W_k y = W_k^T A^T b.$$

The question is, how to choose the basis w_j?

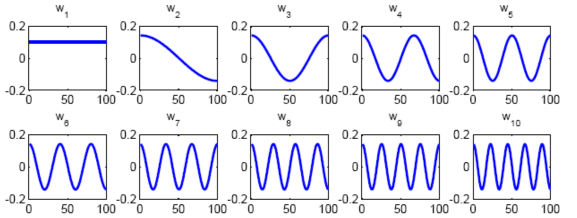
7. Iterative regularization

Projection methods, DCT basis

An example of a suitable basis is the DCT basis

$$w_1 = \sqrt{\tfrac{1}{N}}\,(1, 1, \ldots, 1)^T,$$
$$w_j = \sqrt{\tfrac{2}{N}}\left(\cos\left(\tfrac{(j-1)\pi}{2N}\right), \cos\left(\tfrac{3(j-1)\pi}{2N}\right), \ldots \cos\left(\tfrac{(2N-1)(j-1)\pi}{2N}\right)\right)^T,$$

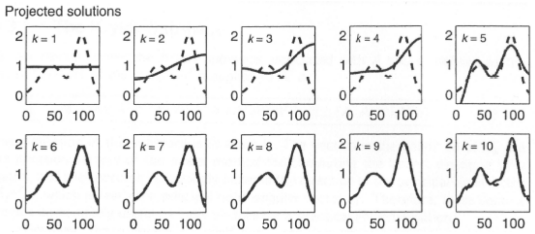
for $j > 1$.



7. Iterative regularization

Projection methods, DCT basis

Solutions computed using the DCT basis $w_1, \ldots, w_k$, $k = 1, \ldots, 10$ ($k = 9$ seems to be the optimal one):



Note: If there are a-priori known certain properties of the true solution (symmetry, periodicity, etc.), use this knowledge to choose basis vectors satisfying these properties.

7. Iterative regularization

Projection methods, Further notes

Note that choosing $w_j = v_j$ (the right singular vectors of A), we get exactly the TSVD mehtod. Thus TSVD is an projection method.

Advantage: With fixed set of basis vectors, computations can be performed quickly. Using e.g. DCT basis we do not have to compute and store the basis vectors (we compute only the DCT and the inverse DCT (IDCT) of a vector).

Disadvantage: The basis vectors are not adapted to the particular problem.

To avoid this disadvatage we introduce the regularizing Krylov subspace iteration.

7. Iterative regularization

Regularizing Krylov subspace iteration

Specific projection methods are the **Krylov subspace methods**. Here the orthonormal basis of a Krylov subspace

$$\mathcal{K}_k(v, M) = \mathrm{span}(v, Mv, \ldots, M^{k-1}v),$$

is used for w_j, $j = 1, \ldots, k$, vectors. For example the choice

$$v = A^T b, \quad M = A^T A,$$

leads to very popular iterative (regularization) methods **CGLS**, **LSQR** or **CGNE**, which are mathematically equivalent to applying CG method on the normal equations $A^T A x = A^T b$.

The regularizing properties of the Krylov subspace methods will be dicussed in ⟨Lecture III⟩ in more details, in particular in the context of hybrid methods.

7. Iterative regularization

Further remarks

In the iterative regularization (using stationary or projection methods), the number of computed iterations k plays the role of the regularization parameter.

As a stopping criterion for the iterative process any of the previously mentioned approaches can be used, e.g.:

- the discrepancy principle,
- the generalized cross validation (GCV),
- the L-curve criterion,
- the normalized cumulative periodograms (NCP).

8. Hybrid methods

The best of both worlds

8. Hybrid methods

Introduction

Hybrid methods combine both previous approaches. Here the regularization is realized in two steps.

First, the original problem is projeted onto a lower dimensional subspace using an iterative (projection) method, which by itself represents a form of regularization by projection, i.e. **outer regularization**.

The small projected problem, however, may inherit a part of the ill-posedness of the original problem and therefore some form of **inner regularization** is applied.

Stopping criteria for the whole process are then based on the regularization of the projected (small) problems.

[O'Leary, Simmons: '81], [Hansen: '98] or [Fiero, Golub, Hansen, O'Leary: '97], [Kilmer, O'Leary: '01], [Kilmer, Español: '06], [O'Leary, Simmnos: '81].

8. Hybrid methods

Projection methods with inner Tikhonov regularization

As an example we introduce the **Projection method with inner Tikhonov regularization**. Consider the ill-posed problem $Ax = b$ and a subspace $\mathcal{W}_k = \mathrm{span}(w_1, \ldots, w_k)$. Denote

$$M_k = W_k^T(A^TA)W_k \in \mathbb{R}^{k\times k}, \qquad \text{where} \quad W_k = [w_1, \ldots, w_k].$$

The system of normal equations $A^TAx = A^Tb$ is projected on $\mathcal{W}_k$,

$$M_k y = W_k^T b, \qquad x = W_k y.$$

The inner Tikhonov regularization can be applied on this small problem

$$y^{\mathrm{Tikhonov}(\lambda)} \;=\; \arg\min_y \{\|W_k^T b - M_k y\| + \lambda\|y\|\}.$$

This leads to a small LS problem that can be easily solved directly for **many** values of λ.

Summary

We have described the following regularization methods:

- the **direct regularization** techniques (TSVD, Tikhonov regularization) suitable for solving **small** ill-posed problems;
- **stationary** regularization methods (Landweber and Cimmino iterations, Kaczmarz's (ART) method);
- **projection** regularization methods including **regularizing Krylov subspace iterations**;
- **hybrid** methods combining the previous techniques.

All regularization techniques require to **choose a good regularization parameter**, that can be find using, e.g., the discrepancy principle, the generalized cross validation, the L-curve criterion, or the normalized cumulative periodograms.

Ill–Posed Inverse Problems in Image Processing

Introduction, Structured matrices, Spectral filtering, Regularization, Noise revealing

I. Hnětynková[1], M. Plešinger[2], Z. Strakoš[3]

hnetynko@karlin.mff.cuni.cz, martin.plesinger@sam.math.ethz.ch, strakos@cs.cas.cz

[1,3]Faculty of Mathematics and Phycics, Charles University, Prague
[2]Seminar of Applied Mathematics, Dept. of Math., ETH Zürich
[1,2,3]Institute of Computer Science, Academy of Sciences of the Czech Republic

SNA '11, January 24—28

Recapitulation of Lecture I and II

Linear system

Consider an ill-posed (square nonsingular) problem

$$Ax = b, \quad b = b^{\text{exact}} + b^{\text{noise}}, \quad A \in \mathbb{R}^{N \times N}, \quad x, b \in \mathbb{R}^N,$$

where

- A is a discretization of a smoothing operator,
- singular values of A decay,
- singular vectors of A represent increasing frequencies,
- b^{exact} is smooth and satisfies the discrete Pickard condition,
- b^{noise} is unknown white noise,

$$\|b^{\text{exact}}\| \gg \|b^{\text{noise}}\|, \qquad \text{but} \qquad \|A^{-1}b^{\text{exact}}\| \ll \|A^{-1}b^{\text{noise}}\|.$$

We want to approximate

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

Recapitulation of Lecture I and II

Linear system

Discrete Picard condition (DPC):

On average, the components $|(b^{\text{exact}}, u_j)|$ of the true right-hand side b^{exact} in the left singular subspaces of A decay faster than the singular values σ_j of A, $j = 1, \ldots, N$.

White noise:

The components $|(b^{\text{noise}}, u_j)|$, $j = 1, \ldots, N$ do not exhibit any trend.

Denote

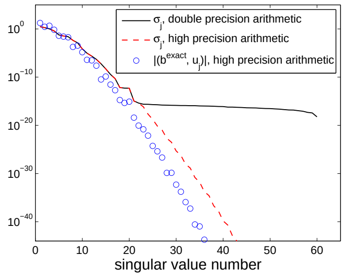
$$\delta^{\text{noise}} \equiv \frac{\|\, b^{\text{noise}} \,\|}{\|\, b^{\text{exact}} \,\|}$$

the **(usually unknown) noise level** in the data.

Recapitulation of Lecture I and II

Linear system

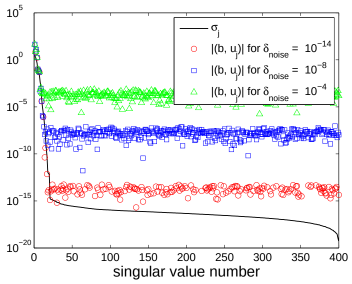
Singular values and DPC (SHAW(400)):



Recapitulation of Lecture I and II

Linear system

Violation of DPC for different noise levels (SHAW(400)):



Recapitulation of Lecture I and II

Naive solution

The components of the naive solution

$$x^{\text{naive}} \equiv A^{-1}b = \underbrace{\sum_{j=1}^{N} \frac{u_j^T b^{\text{exact}}}{\sigma_j} v_j}_{x^{\text{exact}}} + \underbrace{\sum_{j=1}^{N} \frac{u_j^T b^{\text{noise}}}{\sigma_j} v_j}_{\text{amplified noise}}$$

corresponding to small σ_j's are dominated by amplified noise.

Regularization is used to suppress the effect of errors and extract the essential information about the solution.

Recapitulation of Lecture I and II

Regularization methods

Direct regularization (TSVD, Tikhonov regularization): Suitable for solving small ill-posed problems.

Projection regularization: Suitable for solving large ill-posed problems. Regularization is often based on regularizing **Krylov subspace** iterations.

