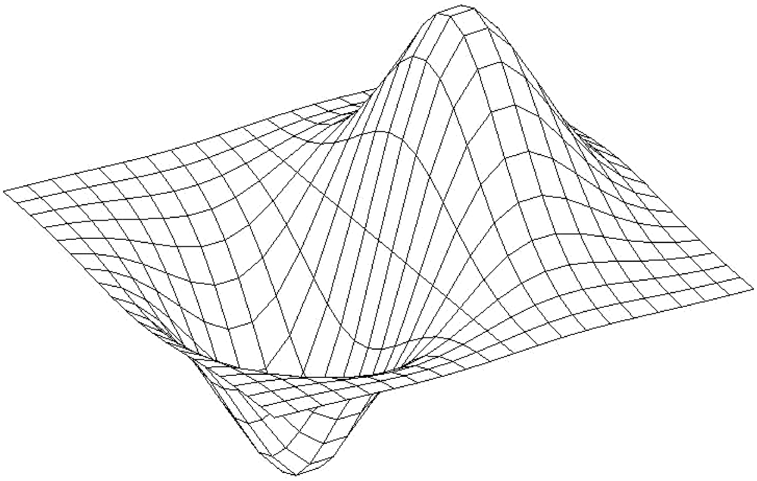
Institute of Geonics of the CAS, Ostrava

SNA'19

Seminar on Numerical Analysis

Modelling and Simulation
of Challenging Engineering Problems



Winter School

Methods of Numerical Mathematics and Modelling,
High-Performance Computing, Numerical Linear Algebra

Ostrava, January 21 – 25, 2019

Programme committee:

Radim Blaheta	Institute of Geonics of the CAS, Ostrava
Stanislav Sysala	Institute of Geonics of the CAS, Ostrava
Dalibor Lukáš	VŠB - Technical University of Ostrava
Jaroslav Kruis	Czech Technical University in Prague
Miroslav Rozložník	Institute of Mathematics of the CAS, Prague
Petr Tichý	Charles University, Prague

Organizing committee:

Radim Blaheta	Institute of Geonics of the CAS, Ostrava
Jiří Starý	Institute of Geonics of the CAS, Ostrava
Stanislav Sysala	Institute of Geonics of the CAS, Ostrava
Dagmar Sysalová	Institute of Geonics of the CAS, Ostrava
Hana Bílková	Institute of Mathematics of the CAS, Prague
Petra Frélichová	VŠB - Technical University of Ostrava

Conference secretary:

Dagmar Sysalová	Institute of Geonics of the CAS, Ostrava

Institute of Geonics of the Czech Academy of Sciences
ISBN 978-80-86407-73-9

Preface

Seminar on Numerical Analysis 2019 (SNA 2019) is the 14$^{\text{th}}$ meeting in a series of SNA events. The previous meetings were held in Ostrava 2003, 2005, Monínec 2006, Ostrava 2007, Liberec 2008, Ostrava 2009, Nové Hrady 2010, Rožnov 2011, Liberec 2012, Rožnov 2013, Nymburk 2014, Ostrava 2015 and 2017.

The first SNA was organized in honour of seventieth birthday of Prof. Ivo Marek. At SNA 2013 we celebrated his eighty but unfortunately this SNA is the first event without him as Ivo Marek passed away in 2017. A memory of his rich and inspiring life can be found in the special issue of the journal Applications of Mathematics devoted to SNA 2017, see Vol. 62(2017), No. 6. Certainly, a continuation of SNA would be a wish of Ivo.

We hope that SNA meetings will successfully continue as one of mostly national events and meetings of the Czech community working in the field of numerical mathematics and computer simulations.

The programme of SNA 2019 includes the traditional Winter School with tutorial lectures focused on selected important topics within the scope of numerical methods and modelling. This year, the Winter School provides a series of lectures on the following topics:

- High-performance variants of Krylov subspace methods *(E. Carson)*
- An introduction to extended finite element methods *(J. Haslinger)*
- On the way from matrix to tensor computations *(M. Plešinger)*
- Guaranteed eigenvalue bounds for elliptic partial differential operators *(T. Vejchodský)*

Beside the Winter School, SNA 2019 includes more than 40 contributions in the form of oral presentations and posters.

SNA 2019 has started with building of a new Programme Committee. On this opportunity, we would like to express many thanks to former members of the Programme Committee, to Prof. Zdeněk Dostál and Prof. Zdeněk Strakoš who contributed a lot to success of previous SNA events. Especially, Zdeněk Strakoš was a promoter of the Winter School becoming a part of SNA since 2005.

SNA 2019 is held again in Ostrava, at the Faculty of Electrical Engineering and Computer Science of the Technical University of Ostrava and at the Institute of Geonics of the Czech Academy of Sciences. We believe that the participants will enjoy the Winter School, the seminar programme consisting of contributed presentations and posters as well as accompanying social events.

On behalf of the Programme and Organizing Committee of SNA 2019,

Radim Blaheta and Jiří Starý

Contents

Winter school lectures

E. Carson:
High-performance variants of Krylov subspace methods

J. Haslinger:
An introduction to extended finite element methods

M. Plešinger:
On the way from matrix to tensor computations

T. Vejchodský:
Guaranteed eigenvalue bounds for elliptic partial differential operators

Guaranteed goal-oriented a posteriori error estimates for elliptic problems

O. Bartoš, V. Dolejší

Faculty of Mathematics and Physics, Charles University in Prague

1 Introduction

Finite element methods for solving partial differential equations give us approximate solutions which on their own need not resemble closely the actual exact solution. This leads us to seek an estimate of how close the approximation is. If the whole computation is carried out knowing that we look for a value of some functional, e.g. an integral over a part of the boundary of the domain or a value at some point, we can estimate the error of this functional. One way we can do this is by solving an adjoint problem and using a dual weighted residual method. If the error estimate is too high, it can then be used locally as an indicator to refine parts of a computational domain and find a more accurate approximate solution.

2 Model problem

Let us consider a Laplace equation $-\Delta u = f$ in Ω, $u = 0$ on $\partial\Omega$, where $\Omega \subset \mathbb{R}^2$ is a polygonal domain and $f \in L^2(\Omega)$ is given. Suppose that we are looking for a value of $J(u)$ rather than for an entire solution u, where J is a given linear functional, e.g. $J(u) = \int_\Omega ju \,\mathrm{d}x$ for $j \in L^2(\Omega)$. The weak solution $u \in H_0^1(\Omega)$ and its continuous piecewise polynomial discretization $u_h \in V_h^p \subset H_0^1(\Omega)$ are defined by

$$\begin{aligned}(\nabla u, \nabla\varphi) &= (f, \varphi) \quad \forall\varphi \in H_0^1(\Omega), \\ (\nabla u_h, \nabla\varphi_h) &= (f, \varphi_h) \quad \forall\varphi_h \in V_h^p.\end{aligned} \tag{1}$$

Similarly, we can formulate an adjoint (dual) problem for a functional J. The weak dual solution $z \in H_0^1(\Omega)$ and its approximation $z_h \in V_h^p$ are defined by

$$\begin{aligned}(\nabla\psi, \nabla z) &= (j, \psi) \quad \forall\psi \in H_0^1(\Omega), \\ (\nabla\psi_h, \nabla z_h) &= (j, \psi_h) \quad \forall\psi_h \in V_h^p.\end{aligned} \tag{2}$$

The sought approximation to $J(u)$ is naturally $J(u_h)$. Furthermore, we can use the Galerkin orthogonality of the approximate solutions to derive

$$\begin{aligned}J(u - u_h) &= (\nabla u - \nabla u_h, \nabla z) = (\nabla u - \nabla u_h, \nabla z - \nabla z_h) \\ &= (f, z - z_h) - (\nabla u_h, \nabla z - \nabla z_h) =: r_h(u_h)(z - z_h) \\ &= J(u - u_h) - (\nabla u - \nabla u_h, \nabla z_h) =: r_h^*(z_h)(u - u_h).\end{aligned}$$

Estimate $|J(u) - J(u_h)| \le |u - u_h|_{H_0^1(\Omega)} |z - z_h|_{H_0^1(\Omega)}$ gives us about double the order of convergence as compared to $|J(u) - J(u_h)| \le \|J\| |u - u_h|_{H_0^1(\Omega)}$, which could be derived without employing any goal-oriented strategy. It still remains to find some way to estimate $|J(u - u_h)|$ without using the unknown weak solutions u and z. The residual forms $r_h(u_h)(z - z_h)$ and $r_h^*(z_h)(u - u_h)$ only use one unknown function each. By replacing u and z in the forms r_h^* and r_h with with some more accurate approximations than u_h and z_h we can find good a posteriori error estimates for our target functional J. This can be done with functions $\tilde{u}_h$ and $\tilde{z}_h$ from a richer space $V_h^{p+1} \supset V_h^p$. The error of the target functional becomes

$$\begin{aligned} J(u - u_h) &= r_h(u_h)(\tilde{z}_h - \Pi\tilde{z}_h) + r_h(u_h)(z - \tilde{z}_h) = r_h(u_h)(\tilde{z}_h - \Pi\tilde{z}_h) + (\nabla(u - \tilde{u}_h), \nabla(z - \tilde{z}_h)) \\ &= r_h^*(z_h)(\tilde{u}_h - \Pi\tilde{u}_h) + r_h^*(z_h)(u - \tilde{u}_h) = r_h^*(z_h)(\tilde{u}_h - \Pi\tilde{u}_h) + (\nabla(u - \tilde{u}_h), \nabla(z - \tilde{z}_h)), \end{aligned}$$

where Π is an interpolation onto V_h^p.The error estimator

$$\eta^I = \frac{1}{2}(r_h(u_h)(\tilde{z}_h - \Pi\tilde{z}_h)) + r_h^*(z_h)(\tilde{u}_h - \Pi\tilde{u}_h)$$

can be further divided to elementwise error contributions

$$\begin{aligned} \eta^{II} = \frac{1}{2} \Bigg(\sum_{K \in \mathcal{T}_h} & R_{K,V} \|\tilde{z}_h - \Pi\tilde{z}_h\|_K + R_{K,B} \|\tilde{z}_h - \Pi\tilde{z}_h\|_{\partial K} \\ & + R_{K,V}^* \|\tilde{u}_h - \Pi\tilde{u}_h\|_K + R_{K,B}^* \|\tilde{u}_h - \Pi\tilde{u}_h\|_{\partial K} \big) . \end{aligned}$$

These can be used for mesh refinement, as was done for a convection-diffusion-reaction equation in [4], and, considering that functions of a type $\|\tilde{z}_h - \Pi\tilde{z}_h\|_K$ are dependent on a shape of K, we can use this estimate to generate anisotropic mesh, see [3]. We know that η^I is a good error estimate on a sufficently refined mesh $\mathcal{T}_h$. The true error could, however, be dominated by the higher order term $(\nabla(u - \tilde{u}_h), \nabla(z - \tilde{z}_h))$ on a coarse mesh. We thus need some upper, not necessarily sharp error estimate for this term to guarantee that we do not stop computing before the true error reaches some given small tolerance. In [1], there is an estimate of a form $C \log(h)^{3/2} \eta_{NVV} \|f\|$, where the constant C is unknown, f is measured in some higher order Sobolev norm, but η_{NVV} is computable and uses functions similar to those in the definition of $R_{K,V}^*$ and $R_{K,B}^*$. In [2], there is a fully computable error estimate of a form $\left(\sum_{K \in \mathcal{T}_h} (\|\sigma_K\|_K + \mathrm{osc}_K)^2\right)^{1/2}$, where σ_K is computed using flux reconstruction and osc_K measures oscillations in the right-hand side (such as $f - \Pi f$). These computations also rely heavily on an assumpion that the considered differential operator is symmetric in the weak form, i.e. $(\nabla \cdot, \nabla \cdot)$ in our case. With all these tools, it is possible to safely continue computations until we arrive at a sufficiently close approximation $J(u_h)$ to $J(u)$.

Acknowledgement: This work has been supported by Charles University, project GAUK 850218.

References

[1] R. H. Nochetto, A. Veeser, M. Verani: *A safeguarded dual weighted residual method.* IMA Journal of Numerical Analysis 2009; **29**(1), pp. 126–140.

[2] M. Ainsworth, R. Rankin: *Guaranteed computable bounds on quantities of interest in finite element computations.* Int. J. Numer. Meth. Engng 2012; **89**, pp. 1605–1634.

[3] V. Dolejší: *Anisotropic hp-adaptive method based on interpolation error estimates in the Lq-norm.* Appl. Numer. Math., **82**, 2014, pp. 80–114.

[4] O. Bartoš, V. Dolejší, G. May, A. Rangarajan, F. Roskovec: *A goal-oriented anisotropic hp-mesh adaptation method for linear convection-diffusion-reaction problems.* Comput. Math. Appl. (submitted).

Simulation of the incompressible turbulent flow using isogeometric analysis

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

Faculty of Applied Sciences, University of West Bohemia in Pilsen

1 Introduction

The Navier-Stokes equations are the basis for computational modeling of the flow of an incompressible Newtonian fluid. The fluid flow behaviour is very complex and depends on the Reynolds number Re, which depends on viscosity ν, geometry and fluid velocity. The Navier-Stokes equations can be used to directly simulate turbulent flows. But the number of grid points in spatial discretization must be proportional to $Re^{9/4}$ and the time step has to be sufficiently small to resolve the movement of the fastest fluctuations, otherwise the simulation becomes unstable. Then the direct numerical simulation based on solving Navier-Stokes equations becomes impossible as the Reynolds number increases and some kind of turbulence modelling is necessary.

In this contribution, we focus on incompressible fluid flow simulation based on RANS equations with LRN (Low Reynolds Number) version of Wilcox (2006) and SST (Shear Stress Transport) k-omega two–equation models. The numerical model is based on Isogeometric Analysis (IgA) which is a recently developed approach based B-spline/NURBS objects and sharing a lot of features and approaches with the well-known Finite Element Method.

Our solver is implemented in an open-source C++ library G+Smo and its functionality will be demonstrated on several examples.

2 Incompressible turbulent flow

The Reynolds–Averaged Navier–Stokes equations (RANS) is the most common approach to simulate turbulent flows. The idea is the modelling of all turbulent scales and only the effect of turbulence on the mean flow behavior is considered, which implies lower memory requirements. The Boussinesq hypothesis is applied in our implementation, which arrives at the Reynolds–Averaged Navier–Stokes equations in the form (see more e.g. in [3])

$$\begin{aligned} \frac{\partial \bar{\mathbf{u}}}{\partial t} - \nabla \cdot [(\nu + \nu_T)\nabla \bar{\mathbf{u}}] + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \nabla \bar{p} - \nabla \cdot (\nu_T (\nabla \bar{\mathbf{u}})^T) &= -\frac{2}{3}\nabla k, \qquad \text{in } \Omega \times (0,T), \\ \nabla \cdot \bar{\mathbf{u}} &= 0, \qquad \text{in } \Omega \times (0,T), \end{aligned} \tag{1}$$

where $\Omega \subset \mathbf{R}^d$ is a bounded domain, d being the number of spatial dimensions, with boundary $\partial\Omega$ consisting of two disjoint parts, Dirichlet $\partial\Omega_D$ and Neumann $\partial\Omega_N$ and $T > 0$ is an upper bound of the time interval of interest $[0, T]$ and ν is the kinematic viscosity. Next, the unknown variables are the mean flow velocity $\bar{\mathbf{u}}$, the mean kinematic pressure $\bar{p}$, the eddy viscosity ν_T and the turbulent kinetic energy k.

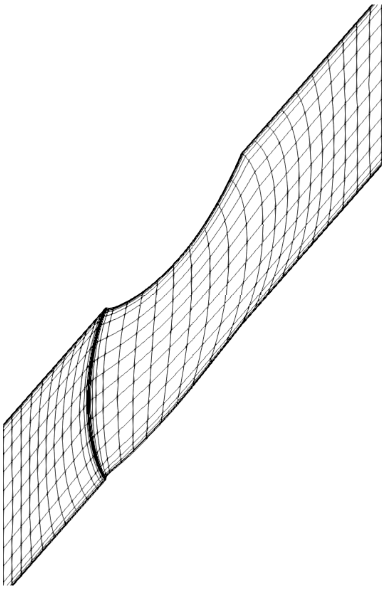


Figure 1: Computational mesh with 7085 DOFs for blade profile.

The initial–boundary value RANS problem is given as a system of (d + 1) equations (1) together with initial and mixed boundary conditions

$$
\begin{aligned}
\bar{\mathbf{u}}(\mathbf{x},0) &= \bar{\mathbf{u}}_0(\mathbf{x}), && \text{in } \Omega, \\
\bar{\mathbf{u}} &= \mathbf{g}, && \text{on } \partial\Omega_D \times [0,T], \\
(\nu + \nu_T)\frac{\partial \bar{\mathbf{u}}}{\partial \mathbf{n}} + \nu_T(\nabla\bar{\mathbf{u}})^T \cdot \mathbf{n} - \mathbf{n}\bar{p} - \frac{2}{3}\mathbf{n}k &= \mathbf{0}, && \text{on } \partial\Omega_N \times [0,T].
\end{aligned}
\tag{2}
$$

A wide range of the turbulent models can be involved to approximate the eddy viscosity and turbulent kinetic energy. The turbulent models vary from relatively simple algebraic models to more complex models, e.g. Spalart–Allmaras model as the one–equation turbulence model and k–ϵ or k–ω models as two–equation turbulence models.

The two–equation models became standard models in engineering practice and research, thus a large number of two–equation models has been derived and are still developing. The variety of the models give us an opportunity to choose the most appropriate model for a wide range of the flows. However, the fluid flow behaviour is necessary to predict properly and it is necessary to understand the formulation of the two–equation models and their assumptions. For example, if the separation region can appear in the flow or if the flow near the walls is important to simulate in details then different turbulent models should be used, because some turbulent models are valid near the boundary - Low Reynolds Number (LRN) models - and others are valid far from the boundary - High Reynolds Number (HRN) models.

The equations are solved through the whole domain up to the solid walls using LRN and hence a very fine grid is necessary near the boundary to resolve the flow variables properly.

On the other hand, the HRN models can be considered instead, also known as wall functions approach. The idea is to bridge the near wall region, i.e., to place the first node not so close to the boundary and solve the RANS problem in the rest of the domain. For more details and discussion about the near wall treatment, the reader is referred to [3].

Considering the implementation difficulties associated with the HRN method in case of isogeometric analysis, low Reynolds number version of Wilcox (2006) $k - \omega$ two–equation model [4] and SST $k - \omega$ two–equation model [2] are considered in this contribution.

Now, we present one of the numerical results for the turbulent flow around the 2D blade profile using LRN Wilcox turbulence model. The computational geometry and the mesh is showed in the Figure 1. The simulation was carried out for a fluid with viscosity $\nu = 10^{-5}$ with time step $\Delta t = 0.001$ for backward Euler time discretization. At the inflow boundary we consider constant velocity profile. The middle parts of the upper and lower boundaries are solid walls with zero velocity and the periodic conditions are considered at the rest of the upper and lower boundaries.

We used a steady Navier–Stokes solution with viscosity $\nu = 0.01$ as the initial condition. Figures 2 and 3 show the mean velocity and pressure distributions at times $T = 0.05s$ and $T = 0.152s$, respectively. The numerical solution at the later time represents result for which the stopping criteria is satisfied.

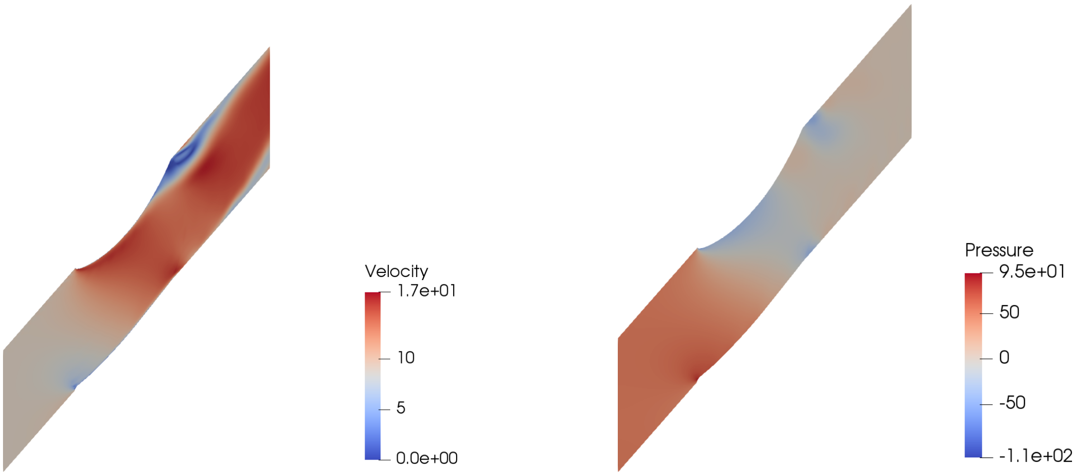


Figure 2: Mean velocity (left) and pressure (right) solution at $T = 0.05s$.

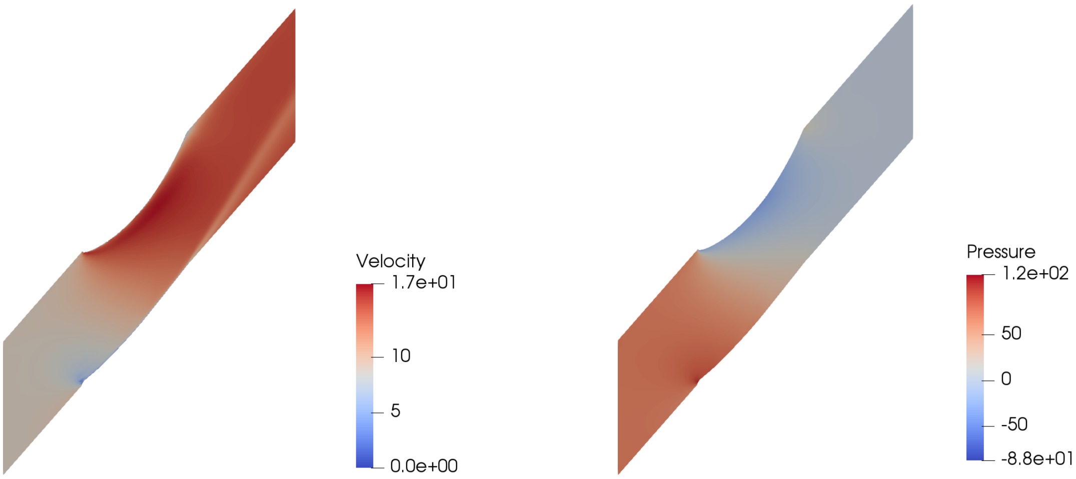


Figure 3: Mean velocity (left) and pressure (right) solution at $T = 0.152s$.

Acknowledgement: This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 678727.

References

[1] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs: *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement.* In: Computer Methods in Applied Mechanics nad Engineering 194 (3941), 2005, pp. 4135–4195.

[2] F.R. Menter: *two–equation eddy-viscosity turbulence models for engineering applications.* In: AIAA Journal 32, 1994, pp. 1598–1605.

[3] H. K. Versteeg, W. Malalasekera: *An introduction to computational fluid dynamics - Second Edition.* Prentice Hall, 2007.

[4] D.C. Wilcox: *Formulation of the $k-\omega$ Turbulence Model Revisited.* In: AIAA Journal 46, 2008, pp. 2823–2838.

Preconditioners for the incompressible Navier–Stokes equations discretized by isogeometric analysis

B. Bastl, M. Brandner, J. Egermaier, H. Horníková, K. Michálková, E. Turnerová, C. Vuik[*]

NTIS, Faculty of Applied Sciences, University of West Bohemia in Pilsen
[*] Delft University of Technology, Delft, The Netherlands

1 Introduction

We deal with flow simulation modeled by the incompressible Navier–Stokes equations. The discretization of the problem is based on isogeometric analysis (IgA) approach resulting in large sparse linear systems of saddle-point type. The most expensive part of the simulation process is the solution of these systems. Direct solvers are inapplicable for large problems because of their very high time and memory requirements, therefore an efficient iterative method is necessary. Among iterative methods, Krylov subspace methods are the most commonly used in applications and can be very efficient if combined with a good preconditioning technique. Since our matrices are nonsymmetric, we have to use a Krylov subspace method for nonsymmetric matrices. The most popular ones are GMRES and BiCGSTAB.

In this work, we focus on a class of block preconditioners for saddle-point type systems developed and studied in recent years, mostly in connection with finite element discretizations. We study their efficiency for systems arising from the IgA discretization, where the matrix is usually less sparse compared to those from finite elements. Our main aim is flow simulation in water turbines, which brings several complications like periodic boundary conditions at nonparallel boundaries and computation in a rotating frame of reference. This makes the system matrix even less sparse with more complicated sparsity pattern.

2 Problem formulation

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, where d is the number of spatial dimensions. The initial boundary value incompressible Navier–Stokes problem is given as a system of $d+1$ equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} + \nabla p &= \mathbf{0} \qquad \text{in } \Omega \times (0,T), \\ \nabla \cdot \mathbf{u} &= 0 \qquad \text{in } \Omega \times (0,T), \end{aligned} \tag{1}$$

together with initial condition and boundary conditions, where $\mathbf{u}$ is the flow velocity, p is the kinematic pressure and ν is the kinematic viscosity.

The problem is discretized in time using backward finite difference, linearized by Picard method and discretized in space using isogeometric analysis approach. IgA is a relatively new discretization approach [1] based on Galerkin method, where the basis of the discrete solution space is taken from the B-spline/NURBS representation of the computational domain Ω. The discretization leads to a sparse linear system of saddle-point type

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{2}$$

usually very large in real world applications.

Here, the matrix A is block diagonal, since there is no coupling between the velocity components. But in the case of flow in the water turbine, the periodic boundary conditions at nonparallel sides (the computational domain is part of a radially symmetric domain) and rotating frame of reference introduce coupling between the velocity components, thus, some off-diagonal blocks of A become nonzero.

3 Block preconditioners

Block preconditioners are based on splitting the system into velocity and pressure part. Their construction is based on the block LDU decomposition of the system matrix in (2)

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}, \tag{3}$$

where $S = -BA^{-1}B^T$ is the Schur complement matrix. This suggests the following choice of the preconditioner matrix

$$P = \begin{bmatrix} A & B^T \\ 0 & S \end{bmatrix}, \tag{4}$$

for which the (right) preconditioned matrix would have all eigenvalues equal to one. The computation of $P^{-1}r$ is performed by solving the linear system

$$\begin{bmatrix} A & B^T \\ 0 & S \end{bmatrix} \begin{bmatrix} z_u \\ z_p \end{bmatrix} = \begin{bmatrix} r_u \\ r_p \end{bmatrix} \tag{5}$$

in the following steps

$$Sz_p = r_p, \tag{6}$$

$$Az_u = r_u - B^T z_p. \tag{7}$$

The explicit construction of the Schur complement S would be impractical, because it would require construction of A^{-1} and it is a dense matrix. Therefore we have to find some inexpensive approximation $\hat{S} \approx S$ first. The choice of the approximation yields different preconditioners. We study several choices which can be found in the literature, namely LSC (least-squares commutator), AL (augmented Lagrangian) and SIMPLE-type preconditioners. An overview of these preconditioners can be found e.g. in [3].

The subsystems with matrices A and $\hat{S}$ are usually solved approximately in practice, e.g. by one or more V-cycles of a multigrid solver. However, we use direct solvers for these subsystems in this contribution.

4 Numerical experiments

In the numerical experiments, we compare convergence of the Krylov subspace methods with the particular preconditioners, study its dependence on the mesh refinement, viscosity etc. and test their efficiency for the case with periodic conditions on nonparallel boundaries. Further, since isogeometric analysis allows the degree of continuity of the solution across the element interfaces to be higher than C^0, we also test the influence of the degree of continuity on the convergence of the iterative solvers.

In Figure 1 we can see a comparison of several preconditioners for a simple ilustrative example of flow over a 2D backward facing step with viscosity $\nu = 0.01$, time step $\Delta t = 0.01$ and a uniform mesh with 42005 degrees of freedom. It shows the evolution of relative residual norm, relative error norm and computational time in seconds during 100 GMRES iterations for one linear system obtained from the discretization of this problem. For the evaluation of the error, the solution computed with direct solver is considered as exact solution.

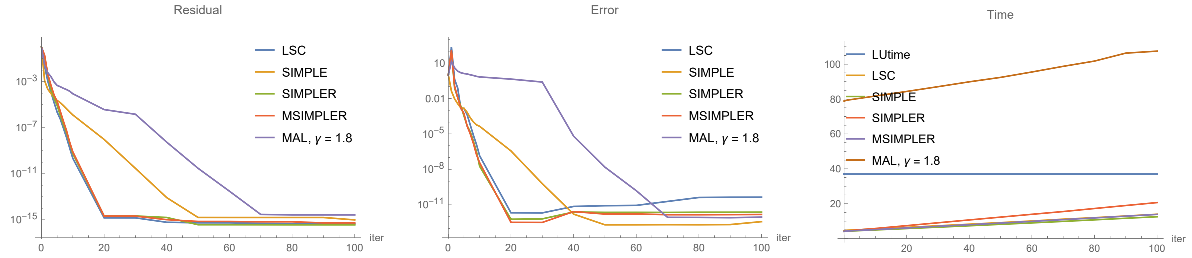


Figure 1: Preconditioners comparison.

Acknowledgement: This work has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 678727.

References

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

[2] Y. Saad: *Iterative methods for sparse linear systems, second edition.* SIAM, 2003.

[3] A. Segal, M. ur Rehman, C. Vuik: *Preconditioners for incompressible Navier–Stokes solvers.* Numer. Math. Theor. Meth. Appl. 3, 2010, pp. 245–275.

On comparison of solution methods for 3D contact shape optimization problems with friction

P. Beremlijski

VŠB - Technical University of Ostrava

The shape optimization of 3D elastic body in contact with rigid obstacle with Coulomb friction can be modelled as a minimization of a composite function generated by the objective and the control-state mapping (see [2, 1, 3]). It has been shown that for small coefficients of Coulomb friction the discretized contact problem with Coulomb friction has a unique solution and this solution is Lipschitzian as a function of a control variable describing the shape of the elastic body. It means that the control-state mapping is single-valued and the 3D contact shape optimization problem with Coulomb friction leads to a minimization of nondifferentiable (nonsmooth) single-valued function.

There are several possibilities how to solve the problem. The easiest one is to neglect the friction and find the optimal shape of the optimized body as the solution of the shape optimization problem of 3D elastic body in contact without friction. The advantage is that we solve the optimization problem with differentiable function. Unfortunately, we find the optimized shape which does not solve the original problem exactly.

Another possibility is to solve the original problem with Coulomb friction. This leads to the optimization of a nonsmooth function. We have to use some method which are working with calculus of Clarke (for details, see [4]) for this case. The most reliable of the nonsmooth methods for this kind of problem are bundle methods. We use bundle trust method proposed by Schramm and Zowe (for details, see [6]). In each step of the iteration process, we must be able to find the solution of the state problem (contact problem with Coulomb friction) and to compute one arbitrary Clarke subgradient. To get subgradient information needed in the used numerical method we use the differential calculus of Mordukhovich (for details, see [5]).

The goal of the contribution is to compute the optimized shape of 3D elastic body in contact with rigid obstacle with Coulomb friction by both previous mentioned solution methods and their comparison.

Acknowledgement: The work is partially supported by Grant of SGS No. SP2018/165, VŠB – Technical University of Ostrava, Czech Republic.

References

[1] P. Beremlijski, T. Brzobohatý, T. Kozubek, A. Markopoulos, J. Outrata: *Parallel solution of contact shape optimization problems with Coulomb friction based on domain decomposition.* In: WIT Transactions on the Built Environment, WIT Press, Southampton, 2012, pp. 285–295.

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

[3] P. Beremlijski, A. Markopoulos: *On solution of 3D contact shape optimization problems with Coulomb friction based on domain decomposition.* In: EngOpt'14, 2014, pp. 465–470.

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

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

[6] J. Outrata, M. Kočvara, J. Zowe: *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results*. Kluwer Acad. Publ., 1998.

An efficient reduced basis construction for stochastic Galerkin matrix equations using deflated conjugate gradients

M. Béreš

Institute of Geonics of the CAS, Ostrava
Department of Applied Mathematics, FEECS, VŠB - Technical University of Ostrava

1 Introduction

The motivation for this work is a laboratory experiment on a rock sample. We assume to know the inner structure (separated subdomains with different types of material) of the sample (e.g. from a Computed Tomography scan) and we want to estimate permeabilities of each distinct subdomain in the sample. This is done by a series of tests, where a fluid is pressed to some part of the sample and let out from another part of the sample, the output volume is then measured. We also admit uncertainties of these measurements, this forms a complex inverse problem. This is usually solved using the Bayesian inversion, which relies on many forward problem (Darcy flow) solutions. In this matter, the stochastic Galerkin (SG) method can be used as a surrogate model substituting the forward solutions.

The work presented in this contribution is an extension of the results presented in [2]. Here we focus on alternative approaches to the rational Krylov approximation and expansion vector proposal, see Sec. 3. Main contribution of this work is the use of the DCG method, which is new in this area of application, see Sec.5. This contribution is based of our paper [1].