Hybrid methods: Here the **outer iterative regularization** is combined with an **inner direct regularization** of the projected small problem (i.e. of the reduced model).

The algorithm is stopped when the regularized solution of the **reduced model** matches some selected **stopping criteria** based, e.g., on the discrepancy principle, the generalized cross validation, the L-curve criterion, or the normalized cumulative periodograms.

Outline of the tutorial

- **Lecture I—Problem formulation:**

  Mathematical model of blurring, System of linear algebraic equations, Properties of the problem, Impact of noise.
- **Lecture II—Regularization:**

  Basic regularization techniques (TSVD, Tikhonov), Criteria for choosing regularization parameters, Iterative regularization, Hybrid methods.
- **Lecture III—Noise revealing:**

  Golub-Kahan iterative bidiagonalization and its properties, Propagation of noise, Determination of the noise level, Noise vector approximation, Open problems.

Outline of Lecture III

- **9. Golub-Kahan iterative bidiagonalization and its properties:**

  Basic algorithm, LSQR method, Connection with the Lanczos tridiagonalization, Approximation of the Riemann-Stieltjes distribution function.
- **10. Propagation of noise:**

  Motivation, Spectral properties of bidiagonalization vectors, Noise amplification.
- **11. Determination of the noise level:**

  Estimate based on distribution functions, Identification of the noise revealing iteration.
- **12. Noise vector approximation:**

  Basic formula, Noise subtraction, Numerical illustration (SHAW and ELEPHANT image deblurring problem).
- **13. Open problems.**

9. Golub-Kahan iterative bidiagonalization and its properties

9. Golub-Kahan iterative bidiagonalization and its properties

Basic algorithm

Golub-Kahan iterative bidiagonalization (GK) of A:

given $w_0 = 0\,,\ s_1 = b\,/\,\beta_1\,,$ where $\beta_1 = \|b\|\,,$ for $j = 1, 2, \ldots$

$$\begin{aligned} \alpha_j\, w_j &= A^T s_j - \beta_j\, w_{j-1}\,, & \|w_j\| = 1\,, \\ \beta_{j+1}\, s_{j+1} &= A\, w_j - \alpha_j\, s_j\,, & \|s_{j+1}\| = 1\,. \end{aligned}$$

Then $w_1, \ldots, w_k$ is an orthonormal basis of $\mathcal{K}_k(A^T A, A^T b)$, and $s_1, \ldots, s_k$ is an orthonormal basis of $\mathcal{K}_k(AA^T, b)$.

[Golub, Kahan: '65].

9. Golub-Kahan iterative bidiagonalization and its properties

Basic algorithm

Let $S_k = [s_1, \ldots, s_k]$, $W_k = [w_1, \ldots, w_k]$ be the associated matrices with orthonormal columns. Denote

$$L_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix}, \quad L_{k+} = \begin{bmatrix} L_k \\ e_k^T \beta_{k+1} \end{bmatrix}$$

the bidiagonal matrices containing the normalization coefficients.

Then GK can be written in the matrix form as

$$A^T S_k = W_k\, L_k^T\,,$$

$$A\, W_k = [\, S_k,\ s_{k+1}\,]\ L_{k+} \ = \ S_{k+1}\, L_{k+}\,.$$

9. Golub-Kahan iterative bidiagonalization and its properties

LSQR method

Regularization based on GK belong among popular approaches for solving **large ill-posed** problems. First the problem is **projected onto a Krylov subspace** using k steps of bidiagonalization (regularization by projection),

$$A\,x \approx b \;\longrightarrow\; L_{k+}\,y \approx \beta_1\,e_1\,.$$

Then, e.g., the **LSQR method** minimizes the residual,

$$\min_{x\in\mathcal{K}_k(A^TA,A^Tb)} \|Ax-b\| = \min_{y\in\mathbb{R}^k} \|L_{k+}y-\beta_1 e_1\|\,,$$

i.e. the approximation has the form $x_k = W_k y_k$, where y_k is a least squares solution of the projected problem, [Paige, Saunders: '82].

9. Golub-Kahan iterative bidiagonalization and its properties

LSQR method

In **hybrid methods**, some form of **inner regularization** (TSVD, Tikhonov regularization) is applied to the (small) projected problem. The method then, however, requires:

- stopping criteria for GK,
- parameter choice method for the inner regularization.

This usually requires solving the problem for **many values** of the regularization parameter and many iterations.

9. Golub-Kahan iterative bidiagonalization and its properties

Connection with the Lanczos tridiagonalization

GK is closely related to the **Lanczos tridiagonalization** [Lanczos: '50] of the symmetric matrix $A\,A^T$ with the starting vector $s_1 = b\,/\,\beta_1$,

$$A\,A^T\,S_k = S_k\,T_k + \alpha_k\,\beta_{k+1}\,s_{k+1}\,e_k^T\,,$$

where

$$T_k = L_k\,L_k^T = \begin{bmatrix} \alpha_1^2 & \alpha_1\,\beta_1 & & \\ \alpha_1\,\beta_1 & \alpha_2^2+\beta_2^2 & \ddots & \\ & \ddots & \ddots & \alpha_{k-1}\,\beta_k \\ & & \alpha_{k-1}\,\beta_k & \alpha_k^2+\beta_k^2 \end{bmatrix}.$$

9. Golub-Kahan iterative bidiagonalization and its properties

Connection with the Lanczos tridiagonalization

Consequently, the matrix L_k from GK represents a **Cholesky factor** of the symmetric tridiagonal matrix T_k from the Lanczos process, [Hnětynková, Strakoš: '07] and the references given there.

9. Golub-Kahan iterative bidiagonalization and its properties

Approximation of the Riemann-Stieltjes distribution function

Consider the **Riemann-Stieltjes distribution function** $\omega(\lambda)$ with the N points of increase associated with the given (SPD) matrix $B\in\mathbb{R}^{N\times N}$ and the normalized initial vector s.

The **Lanczos tridiagonalization** of B with the starting vector s generates at each step k a non-decreasing piecewise constant distribution function $\omega^{(k)}$, with the nodes being the (distinct) eigenvalues of the Lanczos matrix T_k and the weights $\omega_j^{(k)}$ being the squared first entries of the corresponding normalized eigenvectors, [Hestenes, Stiefel: '52].

9. Golub-Kahan iterative bidiagonalization and its properties

Approximation of the Riemann-Stieltjes distribution function

The distribution functions $\omega^{(k)}(\lambda)$, $k = 1, 2, \ldots$ represent Gauss-Christoffel quadrature (i.e. minimal partial realization) approximations of the distribution function $\omega(\lambda)$, [Hestenes, Stiefel: '52], [Fischer: '96], [Meurant, Strakoš: '06].

9. Golub-Kahan iterative bidiagonalization and its properties

Approximation of the Riemann-Stieltjes distribution function

Consider the SVD

$$L_k = P_k \, \Theta_k \, Q_k{}^T ,$$

$P_k = [p_1^{(k)}, \ldots, p_k^{(k)}] , \; Q_k = [q_1^{(k)}, \ldots, q_k^{(k)}] ,$
$\Theta_k = \mathrm{diag}\,(\theta_1^{(k)}, \ldots, \theta_n^{(k)}) ,$
with the singular values ordered in the **increasing** order,

$$0 < \theta_1^{(k)} < \ldots < \theta_k^{(k)} .$$

Then $T_k = L_k L_k^T = P_k \Theta_k^2 P_k^T$ is the spectral decomposition of T_k ,

$(\theta_\ell^{(k)})^2$ are its **eigenvalues** (the Ritz values of AA^T) and

$p_\ell^{(k)}$ its **eigenvectors** (which determine the Ritz vectors of AA^T), $\ell = 1, \ldots, k$.

9. Golub-Kahan iterative bidiagonalization and its properties

Approximation of the Riemann-Stieltjes distribution function

Consequently, the GK bidiagonalization generates at each step k the distribution function

$$\omega^{(k)}(\lambda) \quad \text{with nodes} \quad (\theta_\ell^{(k)})^2 \quad \text{and weights} \quad \omega_\ell^{(k)} = |(p_\ell^{(k)}, e_1)|^2$$

that approximates the distribution function

$$\omega(\lambda) \quad \text{with nodes} \quad \sigma_j^2 \quad \text{and weights} \quad \omega_j = |(b/\beta_1, u_j)|^2 ,$$

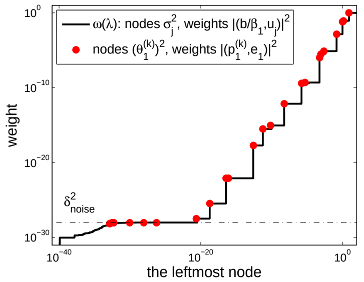
where $\sigma_j^2, \, u_j$ are the eigenpairs of $A A^T$, for $j = N, \ldots, 1$, [Hestenes, Stiefel: '52], [Fischer: '96], [Meurant, Strakoš: '06].

Note that unlike the Ritz values $(\theta_\ell^{(k)})^2$, the squared singular values σ_j^2 are enumerated in *descending* order.

9. Golub-Kahan iterative bidiagonalization and its properties

Approximation of the Riemann-Stieltjes distribution function

Discrete ill-posed problem, the smallest node and weight in approximation of $\omega(\lambda)$:



10. Propagation of noise

10. Propagation of noise

Motivation

If the noise level δ^{noise} in the data is known, many different approaches can be used for the stopping criterion in GK [Kilmer, O'Leary: '01], e.g., the discrepancy principle [Morozov: '66], [Morozov: '84], [Hansen: '98].

However, in most applications such apriory information is not available.