1.1 Problem setting

We assume a Darcy flow problem on 2D square $\langle 0,1\rangle^2$ domain with no forcing term:

$$\begin{cases} -\nabla_x \cdot \left(k\left(x;\boldsymbol{Z}\right) \nabla_x u\left(x;\boldsymbol{Z}\right)\right) = 0 & \forall x \in \Omega,\ \boldsymbol{Z} \in \mathbb{R}^M, \\ u\left(x;\boldsymbol{Z}\right) = g\left(x\right) & \forall x \in \Gamma^D, \boldsymbol{Z} \in \mathbb{R}^M, \\ n\left(x\right) \cdot k\left(x;\boldsymbol{Z}\right) \nabla_x u\left(x;\boldsymbol{Z}\right) = 0 & \forall x \in \Gamma^N, \boldsymbol{Z} \in \mathbb{R}^M. \end{cases} \tag{1}$$

Here the random material field takes the form

$$k\left(x;\boldsymbol{Z}\right) = \sum_{i=1}^{M} \chi_{\Omega_i}\left(x\right) \exp\left(\sigma_i Z_i + \mu_i\right), \tag{2}$$

where $\chi_{\Omega_i}\left(x\right)$ is a characteristic function of the subdomain Ω_i and $\exp\left(\sigma_i Z_i + \mu_i\right)$ describes the distribution of the permeability on subdomain Ω_i. μ_i, σ_i are the mean value and the standard deviation of the underlying normal distribution. Then the components of $\boldsymbol{Z}$ are independent standard normal random variables.

2 Stochastic Galerkin method

The SGM assumes a discretization of both the physical space $\langle \varphi_1\left(x\right), \ldots, \varphi_{N_d+N_{Dd}}\left(x\right)\rangle \subset H^1\left(\Omega\right)$ (linear elements) and the stochastic/parametric space $\langle \psi_1\left(\boldsymbol{Z}\right), \ldots, \psi_{N_s}\left(\boldsymbol{Z}\right)\rangle \subset L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M\right)$ (Hermite polynomials).

Due to the separable nature of the material field, the values of bilinear form on the elements of tensor product basis can be written as

$$a\left(\varphi_i\psi_j,\varphi_k\psi_l\right)=\sum_{m=1}^{M}\int_{\mathbb{R}^M}\psi_j\psi_l\exp\left(a_mZ_m+b_m\right)\mathrm{d}F\boldsymbol{Z}\int_{\Omega_m}\nabla\varphi_i\nabla\varphi_k\mathrm{d}x. \tag{3}$$

This leads to a large system $(N_s \times N_d)$ of linear equations in the form of

$$A\cdot\overline{u}_h=\overline{b},\;\; A=\sum_{m=1}^{M}G_m\otimes K_m,\;\; b=\sum_{m=1}^{M}g_m\otimes f_m, \tag{4}$$

where G_m/g_m are matrices/vectors of the parameter space and K_m/f_m are matrices/vectors of the physical space.

3 Reduced basis method

We can view the system (4) as matrix equations

$$\sum_{m=0}^{M}K_mxG_m^T=\sum_{m=1}^{M}f_mg_m^T. \tag{5}$$

We assume that there exist a low-rank approximation $x_k = W_ky_k$ of the solution x, where $W_k = [w_1,\dots,w_k] \in \mathbb{R}^{N_d\times k}$ is given reduced basis (RB) and y_k is a reduced solution matrix. The matrix y_k can be obtained from (5) using the Galerkin condition on the residual of x_k

$$W_k^TR_k=0\Rightarrow\sum_{m=0}^{M}W_k^TK_mW_ky_kG_m^T=\sum_{m=1}^{M}W_k^Tf_mg_m^T. \tag{6}$$

The RB approach can be viewed as a standard iterative method. In each iteration we expand the RB and control the relative residual error.

3.1 Rational Krylov subspace methods

We aim to build the RB using an approach from [2] called the rational Krylov subspace approximation. Briefly, the rational Krylov subspace approximation can be performed for a series of symmetric positive definite (SPD) matrices $\{K_m\}_{m=1,\dots,M}$ and a nonzero vector v. We use only simplest rational functions $\frac{1}{K_m}$. In the first iteration, it generates the basis $\left\langle K_1^{-1}v,\dots,K_M^{-1}v\right\rangle$, in the second the basis $\left\langle K_1^{-1}K_1^{-1}v,K_1^{-1}K_2^{-1}v,\dots,\;K_M^{-1}K_{M-1}^{-1}v,K_M^{-1}K_M^{-1}v\right\rangle$ and so on, for details see [2]. The reduced basis will be the union of v and all these bases.

In our case the matrices $\{K_m\}_{m=1,\dots,M}$ are not SPD and we need to transform the system. This leads to the use of these matrices $\left\{K_0^{-1}\left(K_m-\alpha K_0\right)\right\}_{m=1,\dots,M}$ and these starting vectors $\left\{K_0^{-1}f_m\right\}_{m=1,\dots,M}$.

3.2 Adaptive selection of space expansion

The process of building the rational Krylov subspace is impractical because in each subsequent iteration we need to construct larger bases. The remedy to this is to iteratively select a vector v from the current basis and expand the basis by $\langle K_1^{-1}v, \dots, K_M^{-1}v\rangle$.

We propose an approach, which calculates weights for the vectors during the orthogonalisation step (with insignificant additional costs). And choose v for the next step accordingly.

4 Deflated conjugate gradients

The deflated conjugate gradients (DCG) method is an extension of the standard conjugate gradient (CG or PCG if using a preconditioner) method, see [3]. The DCG method takes an additional parameter in the form of the deflation basis W. The deflation basis W should be able to describe the sought solution reasonably well.

We choose the RB available in each iteration (it gradually grows) as the deflation basis for the DCG.

5 Numerical testing

Here, we present an outline of the obtained results. We tested both the convergence of the RB solver (i.e. convergence of the whole SG solution) and the convergence of the DCG with varying size of the deflation basis (i.e. in different iterations of the RB solver).

Our model problem consists of $M = 7$ subdomains, with stochastic properties $\mu_i \in (9,5,9,1,5,1,5)$ and $\sigma_i = 0.5$. The used discretizations consist of complete polynomials on 7 variables up to a given degree and an uniform finite element grid on unit square, where grid lvl l equals to the discretization $10l$ on each side.

Reduced basis convergence: we compare different settings with 3 different parameters of the RB construction:

- the approach from [2] using the Cholesky factor with the proposed approach without it
- the choice of the expansion candidate vectors from [2] using the SVD based approach with our approach incorporated to the orthogonalisation step
- initial vectors for the RB creation: single vector as a sum of f_m and zero iteration of the RB $W_0 = \langle f_m\rangle_{m=1}^{M}$

The performance of all of these approaches are comparable with slight advantage to the version without Cholesky factor + adaptive selection during the orthogonalisation step + initial vector as a sum of f_m.

Deflated CG convergence: we compare the impact of the RB used as a deflation basis together with some preconditioners (Schwarz, diagonal, ichol). The summary of collected results is in Tab. 1. The RB solver running up to the precision 10^{-9} (higher targeted precision would lead to higher efficiency) saves more than 70% of the time spent on the PCG if the DCG is used.

	Additive Schwarz p.	**diagonal p.**	**ichol (nofill) p.**
sum of saved iterations	18335	75191	26558
savings in percents	72.32%	73.47%	73.33%

Table 1: Computational savings using DCG with the RB as a deflation basis.

6 Conclusions

We examined new approach without the use of the Cholesky factor, which performs similarly (or slightly better) in comparison to the original one from [2]. Additionally we proposed a cheaper alternative to the RB expansion vector choice using SVD from [2], which performed slightly better than the SVD approach.

The main contribution is the use of the DCG method with the current build of the RB as a deflation basis. With the use of the DCG method, we saved more than 70% of the computational effort during the construction of the RB (independent of the choice of the preconditioner). The solution of the reduced problem using the PCG with the Kronecker preconditioner was also effective, due to variable accuracy (10 times higher than the last relative residual of the RB solver) and almost precise initial guess based on the solution from the previous RB iteration.

We also aimed at the algorithm, where every step is not bound by memory (like Cholesky decomposition) and can be effectively parallelized. This was fully achieved e.g. with the use of additive Schwarz preconditioner, we can scale up to large supercomputer clusters. Such implementation is the aim of our future work.

Acknowledgments: This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project “IT4Innovations excellence in science - LQ1602”. The work was also partially supported by Grant of SGS No. SP2018/68 and by Grant of SGS No. SP2018/161, VŠB - Technical University of Ostrava, Czech Republic.

References

[1] M. Béreš: *An efficient reduced basis construction for stochastic Galerkin matrix equations using deflated conjugate gradients.* AETA 2018: The 5th International Conference on Advanced Engineering Theory and Applications 2018, 2018.

[2] C.E. Powell, D. Silvester, V. Simoncini: *An Efficient Reduced Basis Solver for Stochastic Galerkin Matrix Equations.* SIAM Journal on Scientific Computing, 39(1), A141–A163, Jan. 2017.

[3] Y. Saad, M. Yeung, J. Erhel, F. Guyomarc’h: *A Deflated Version of the Conjugate Gradient Algorithm.* SIAM Journal on Scientific Computing, 21(5), pp. 1909–1926, Jan. 2000.

Approximation of PDF using maximal entropy multilevel Monte Carlo method

J. Březina, M. Špetlík, P. Exner, J. Stebel

Technical University of Liberec

Introduction

Classical Monte Carlo method estimates a mean of a random variable X by the average

$$\langle X\rangle_N = \frac{1}{N}\sum_{i=1}^{N} X(\omega_i).$$

using N samples of X. This is attractive for applications where X is result of complex calculations as it does not require changes into the simulation software. On the other hand the convergence rate its slow convergence rate:

$$\mathrm{Var}\langle X\rangle_N = \frac{\mathrm{Var}\, X}{\sqrt{N}}$$

with respect to number of samples N prevents usage of this approach for realistic simulations.

The idea of multilevel Monte Carlo estimators (MLMC, see [3]) is to diminish the error by reduction of the variance by a sequence of approximations X^n that are cheaper to sample. In particular we write L-level MLMC estimator as

$$\langle X^L\rangle_{\mathbf{N}} = \sum_{l=1}^{L}\langle X^l - X^{l-1}\rangle_{N_l} = \sum_{l=1}^{L}\langle \Delta^l X\rangle_{N_l} \tag{1}$$

where $\mathbf{N} = (n_1, \dots, n_L)$ is the *sampling vector* of the number of samples on individual levels and by $\Delta^l X(\omega) = X^l(\omega) - X^{l-1}(\omega)$ we denote level differences. This estimator have variance:

$$\mathrm{Var}\langle X^L\rangle_{\mathbf{N}} = \sum_{l=1}^{L}\frac{V_l}{N_l}, \quad V_l = \mathrm{Var}\langle \Delta^l X\rangle_{N_l}. \tag{2}$$

where the level variances V_l can be estimated using the standard unbiased estimator:

$$V_l \approx \widehat{V}_l = \frac{1}{N_l - 1}\sum_{i=1}^{N_l}\left(\Delta^l X_i - \langle \Delta^l X\rangle_{N_l}\right)^2$$

For many applications the level variances decay rapidly. In particular, when X is based on numerical PDE solution, we can approximate X by solutions based on coarser grids. For an elliptic PDE, numerical scheme of order s and assuming linear computational complexity (e.g. multigrid solver) the optimal choice of the sample vector is:

$$N_l \propto h_l^{2s+d}$$

where h_l is the mesh step at level l and d is the dimension of the domain. So we need a work corresponding to the calculation of just few fine resolution samples of X to obtain mean estimate at quality that would require millions of samples using the classical MC method. In practice, however, the speedup is limited by the fact that PDE solvers are usually not optimized for performance on coarse grids.

Maximal entropy method

For the approximation of the density function of the random variable X based on the MLMC we use generalized moments:

$$\mu_m = \mathsf{E}_\rho[\phi_m(X)], \tag{3}$$

where $\phi_m : R \to R$ are smooth functions. To deal with normalization we assume $\phi_1(x) = 1$. It can be shown that the minimum entropy approximation with moments $\boldsymbol{\mu}$ have density function:

$$\rho_{\boldsymbol{\lambda}}(x) = e^{\boldsymbol{\lambda}\cdot\boldsymbol{\phi}}$$

where the parameter vector $\boldsymbol{\lambda}$ is the unique solution to the convex minimization problem:

$$\text{minimize}(\boldsymbol{\lambda}): \quad F(\boldsymbol{\lambda}) = \int_\Omega \rho_\lambda - \boldsymbol{\lambda}\cdot\boldsymbol{\mu}. \tag{4}$$

Where $\Omega \subset \operatorname{supp}\rho$ is assumed to be bounded.

In contradiction to previous works, namely [1], we relax the normalization condition $\int_\Omega \rho_\lambda = 1$, which leads to natural convex problem (4) for $\boldsymbol{\lambda}$. This approach also extends the stability result [1, Theorem 3] while keeping the proof very simple. In particular, we can estimate the approximation error in the sense of Kullback-Leibler divergence $D_{KL}(f\|g) = \mathsf{E}_f[\log(f/g)]$ as follows:

Theorem 1. *Let $\rho_{\boldsymbol{\lambda}}$ be the approximation of the exact density ρ based on exact moments $\boldsymbol{\mu} = \mathsf{E}_\rho\boldsymbol{\phi}$. The MLMC estimator* (1) *provides estimated moments $\hat{\boldsymbol{\mu}}$, we denote $\hat\rho = \rho_{\hat{\boldsymbol{\lambda}}}$ corresponding density approximation with parameters $\hat{\boldsymbol{\lambda}}$ given as solution to* (4)*. Assuming the error of the estimator $V = \mathsf{E}_\rho|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}|^2$, then for any given (small) probability π we have*

$$P_\rho\Big(D_{KL}(\rho_\lambda\|\hat\rho) \le \eta\Big) \ge 1 - \pi, \quad \textit{with } \eta = \frac{V}{\alpha_0\pi},$$

where α_0 is the smallest eigenvalue of the Hessian matrix $\boldsymbol{H}(\hat{\boldsymbol{\lambda}}) = \partial^2 F(\hat{\boldsymbol{\lambda}})$.

Numerical issues

For usual choices of the moments $\boldsymbol{\phi}$ as Legendre polynomials and Fourier basis the corresponding Hessian matrix $\boldsymbol{H}(\boldsymbol{\lambda})$ is poorly conditioned when ρ is small on substantial part of Ω. In this case the moment functions are close to be dependent with respect to the scalar product:

$$(f, g)_\rho = \mathsf{E}_\rho[fg] \tag{5}$$

In order to improve the conditioning we can observe that Hessian for the exact $\boldsymbol{\lambda}$ can be expressed as expectation and therefore can be estimated by the MLMC estimator. Using the eigenvalue decomposition of the estimate we can find the set of moment functions that is orthonormal with respect to the scalar product (5). Then the Hessian matrices are close to the identity matrix which provides fast and stable solution to the minimization problem (4).

Numerical results

Ideas mentioned above have been applied to a test Darcy problem, where X was total flux through a unit square with correlated random conductivity field and pressure gradient boundary

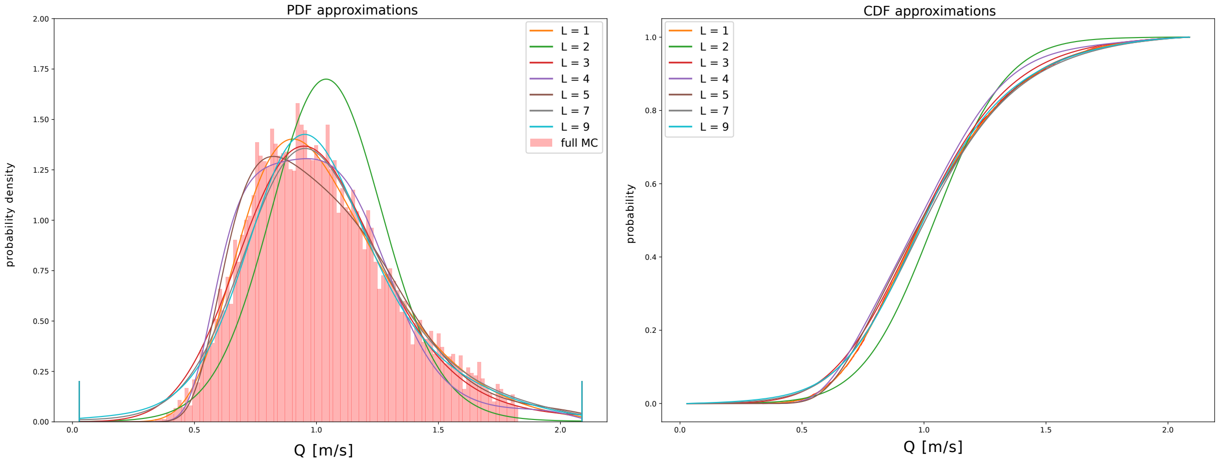


Figure 1: PDF and CDF approximation for the flux through a square domain with a random conductivity. Using 11 polynomial moments and varying number of levels in the MLMC estimator.

condition. Realizations have been calculated using Flow123d simulator [2]. Results of PDF approximation using 11 polynomial moments and various number of levels in MLMC is depicted at Figure. 1.

Conclusion

Combination of the maximal entropy method with multilevel Monte Carlo method provides an effective way for approximation of the PDF. Improved probabilistic characterization of the approximation error in probabilistic sense has been provided. A MLMC estimate of the covariance matrix has been used to construct natural moment functions for approximated density.

Acknowledgement: This work has been supported by the project “Geopax - Software pro komplexní a stochastické hydrogeologické modely” no. TH03010227 funded by TAČR.

References

[1] A.R. Barron, Ch.-H. Sheu: *Approximation of Density Functions by Sequences of Exponential Families. The Annals of Statistics*, 19(3):1347–1369, September 1991.

[2] J. Březina, J. Stebel, P. Exner, J. Hybš: *Flow123d.* `http://flow123d.github.com`, repository: `http://github.com/flow123d/flow123d`, 2011–2016.

[3] M.B. Giles: *Multilevel Monte Carlo methods. Acta Numerica*, 24:259–328, May 2015.

On vectorized MATLAB implementation of elastoplastic problems

M. Čermák[1,2,3,4], *S. Sysala*[4], *J. Valdman*[5,6]

[1] Department of Mathematics, Faculty of Civil Engineering, VŠB-TU Ostrava, Ostrava
[2] Department of Applied Mathematics, FEEIC, VŠB-TU Ostrava, Ostrava
[3] ENET Centre, VŠB-TU Ostrava, Ostrava
[4] Institute of Geonics of the Czech Academy of Sciences, Ostrava
[5] Institute of Mathematics and Biomathematics, University of South Bohemia, České Budějovice
[6] Institute of Information Theory and Automation of the Czech Academy of Sciences, Prague

1 Introduction

Vectorization in MATLAB replaces inefficient loops over long arrays by operations with matrices, mainly with sparse matrices. Vectorized codes are then reasonably scalable and fast for large size problems. In this contribution, we deal with a vectorized MATLAB implementation in 2D and 3D proposed in [1] for solution of elastoplastic problems. The related codes are available for download in [2].

Our implementation arises from a current elastoplastic solution scheme including time discretization by the implicit Euler method, construction of a constitutive operator and its generalized derivatives by the return-mapping algorithm, space discretization by the finite element method, and solution of nonlinear systems of equations by the semismooth Newton method. In [1], there is described in detail the implementation for models including von Mises and Drucker-Prager yield criteria. Similar implementation has been used for other yield criteria within numerical examples introduced in recent papers [3, 4, 5].

Further, one can optionally choose P1, P2, Q1 and Q2 finite elements with convenient quadrature rule for numerical integration. To be the codes universal, crucial functions are written uniformly regardless on the choice of elastoplastic models, finite elements or geometries.

The rest of this abstract describes main features of elastoplastic systems of nonlinear equations, assembling of the elastic and tangent stiffness matrices, and illustrative numerical results.

2 Elastoplastic system of nonlinear equations and its solution

Broadly speaking, in each time step of elastoplastic problems we solve a system of nonlinear equations of the following type:

$$\text{find } \ \boldsymbol{u} \in \mathbb{R}^n : \quad F(\boldsymbol{u}) = \boldsymbol{f}, \tag{1}$$

where $\boldsymbol{u}$ denotes the unknown displacement vector, $\boldsymbol{f} \in \mathbb{R}^n$ is the vector of external forces, and $F\colon \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function representing internal forces which is usually Lipschitz continuous and semismooth but nonsmooth in $\mathbb{R}^n$. Therefore, it is necessary to use the semismooth variant of the Newton method, see, e.g., [5]. In each Newton iteration $\ell = 1, 2, \ldots$, we solve a linear system of equations

$$\text{find } \ \delta\boldsymbol{u}^\ell \in \mathbb{R}^n : \quad \boldsymbol{K}_{tangent}\delta\boldsymbol{u}^\ell = \boldsymbol{f} - F(\boldsymbol{u}^\ell), \tag{2}$$

where $\delta\boldsymbol{u}^\ell \in \mathbb{R}^n$ is an unknown incremental vector, $\boldsymbol{u}^\ell \in \mathbb{R}^n$ is a previous iteration of $\boldsymbol{u}$, and $\boldsymbol{K}_{tangent} \in \mathbb{R}^{n\times n}$ is a tangential stiffness matrix representing a generalized derivative of F at $\boldsymbol{u}^\ell \in \mathbb{R}^n$.

The systems of nonlinear equations (1) have other specific features. First, if the load $\boldsymbol{f}$ is sufficiently small then solutions of elastic and elastoplastic problems usually coincide, i.e., $F(\boldsymbol{u}) = \boldsymbol{K}_{elast}\boldsymbol{u}$, where $\boldsymbol{K}_{elast} \in \mathbb{R}^{n\times n}$ is the corresponding elastic stiffness matrix. Further, for larger loads these solutions significantly differ and in addition, the solution $\boldsymbol{u}$ need not exist for some elastoplastic models due to the presence of limit loads [3, 4, 5]. In vicinity of the limit load, one can also observe locking phenomena and higher order finite elements are recommended. Then, assemblies of F and $\boldsymbol{K}_{tangent}$ require suitable quadrature rules of higher order. Finally, the definition of F is based on solution of the elastoplastic constitutive problems at each integration point of the investigated body. Such solutions (constitutive operators) are given in an implicit form and depend on history of loading. Therefore, constructions of F and $\boldsymbol{K}_{tangent}$ are technically complicated and not straightforward [4, 5].

3 Assembly of stiffness matrices $\boldsymbol{K}_{elast}$ and $\boldsymbol{K}_{tangent}$

Stiffness matrices based on the finite element method are usually assembled elementwisely by using local stiffness matrices. For example, one can write

$$\boldsymbol{K}_{elast} = \sum_{e=1}^{n_e} \boldsymbol{R}_e^\top \boldsymbol{K}_{e,elast} \boldsymbol{R}_e, \tag{3}$$

where n_e denotes a number of finite elements, $\boldsymbol{R}_e$ is a matrix restricting the displacement vector into its components belonging to a finite element and $\boldsymbol{K}_{e,elast}$ is the local stiffness matrix of the form

$$\boldsymbol{K}_{e,elast} = \sum_{q=1}^{n_q} \omega_{e,q} \boldsymbol{B}_{e,q}^\top \boldsymbol{C}_{e,q} \boldsymbol{B}_{e,q}. \tag{4}$$

Here, n_q is a number of quadrature points at any element, $\omega_{e,q}$ denotes quadrature weights, $\boldsymbol{B}_{e,q}$ is the strain-displacement matrix, and $\boldsymbol{C}_{e,q}$ is the elastic constitutive matrix following from the Hooke's law ($\boldsymbol{C}_{e,q} \in \mathbb{R}^{3\times 3}$ in 2D and $\boldsymbol{C}_{e,q} \in \mathbb{R}^{6\times 6}$ in 3D). For homogeneous materials, $\boldsymbol{C}_{e,q}$ is fixed for any element and any quadrature point.

The assembly of $\boldsymbol{K}_{elast}$ introduced in [1] arises from the following split:

$$\boldsymbol{K}_{elast} = \boldsymbol{B}^\top \boldsymbol{D}_{elast} \boldsymbol{B}, \tag{5}$$

where

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_{1,1}\boldsymbol{R}_1 \\ \boldsymbol{B}_{1,2}\boldsymbol{R}_1 \\ \vdots \\ \boldsymbol{B}_{1,n_q}\boldsymbol{R}_1 \\ \boldsymbol{B}_{2,1}\boldsymbol{R}_2 \\ \vdots \\ \vdots \\ \boldsymbol{B}_{n_e,n_q}\boldsymbol{R}_{n_e} \end{pmatrix}, \quad \boldsymbol{D}_{elast} = \begin{pmatrix} \tilde{\boldsymbol{C}}_{1,1} &&&&&& \\ & \tilde{\boldsymbol{C}}_{1,2} &&&&& \\ && \ddots &&&& \\ &&& \tilde{\boldsymbol{C}}_{1,n_q} &&& \\ &&&& \tilde{\boldsymbol{C}}_{2,1} && \\ &&&&& \ddots & \\ &&&&&& \ddots & \\ &&&&&&& \tilde{\boldsymbol{C}}_{n_e,n_q} \end{pmatrix},$$

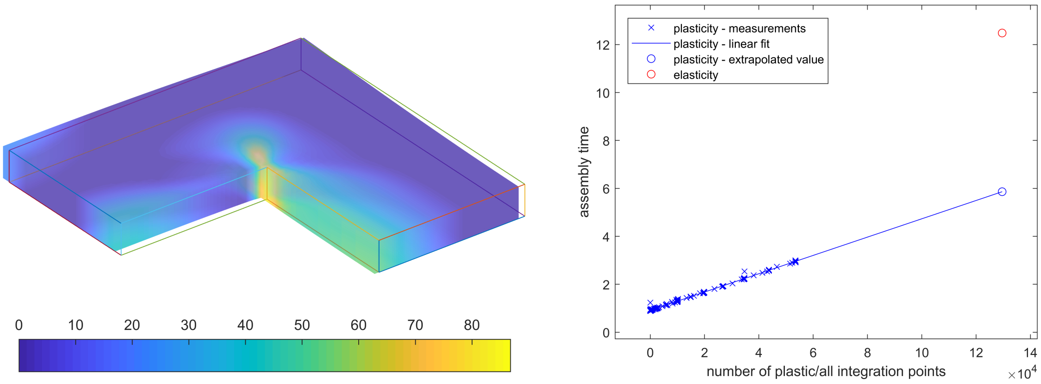


Figure 1: 3D problem with the von Mises yield criterion and kinematic hardening. Hardening field (left), assembly times of tangential stiffness matrix versus number of plastic integration points (right).

with $\tilde{\boldsymbol{C}}_{e,q} = \omega_{e,q}\boldsymbol{C}_{e,q}$, $e = 1, 2, \ldots, n_e$, $q = 1, 2, \ldots, n_q$. The matrices $\boldsymbol{B}$ and $\boldsymbol{D}_{elast}$ are large and sparse. Moreover, we see that $\boldsymbol{D}_{elast}$ is block diagonal. The multiplications in (5) are possible and convenient in MATLAB if these matrices are defined as sparse.

Similarly, one can assemble the tangent stiffness matrix for an elastoplastic problem:

$$\boldsymbol{K}_{tangent} = \boldsymbol{B}^\top \boldsymbol{D}_{tangent}\boldsymbol{B}. \tag{6}$$

Here, the matrix $\boldsymbol{D}_{tangent}$ has the same size and structure as $\boldsymbol{D}_{elast}$. Each block of $\boldsymbol{D}_{tangent}$ represents a generalized derivative of the elastoplastic constitutive operator at any integration point. Moreover, one can write [1]:

$$\boldsymbol{K}_{tangent} = \boldsymbol{K}_{elast} + \boldsymbol{B}^\top(\boldsymbol{D}_{tangent} - \boldsymbol{D}_{elast})\boldsymbol{B}, \tag{7}$$

Although (6) and (7) are algebraically identical, the form (7) is more convenient for MATLAB implementation since the sparse matrix $\boldsymbol{D}_{tangent} - \boldsymbol{D}_{elast}$ is typically sparser than $\boldsymbol{D}_{tangent}$. This occurs when most of integration points remains in the elastic phase. Therefore, for problems with smaller plastic regions, the assembly of the tangential stiffness matrix can be faster than for problems with larger plastic regions, see Figure 1 (left).

Finally, it is important to note that the matrices $\boldsymbol{K}_{elast}, \boldsymbol{B}, \boldsymbol{D}_{elast}$ can be precomputed and only the matrix $\boldsymbol{D}_{tangent}$ depends on a particular plasticity model and needs to be partially reassembled in each Newton iteration. Additionally, $\boldsymbol{B}$ can be also used for the assembly of the function F.

4 Illustrative numerical results

The first illustrative result is depicted in Figure 1. It is considered a 3D problem with L-shaped geometry and cycling loading. The body obeys the associative plastic flow rule and the linear kinematic hardening law. The von Mises yield criterion is used. The left figure visualizes zones with inelastic material response. The right figure compares assembly times of $\boldsymbol{K}_{tangent}$ at particular time steps and Newton iterations. We see that the assembly times linearly depend on numbers of elements with plastic response and are less than the assembly time of $\boldsymbol{K}_{elast}$.

The second illustrative result is depicted in Figure 2. It is considered a strip-footing problem under the plane strain assumption. The aim is to analyze bearing capacity of a soil foundation

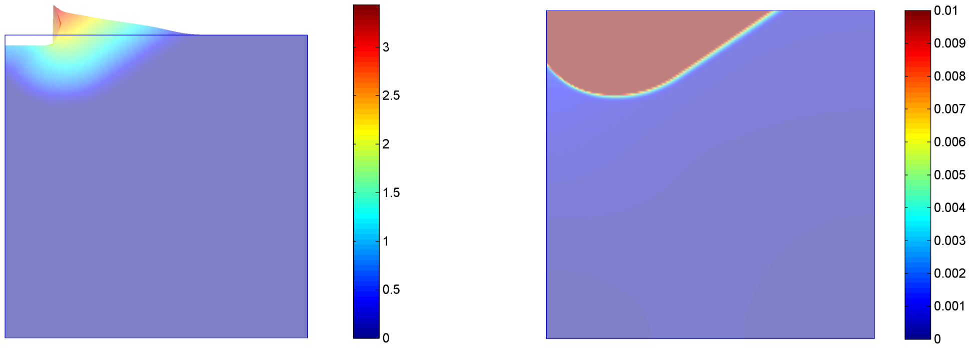


Figure 2: Strip-footing 2D problem solved by perfect plasticity with the Drucker-Prager yield criterion. Failure mechanism is visualized by the deform shape (left) and jumps in displacement fields (right).

and visualize the plastic collapse of the body. Monotone displacement loading is prescribed on the left part of the top. The body is perfectly plastic with the Drucker-Prager yield criterion. Failure mechanism is visualized by displacement fields and deformed shape. We observe significant jumps in displacement fields. The interface between the blue and red regions defines the expected failure zone.

Acknowledgement: This work has been supported by the Czech Science Foundation, project No. 19-11441S.

References

[1] M. Čermák, S. Sysala, J. Valdman: *Efficient and flexible Matlab implementation of 2D and 3D elastoplastic problems.* Applied Mathematics and Computation, 2019. Submitted article, available at arXiv:1805.04155.

[2] M. Čermák, S. Sysala, J. Valdman: *Matlab FEM package for elastoplasticity.* https://github.com/matlabfem/matlab_fem_elastoplasticity, 2018.

[3] S. Repin. S. Sysala, J. Haslinger: *Computable majorants of the limit load in Hencky's plasticity problems.* Computer and Mathematics with Applications **75** (2018), pp. 199–217.

[4] S. Sysala, M. Čermák, T. Koudelka, J. Kruis, J. Zeman, R. Blaheta: *Subdifferential- based implicit return-mapping operators in computational plasticity.* Zeitschrift für Angewandte Mathematik und Mechanik **96** (2016), pp. 1318–1338.

[5] S. Sysala, M. Čermák, T. Ligurský: *Subdifferential-based implicit return-mapping op- erators in Mohr-Coulomb plasticity.* Zeitschrift für Angewandte Mathematik und Mechanik, 97 (2017), pp. 1502–1523.

A posteriori error estimates in greedy reduced basis algorithms

M. Čertíková, L. Gaynutdinova, I. Pultarová

Czech Technical University in Prague

1 Introduction

The goal of reduced basis (RB) algorithms is to provide a *relatively small* set of functions which can serve as a basis for sufficiently accurate numerical solutions of some parametrized problem for *any* choice of parameters. An important part of *greedy* RB (GRB) algorithms is to estimate the difference between the exact solution of a discretized problem and its projection onto the space spanned by a reduced basis. We introduce a new kind of the estimate, which is based on a multilevel splitting of a discretized solution space, as an alternative to a widely used estimate based on bounds to coercivity and continuity constants. In our conference presentation, we introduce both algorithms, compare the guaranteed two-sided bounds obtained for both types of the error estimates and discuss the numerical complexity, accuracy and localization of errors.