GK starts with the normalized **noisy right-hand side** $s_1 = b \, / \, \|b\|$. Consequently, vectors s_j contain information about the noise.

Can this information be used to determine the (unknown) noise level?

10. Propagation of noise

Spectral properties of bidiagonalization vectors

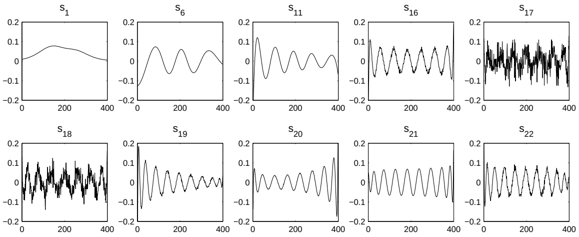
Consider the problem SHAW(400) from [Regularization Toolbox] with a noisy right-hand side (the noise was artificially added using the MatLab function `randn`). As an example we set

$$\delta^{\mathrm{noise}} \equiv \frac{\| \, b^{\mathrm{noise}} \, \|}{\| \, b^{\mathrm{exact}} \, \|} = 10^{-14} .$$

10. Propagation of noise

Spectral properties of bidiagonalization vectors

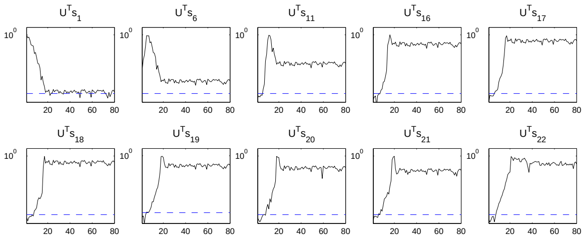
Components of several bidiagonalization vectors s_j computed via GK with double reorthogonalization:



10. Propagation of noise

Spectral properties of bidiagonalization vectors

The first 80 spectral coefficients of the vectors s_j in the basis of the left singular vectors u_j of A:



10. Propagation of noise

Spectral properties of bidiagonalization vectors

Using the three-term recurrences,

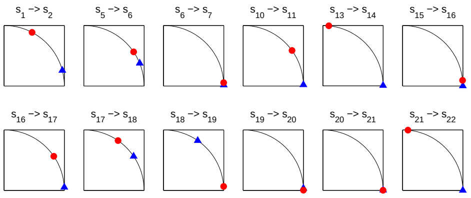
$$\beta_2\alpha_1\, s_2 \;=\; \alpha_1(\,Aw_1 - \alpha_1 s_1) \;=\; AA^T s_1 - \alpha_1^2 s_1 ,$$

where AA^T has smoothing property. The vector s_2 is a linear combination of s_1 contaminated by the noise and $A\,A^T\,s_1$ which is smooth. Therefore the contamination of s_1 by the **high frequency part** of the noise is transferred to s_2, while a portion of the smooth part of s_1 is subtracted by orthogonalization of s_2 against s_1. **The relative level of the high frequency part of noise in s_2 must be higher than in s_1.**
In subsequent vectors $s_3, s_4, \ldots$ the relative level of the high frequency part of noise gradually increases, until the low frequency information is projected out.

10. Propagation of noise

Spectral properties of bidiagonalization vectors

Signal space – noise space diagrams:



s_k (triangle) and s_{k+1} (circle) in the signal space $\mathrm{span}\{u_1, \ldots, u_{k+1}\}$ (horizontal axis) and the noise space $\mathrm{span}\{u_{k+2}, \ldots, u_n\}$ (vertical axis).

10. Propagation of noise

Noise amplification

Noise is amplified with the ratio α_k/β_{k+1}:

GK for the spectral components:

$$\begin{aligned}\alpha_1\,(V^T w_1) &= \Sigma\,(U^T s_1)\,,\\ \beta_2\,(U^T s_2) &= \Sigma\,(V^T w_1) - \alpha_1\,(U^T s_1)\,,\end{aligned}$$

and for $k = 2, 3, \ldots$

$$\begin{aligned}\alpha_k(V^T w_k) &= \Sigma\,(U^T s_k) - \beta_k(V^T w_{k-1})\,,\\ \beta_{k+1}(U^T s_{k+1}) &= \Sigma\,(V^T w_k) - \alpha_k(U^T s_k)\,.\end{aligned}$$

See [Hnětynková, Plešinger, Strakoš: '10] for a detailed derivation.

10. Propagation of noise

Noise amplification

Since dominance in $\Sigma(U^T s_k)$ and $(V^T w_{k-1})$ is shifted by one component, in $\alpha_k\,(V^T w_k) = \Sigma\,(U^T s_k) - \beta_k\,(V^T w_{k-1})$, one can not expect a significant cancellation, and therefore

$$\alpha_k \;\approx\; \beta_k\,.$$

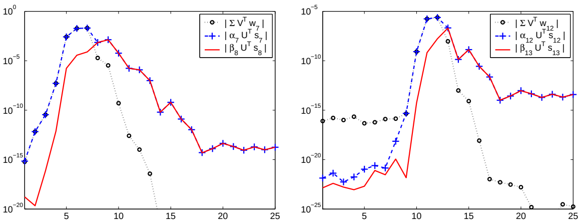
Whereas $\Sigma\,(V^T w_k)$ and $(U^T s_k)$ do exhibit dominance in the direction of the same components. If this dominance is strong enough, then the required orthogonality of s_{k+1} and s_k in $\beta_{k+1}\,(U^T s_{k+1}) = \Sigma\,(V^T w_k) - \alpha_k\,(U^T s_k)$ can not be achieved without a significant cancellation, and one can expect

$$\beta_{k+1} \;\ll\; \alpha_k\,.$$

10. Propagation of noise

Noise amplification

Absolute values of the first 25 components of $\Sigma(V^T w_k)$, $\alpha_k(U^T s_k)$, and $\beta_{k+1}(U^T s_{k+1})$ for $k = 7$ (left) and for $k = 12$ (right), SHAW(400) with the noise level $\delta_{\text{noise}} = 10^{-14}$:



11. Determination of the noise level

11. Determination of the noise level

Estimate based on distribution functions

Back to the distribution function:

The large nodes $\sigma_1^2, \sigma_2^2, \ldots$ of $\omega(\lambda)$ are well-separated (relatively to the small ones) and their weights on average decrease faster than σ_1^2, σ_2^2 due to the DPC. Therefore the **large nodes** essentially **control the behavior of the early stages of the Lanczos process.**

11. Determination of the noise level

Estimate based on distribution functions

Depending on the noise level, the weights corresponding to **smaller nodes** are completely dominated by noise, i.e., there exists an index J_{noise} such that

$$|(b/\beta_1, u_j)|^2 \approx |(b^{\text{noise}}/\beta_1, u_j)|^2 , \qquad \text{for } j \geq J_{\text{noise}} .$$

The **weight of the set of the associated nodes** is given by

$$\delta^2 \equiv \sum_{j=J_{\text{noise}}}^{n} |(b^{\text{noise}}/\beta_1, u_j)|^2 \approx \delta_{\text{noise}}^2 .$$

11. Determination of the noise level

Estimate based on distribution functions

At **any** iteration step, the weight of $\omega^{(k)}(\lambda)$ corresponding to the **smallest node** $(\theta_1^{(k)})^2$ must be larger than the sum of weights of all σ_j^2 smaller than this $(\theta_1^{(k)})^2$, see [Fischer, Freund: '94].

As k increases, some $(\theta_1^{(k)})^2$ eventually approaches (or becomes smaller than) the node $\sigma_{J_{\text{noise}}}^2$, and its weight becomes

$$|(p_1^{(k)}, e_1)|^2 \approx \delta^2 \approx \delta_{\text{noise}}^2 .$$

11. Determination of the noise level

Estimate based on distribution functions

Summarizing:

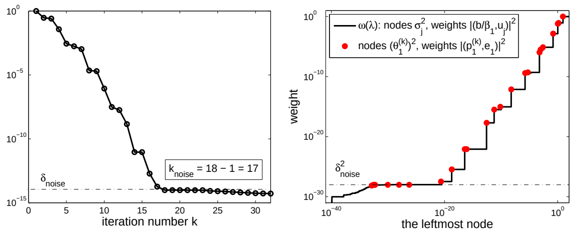
The weight $|(p_1^{(k)}, e_1)|^2$ corresponding to the smallest Ritz value $(\theta_1^{(k)})^2$ is strictly decreasing. At some iteration step it sharply **starts to (almost) stagnate close to the squared noise level** δ_{noise}^2, see [Hnětynková, Plešinger, Strakoš: '10].

The **last iteration before** this happens is called the **noise revealing iteration** k_{noise}.

11. Determination of the noise level

Estimate based on distribution functions

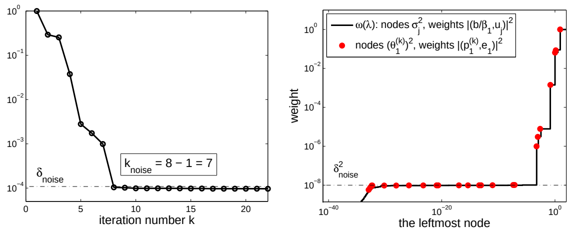
Square roots of the weights $|(p_1^{(k)}, e_1)|^2$, $k = 1, 2, \ldots$ (left), and the smallest node and weight in approximation of $\omega(\lambda)$ (right), SHAW(400) with the noise level $\delta_{\text{noise}} = 10^{-14}$:



11. Determination of the noise level

Estimate based on distribution functions

Square roots of the weights $|(p_1^{(k)}, e_1)|^2$, $k = 1, 2, \ldots$ (left), and the smallest node and weight in approximation of $\omega(\lambda)$ (right), SHAW(400) with the noise level $\delta_{\text{noise}} = 10^{-4}$:



11. Determination of the noise level

Identification of the noise revealing iteration

In order to estimate δ_{noise}, the iteration k_{noise} must be identified. This can be done by an **automated procedure** that does not rely on human interaction.