2 The problem

We solve the problem to find $u : D \times \Gamma \to \mathbb{R}$ such that

$$\int_D a(\boldsymbol{x},\boldsymbol{\xi})\nabla u(\boldsymbol{x},\boldsymbol{\xi})\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_D f(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \text{for all } v \in V,\ \boldsymbol{\xi} \in \Gamma, \tag{1}$$

where $D \subset \mathbb{R}^2$ is a bounded polygonal domain, $V = W_0^{1,2}(D)$, $u(\boldsymbol{x},\boldsymbol{\xi}) = 0$ on $\delta D \times \Gamma$, $f \in L^2(D)$ and $a(\cdot,\boldsymbol{\xi}) \in L^\infty(D)$ for $\boldsymbol{\xi} \in \Gamma$. The gradient ∇ is considered with respect to the physical variable $\boldsymbol{x}$. The set $\Gamma \subset \mathbb{R}^K$ is usually a set of outcomes of K independent and identically distributed random variables which induce a metric in Γ. The coefficient $a(\boldsymbol{x},\boldsymbol{\xi})$ is considered in the affine form

$$a(\boldsymbol{x},\boldsymbol{\xi}) = a_0(\boldsymbol{x}) + \sum_{k=1}^{K} \xi_k a_k(\boldsymbol{x}), \tag{2}$$

where $a_k(\boldsymbol{x}) \in L^\infty(D)$, $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_K)$, $\xi_k \in \Gamma_k$, $\Gamma = \prod_{k=1}^K \Gamma_k$. We assume that there exist constants $0 < \alpha_1 \le \alpha_2 < \infty$, such that

$$\alpha_1 \le a(\boldsymbol{x},\boldsymbol{\xi}) \le \alpha_2 \quad \text{for all } \boldsymbol{x} \in D,\ \boldsymbol{\xi} \in \Gamma.$$

For the discretization of problem (1) with respect to the physical variable $\boldsymbol{x} \in D$, we employ the finite element (FE) method with continuous piece-wise bilinear basis functions $\psi_n(\boldsymbol{x})$, $n = 1,\ldots,N$, using a grid of N inner nodes in D. Let us denote the N-dimensional span of these functions by V_N. The discretized problem reads to find $u \in V_N$ such that

$$\int_D a(\boldsymbol{x},\boldsymbol{\xi})\nabla u(\boldsymbol{x},\boldsymbol{\xi})\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_D f(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \text{for all } v \in V_N,\ \boldsymbol{\xi} \in \Gamma. \tag{3}$$

Due to (2), the discretized problem (3) can be expressed in the matrix-vector form

$$\boldsymbol{A}(\boldsymbol{\xi})\boldsymbol{u}(\boldsymbol{\xi}) := \left(\boldsymbol{A}_0 + \sum_{k=1}^{K} \xi_k \boldsymbol{A}_k\right)\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{b}, \tag{4}$$

where $(\boldsymbol{A}_k)_{ns} = \int_D a_k(\boldsymbol{x})\nabla\psi_n(\boldsymbol{x})\nabla\psi_s(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, $\boldsymbol{b}_n = \int_D f(\boldsymbol{x})\psi_n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, $k = 0,1,\dots,K$, $n,s = 1,\dots,N$. The solution $\boldsymbol{u}(\boldsymbol{\xi}) \in \mathbb{R}^N$ of (4) is the coefficient vector of the solution $u(\boldsymbol{x},\boldsymbol{\xi}) \in V_N$ of (3), which means that they are connected via $u(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{n=1}^N u_n(\boldsymbol{\xi})\psi_n(\boldsymbol{x})$.

During the *offline phase*, the GRB algorithm finds a relatively small set (the *reduced basis*) of solutions of (3), $u_1,\dots,u_M \in V_N$, $M < N$, such that the Galerkin solutions of (3) found in the span of them are sufficiently accurate for all $\boldsymbol{\xi} \in \Gamma$. In the matrix-vector form, the reduced basis can be represented by the matrix $\boldsymbol{U}_M \in \mathbb{R}^{N\times M}$ with columns $\boldsymbol{u}_1,\dots,\boldsymbol{u}_M$ formed by the coefficient vectors with respect to the FE basis functions $\psi_1(\boldsymbol{x}),\dots,\psi_N(\boldsymbol{x})$ of the basis $\mathcal{U}_M = \{u_1(\boldsymbol{x}),\dots u_M(\boldsymbol{x})\}$. The name *reduced basis* will be used for $\mathcal{U}_M$ as wll as for the coefficient vectors $\boldsymbol{U}_M$. In the *online phase* of the GBR algorithm, for each $\boldsymbol{\xi} \in \Gamma$, the high-fidelity solution $\boldsymbol{u}(\boldsymbol{\xi})$ of (4) can be approximated by $\boldsymbol{U}_M\boldsymbol{u}_M^{\mathrm{RB}}(\boldsymbol{\xi})$ where $\boldsymbol{u}_M^{\mathrm{RB}}(\boldsymbol{\xi})$ is the RB solution of the RB problem

$$\boldsymbol{A}^{\mathrm{RB}}(\boldsymbol{\xi})\boldsymbol{u}^{\mathrm{RB}}(\boldsymbol{\xi}) := \left(\boldsymbol{A}_0^{\mathrm{RB}} + \sum_{k=1}^K \xi_k\boldsymbol{A}_k^{\mathrm{RB}}\right)\boldsymbol{u}^{\mathrm{RB}}(\boldsymbol{\xi}) = \boldsymbol{b}^{\mathrm{RB}}, \tag{5}$$

where $\boldsymbol{A}_k^{\mathrm{RB}} \in \mathbb{R}^{M\times M}$, $\boldsymbol{u}^{\mathrm{RB}}(\boldsymbol{\xi}), \boldsymbol{b}^{\mathrm{RB}} \in \mathbb{R}^M$, $\boldsymbol{A}_k^{\mathrm{RB}} = \boldsymbol{U}_M^T\boldsymbol{A}_k\boldsymbol{U}_M$, $k = 0,1,\dots,K$, and $\boldsymbol{b}^{\mathrm{RB}} = \boldsymbol{U}_M^T\boldsymbol{b}$.

3 A posteriori error estimates

Let us emphasize that the name *a posteriori error estimate* here and also, for example, in [3, 4, 5, 6], means the estimate of a difference (measured in some norm) between the Galerkin solutions $u_M(\boldsymbol{x},\boldsymbol{\xi}) \in V_M \subset V_N$ and $u_N(\boldsymbol{x},\boldsymbol{\xi}) \in V_N$. These solutions are the $(\cdot,\cdot)_{\boldsymbol{\xi}}$-orthogonal projections of the exact solution $u(\boldsymbol{x},\boldsymbol{\xi}) \in V$ of problem (1) onto V_M and V_N, respectively. However, a widely used meaning of the *a posteriori error estimate* is connected to an estimate of a distance of some approximate solution $u_M(\boldsymbol{x},\boldsymbol{\xi}) \in V_M$ from the *exact* solution $u(\boldsymbol{x},\boldsymbol{\xi}) \in V$ of (1). Such an estimate, however, cannot be obtained using only the discretized form of the problem. Some sophisticated construction is needed unless some other properties of the solutions are employed, such as in [1].

Let us denote by $E_M(\boldsymbol{\xi})$ the squared energy norm of the error $\boldsymbol{e}_M(\boldsymbol{\xi})$ of the approximate solution of the problem (4) found in the M-dimensional RB space for some parameter $\boldsymbol{\xi}$. Thus for the residual vector $\boldsymbol{r}_M(\boldsymbol{\xi}) = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{\xi})\boldsymbol{u}_M(\boldsymbol{\xi})$, we have the guaranteed error bounds

$$E_M(\boldsymbol{\xi}) = \boldsymbol{e}_M(\boldsymbol{\xi})^T\boldsymbol{A}(\boldsymbol{\xi})\boldsymbol{e}_M(\boldsymbol{\xi}) = \boldsymbol{r}_M(\boldsymbol{\xi})^T\boldsymbol{A}(\boldsymbol{\xi})^{-1}\boldsymbol{r}_M(\boldsymbol{\xi}).$$

3.1 Mean based a posteriori error estimate

A widely used guaranteed estimate of the energy norm $E_M(\boldsymbol{\xi})$ of the error is called the *mean-based estimate* (MB)

$$\frac{1}{\alpha_2(\boldsymbol{\xi})}\,\boldsymbol{r}_M(\boldsymbol{\xi})^T\boldsymbol{A}_0^{-1}\boldsymbol{r}_M(\boldsymbol{\xi}) \le E_M(\boldsymbol{\xi}) \le \frac{1}{\alpha_1(\boldsymbol{\xi})}\,\boldsymbol{r}_M(\boldsymbol{\xi})^T\boldsymbol{A}_0^{-1}\boldsymbol{r}_M(\boldsymbol{\xi}) \tag{6}$$

see, e.g. [5], where we can set

$$\alpha_2(\boldsymbol{\xi}) = \operatorname{ess\,inf}_{\boldsymbol{x}\in D}\frac{a(\boldsymbol{x},\boldsymbol{\xi})}{a_0(\boldsymbol{x})}, \qquad \alpha_2(\boldsymbol{\xi}) = \operatorname{ess\,sup}_{\boldsymbol{x}\in D}\frac{a(\boldsymbol{x},\boldsymbol{\xi})}{a_0(\boldsymbol{x})}.$$

In the GRB algorithms, instead of computing the constants $\alpha_1(\boldsymbol{\xi})$ and $\alpha_2(\boldsymbol{\xi})$ for every $\boldsymbol{\xi} \in \Gamma$ separately, their uniform lower and upper bounds can be used in (6). However, if the variation of $a(\boldsymbol{x},\boldsymbol{\xi})$ grows with respect to $\boldsymbol{x}$ and $\boldsymbol{\xi}$, these uniform bounds may become useless.

3.2 Multi-level a posteriori error estimate

The estimate of $E_M(\boldsymbol{\xi})$ suggested in this section is based on a hierarchy designed in the FE solution space. Let us consider a hierarchical two-level splitting of the solution space V_N into two subspaces, the coarse and the fine one. Let us denote the coefficient vectors of a function $u \in V_N$ with respect to the original FE space by $\boldsymbol{u} \in \mathbb{R}^N$ and with respect to the hierarchical two-level basis by

$$\boldsymbol{u}^{\mathrm{ML}} = \begin{pmatrix} \boldsymbol{u}^{\mathrm{ML,C}} \\ \boldsymbol{u}^{\mathrm{ML,F}} \end{pmatrix} \in \mathbb{R}^N, \quad \boldsymbol{P}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{u},$$

respectively, where $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ is a transformation matrix. Any system of linear equations $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}$ with respect to the original FE basis can be transformed into the system

$$\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{u}^{\mathrm{ML}} = \begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}} & \boldsymbol{A}^{\mathrm{ML,CF}} \\ \boldsymbol{A}^{\mathrm{ML,CF}T} & \boldsymbol{A}^{\mathrm{ML,F}} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{\mathrm{ML,C}} \\ \boldsymbol{u}^{\mathrm{ML,CF}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}^{\mathrm{ML,C}} \\ \boldsymbol{b}^{\mathrm{ML,CF}} \end{pmatrix} = \boldsymbol{b}^{\mathrm{ML}}$$

where we have

$$\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{P}^T\boldsymbol{b} = \boldsymbol{b}^{\mathrm{ML}}.$$

The *multi-level* (ML) guaranteed error bounds are defined as

$$E_M^{\mathrm{ML},1}(\boldsymbol{\xi}) \le E_M(\boldsymbol{\xi}) \le E_M^{\mathrm{ML},2}(\boldsymbol{\xi}),$$

where

$$\begin{aligned} E_M^{\mathrm{ML},1}(\boldsymbol{\xi}) &= \frac{1}{(1+\gamma)}\left(\boldsymbol{r}_M^{\mathrm{ML,C}}(\boldsymbol{\xi})^T\boldsymbol{A}^{\mathrm{ML,C}}(\boldsymbol{\xi})^{-1}\boldsymbol{r}_M^{\mathrm{ML,C}}(\boldsymbol{\xi}) + \frac{1}{\beta_2}\boldsymbol{r}_M^{\mathrm{ML,F}}(\boldsymbol{\xi})^T\boldsymbol{D}^{\mathrm{ML,F}}(\boldsymbol{\xi})^{-1}\boldsymbol{r}_M^{\mathrm{ML,F}}(\boldsymbol{\xi})\right) \\ E_M^{\mathrm{ML},2}(\boldsymbol{\xi}) &= \frac{1}{(1-\gamma)}\left(\boldsymbol{r}_M^{\mathrm{ML,C}}(\boldsymbol{\xi})^T\boldsymbol{A}^{\mathrm{ML,C}}(\boldsymbol{\xi})^{-1}\boldsymbol{r}_M^{\mathrm{ML,C}}(\boldsymbol{\xi}) + \frac{1}{\beta_1}\,\boldsymbol{r}_M^{\mathrm{ML,F}}(\boldsymbol{\xi})^T\boldsymbol{D}^{\mathrm{ML,F}}(\boldsymbol{\xi})^{-1}\boldsymbol{r}_M^{\mathrm{ML,F}}(\boldsymbol{\xi})\right), \end{aligned}$$

where $\boldsymbol{D}^{\mathrm{ML,F}}(\boldsymbol{\xi})$ is a diagonal matrix spectrally equivalent with $\boldsymbol{A}^{\mathrm{ML,F}}(\boldsymbol{\xi})$, i.e. there exist constants $0 < \beta_1 \le 1 \le \beta_2 < \infty$ such that

$$\beta_1\,\boldsymbol{v}^T\boldsymbol{D}^{\mathrm{ML,F}}(\boldsymbol{\xi})\boldsymbol{v} \le \boldsymbol{v}^T\boldsymbol{A}^{\mathrm{ML,F}}(\boldsymbol{\xi})\boldsymbol{v} \le \beta_2\,\boldsymbol{v}^T\boldsymbol{D}^{\mathrm{ML,F}}(\boldsymbol{\xi})\boldsymbol{v} \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^N.$$

The constants γ, β_1 and β_2 can be easily quantified or estimated under many practical and theoretical settings.

4 Discussion

In our contribution we introduce a new a posteriori error estimate for the GRB algorithm based on the multilevel splitting of the solution FE space which can serve as an alternative to the popular mean based estimate. Many numerical examples will complement this during the conference presentation. The two algorithms will be compared from the point of view of accuracy, complexity and memory efficiency. In this abstract, let us only introduce two graphs of differences between some exact error function $\boldsymbol{e}_M$ and its MB ($\boldsymbol{e}_M^{\mathrm{MB}}$) and ML ($\boldsymbol{e}_M^{\mathrm{ML}}$) estimates, respectively. See $\boldsymbol{e}_M$, $\boldsymbol{e}_M - \boldsymbol{e}_M^{\mathrm{MB}}$, and $\boldsymbol{e}_M - \boldsymbol{e}_M^{\mathrm{ML}}$ in Figure 1.

Acknowledgement: This work was supported by the Czech Science Foundation under the contract No. 17-04150J and by the Center of Advanced Applied Sciences, the European Regional Development Fund, project No. CZ.02.1.01/0.0/0.0/16_019/0000778.

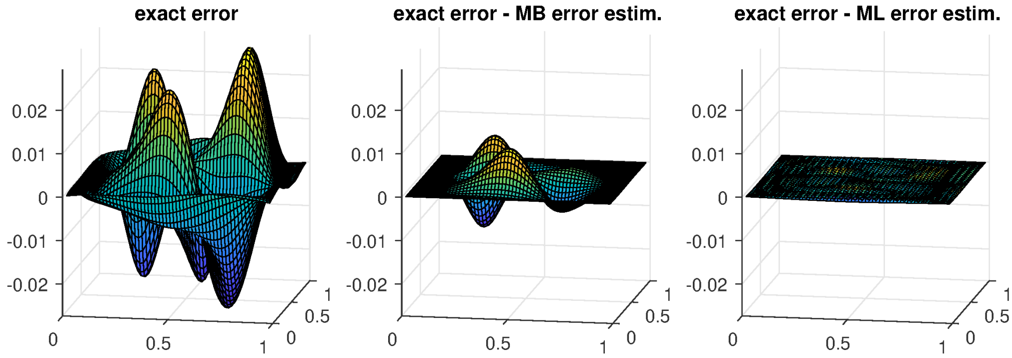


Figure 1: An example of an exact error function $\boldsymbol{e}_M$ (left), the difference between the exact error function and the MB estimate $\boldsymbol{e}_M - \boldsymbol{e}_M^{\mathrm{MB}}$ (middle), and the difference between the exact error function and the ML estimate $\boldsymbol{e}_M - \boldsymbol{e}_M^{\mathrm{ML}}$ (right).

References

[1] A. Buffa, Y. Maday, A.T. Patera, Ch. Prud'homme, G. Turinici: *A priori convergence of the greedy algorithm for the parametrized reduced basis method.* In: ESAIM: Mathematical Modelling and Numerical Analysis, 46, 2012, pp. 595–603.

[2] M. Čertíková, L. Gaynutdinova, I. Pultarová: *A posteriori error estimates in greedy reduced basis algorithms.* Submitted.

[3] L. Giraldi, A. Litvinenko, D. Liu, H.G. Matthies, A. Nouy: *To be or not to be intrusive? The solution of parametric and stochastic equations–the "plain vanilla" Galerkin case.* In: SIAM Journal on Scientific Computing, 36(6), 2014, pp. A2720–A2744.

[4] P. Chen, A. Quarteroni, G. Rozza: *Comparison between reduced basis and stochastic collocation methods for elliptic problems.* In: Journal of Scientific Computing, 59(1), 2014, pp. 187–216.

[5] P. Chen, A. Quarteroni, G. Rozza: *Reduced basis mehod for uncertainty quantification.* In: SIAM/ASA Journal on Uncertainty Quantification, 5(1), 2017, pp. 813–869.

[6] G. Rozza, D.B.P. Huynh, A.T. Patera: *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations.* In: Archives of Computational Methods in Engineering, 15, 229, 2008.

The effect of uncertain input data on drying creep and drying shrinkage of concrete

L. Dohnalová, J. Chleboun

Faculty of Civil Engineering, Czech Technical University in Prague

1 Introduction

Although the short-term evolution of principal mechanical properties of concrete is the subject of numerous measurements and, consequently, can be considered sufficiently understood (at least in common situations), much less is known about the long-term behavior of concrete structures. Since their lifespan is expected to reach at least several decades or, better, a century, reliable predictive models are highly desirable.

Owing to the complexity of phenomena and parameters contributing to the ageing of concrete, phenomenological features of long-term models as well as model calibration are closely interconnected. The latter is quite difficult and burdened with uncertainty due to the lack of relevant and high-quality measurements. Indeed, long-term tests of concrete samples stored under controlled conditions (different humidity levels, for instance) are expensive and exceeding the duration of common research projects. Moreover, both the variability of factors that have a significant impact on the material parameters of concrete and different designs of experiments have resulted in output data sets that are only partially compatible.

As a consequence, the amount of data suitable for modeling the long-term behavior of concrete is limited. For example, the database [4] contains more than 60 thousand records coming from 362 experimental surveys but for the specific purposes of [2, 3] only a few surveys and a few dozen records were relevant and sufficiently compatible.

Two fundamental phenomena are in the focus of civil engineers, namely concrete creep and shrinkage. The former appears in loaded structural elements and two forms are generally distinguished: basic creep in a high humidity environment (no drying) and drying creep, a contribution added to the basic creep and caused by the drying of concrete. The shrinkage phenomenon is not related to loading and is demonstrated through volumetric changes of structural elements and the formation of mechanical stress resulting in cracks. Again, different shrinkage sources can be identified. We will limit our attention to drying creep and drying shrinkage only.

In modeling drying creep and drying shrinkage, two respective functions are used: $J_{\mathrm{d}}(t, t', t_0)$ (drying creep compliance function) and $\varepsilon_{\mathrm{sh}}(t, t_0)$ (drying shrinkage function), where t is the current time (i.e., the age of the concrete specimen), t' and t_0 are the respective times of the origin of loading and drying. All times are in days.

Among several models and codes widely used in long-term creep and shrinkage predictions, Model B3 [1] has gained wide recognition. In this model,

$$J_{\mathrm{d}}(t, t', t_0) = q\sqrt{\mathrm{e}^{-8H(t)} - \mathrm{e}^{-8H(t_0')}},$$

where q is an aggregate parameter comprising a number of other parameters, $t_0' = \max\{t', t_0\}$, and $H(t) = 1 - (1 - h)S(t)$, where h stands for the relative humidity of the environment, and $S(t) = \tanh\sqrt{\frac{t-t_0}{\tau_{\mathrm{sh}}}}$. In the last expression, $\tau_{\mathrm{sh}} = 8.5 \times 10^4 t_0^{-0.08} \overline{f}_c^{\,-1/4} (k_s D)^2$, where $\overline{f}_c^{\,-1/4}$ is the 28 day mean cylinder compression strength of concrete and the product $k_s D$ represents some geometric features of the structural element.

In Model B3, the shrinkage function is defined as follows:

$$\varepsilon_{\mathrm{sh}}(t, t_0) = -\varepsilon_{\mathrm{sh}}^{\infty} k_h S(t),$$

where $\varepsilon_{\mathrm{sh}}^{\infty}$ is the ultimate shrinkage parameter and k_h is a humidity dependent parameter.

2 Uncertainty quantification

The application of the functions J_{d} and $\varepsilon_{\mathrm{sh}}$ in a long-term creep and shrinkage analysis is accompanied by uncertainty. For instance, the parameters that constitute the aggregate factors q, τ_{sh}, and $\varepsilon_{\mathrm{sh}}^{\infty}$ are not known exactly or exhibit natural variability. Although the model is presented in a deterministic way in [1], the reader is warned that the input parameters should be considered as normally distributed mutually independent probabilistic variables; their variability is estimated to give the analyst some guidance for a robust design of concrete structural elements.

The amount of uncertainty is even higher if predictions are to be made for a structure with incomplete records about the technological history of concrete it is made of. This and concerns about underestimated results if independent probabilistic parameters were used in predictions have led us to the application of a fuzzy set approach.

The constituents of J_{d} and $\varepsilon_{\mathrm{sh}}$ are modeled by fuzzy numbers and the fuzzy functions representing the drying creep and drying shrinkage are inferred through standard technique of finding the worst- and best-case scenarios on a series of α-level subsets of a fuzzy set of admissible input parameters. An interplay between the creep and shrinkage functions is also determined.

Acknowledgement: This research has been performed in the Center of Advanced Applied Sciences and (partially) supported by the European Regional Development Fund under project No. CZ.02.1.01/0.0/0.0/16_019/0000778.

References

[1] Z.P. Bažant, S. Baweja: *Creep and shrinkage prediction model for analysis and design of concrete structures: Model B3 – Short Form.* In: A. Al-Manaseer (ed.): Adam Neville Symposium: Creep and Shrinkage – Structural Design Effects, American Concrete Institute SP–194, 2000, pp. 85-100. An extended form is available at `http://www.civil.northwestern.edu/people/bazant/PDFs/Papers/S39.pdf` (last visited Dec. 16, 2018).

[2] L. Dohnalová: *Comparison of shrinkage and drying creep kinetics of concrete.* Bachelor's thesis (in Czech), Faculty of Civil Engineering, Czech Technical University in Prague, 2018.

[3] L. Dohnalová, P. Havlásek: *Comparison of drying shrinkage and drying creep kinetics in concrete.* Acta Polytechnica (to appear).

[4] M.H. Hubler, R. Wendner, Z.P. Bažant: *Comprehensive database for concrete creep and shrinkage: Analysis and recommendations for testing and recording.* ACI Materials Journal 112 (2015), pp. 547–558.
The database is available at `http://www.civil.northwestern.edu/people/bazant/` under NU Database of Laboratory Creep and Shrinkage Data (last visited Dec. 16, 2018).

On a numerical solution of degenerated parabolic problems

V. Dolejší[1], *M. Kuráž*[2]

[1] Faculty of Mathematics and Physics, Charles University in Prague
[2] Faculty of Environmental Sciences, Czech University of Life Sciences, Prague

1 Introduction

We deal with the numerical solution of nonlinear parabolic equations where the diffusion as well as time derivative term can degenerate. Such type of equations describe, e.g., water flow in unsaturated/saturated porous media, water and soil pollution, CO_2 storage, enhanced oil recovery and nuclear waste management. The degeneracies can cause troubles in the proposals of suitable numerical schemes and in the convergence of solvers for the arising algebraic systems. In this contribution, we focus on the presentation of arising troubles and their possible solution.

2 Model problem

A typical example of a *degenerate parabolic equation* is the Richards equation [12] describing water flow in a unsaturated/saturated porous medium. It can be written in the form

$$\frac{\partial \vartheta(\psi)}{\partial t} - \nabla \cdot (\mathbf{K}(\psi)\nabla\Psi) = 0, \tag{1}$$

where Ψ is the hydraulic head [L], ψ is the pressure head, the relation between the pressure head and the hydraulic head states as $\Psi = \psi + z$, where z is the geodetic head [L] (distance from the reference level), $\mathbf{K}(\psi)$ is the unsaturated hydraulic conductivity tensor of the second order [L.T^{-1}] and the derivative of ϑ satisfies

$$\vartheta'(\psi) := \frac{\mathrm{d}\vartheta(\psi)}{\mathrm{d}\psi} = \frac{\mathrm{d}\theta(\psi)}{\mathrm{d}\psi} + \frac{\theta(\psi)}{\theta_S} S_s, \tag{2}$$

where $\theta(\psi)$ is the water content function [-], S_s is the specific aquifer storage [L^{-1}], θ_S is the saturated water content [-]. A constitutive relation for the function $\theta(\psi)$ is given by van Genuchten's law [13], and for the function $\mathbf{K}(\psi) = \mathbf{K}_s K_r(\theta(\psi))$ by Mualem's law [10] ($\mathbf{K}_s$ is the saturated hydraulic conductivity and $K_r(\theta(\psi))$ is the relative hydraulic conductivity). Figure 1 shows an example of functions $\vartheta'(\psi)$ and $\mathbf{K}(\psi)$ for three different materials for $\psi \in [-30, 10]$. The constitutive relations for the function ϑ and $\mathbf{K}$ satisfy the following assumptions which can lead to several *degeneracies* of (1):

(A1) the function $\vartheta : \mathbb{R} \to \mathbb{R}$, $\vartheta(\psi) = \vartheta(\Psi - z)$ is a Hölder continuous and non-decreasing with a non-negative derivative $\vartheta'(\psi) \geq 0$; if $S_s = 0$ then $\vartheta'(\psi) = 0$ in a fully saturated flow regime ($\psi \geq 0$) and consequently equation (1) degenerates to an elliptic one (the *fast-diffusion* type of degeneracy),

(A2) for a particular choice of the material parameters in the constitutive relations, one can have $\vartheta'(\psi) \to \infty$ as $\psi \to 0$ (the *slow-diffusion* type of degeneracy); however, for the most real materials, $\vartheta'(\psi)$ can be large but bounded,

(A3) the function $\mathbf{K} : \mathbb{R} \to \mathbb{R}^{2\times 2}$ is a positive, Lipschitz continuous and nondecreasing, it can vanish, typically $\mathbf{K}(\psi) \to 0$ for $\psi \to -\infty$ (the *slow-diffusion* degeneracy),

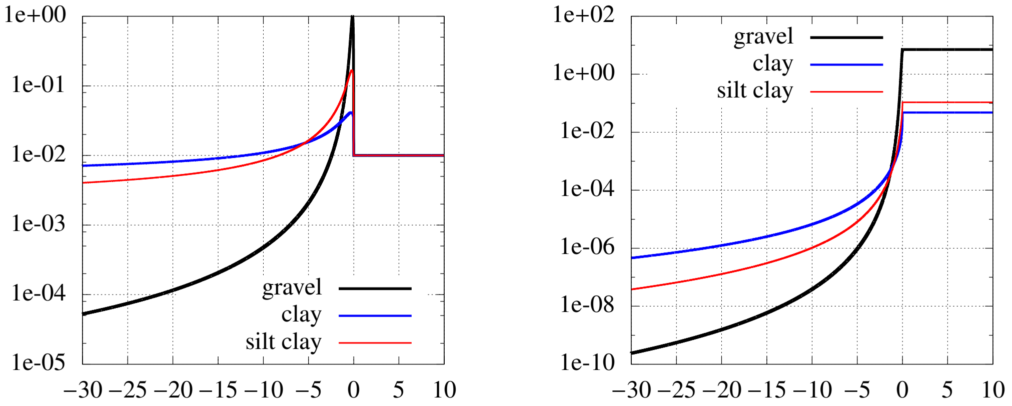


Figure 1: Examples of $\vartheta'(\psi)$ (left) and $\mathbf{K}(\psi)$ (right) for three different materials.

3 Discretization

The properties mentioned above make the solution of (1) a challenging task. Nowadays, there is a variety of numerical methods which have been proposed, analyzed and tested in the last decays for the solution of degenerate parabolic problems. Let us mention the conforming finite element methods developed in [6], the mixed finite element methods treated in [1, 11] and the finite volume based technique developed by [8]. Finally, papers [2, 7] deals with the numerical solution of two-phase flow in porous media by the discontinuous Galerkin method [4]. Due to the stiffness character of the governing equations, the (semi-)implicit time discretization is advantageous. Usually, the lowest order backward Euler method is employed. In [2], the diagonally implicit Runge-Kutta schemes of order two and three are used.

For the discretization of (1) we employ the *space-time discontinuous Galerkin method* which offer a high-order approximation with respect to the time and space. Let (1) be considered on the computational domain Ω and time interval $(0, T)$, $T > 0$. Let $0 = t_0 < t_1 < \ldots < t_r = T$ be a partition of $(0, T)$ generating time intervals $I_m = (t_{m-1}, t_m]$, $m = 1, \ldots, r$. For every time interval I_m, $m = 1, \ldots, r$ we consider generally different space partition $\mathscr{T}_{h,m}$ of Ω consisting of a finite number of closed triangles K.

The approximate solution is sought in the space of discontinuous piecewise-polynomial functions

$$S_{h,p}^{\tau,q} := \{\varphi : \Omega \times (0, T) \to \mathbb{R};\ \varphi|_{K \times I_m} \in P^{p_K}(K) \times P^q(I_m),\ K \in \mathscr{T}_{h,m},\ m = 1, \ldots, r\},$$

where $P^{p_K}(K) \times P^q(I_m)$ is the space of polynomials on $K \times I_m$ of the degree $\leq p_K$ with respect to $x \in K$ and the degree $\leq q$ with respect to $t \in I_m$ for $K \in \mathscr{T}_{h,m}$ and $m = 1, \ldots, r$.

The function $\Psi_{h\tau} \in S_{h,p}^{\tau,q}$ is an *approximate solution* of (1) if

$$A_{h,m}(\Psi_{h\tau}, \varphi) = 0 \quad \forall \varphi \in S_{h,p}^{\tau,q},\ m = 1, \ldots, r, \tag{3}$$

where

$$A_{h,m}(\Psi, \varphi) = \int_{I_m} \left((\partial_t \vartheta(\Psi - z), \varphi) + a_{h,m}(\Psi, \varphi) \right) \mathrm{d}t \ + \left(\{\vartheta(\Psi - z)\}_{m-1}, \varphi|_{m-1}^{+} \right), \tag{4}$$

$a_{h,m}(\cdot, \cdot)$ represents the usual discretization of the term $-\nabla \cdot (\mathbf{K}(\Psi - z)\nabla\Psi)$ by the discontinuous Galerkin method ([4, Chapters 2 and 6]), $(\cdot, \cdot)$ denotes the L^2-scalar product over Ω and $\{\cdot\}_m$ denotes the jump of the argument with respect to the time at $t = t_m$.

4 Solution of the resultig algebraic systems

The relation (4) exhibits a system of strongly nonlinear algebraic equations whose numerical solution is a difficult task. The popular method Newton method often fails for parabolic degenerate problems since the Jacobian might become singular, see, e.g., [9]. In [3], the modified Picard method was developed, it is more robust but still requires the derivatives of ϑ and it might also fail to converge, see [9]. This drawback can be overcome using the L-scheme developed in [11], which exploits the monotonicity of ϑ. In comparison to the modified Picard method the derivative $\vartheta'(\psi)$ is replaced by the constant $L \geq \max_\psi |\vartheta'(\psi)|$. These techniques together with their combinations were analyses and numerically compared in [9].

Due to the properties of $A_{h,m}$, $m = 1, \dots,$ given by (4), its is possible to construct forms $A^{\mathrm{L}}_{h,m}(\cdot,\cdot,\cdot) : S^{\tau,q}_{h,p} \times S^{\tau,q}_{h,p} \times S^{\tau,q}_{h,p} \to \mathbb{R}$, $m = 1, \dots,$ which are linear with respect to their second and third arguments and are consistent with $A_{h,m}$ by

$$A_{h,m}(\Psi,\varphi) \approx A^{\mathrm{L}}_{h,m}(\Psi,\Psi,\varphi) - D_{h,m}(\Psi,\varphi) \quad \forall \Psi,\varphi \in S^{\tau,q}_{h,p},\ m = 1,\dots,r, \tag{5}$$

where $D_{h,m}(\Psi,\varphi)$ is a form vanishing for most $\varphi \in S_{hp}$.

The linearization (5) is a base of two iterative techniques derived in [5]. The first one is the damped Newton-like method where the Jacobian is replaced by the *flux matrix* defined by $A^{\mathrm{L}}_{h,m}$. The second approach is the adoption of the *Anderson acceleration for Picard method* [14] which (for linear problems) is equivalent to the GMRES method.

5 Regularization

The approach briefly described above is no able to avoid the troubles arising in the degeneracy mentioned at the end of §2. In order to improve the robustness of the method some *regularizations* of the problem are required.

Case (A1) (fast diffusion) For a vanishing specific aquifer storage $S_S = 0$ and a fully saturated medium $\psi \geq 0$ ($\Rightarrow \ \theta(\psi) = \text{const}$), one has $\vartheta'(\psi) = 0$ and then (1) degenerates to (time-independent) elliptic equation. This type of degeneracy is often solved using the replacing $\theta(\psi)$ by $\theta(\psi) + \epsilon\psi$, where $\epsilon > 0$ is a small regularization parameter. The adding of factor $\epsilon\psi$ can be interpreted as an *artificial specific storage.*

Case (A2) (slow diffusion) For some (realistic) values of material parameters in van Genuchten constitutive relation [13], $\vartheta'(\psi)$ has steep gradient for $\psi \to 0$. In order to avoid this troubles, we slightly modify the relations for $\theta(\psi)$ in order to improve the convergence properties but keep the accuracy as much as possible. For a small $\psi_{\mathrm{R}} > 0$, we replace $\theta(\psi)$ on the interval $(-\psi_{\mathrm{R}}, 0)$ by a *cubic polynomial function* which is uniquely defined by four values: $\theta(0)$, $\theta(-\psi_{\mathrm{R}})$, $\theta'(0)$, $\theta'(-\psi_{\mathrm{R}})$. Obviously, we put $\theta'(0) = 0$ and the $\theta'(-\psi_{\mathrm{R}})$ is approximated numerically by a central difference using an explicit knowledge of $\theta(\psi)$.