For example, in our experiments k_{noise} was determined as the first iteration for which

$$\frac{|(p_1^{(k+1)}, e_1)|}{|(p_1^{(k+1+step)}, e_1)|} < \left(\frac{|(p_1^{(k)}, e_1)|}{|(p_1^{(k+1)}, e_1)|} \right)^{\zeta},$$

where ζ was set to 0.5 and *step* was set to 3.

11. Determination of the noise level

Identification of the noise revealing iteration

Noise level δ_{noise} in the data, iteration k_{noise}, and the estimated noise level $|(p_1^{(k_{\text{noise}}+1)}, e_1)|$, for two problems from [Regularization Toolbox]. The estimates represent average values computed using 1000 randomly chosen vectors b^{noise}:

SHAW(400)				
δ_{noise}	1×10^{-14}	1×10^{-6}	1×10^{-4}	1×10^{-2}
k_{noise}	16	9	7	4
estimate	1.80×10^{-14}	1.31×10^{-6}	1.01×10^{-4}	1.03×10^{-2}
ILAPLACE(100,1)				
δ_{noise}	1×10^{-13}	1×10^{-7}	1×10^{-2}	1×10^{-1}
k_{noise}	22	15.30	6.02	2
estimate	9.12×10^{-14}	1.34×10^{-7}	1.02×10^{-2}	1.11×10^{-1}

12. Noise vector approximation

12. Noise vector approximation

Basic formula

In the noise revealing iteration

$$\delta_{\text{noise}} \approx |(p_1^{(k_{\text{noise}}+1)}, e_1)|,$$

and the bidiagonalization vector $s_{k_{\text{noise}}}$ **is fully dominated by the high frequency noise**. Thus

$$b^{\text{noise}} \approx \|b^{\text{noise}}\| \; s_{k_{\text{noise}}} \approx \beta_1 \, |(p_1^{(k_{\text{noise}}+1)}, e_1)| \, s_{k_{\text{noise}}},$$

represents an approximation of the unknown noise.

We can **subtract the reconstructed noise from the noisy observation vector** b. Hopefully, the noise level in the corrected system will be **lower** than in the original one.

What happens if we **repeat** this process several times?

12. Noise vector approximation

Noise subtraction

```
Algorithm:  Given $A$, $b$;  $b^{(0)} := b$;
for $j = 1, \dots, t$
  •  GK bidiagonalization of $A$ with the starting vector $b^{(j-1)}$;
  •  identification of the noise revealing iteration $k_{\rm noise}$;
  •  $\delta^{(j-1)} := |(p_1^{(k_{\rm noise})}, e_1)|$;
  •  $b^{{\rm noise},(j-1)} := \beta_1 \, \delta^{(j-1)} \, s_{k_{\rm noise}}$;      // noise approximation
  •  $b^{(j)} := b^{(j-1)} - b^{{\rm noise},(j-1)}$;                          // correction
end;
```

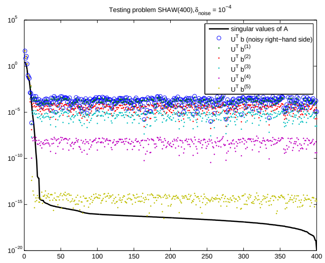
The **accumulated noise approximation** is

$$\hat{b}^{\rm noise} \equiv \sum_{j=0}^{t-1} b^{{\rm noise},(j)} \, .$$

12. Noise vector approximation

Numerical illustration - SHAW problem

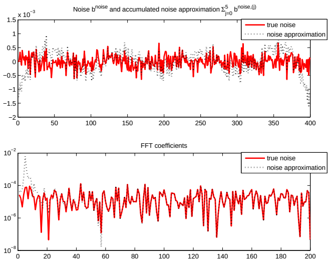
Singular values of A, and spectral coeffs. of the original and corrected observation vector $b^{(j)}$, $j = 1, \dots, 5$, SHAW(400) with the noise level $\delta_{\rm noise} = 10^{-4}$ ($k_{\rm noise} = 10$ is **fixed**):



12. Noise vector approximation

Numerical illustration - SHAW problem

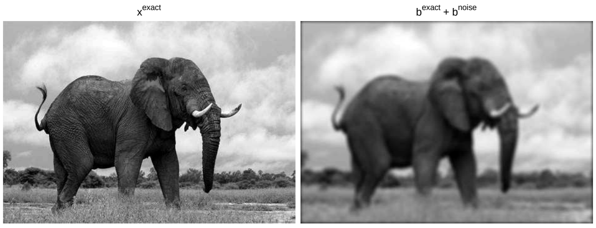
Individual components (top) and Fourier coeffs. (bottom) of $\hat{b}^{\rm noise}$, SHAW(400) with the noise level $\delta_{\rm noise} = 10^{-4}$:



12. Noise vector approximation

Numerical illustration - ELEPHANT image deblurring problem

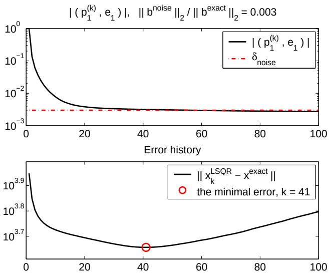
Elephant image deblurring problem: image size 324 × 470 pixels, problem dimension $N = 152280$, the exact solution (left) and the noisy right-hand side (right), $\delta_{\rm noise} = 3 \times 10^{-3}$:



12. Noise vector approximation

Numerical illustration - ELEPHANT image deblurring problem

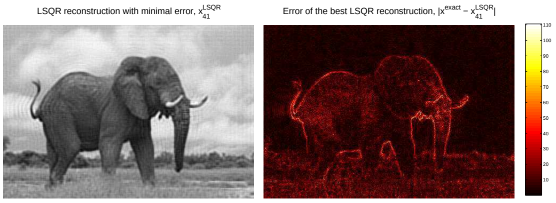
Square roots of the weights $|(p_1^{(k)}, e_1)|^2$, $k = 1, 2, \dots$ (top) and error history of LSQR solutions (bottom):



12. Noise vector approximation

Numerical illustration - ELEPHANT image deblurring problem

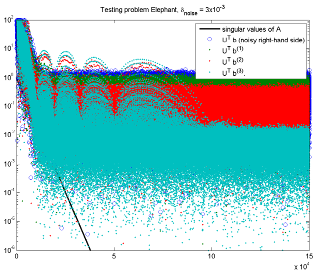
The best LSQR reconstruction (left), $x_{41}^{\rm LSQR}$, and the corresponding componentwise error (right). GK without any reorthogonalization:


LSQR reconstruction with minimal error, x_41^LSQR
Error of the best LSQR reconstruction, |x^exact − x_41^LSQR|
110
100
90
80
70
60
50
40
30
20
10

12. Noise vector approximation

Numerical illustration - ELEPHANT image deblurring problem

Singular values of A, and spectral coeffs. of the original and corrected observation vector $b^{(j)}$, $j = 1, \ldots, 3$, Elephant image deblurring problem with $\delta_{\text{noise}} = 3 \times 10^{-3}$ (k_{noise} corresponds to the best LSQR approximation of x):



13. Open problems

Message:

Using GK, information about the noise can be obtained in a straightforward and cheap way.

Open problems:

- Large scale problems (determining k_{noise});
- Behavior in finite precision arithmetic (GK without reorthogonalization);
- Regularization;
- Denoising;
- Colored noise.

Thank you for your kind attention!

Základy algebraického multigridu založeného na zhlazených agregacích

P. Vaněk

Západočeská univerzita v Plzni

Cílem přednášky je poskytnout posluchači základní informace o metodě zhlazených agregací. Přednáška obsahuje detailní popis algoritmu v jeho podobě vhodné pro řešení neskalárních eliptických problémů jako jsou problémy pružnosti a tenké pružnosti (desky a skořepiny). Výklad je držen v elementárních mezích. V závěru přednášky bude prezentován klíčový konvergenční výsledek o víceúrovňové metodě zhlazených agregací (bez důkazu).

Metoda zhlazených agregací

Tato přednáška se opírá o tyto výsledky:

[**1**] P. Vaněk, M. Brezina, J. Mandel *Convergence of Algebraic Multigrid Based on Smoothed Aggregations*Numer. Math. 88(2001), no. 3 pp. 559–579
[**2**] P. Vaněk, J. Mandel, M. Brezina *Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems* Computing 56(1996) pp. 179–196

- Metoda pro řešení soustav lineárních algebraických rovnic pro řešení okrajových úloh pro eliptické parciální diferenciální rovnice
- zhrubovací technika v algebraickém multigridu
- umožňuje řešení problémů na vysoce nestrukturovaných sítích
- vhodná pro neskalární problémy (elasticita, tenká elasticita)

Co je multigrid ?

- Řešíme soustavu
$$A\mathbf{x} = \mathbf{f}$$
se symetrickou positivně definitní maticí vzniklou diskretizací okrajové úlohy pro parciální diferenciální rovnici
- Metoda více sítí se odehrává ve dvou základních krocích:
  - přípravná fáze
  - iterace
- V přípravné fázi se vytváří systém prolongátorů I^l_{l+1} a hierarchie hrubých matic A_l,

- *$l := 1$ a $A_1 := A$,*
- *opakuj*
  - *zkonstruuj $I^l_{l+1} : \mathbb{R}^{n_{l+1}} \to \mathbb{R}^{n_l}$, $n_{l+1} < n_l$,*
  - *vypočti*
$$A_{l+1} = (I^l_{l+1})^T A_l I^l_{l+1}, \tag{0.1}$$
  - *$l \leftarrow l + 1$*
- *dokud A_l není dostatečně malá, aby umožňovala efektivní finitní řešení,*
- *$L := l$.*

iterace: jsou dány:

- produkty přípravné fáze
  - prolongátory I^l_{l+1}, $l = 1, \ldots, L-1$
  - hierarchie matic A_l, $l = 1, \ldots, L$, $A_1 = A$
- hladící iterační procedura
$$\mathbf{x}_l \leftarrow \mathcal{S}_l(\mathbf{x}_l, \mathbf{f}_l),\ \mathbf{x}_l, \mathbf{f}_l \in \mathbb{R}^{n_l}$$
- iterační parametry
  - ν_1: počet pre–smoothing hladicích kroků
  - ν_2: počet post–smoothing hladicích kroků
  - γ: parametr cyklu, $\gamma = 1$ nebo $\gamma = 2$

Algoritmus 1. *$\mathbf{x}_1 := \mathbf{x}, \mathbf{f}_1 := \mathbf{f}$ a $MG(\cdot,\cdot) := MG_1(\cdot,\cdot)$, kde $MG_l(\cdot,\cdot)$ je definováno takto:*

```
• pro $i = 1, \ldots, \nu_1$ proveď $\mathbf{x}_l \leftarrow \mathcal{S}_l(\mathbf{x}_l, \mathbf{f}_l)$,
• $\mathbf{d}_l = A_l\mathbf{x}_l - \mathbf{f}_l$,
• $\mathbf{d}_{l+1} = (I^l_{l+1})^T \mathbf{d}_l$,
• Je-li $l + 1 = L$, řeš soustavu $A_{l+1}\mathbf{v} = \mathbf{d}_{l+1}$, $\mathbf{v} \in \mathbb{R}^{n_{l+1}}$, finitně, jinak
    – polož $\mathbf{v} = 0$,
    – pro $i = 1, \ldots, \gamma$ proveď $\mathbf{v} \leftarrow MG_{l+1}(\mathbf{v}, \mathbf{d}_{l+1})$
• $\mathbf{x}_l \leftarrow \mathbf{x}_l - I^l_{l+1}\mathbf{v}$,
• pro $i = 1, \ldots, \nu_2$ proveď $\mathbf{x}_l \leftarrow \mathcal{S}_l(\mathbf{x}_l, \mathbf{f}_l)$.
```

Základní informace o konvegenční teorii metody více sítí [BPWX]:

Nejprve definujeme
$$I^1_l = I^1_2 \ldots I^{l-1}_l, \quad I^1_1 = I.$$

Dále definujme hierarchii hrubých prostorů s normou a skalárním součinem
$$U_l = \text{ Range } (I^1_l)$$
$$(\cdot,\cdot)_l : I^1_l\mathbf{x}, I^1_l\mathbf{y} \mapsto \sum_{i=1}^{n_l} x_i y_i,$$
$$\|\cdot\|_l = (\cdot,\cdot)^{1/2}_l.$$

- Přirozenou bází prostoru $U_l =$ Range (I^1_l) jsou slupce matice I^1_l.
- V algoritmu počítáme s reprezentacemi vektorů $I^1_l\mathbf{x} \in U_l$ vzhledem k bázi dané sloupci I^1_l, tedy vektory $\mathbf{x}$.
- Normou vektoru $I^1_l\mathbf{x} \in U_l$ je Eukleidovská norma vektoru $\mathbf{x}$, tedy Eukleidovská norma reprezentace vektoru $I^1_l\mathbf{x}$ vzhledem k sloupcům matice I^1_l.
- Skalárním součinem vektorů $I^1_l\mathbf{x}, I^1_l\mathbf{y} \in U_l$ je Eukleidovský skalární součin vektorů $\mathbf{x}, \mathbf{y}$, tedy Eukleidovský skalární součin reprezentací vektorů $I^1_l\mathbf{x}, I^1_l\mathbf{y}$ vzhledem k sloupcům matice I^1_l.

THEOREM 0.1. *Předpokládáme existenci lineárních zobrazení*

$$Q_l,\ l = 1, \dots, L,\ Q_1 = I,$$

takových, že

$$\|(Q_l - Q_{l+1})\mathbf{u}\|_l^2 \le \frac{C_1}{\varrho(A_l)}\|\mathbf{u}\|_A^2 \quad \forall \mathbf{u} \in U_1,\ l = 1, \dots, L-1 \tag{0.2}$$

a

$$\|Q_l\|_A \le C_2 \quad \forall l = 1, \dots, L. \tag{0.3}$$

Dále uvažujeme hladiče ve tvaru

$$\mathcal{S}_l(\mathbf{x}_l, \mathbf{f}_l) = (I - R_l A_l)\mathbf{x}_l + R_l \mathbf{f}_l,$$

kde R_l jsou symetrické pozitivně semidefinitní matice takové, že matice $I - R_l A_l$ jsou A_l-symetrické pozitivně semidefinitní a

$$C_R (R_l \mathbf{u}, \mathbf{u})_{\mathbb{R}^l} \ge \frac{\|\mathbf{u}\|_{\mathbb{R}^{n_l}}^2}{\varrho(A_l)} \quad \forall \mathbf{u} \in \mathbb{R}^{n_l}, l = 1, \dots, L-1, \tag{0.4}$$

kde $\|\cdot\|_{\mathbb{R}^{n_l}}, (\cdot,\cdot)_{\mathbb{R}^{n_l}}$ značí Eukleidovskou normu a skalární součin v $\mathbb{R}^{n_l}$. Potom pro operátor šíření chyby E metody více sítí platí

$$\|E\|_A \le 1 - \frac{1}{CL}, \quad C = \left(1 + C_2^{1/2} + (C_R C_1)^{1/2}\right)^2.$$

POZNÁMKA 0.2. *Z definice prostoru U_l plyne, že operátor Q_l je možno psát ve tvaru*

$$Q_l = I_l^1 \tilde{Q}_l, \quad \tilde{Q}_l : U_1 \to \mathbb{R}^{n_l}.$$

Tato skutečnost, rovnost $A_l = (I_l^1)^T A I_l^1$, $Q_1 = I$ a rozklad

$$\begin{aligned} Q_l &= Q_l - Q_{l-1} + Q_{l-1} - Q_{l-2} + \dots + Q_2 - Q_1 + Q_1 \\ &= \sum_{j=1}^{l-1}(Q_{j+1} - Q_j) + Q_1 \end{aligned}$$

nám umožňuje odhadovat

$$\begin{aligned} \|Q_l\mathbf{u}\|_A &= \|\sum_{j=1}^{l-1}(Q_{j+1} - Q_j)\mathbf{u} + Q_1\mathbf{u}\|_A \\ &\le \sum_{j=1}^{l-1} \|(Q_j - Q_{j+1})\mathbf{u}\|_A + \|Q_1\mathbf{u}\|_A \\ &= \sum_{j=1}^{l-1} \|(I_j^1\tilde{Q}_j - I_{j+1}^1\tilde{Q}_{j+1})\mathbf{u}\|_A + \|\mathbf{u}\|_A \\ &= \sum_{j=1}^{l-1} \|I_j^1(\tilde{Q}_j - I_{j+1}^j\tilde{Q}_{j+1})\mathbf{u}\|_A + \|\mathbf{u}\|_A \\ &= \sum_{j=1}^{l-1} \|(\tilde{Q}_j - I_{j+1}^j\tilde{Q}_{j+1})\mathbf{u}\|_{A_j} + \|\mathbf{u}\|_A \\ &\le \sum_{j=1}^{l-1} \sqrt{\varrho(A_j)}\|(\tilde{Q}_j - I_{j+1}^j\tilde{Q}_{j+1})\mathbf{u}\|_{\mathbb{R}^{n_j}} + \|\mathbf{u}\|_A \\ &= \sum_{j=1}^{l-1} \sqrt{\varrho(A_j)}\|I_j^1(\tilde{Q}_j - I_{j+1}^j\tilde{Q}_{j+1})\mathbf{u}\|_j + \|\mathbf{u}\|_A \\ &= \sum_{j=1}^{l-1} \sqrt{\varrho(A_j)}\|(Q_j - Q_{j+1})\mathbf{u}\|_j + \|\mathbf{u}\|_A. \end{aligned}$$

Výše uvedený odhad spolu s aproximační podmínkou (0.2) dává

$$\|Q_l\mathbf{u}\|_A \le \sum_{j=1}^{l-1} C_1^{1/2}\|\mathbf{u}\|_A + \|\mathbf{u}\|_A = (1 + C_1^{1/2}(l-1)\|\mathbf{u}\|_A.$$

Odtud vidíme, že podmínka (0.3) plyne z aproximační podmínky (0.2) s kvazioptimální konstantou. Aproximační podmínka (0.2) je tudíž podmínkou klíčovou. Z podmínky (0.2)

plyne, že při konstrukci hrubých prostorů je třeba sledovat dva cíle:

- konstruovat prolongátory tak, že levá strana (0.2) je co možná nejmenší (aproximace),
- a tak, že spektrální poloměry hrubých matic jsou tak malé, jak je jen možné, s cílem učinit aproximační podmínku (0.2) co nejslabší (nejsnažší splnit).