Case (A3) (slow diffusion) Figure 1 shows that $\mathbf{K}(\psi) \to 0$ for $\phi \to -\infty$. However, this type of degeneracy causes mainly the troubles in numerical analysis when the absence of lower bound has to be overcome, e.g., by a suitable regularization. In practical computations we did not meet any trouble concerning this type of slow diffusion degeneracy.

Numerical experiments demonstrate that the regularizations do not seriously affect the results but significantly accelerate the computational process. A similar type of regularization can be employed for the *seepage face boundary condition* which can be formulated as the Signorini type boundary condition or as the nonlinear Robin boundary condition.

Acknowledgement: This work has been supported by the grant No. 17-01747S of the Czech Science Foundation.

References

[1] T. Arbogast, M.F. Wheeler, N.-Y. Zhang: *A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media*, SIAM J. Numer. Anal. **33** (1996), no. 4, pp. 1669–1687.

[2] P. Bastian: *A fully-coupled discontinuous Galerkin method for two-phase flow in porous media with discontinuous capillary pressure*, Comput. Geosci. **18** (2014), no. 5, pp. 779–796.

[3] M.A. Celia, E.T. Bouloutas, R.L. Zarba: *A general mass-conservative numerical-solution for the unsaturated flow equation*, Water Resources Research **26** (1990), no. 7, pp. 1483–1496.

[4] V. Dolejší, M. Feistauer: *Discontinuous galerkin method – analysis and applications to compressible flow*, Springer Series in Computational Mathematics 48, Springer, Cham, 2015.

[5] V. Dolejší, M. Kuráž, P. Solin: *Adaptive higher-order space-time discontinuous Galerkin method for the computer simulation of variably-saturated porous media flows*, Applied Mathematical Modelling ((submitted)).

[6] C. Ebmeyer: *Error estimates for a class of degenerate parabolic equations*, SIAM J. Numer. Anal. **35** (1998), no. 3, pp. 1095–1112.

[7] Y. Epshteyn, B. Rivière: *Analysis of hp discontinuous Galerkin methods for incompressible two-phase flow*, J. Comput. Appl. Math. **225** (2009), no. 2, pp. 487–509.

[8] R. Eymard, D. Hilhorst, M. Vohralík: *A combined finite volume-nonconforming/mixed-hybrid finite element scheme for degenerate parabolic problems*, Numerische Mathematik **105** (2006), no. 1, pp. 73–131.

[9] F. List, F.A. Radu: *A study on iterative methods for solving Richards' equation*, Comput. Geosci. **20** (2016), no. 2, pp. 341–353.

[10] Y. Mualem: *A new model for predicting the hydraulic conductivity of unsaturated porous media*, Water Resources Research **12** (1976), no. 3, pp. 513–522.

[11] I.S. Pop, F. Radu, P. Knabner: *Mixed finite elements for the Richards' equation: linearization procedure*, J. Comput. Appl. Math. **168** (2004), no. 1-2, SI, pp. 365–373.

[12] L.A. Richards: *Capillary conduction of liquids through porous mediums*, Journal of Applied Physics **1** (1931), no. 5, pp. 318–333.

[13] M.Th. van Genuchten: *Closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, Soil Science Society of America Journal **44** (1980), no. 5, pp. 892–898.

[14] H.F. Walker, P. Ni: *Anderson acceleration for fixed-point iterations*, SIAM J. Numer. Anal. **49** (2011), no. 4, pp. 1715–1735. MR 2831068

Bayesian inversion using surrogate models with applications in porous media flow

S. Domesová[1,2], *M. Béreš*[1,2], *R. Blaheta*[1]

[1] Institute of Geonics of the CAS, Ostrava
[2] Department of Applied Mathematics, FEECS, VŠB - Technical University of Ostrava

1 Introduction to the Bayesian inversion

In mathematical modeling, we encounter two different kinds of problems - direct and inverse. Let us consider a simulation of processes described by the following elliptic boundary value problem in the domain Ω with boundary $\partial\Omega$,

$$\begin{aligned} -\operatorname{div}\left(k\nabla v\right) &= f \ \text{ in } \Omega, \\ v \text{ fulfils boundary conditions } & \text{ on } \partial\Omega. \end{aligned}$$

In this case, a *direct problem* means a solution of the boundary value problems with given data, i.e. the coefficient k, the right hand side f and the coefficients inside boundary conditions. Under natural assumptions, the direct problem is well posed and can be numerically solved after a suitable discretization (e.g. using the finite element method). We assume that all input data with the exception of the coefficient k are fixed and $k = k\left(\mathbf{u}\right)$ depends on a vector of parameters $\mathbf{u} \in \mathbb{R}^n$ (e.g. values of k in defined subdomains $\Omega_i \subset \Omega$). In addition, we consider the mapping $G\left(\mathbf{u}\right) = \mathbf{y}$, where $\mathbf{y} \in \mathbb{R}^m$ are outputs derived from the finite element solution. The mapping G will be called a forward model. If the output values $\mathbf{y}$ or their approximations $\mathbf{y}_\mathrm{m} \approx \mathbf{y}$ are given (e.g. obtained by measurements), then an *inverse problem* can be formulated as “Find $\mathbf{u} \in \mathbb{R}^n$ such that $G(\mathbf{u}) = \mathbf{y}_\mathrm{m}$,” or more generally as “Seek for $\mathbf{u} \in \mathbb{R}^n$ such that $\|G(\mathbf{u}) - \mathbf{y}_\mathrm{m}\|$ is minimal.”

However, the inverse problems are generally not well posed and straightforward optimization may fail (especially when the measurements are corrupted by noise). This is a motivation for the Bayesian approach, which naturally includes statistical characterization of the measurements. This stochastic approach doesn’t aim at determining $\mathbf{u} \in \mathbb{R}^n$ as a point value but at a statistical characterization of $\mathbf{u}$. The vector $\mathbf{u}$ is treated as a random vector and uncertainties in the observed data are included in the form of a probability distribution of the measurement error. Furthermore, a prior knowledge of the parameters available from experience (independent of the measurements) can also be included.

Briefly, the aim of the Bayesian approach is to describe the joint probability distribution of the random vector $\mathbf{u} \in \mathbb{R}^n$ called posterior distribution. The posterior probability density function (pdf) $\pi\left(\mathbf{u}|\mathbf{y}_\mathrm{m}\right)$ is given by the Bayes’ theorem as

$$\pi\left(\mathbf{u}|\mathbf{y}_\mathrm{m}\right) \propto f_{\boldsymbol{\eta}}\left(\mathbf{y}_\mathrm{m} - G\left(\mathbf{u}\right)\right)\pi_0\left(\mathbf{u}\right),$$

where $f_{\boldsymbol{\eta}}$ is the pdf of the noise, π_0 is the prior pdf and $\propto$ denotes a proportionality. Notice that this model considers additive noise $\boldsymbol{\eta} \in \mathbb{R}^m$ such that $\mathbf{y}_\mathrm{m} = G\left(\mathbf{u}\right) + \boldsymbol{\eta}$, typically from $\mathcal{N}\left(\mathbf{0}, \Sigma\right)$. An alternative would be e.g. multiplicative noise $\boldsymbol{\eta} \in \mathbb{R}^m$ such that $\mathbf{y} = G\left(\mathbf{u}\right) \cdot \boldsymbol{\eta}$, typically from $\mathcal{N}\left(\mathbf{1}, \Sigma\right)$. Here, Σ denotes a covariance matrix of a multivariate Gaussian distribution. The computational complexity of the Bayesian inversion is given by the process of generating samples from the posterior distribution using Markov chain Monte Carlo methods.

In comparison to standard deterministic approaches using optimization methods, the Bayesian approach provides more information about the unknown parameters, it is more robust, but also more expensive. For more details see e.g. [1]. This contribution is devoted to the numerical realization of the Bayesian inversion and especially to the use of surrogate models for the acceleration of the computations.

2 Posterior sampling using surrogate models

To provide samples from the posterior distribution, we use the delayed acceptance Metropolis-Hastings (DAMH) algorithm, see [2]. In comparison to the standard Metropolis-Hastings algorithm (see [3]), the target pdf (here $\pi\left(\mathbf{u}|\mathbf{y}_m\right)$) is not evaluated in each step. DAMH works both with $\pi\left(\mathbf{x}|\mathbf{y}_m\right)$ and with its approximation $\widetilde{\pi}\left(\mathbf{x}|\mathbf{y}_m\right)$, see Alg. 1. We construct this approximation (up to a multiplicative constant) as $f_{\boldsymbol{\eta}}\left(\mathbf{y}_\mathrm{m}-\widetilde{G}\left(\mathbf{u}\right)\right)\pi_0\left(\mathbf{u}\right)$, where $\widetilde{G}:\mathbb{R}^n\to\mathbb{R}^m$ is a surrogate model of G. As proposal distribution, the symmetric Gaussian random walk distribution is chosen; see Sec. 3 for the choice of its standard deviation (std).

```
• Choose an initial sample $\mathbf{u}^{(1)}$.
• For $t = 1, 2, \ldots, T$
    – generate $\mathbf{x}$ from proposal pdf $q\left(\mathbf{x}|\mathbf{u}^{(t)}\right)$,
    – pre-accept $\mathbf{x}$ with probability $\widetilde{\alpha}\left(\mathbf{u}^{(t)},\mathbf{x}\right) = \min\left\{1, \frac{\widetilde{\pi}(\mathbf{x}|\mathbf{y}_m)}{\widetilde{\pi}\left(\mathbf{u}^{(t)}|\mathbf{y}_m\right)}\right\}$,
        ∗ set $\mathbf{u}^{(t+1)} = \mathbf{x}$ (i.e. accept $\mathbf{x}$) with probability
          $\alpha\left(\mathbf{u}^{(t)},\mathbf{x}\right) = \min\left\{1, \frac{\pi(\mathbf{x}|\mathbf{y}_m)}{\pi\left(\mathbf{u}^{(t)}|\mathbf{y}_m\right)}\frac{\widetilde{\pi}\left(\mathbf{u}^{(t)}|\mathbf{y}_m\right)}{\widetilde{\pi}(\mathbf{x}|\mathbf{y}_m)}\right\}$,
        ∗ otherwise set $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)}$,
    – otherwise set $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)}$.
```

Algorithm 1: DAMH algorithm with symmetric proposal distribution

For the construction of surrogate models, we use either the stochastic collocation method (SCM) or the radial basis functions interpolation (RBF). Intrusive approaches (such as the stochastic Galerkin method) can be also used, see [1]. However, here we focus on non-intrusive approaches; this allows us to update the surrogate model during the sampling process.

Let us briefly describe the use of SCM. Consider $L^2_{\mathrm{d}F\mathbf{Z}}\left(\mathbb{R}^n\right)$ space with inner product $\left(g,f\right)_{L^2_{\mathrm{d}F\mathbf{Z}}}=\int_{\mathbb{R}^n} g\left(\mathbf{Z}\right)f\left(\mathbf{Z}\right)\mathrm{d}F\mathbf{Z}$ and its subspace S with a basis of polynomials $p_1,\ldots,p_N$ (not necessarily orthogonal). $G\in L^2_{\mathrm{d}F\mathbf{Z}}\left(\mathbb{R}^n\right)$ can be approximated with its orthogonal projection $\widetilde{G}_l=\sum_{i=1}^N\alpha_i p_i\in S$, such as $\left(G_l-\widetilde{G}_l,p_j\right)_{L^2_{\mathrm{d}F\mathbf{Z}}}=0\ \forall j\in\{1,\ldots,N\}$. The coefficients α_i are then determined by

$$\left(G_l,p_j\right)_{L^2_{\mathrm{d}F\mathbf{Z}}}=\sum_{i=1}^N\alpha_i\left(p_i,p_j\right)_{L^2_{\mathrm{d}F\mathbf{Z}}}\quad\forall j\in\left\{1,\ldots,N\right\}.$$

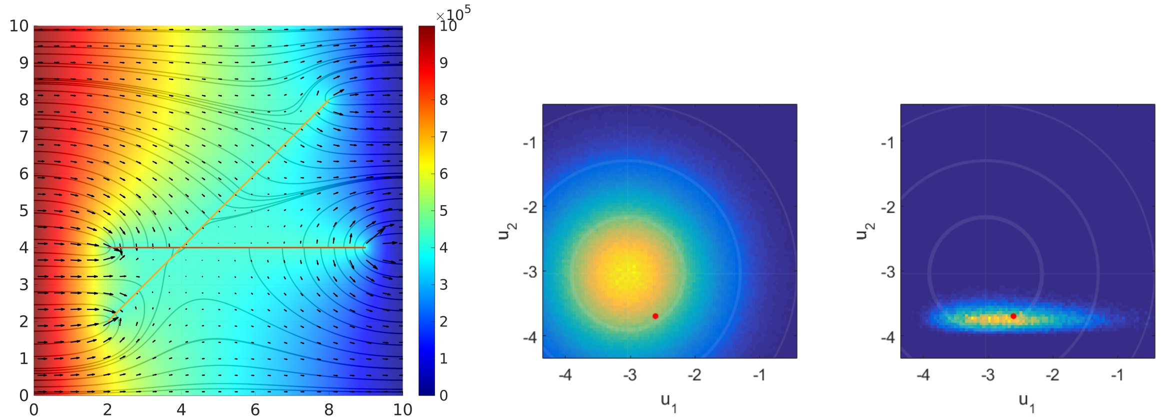
The elements of the matrix and the right-hand-side are estimated using the Monte Carlo estimator $\left(g,f\right)_{L^2_{\mathrm{d}F\mathbf{Z}}}\approx\frac{1}{K}\sum_{i=1}^K g\left(\mathbf{u}_i\right)f\left(\mathbf{u}_i\right)$, assuming that $\mathbf{u}_i$ are generated from the distribution of $\mathbf{Z}$.

For the description of the surrogate model construction using RBF see [4].

3 Applications and numerical experiments

In considered inverse problems, the forward model describes the Darcy flow in porous (possibly fractured) materials. The random vector $\mathbf{u}$ represents unknown parameters such as hydraulic conductivity, fracture aperture, etc. As measurements, we consider total flow through chosen boundary parts (as in the following example) or pressure in chosen boreholes inside of the domain.

In this example, $\mathbf{u} = (u_1, u_2)$ represents parameters of the aperture of two fractures in a two-dimensional domain, see Fig. 1a. Aperture of each fracture is assumed to be constant: $\exp(u_1)$ and $\exp(u_2)$. Measurements are total flows through 3 chosen boundary parts: one inflow window (left side) and two outflow windows (two halves of the right side). There is no flow on the rest of the boundary. The measurements $\mathbf{y}_m$ were calculated artificially as $G(\mathbf{u}_{real})$ and corrupted by additive Gaussian noise. Prior distribution of $\mathbf{u}$ is also Gaussian, see Fig. 1b. Posterior pdf of $\mathbf{u}$ conditioned by $\mathbf{y}_m$ was estimated using DAMH sampling, see Fig. 1c.



(a) Pressure and flow in a domain with two fractures

(b) Prior pdf of $\mathbf{u}$ and artificial real parameters $\mathbf{u}_{real}$ (red dot)

(c) Posterior pdf estimated using DAMH sampling

Figure 1: Visualization of the model inverse problem

Our previous results show that the sampling efficiency highly depends on the choice of the proposal std, see [1] and [5]. In this simulation, the choice of the proposal std was based on short preliminary runs of the DAMH algorithm. For each run, the autocorrelation time was estimated and the cost per one almost uncorrelated sample (CpUS) was calculated, see 2. The calculation of CpUS includes the computation time of the surrogate model $\widetilde{G}$; a unit is one evaluation of G. Note that in the case of the standard MH algorithm with one G evaluation in each step, the value of CpUS is equal to the autocorrelation time.

The sampling efficiency is also influenced by the quality of the surrogate model. According to Alg. 1, lower accuracy of the surrogate model leads to high amount of useless evaluations of G (the case of proposed samples that were pre-accepted and that rejected). Table 1 shows the dependence of the percentage of rejected samples on the size N of polynomial basis used for the construction of the SCM surrogate model.

max. pol. degree (number of polynomials)	1 (3)	2 (6)	3 (10)	4 (15)	5 (21)	6 (28)
rejected samples (%)	1.79	0.66	0.11	0.07	0.01	0.01

Table 1: Percentage of rejected samples (SCM surrogate model)

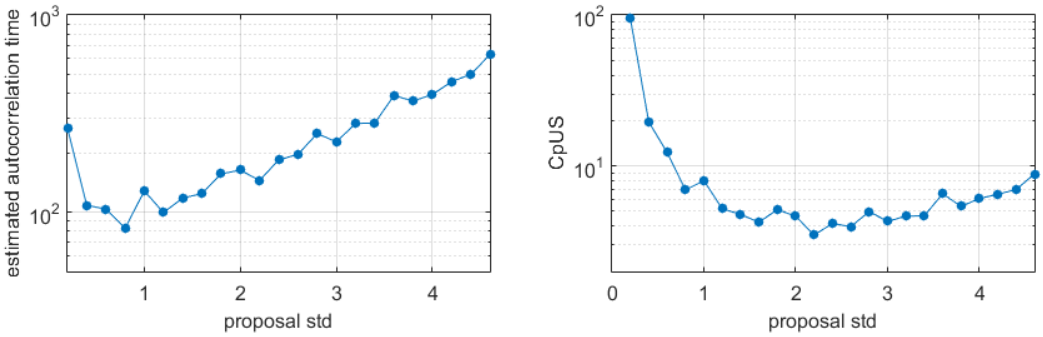


Figure 2: Dependence of autocorrelation time (left) and sampling cost (right) on proposal std

4 Conclusions

The DAMH algorithm with the use of surrogate models was applied to the Bayesian inverse approach to the estimation of the fracture aperture. In comparison to the standard MH algorithm, this sampling procedure significantly reduces the number of evaluations of the forward model G. Therefore, it allows us to solve inverse problems governed by computationally expensive forward models in the Bayesian way.