Aproximační podmínka (0.2):

$$\|(Q_l - Q_{l+1})\mathbf{u}\|_l^2 \le \frac{C_1}{\varrho(A_l)}\|\mathbf{u}\|_A^2$$

Metoda zhlazených agregací – základní koncept

Zde popíšeme metodu zhlazených agregací, tedy metodu, kde prolongátor je konstruován ve tvaru

$$I_{l+1}^l = S_l P_{l+1}^l,$$

kde

- S_l je polynom v A_l volený tak, aby
$$\varrho(A_{l+1}) = \varrho((I_{l+1}^l)^T A_l I_{l+1}^l) = \varrho((S_l P_{l+1}^l)^T A_l S_l P_{l+1}^l)$$
byl co možná nejmenší a
- P_{l+1}^l je ortogonální matice vytvořená metodou zobecněných agregací. Jejím úkolem je zajistit aproximaci.

Jak jsme již řekli, ve snaze splnit klíčovou podmínku konvergenční věty (0.2) t.j.

$$\|(Q_l - Q_{l+1})\mathbf{u}\|_l^2 \le \frac{C_a}{\varrho(A_l)}\|\mathbf{u}\|_A^2,$$
$$\|\cdot\|_l : I_l^1\mathbf{x} \mapsto (\mathbf{x}^T\mathbf{x})^{1/2}.$$

usilujeme o dvě věci:

- minimalizovat levou stranu aproximační podmínky (aproximace),
- minimalizovat $\varrho(A_l)$, $l = 2, \dots, L$, a tím učinit aproximační podmínku co nejslabší (nejsnažší splnit).

Takže,

- P_{l+1}^l má za úkol minimalizovat levou stranu (0.2)
- *prolongátorový hladič* S_l má za úkol minimalizovat $\varrho(A_l)$.

Vyložme nyní efekt hlazení prolongátoru. Protože P^l_{l+1} je ortogonální matice, je

$$\|\mathbf{x}\|_{\mathbb{R}^{n_{l+1}}} = \|P^l_{l+1}\mathbf{x}\|_{\mathbb{R}^{n_l}} \quad \forall \mathbf{x} \in \mathbb{R}^{n_{l+1}},$$

a můžeme odhadovat

$$\begin{aligned}
\varrho(A_{l+1}) &= \max_{\mathbf{x}\in\mathbb{R}^{n_{l+1}}} \frac{\left((I^l_{l+1})^T A_l I^l_{l+1}\mathbf{x}, \mathbf{x}\right)_{\mathbb{R}^{n_{l+1}}}}{\|\mathbf{x}\|^2_{\mathbb{R}_{n_{l+1}}}} \\
&= \max_{\mathbf{x}\in\mathbb{R}^{n_{l+1}}} \frac{\left((S_l P^l_{l+1})^T A_l S_l P^l_{l+1}\mathbf{x}, \mathbf{x}\right)_{\mathbb{R}^{n_{l+1}}}}{\|\mathbf{x}\|^2_{\mathbb{R}_{n_{l+1}}}} \\
&= \max_{\mathbf{x}\in\mathbb{R}^{n_{l+1}}} \frac{\left(S_l^T A_l S_l P^l_{l+1}\mathbf{x}, P^l_{l+1}\mathbf{x}\right)_{\mathbb{R}^{n_l}}}{\|P^l_{l+1}\mathbf{x}\|^2_{\mathbb{R}^{n_l}}} \\
&= \max_{\mathbf{x}\in \text{ Range } P^l_{l+1}} \frac{\left(S_l^T A_l S_l \mathbf{x}, \mathbf{x}\right)_{\mathbb{R}^{n_l}}}{\|\mathbf{x}\|^2_{\mathbb{R}^{n_l}}} \\
&\leq \max_{\mathbf{x}\in\mathbb{R}^{n_l}} \frac{\left(S_l^T A_l S_l \mathbf{x}, \mathbf{x}\right)_{\mathbb{R}^{n_l}}}{\|\mathbf{x}\|^2_{\mathbb{R}^{n_l}}} \\
&= \varrho(S_l^T A_l S_l).
\end{aligned}$$

Závěr:

- $\varrho(A_{l+1}) \leq \varrho(S_l^T A_l S_l)$, takže S_l volíme tak, abychom minimalizovali $\varrho(S_l^T A_l S_l)$.
- Jako S_l volíme polynom v A_l minimalizující

$$\varrho(S_l^T A_l S_l) = \varrho(S_l^2 A_l).$$

Jako prolongátorový hladič volíme polynom v A_l

$$S_l = I - \frac{4}{3}\frac{1}{\bar{\lambda}_l} A_l, \quad \bar{\lambda}_l \geq \varrho(A_l). \tag{0.5}$$

Tuto specifickou volbu zdůvodníme za chvíli. Pro $\varrho(A_{l+1})$ máme odhad

$$\begin{aligned}
\varrho(A_{l+1}) &\leq \varrho(S_l^T A_l S_l) = \varrho(S_l^2 A_l) = \max_{t\in\sigma(A_l)} t\left(1 - \frac{4}{3}\frac{1}{\bar{\lambda}_l} t\right)^2 \\
&\leq \max_{t\in[0,\bar{\lambda}_l]} t\left(1 - \frac{4}{3}\frac{1}{\bar{\lambda}_l} t\right)^2 = \frac{1}{9}\bar{\lambda}_l,
\end{aligned}$$

takže za $\bar{\lambda}_{l+1} \geq \varrho(A_{l+1})$ můžeme vzít

$$\bar{\lambda}_{l+1} = \frac{1}{9}\bar{\lambda}_l. \tag{0.6}$$

Důvodem volby prolongátorového hladiče (0.5) je skutečnost, že

$$\min_{\omega\in\mathbb{R}} \max_{t\in[0,\bar{\lambda}_l]} t\left(1 - \omega\frac{1}{\bar{\lambda}_l} t\right)^2 = \max_{t\in[0,\bar{\lambda}_l]} t\left(1 - \frac{4}{3}\frac{1}{\bar{\lambda}_l} t\right)^2.$$

Metoda zobecněných agregací

Standardní agregace:

- Nejjednodušší prolongátor založený na agregacích pro jednodimenzionální příklad
- P^1_2 pro jednodimenzionální Laplaceovu rovnici diskretizované na pravidelné síti sestávající z $n_1 = 3n_2$ nodů
- Máme agregáty stupňů volnosti

$$\{1,2,3\},\ \{4,5,6\},\ldots,\{n_1-2, n_1-1, n_1\}.$$

- Sloupce P^1_2 definujeme jako restrikce vektoru jedniček na příslušné agregáty:

$$P^1_2 = \begin{pmatrix}
1 & & \cdot & \\
1 & & \cdot & \\
1 & & \cdot & \\
 & 1 & \cdot & \\
 & 1 & \cdot & \\
 & 1 & \cdot & \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
 & & \cdot & 1 \\
 & & \cdot & 1 \\
 & & \cdot & 1
\end{pmatrix}.$$

- Operátor P^1_2 odpovídá disagregaci dané agregáty stupňů volnosti

$$\{1,2,3\},\ \{4,5,6\},\ldots,\{n_1-2, n_1-1, n_1\}.$$

- Sloupce P^1_2 sestávající z 0–1 vektorů s disjunktní nenulovou strukturou
- P^1_2 odpovídá diskrétní po částech konstantní interpolaci

Metoda zobecněných agregací

- Naším cílem je vytvořit hierarchii pomocných prolongátorů P^l_{l+1} takových, že pro danou $n_1 \times r$ matici B^1

$$\text{Range } B^1 \subset \text{ Range } P^1_l, \quad P^1_l = P^1_2 \ldots P^{l-1}_l \qquad l = 1,\ldots,L-1. \tag{0.7}$$

- Obor hodnot matice B^1 specifikuje, které funkce (vektory na nejjemnější úrovni) budou přesně reprezentovány na všech úrovních. Podobně jako v [2], volíme B^1 jako generátor módů s nulovou energií, tedy kernel matice tuhosti bez esenciálních okrajových podmínek.
- Módy s nulovou energií získané z geometrie a definice elementů jsou dostupné ve většině konečněprvkových řešičů.
- Předpokládáme že máme matice $P^1_2,\ldots,P^{l-1}_l$ a B^l takové, že

$$P^1_l B^l = B^1.$$

Abychom splnili (0.7), tvoříme souběžně P^l_{l+1} a $n_{l+1}\times r$ matici B^{l+1} tak, že

$$P^l_{l+1} B^{l+1} = B^l, \tag{0.8}$$

kde B^l bylo vytvořeno spolu s P^{l-1}_l (a dáno na úrovni $l = 1$). Tím je zaručeno, že

$$P^1_{l+1} B^{l+1} = B^1.$$

Agregáty:

- Naše konstrukce je založena na agregaci supernodů. Na každé úrovni, stupně volnosti jsou organizovány v malých disjunktních množinách zvaných supernody (supernody tvoří disjunktní pokrytí množiny všech stupňů volnosti.) Na nejjemnější úrovni supernody musí být specifikovány, například jako množiny stupňů volnosti odpovídající konečněprvkovým vertexům. Na hrubších úrovních jsou supernody definovány naším algoritmem.
- Prolongátor P^l_{l+1} je konstruován z daného systému agregátů $\{\mathcal{A}^l_i\}_{i=1}^{N_l}$, které tvoří disjunktní pokrytí supernodů na úrovni l.
- Agregáty jsou malé množiny supernodů, které tvoří disjunktní pokrytí množiny všech supernodů. V ideálním případě jsou agregáty tvořeny jako nodální okolí vybraných supernodů

  $$\mathcal{N}(i) = \{j : A_{ij} \neq 0\}\,,$$

  kde i, j jsou supernody a A_{ij} je blok matice A_l odpovídající supernodům i, j.

- V praxi je mnohdy nemožné vytvořit disjunktní prokrytí z nodálních okolí, proto jsou agregáty obohacovány supernody

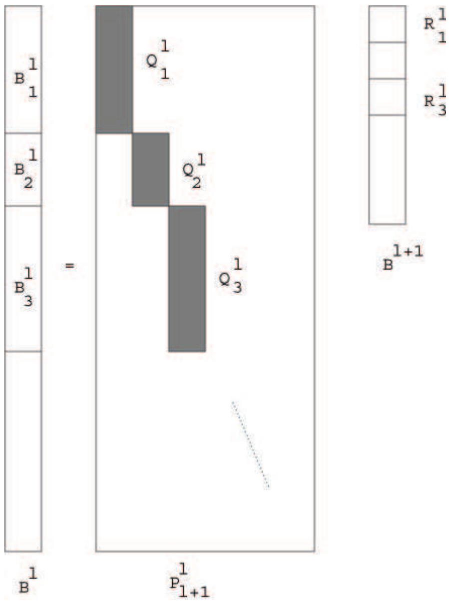
  $$k : A_{kj} \neq 0 \quad \text{pro nějaké } j \in \mathcal{N}(i).$$

- Algoritmus tvorby agregátů lze v hrubých rysech popsat takto:

  ALGORITMUS 2.

```
– polož $k = 1$
– Definuj $C$ jako množinu všech supernodů na úrovni $l$.
– Pro všechny supernody $i \in C$
    ∗ Je-li $\mathcal{N}(i) \subset \mathcal{C}$, polož $\mathcal{A}^l_k = \mathcal{N}(i)$, $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{N}(i)$ a $k \leftarrow k + 1$.
– Pro všechny supernody $i \in C$
    ∗ Najdi agregát $\mathcal{A}^l_k$ jehož supernody $j \in \mathcal{A}^l_k$ jsou se supernodem $i$ vázány největšími bloky $A_{ij}$ a polož $\mathcal{A}^l_k \leftarrow \mathcal{A}^l_k \cup \{i\}$.
```

- Vlastnost (0.8) je vynucována agregát po agregátu; sloupce P^l_{l+1} odpovídající agregátu $\mathcal{A}^l_i$ jsou tvořeny ortonormalizovanými restrikcemi sloupců B^l na agregát $\mathcal{A}^l_i$. Pro každý agregát tato konstrukce dá vznik r stupňům volnosti na hrubé úrovni, které tvoří supernode. Každý agregát na úrovni l dá tudíž vznik jednomu supernodu na úrovni $l + 1$.



OBRÁZEK 0.1. *Pomocný prolongátor založený na zobecněných agregacích.*

ALGORITMUS 3. *Pro daný systém agregátů $\{\mathcal{A}^l_i\}_{i=1}^{N_l}$ a $n_l \times r$ matici B^l splňující $P^1_l B^l = B^1$, vytvoříme prolongátor P^l_{l+1}, matici B^{l+1} splňující (0.8) a supernody na úrovni $l + 1$ následovně:*

1. *Nechť d_i značí počet stupňů volnosti odpovídající agregátu $\mathcal{A}^l_i$. Rodzěl $n_l \times r$ matici B^l do $d_i \times r$ bloků B^l_i, $i = 1, \dots, N_l$, z nichž každý odpovídá množině stupňů volnosti agregátu $\mathcal{A}^l_i$ (viz Obr. 0.1).*
2. *Rozlož $B^l_i = Q^l_i R^l_i$, kde Q^l_i je $d_i \times r$ ortogonální matice a R^l_i je $r \times r$ horní trojúhelníková matice.*
3. *Polož $P^l_{l+1} = \operatorname{diag}(Q^l_i)$, a (viz Obr. 0.1)*

   $$B^{l+1} = \begin{pmatrix} R^l_1 \\ R^l_2 \\ \dots \\ R^l_{N_l} \end{pmatrix}.$$

4. *Pro každý agregát $\mathcal{A}^l_i$, zhrubování dává vznik r stupňů volnosti na hrubé úrovni ($i-$tý blokový sloupec P^l_{l+1}). Tyto stupně volnosti definují $i-$tý supernode na hrubé úrovni.*

Konvergence metody zhlazených agregací

kompozitní agregáty

- Kompozitní agregát $\tilde{\mathcal{A}}_i^l$ je agregát $\mathcal{A}_i^l$ chápaný jako množina stupňů volnosti na nejjemější úrovni
- Formálně je možno kompositní agregáty zavést takto:

$$\tilde{\mathcal{A}}_i^l = \mathcal{A}_i^{l,1}, \quad \text{kde} \quad \mathcal{A}_i^{l,l} = \mathcal{A}_i^l, \qquad \mathcal{A}_i^{l,j-1} = \bigcup_{k \in \mathcal{A}_i^{l,j}} \mathcal{A}_k^{j-1}.$$

- Alternativní způsob definice kompozitních agregátů:

$$\tilde{\mathcal{A}}_i^l = \text{supp } P_l^1 \chi(\mathcal{A}_i^l)$$

THEOREM 0.3. *Nechť prolongátorové hladiče S_l jsou dány formulí*

$$S_l = I - \frac{4}{3}\frac{1}{\bar{\lambda}_l} A_l,$$

kde

$$\bar{\lambda}_l = \frac{1}{9^{l-1}}\bar{\lambda}, \ \bar{\lambda} \geq \varrho(A).$$

Předpokládáme že $C_1 > 0$ je konstanta taková, že existují lineární zobrazení

$$\tilde{Q}_l : \mathbb{R}^{n_1} \to \mathbb{R}^{n_l}, \quad l = 1, \ldots, L, \quad \tilde{Q}_1 = I,$$

taková, že

$$(0.9) \qquad \|P_l^1 \tilde{Q}_l \mathbf{u} - P_{l+1}^1 \tilde{Q}_{l+1} \mathbf{u}\|_{\mathbb{R}^{n_1}}^2 \leq C_1^2 \frac{9^{l-1}}{\bar{\lambda}} \|\mathbf{u}\|_A^2$$
$$\forall \mathbf{u} \in \mathbb{R}^{n_1}, \ l = 1, \ldots, L-1.$$

Dále předpokládáme že R_l je symetrická pozitivně definitní matice splňující (0.4) s konstantou $c_R > 0$ nezávislou na úrovni.

Potom

$$\|\hat{\mathbf{x}} - MG(\mathbf{x}, \mathbf{b})\|_A \leq \left(1 - \frac{1}{c_0}\right) \|\hat{\mathbf{x}} - \mathbf{x}\|_A \quad \forall \mathbf{x} \in \mathbb{R}^{n_1},$$

kde $A\hat{\mathbf{x}} = \mathbf{b}$, a

$$c_0 = \left(2 + C_1 c_R + \frac{4}{3}c_R + \frac{1}{3}C_1\left(1 + \frac{4}{3}c_R\right)(L-1)\right)^2 (L-1)$$

Navíc, je–li $P : \mathbf{u} \mapsto MG(\mathbf{0}, \mathbf{u})$, pak P je symetrická matice a $\text{cond}(A, P) \leq c_0$.

THEOREM 0.4. *Nechť prolongátorový hladič S_l je dán formulí*

$$S_l = I - \frac{4}{3}\frac{1}{\bar{\lambda}_l} A_l,$$

kde

$$\bar{\lambda}_l = \frac{1}{9^{l-1}}\bar{\lambda}, \ \bar{\lambda} \geq \varrho(A)$$

a pomocný prolongátor P_{l+1}^l je vytvořen Algoritmem 3 pomocí $n_1 \times r$ matice B^1 a agregátů $\{\mathcal{A}_i^l\}_{i=1}^{N_l}$, $l = 1, \ldots, L-1$. Předpokládáme, že existuje konstanta $C_{\mathcal{A}} > 0$ taková, že pro každý vektor $\mathbf{u} \in \mathbb{R}^{n_1}$ a každé $l = 1, \ldots, L-1$ platí

$$(0.10) \qquad \sum_{i=1}^{N_l} \min_{\mathbf{w} \in \mathbb{R}^r} \|\mathbf{u} - B^1 \mathbf{w}\|_{l^2(\tilde{\mathcal{A}}_i^l)}^2 \leq C_{\mathcal{A}} \frac{9^{l-1}}{\bar{\lambda}} \|\mathbf{u}\|_A^2.$$

Dále předpokládáme že R_l je symetrická pozitivně definitní matice splňující (0.4) s konstantou $c_R > 0$ nezávislou na úrovni. Potom,

$$\|\hat{\mathbf{x}} - MG(\mathbf{x}, \mathbf{b})\|_A \leq \left(1 - \frac{1}{c_0}\right) \|\hat{\mathbf{x}} - \mathbf{x}\|_A \ \forall \mathbf{x} \in \mathbb{R}^{n_1},$$

kde $A\hat{\mathbf{x}} = \mathbf{b}$, a

$$c_0 = (2 + C_{\mathcal{A}} c_R + (4/3)c_R + (1/3)C_{\mathcal{A}}(1 + (4/3)c_R)(L-1))^2 (L-1).$$

Dále, pokud $P : \mathbf{u} \mapsto MG(\mathbf{0}, \mathbf{x})$, pak P je symetrická matice a $\text{cond}(A, P) \leq c_0$.

REFERENCE

[1] P. Vaněk, M. Brezina, J. Mandel *Convergence of Algebraic Multigrid Based on Smoothed Aggregations*Numer. Math. 88(2001), no. 3 pp. 559–579

[2] P. Vaněk, J. Mandel, M. Brezina *Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems* Computing 56(1996) pp. 179–196

Analysis of a rate-independent model of non-local damage and its numerical approximation

J. Zeman, A. Mielke, T.Roubíček

[1] Faculty of Civil Engineering, Czech Technical University in Prague
[2] Institut für Mathematik, Humboldt Universität zu Berlin
[3] Charles University in Prague & Institute of Thermomechanics AS CR, Prague

1 Introduction

Damage presents an inelastic load-induced response of solid bodies, which is typical of quasi-brittle materials. From the physical point of view, it is interpreted as a collective effect of microstructural failures, leading finally to the macroscopic collapse of the structure. Due to obvious reasons, the damage theories have received a great attention in the engineering literature and a considerable amount of theoretical, numerical and experimental work has been invested into understanding and prediction of damage processes. In this contribution, we present an overview of available results related to a specific non-local rate-independent isotropic damage model and its numerical treatment. The major difference of the current work and the existing approaches is the fact that the reported numerical simulations are supported by a number of rigorous mathematical results obtained recently in [1, 5, 7, 8].

2 The model setup

The common theoretical framework for both the analysis and numerics is provided by recent advances in the mathematical theory of rate-independent processes; see [4] for a review. In this setting, the state of a system is described by kinematics (displacement field $\boldsymbol{u}$) and an internal variables (damage level ζ). The time evolution of the system is then governed by the global minimization of total energy of the system, consisting of the globally stored and the dissipated energy specified later.

The global energy minimizer in space and times is then referred to as the *energetic solution* to the damage problem. Its existence for a specific damage model of the Frémond-Nedjar type [3] was proven in [1, 6, 8] under mild assumptions the problem data. In general, the procedure involves the introduction of the ϵ-regularized problem, preventing the complete disintegration of the material, and the semi-discretization in time. For a given partition of the time interval $0 = t_0 < t_1 + \tau \ldots < t_N = T$, the time-incremental problem reads as

$$(\boldsymbol{u}^\epsilon(t_k), \zeta^\epsilon(t_k)) \in \operatorname{Arg} \min_{(\widehat{\boldsymbol{u}}, \widehat{\zeta}) \in \mathbb{K} \times \mathbb{Z}} \left[\mathcal{E}^\epsilon(t_k, \widehat{\boldsymbol{u}}, \widehat{\zeta}) + \mathcal{D}(\zeta(t_{k-1}), \widehat{\zeta}) \right] \text{ for } k = 1, 2, \ldots, N, \qquad (1)$$

where $\mathbb{K}$ denotes the set of kinematically admissible displacements, $\mathbb{Z}$ is the set of admissible internal variables and the energetic contributions attain the from

$$\mathcal{E}^{\epsilon}(t,\widehat{\boldsymbol{u}},\widehat{\zeta}) = \int_{\Omega} \frac{\epsilon+\widehat{\zeta}}{2}\boldsymbol{\varepsilon}(\widehat{\boldsymbol{u}}+\boldsymbol{u}_{\mathrm{D}}(t)) : \boldsymbol{C} : \boldsymbol{\varepsilon}(\widehat{\boldsymbol{u}}+\boldsymbol{u}_{\mathrm{D}}(t)) + \frac{1}{2}\kappa \left|\nabla\widehat{\zeta}\right|^2 \,\mathrm{d}\Omega, \tag{2}$$

$$\mathcal{D}(\widehat{\zeta}^1,\widehat{\zeta}^2) = \begin{cases} \displaystyle\int_{\Omega} a(\boldsymbol{x}) \left(\widehat{\zeta}^1(\boldsymbol{x}) - \widehat{\zeta}^2(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x} & \text{if } \widehat{\zeta}^1 \geq \widehat{\zeta}^2 \text{ a.e. in } \Omega \\ +\infty & \text{otherwise} \end{cases} \tag{3}$$

where $\boldsymbol{u}_{\mathrm{D}}$ denote the time-dependent Dirichlet boundary data, $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{u}})$ is the linearized strain corresponding to a displacement field $\widehat{\boldsymbol{u}}$, $\boldsymbol{C}$ is a fourth-order tensor of elastic stiffness, κ is an influence factor introducing an internal length into the formulation, a denotes an activation threshold (related to strength of a material) and the term "$+\infty$" ensures unidirectionality of the damage evolution. An energetic solution to the complete damage is then obtained by the limit passage in time ($\tau \to 0$) and regularization parameter ϵ, with an appropriate re-interpretation of kinematics at fully damaged regions [1, 5].

3 Numerical aspects

In the numerical treatment, the formulation is converted to the discrete form by performing the spatial discretization using the conforming finite element method. The incremental time problem then transforms into a non-convex large-scale optimization program posed in terms of nodal displacements and nodal damage values. Following Bourdin [2], the special structure of the problem is exploited to apply sequential convex optimization procedure, converging to a critical point of the objective function. To ensure that the critical point is a good approximation to the global minimizer, a simple variant of a time back-tracking algorithm, based on two-sided energetic estimates derived in [6], is introduced [7].

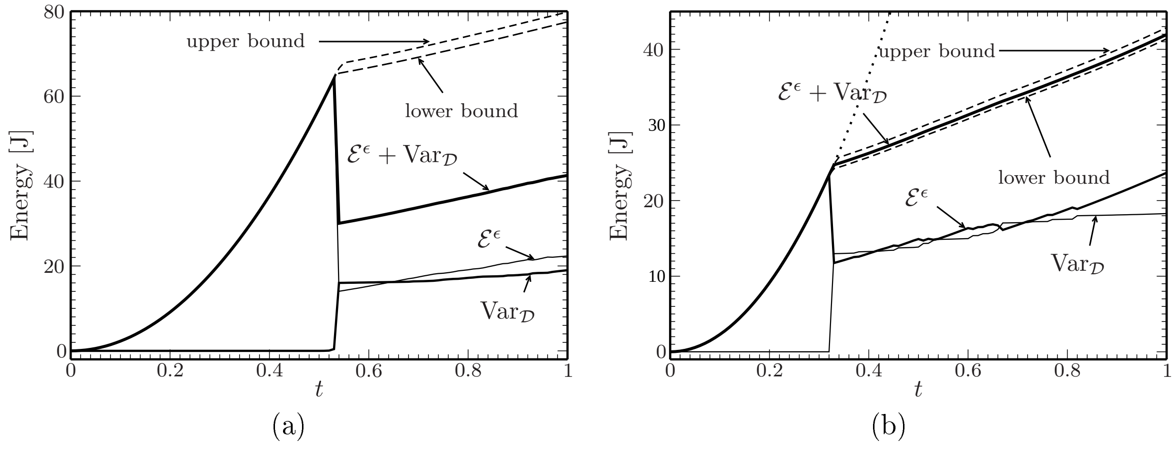


Figure 1: Example of energetics for a dog-bone shape specimen; (a) without backtracking (energy balance fails), (b) with backtracking (an approximate energetic solution), $\mathcal{E}^{\epsilon}$ is the globally stored energy, Var$\mathcal{D}$ denotes the cumulative dissipative energy.

To illustrate the performance of the proposed algorithm, in Figure 1 we present energetics of a uniaxial tension experiment for a dog-bone shape specimen. The results confirm that the proposed backtracking algorithm is capable of delivering a solution with lower energies then the basic scheme. Moreover, it can be shown that the resulting response is (almost) independent of spatial and temporal discretization. An interested reader is referred to [7] for additional details.

Acknowledgment: This work has been supported by the grant No. 201/10/0357 awarded by the Czech Science Foundation.

References

[1] G. Bouchitté, A. Mielke, T. Roubíček: *A complete-damage problem at small strains.* Zeit. für angew. Math. Phys. 60, 2009, 205–236.

[2] B. Bourdin: *Numerical implementation of the variational formulation of brittle fracture.* Interface Free Bound. 9, 2007, 411–430.

[3] M. Frémond, B. Nedjar: *Damage, gradient of damage and principle of virtual power.* Int. J. Solid Struct. 33, 1996, 1083–1103.

[4] A. Mielke: *Evolution of rate-independent systems.* In: Handbook of Differential Equations: Evolutionary Diff. Eqs. C. Dafermos, E. Feireisl (Eds.), pp. 461–559, Elsevier, Amsterdam, 2005.

[5] A. Mielke: *Complete-damage evolution based on energies and stresses.* Discrete Cont. Dyn. Syst. Ser. S 4, 2011, 423–439.

[6] A. Mielke, T. Roubíček: *Rate-independent damage processes in nonlinear elasticity.* Math. Models and Meth. in Appl. Sci. 16, 2006, 177–209.

[7] A. Mielke, T. Roubíček, J. Zeman: *Complete damage in elastic and viscoelastic media and its energetics.* Comput. Meth. Appl. Mech. Eng. 199, 2010, 1242–1253.

[8] M. Thomas, A. Mielke: *Damage of nonlinearly elastic materials at small strain - Existence and regularity results.* Zeit. Angew. Math. Mech. 90, 2010, 88–112.

Title: SEMINAR ON NUMERICAL ANALYSIS & WINTER SCHOOL

Proceedings of the conference SNA'11
Rožnov pod Radhoštěm
January 24 – 28, 2011

Editors: Radim Blaheta, Jiří Starý, Hana Bílková

Published by: Institute of Geonics AS CR, Ostrava

Printed by: Publishing Centre of VŠB – Technical University, Ostrava

First edition

Ostrava, 2011

copies 90

ISBN 978-80-86407-19-7