The discussed sampling procedure can be applied to a wide range of problems, since both of the aforementioned surrogate models (SCM and RBF) are non-intrusive. Therefore, to solve an inverse problem using this framework, it is sufficient to have a black-box solver of the forward model available and to specify the distribution of the noise and the prior distribution.

Acknowledgement: This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project "IT4Innovations excellence in science - LQ1602". The work was also partially supported by Grant of SGS No. SP2018/68 and by Grant of SGS No. SP2018/161, VSB-TU Ostrava, Czech Republic.

References

[1] R. Blaheta, M. Béreš, S. Domesová, P. Pan: *A comparison of deterministic and Bayesian inverse with application in micromechanics.* Applications of Mathematics, 2018.

[2] J.A. Christen, C. Fox: *Markov chain Monte Carlo using an approximation.* Journal of Computational and Graphical statistics, 2005.

[3] C. Robert: *The Bayesian choice: from decision-theoretic foundations to computational implementation.* Springer Science & Business Media, 2007.

[4] S. Domesová: *The use of radial basis function surrogate models for sampling process acceleration in Bayesian inversion.* AETA, 2018.

[5] S. Domesová, M. Béreš: *A Bayesian approach to the identification problem with given material interfaces in the Darcy flow.* HPCSE, 2017.

Towards numerical simulation of the macroalgae movement and photosynthetic growth within IMTA-RAS systems

R. Filip[1], *K. Petera*[1], *Š. Papáček*[2]

[1] Czech Technical University in Prague, Faculty of Mechanical Engineering,
Technická 4, 160 00 Prague 6, Czech Republic
[2] Institute of Complex Systems, University of South Bohemia in České Budějovice,
FFPW USB, CENAKVA, Zámek 136, 373 33 Nové Hrady, Czech Republic

1 Introduction

This work aims to contribute to the research and development of Integrated MultiTrophic Aquaculture (IMTA) production systems. Such systems, where the synergic effect of an aquaculture (usually in form of a Recirculated Aquaculture System – RAS) and a macroalgae (seaweed) culture system is exploited, could reduce the pressure on both, the open sea fishing and the use of terrestrial and inland water resources. IMTA-RAS systems represent an emerging research topic due to their biotechnological potential with an impact on human health and wellbeing [1].

2 Problem motivation

IMTA-RAS is currently one of the most promising lines of action to increase sustainability of fish farms [2]. However, there are some limiting factors or drawbacks in case of seaweed, e.g., *Ulva* sp. cultivation, integrated within recirculating aquaculture systems: (i) the large area required, (ii) the energy cost, (iii) lack of reliable mathematical models.

Concerning the second point, the major energy sinks in land-based seaweed culture systems is the system designed to make move (to tumble) seaweeds either by the bottom aeration or by the jet array, see Fig. 1.

The energy issue was studied experimentally in the work [3]. Here, we shall treat the third point, i.e., we aim to make one step towards modeling and *in silico* simulation of both multiphase flow in tanks for macroalgae cultivation and macroalgae photosynthetic growth. The first point in our scope is the analysis of flow pattern of liquid medium and seaweeds motion within the vessel. Once having determined the complex seaweed motion (depending on the intensity and type of tumbling-mixing), the assessment of photosynthetic growth is possible and eventually an **optimization** problem could be formulated and resolved. Here, we only point out, that the optimization factors are mainly (i) the tank operating conditions (e.g., aeration rate), and (ii) the culture conditions (e.g., stocking density of macroalgae fragments).

3 Expected results

The graphical results from a mathematical (numerical) model have to describe the relation between (i) the tank design parameters and operational conditions, and (ii) hydrodynamics or flow pattern (or time period in case of cyclic-rotational macroalgae motion), as can be seen on

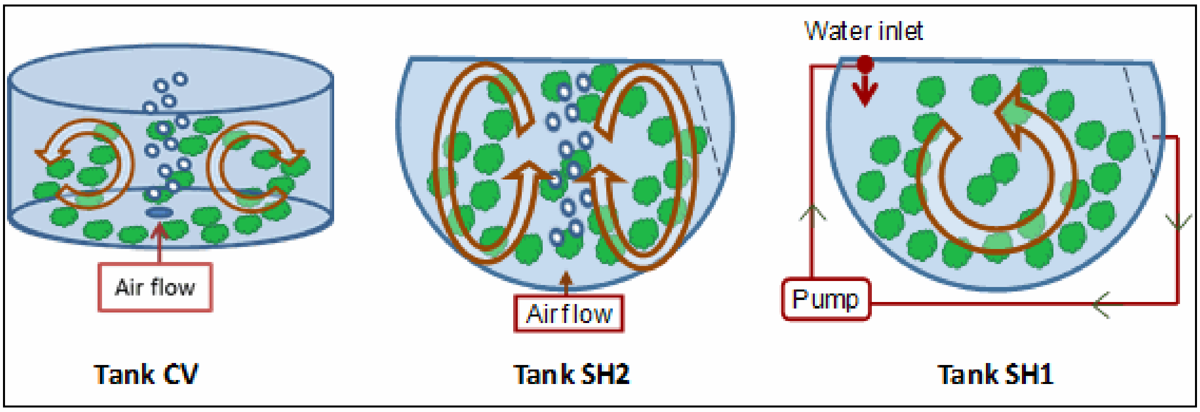


Figure 1: Three laboratory experimental systems: vertical cylindrical (CV) tank, semi-spherical tank bottom aerated (SH2), and semi-spherical tank with water jet system (SH1) [3].

Fig. 1. In this study, we are looking for a similar result as in Fig. 2, where the trajectory of an individual **microalgae** cell was calculated and subsequently used for the "irradiance history" identification by a simple concatenation of cell trajectory and the irradiance field $I = f(R, t)$ within the device [4]. We underline that while the *Eulerian* (immobile control volume) approach was preferred for microalgae growth description [4, 5], the *Lagrangian* approach describing the situation of an individual moving object (in our case a determined seaweed growth) is the right method here.

In this moment, we have got promising results using the computational fluid dynamics code STAR-CCM+, which offers an efficient and accurate set of fluid dynamics models and solvers with excellent parallel performance and scalability [6]. The circular tank (with diameter of 20 cm) with bottom air injection, i.e., the type *Tank CV* on Fig. 1, and a height of water equal to one radius is used for our analysis. This set up ensures the formation of two rotating flow cells placed, in the vertical section of the tank, at both sides of the aeration inlet, see Fig. 4. Both 2D axi-symmetric and full 3D simulation of seaweeds-like particles clumps movement are being performed. Based on the selected clumps trajectories, the probabilistic description of the random variable T_{cycle} (describing one period of rotational movement, detected by passing through a horizontal plane) was assessed, see Fig. 3. Obviously, an experiment with a real macroalgae strain, e.g. *Ulva* sp., shall be prepared in order to validate our numerical results.[1]

Once having described the macroalgae motion, the problem of light absorption by a growth model, e.g., the model of "photosynthetic factory" [7, 8] mounted on the macroalgae frond can be solved. Of course, the PSF model parameters have to be identified previously, e.g., using the method published in [9]. As well as in the case of parameter identification problem, our hope resides in the possibility to apply our previously developed algorithms for microalgae culture systems, namely closed photobioreactors [5].

Finally, we confess that one particular research topic attracts our interest: The inquiry if does it exist for seaweeds something similar to **flashing light enhancement**, see e.g., [10] and references within there, as in case of microalgae.

Acknowledgement: This work was supported by the Ministry of Education, Youth and Sport of the Czech Republic – projects CENAKVA (No. CZ.1.05/2.1.00/01.0024) and CENAKVA II (No. LO1205 under the NPU I program).

[1] There is an appealing alternative to the real macroalgae strain if it is not available – using synthetic sheets with similar thickness, size and density instead.

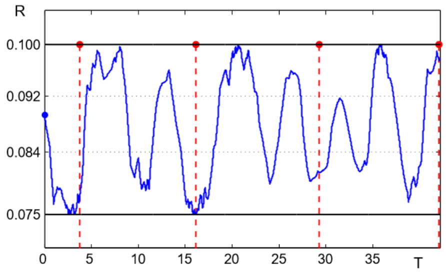


Figure 2: Time course (in seconds) of one single microalgae cell radial position R (in meters) within a Couette-Taylor bioreactor [4], simulated by the CFD code ANSYS Fluent.

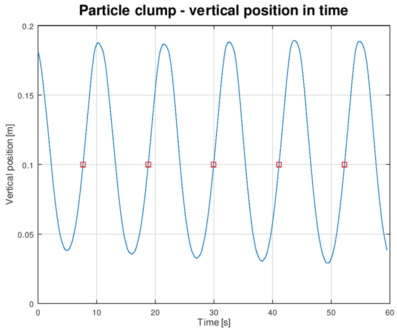


Figure 3: Time course [s] of one particle clump (representing one macroalgae) radial position R [m] in a cylindrical vessel (Tank CV - Fig. 1), simulated by the CFD code STAR-CCM+.

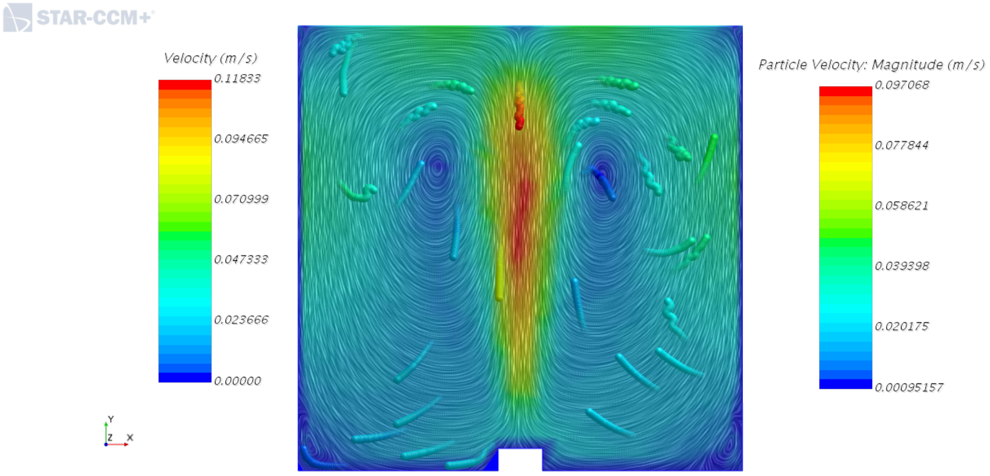


Figure 4: Particle clumps and 2D flow description within the Tank CV, simulated by the CFD code STAR-CCM+. Two nearly symmetrical rotating flow cells are clearly visible.

References

[1] A. Neori: *Essential role of seaweed cultivation in integrated multi-trophic aquaculture farms for global expansion of mariculture: An analysis.* Journal of Applied Phycology, 20(5), 2008, pp. 567–570. DOI: 10.1007/s10811-007-9206-3

[2] S. Hadley, K. Wild-Allen, C. Johnson, C. Macleod: *Modeling macroalgae growth and nutrient dynamics for integrated multi-trophic aquaculture.* Journal of Applied Phycology, 27(2), 2015, pp. 901–916.

[3] J. Oca, S. Machado, P. Jimenez de Ridder, J. Cremades, J. Pintado, I. Masaló: *Comparison of two water agitation methods in seaweed culture tanks: influence of the rotating velocity in the seaweed growth and energy requirement.* EAS2016 - Food for thought, 2016, pp. 716–717. Edinburgh, Oct 2016.

[4] Š. Papáček, C. Matonoha, K. Petera: *Modeling and Simulation of Microalgae Growth in a Couette-Taylor Bioreactor.* In: T. Kozubek, M. Čermák, P. Tichý, R. Blaheta, J. Šístek, D. Lukáš, J. Jaroš (eds): Lecture Notes in Computer Science 11087, 2018. 3rd International Conference on High Performance Computing in Science and Engineering, HPCSE 2017, Karolinka, Czech Republic, pp. 174–187.

[5] Š. Papáček, J. Jablonsky, K. Petera: *Advanced integration of fluid dynamics and photosynthetic reaction kinetics for microalgae culture systems.* BMC Systems Biology 201812 (Suppl 5):93, 2018, https://doi.org/10.1186/s12918-018-0611-9

[6] STAR-CCM+ product documentation,
https://mdx.plm.automation.siemens.com/star-ccm-plus

[7] P.H.C. Eilers, J.C.H. Peeters: *A model for the relationship between light intensity and the rate of photosynthesis in phytoplankton.* Ecological Modelling, 42, 1998, pp. 199–215.

[8] X. Wu, J.C. Merchuk: *Simulation of Algae Growth in a Bench-Scale Bubble Column Reactor.* Biotechnology & Bioengineering 80, 2002, pp. 156–168.

[9] B. Rehák, S. Čelikovský, Š. Papáček: *Model for Photosynthesis and Photoinhibition: Parameter Identification Based on the Harmonic Irradiation O_2 Response Measurement*, Joint Special Issue of TAC IEEE and TCAS IEEE, pp. 101–108.

[10] L. Nedbal, V. Tichý, F. Xiong, J.U. Grobbelaar: *Microscopic green algae and cyanobacteria in high-frequency intermittent light.* Journal of Applied Phycology, 8, 1996, pp. 325–333.

Laplacian preconditioning of elliptic PDEs: Localization of the eigenvalues of the discretized operator

T. Gergelits[1,4], *K.-A. Mardal*[2], *B.F. Nielsen*[3], *Z. Strakoš*[4]

[1] Institute of Computer Science of the CAS, Prague
[2] University of Oslo, Norway
[3] Norwegian University of Life Sciences, Ås, Norway
[4] Charles University in Prague

1 Introduction

In the paper [1], the authors study the operator generated by using the inverse of the Laplacian as preconditioner for second order elliptic PDEs $-\nabla \cdot (k(x)\nabla u) = f$. They prove that the range of $k(x)$ is contained in the spectrum of the preconditioned operator, provided that $k(x)$ is continuous. Their rigorous analysis only addresses mappings defined on infinite dimensional spaces, but the numerical experiments in the paper suggest that a similar property holds in the discrete case.

In this contribution we present the results obtained in the submitted paper [2], where we analyze the eigenvalues of the matrix $\mathbf{L^{-1}A}$, where $\mathbf{L}$ and $\mathbf{A}$ are the stiffness matrices associated with the Laplace operator and second order elliptic operators with a scalar coefficient function, respectively. Using only technical assumptions on $k(x)$, we prove the existence of a one-to-one pairing between the eigenvalues of $\mathbf{L^{-1}A}$ and the intervals determined by the images under $k(x)$ of the supports of the FE nodal basis functions. As a consequence, we can show that the nodal values of $k(x)$ yield accurate approximations of the eigenvalues of $\mathbf{L^{-1}A}$. In this contribution, the obtained theoretical results will be illustrated by several numerical experiments.

2 Setting of problem and notation

We consider a self-adjoint second order elliptic PDE in the form

$$\begin{aligned} -\nabla \cdot (k(x)\nabla u) &= f \quad \text{for } x \in \Omega, \\ u &= 0 \quad \text{for } x \in \partial\Omega, \end{aligned} \tag{1}$$

and the corresponding generalized eigenvalue problem

$$\begin{aligned} \nabla \cdot (k(x)\nabla u) &= \lambda \Delta u \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{2}$$

with the domain $\Omega \subset \mathbb{R}^d$, $d \in \{1,2,3\}$ and the given function $f \in L^2(\Omega)$. We assume that the real valued scalar function $k(x) : \mathbb{R}^d \to \mathbb{R}$ is bounded and piecewise continuous and that it is uniformly positive, i.e.,

$$k(x) \geq \alpha > 0, \quad x \in \Omega.$$

Let $V \equiv H_0^1(\Omega)$ denote the Sobolev space of functions defined on Ω with zero trace at $\partial\Omega$ and with the standard inner product. The weak formulations of the problems (1) and (2) are to seek $u \in V$, respectively $u \in V$ and $\lambda \in \mathbb{R}$, such that

$$\mathcal{A}u = f, \qquad \text{respectively} \qquad \mathcal{A}u = \lambda \mathcal{L}u \tag{3}$$

where $\mathcal{A}$, $\mathcal{L} : V \to V^{\#}$, $f \in V^{\#}$ are defined as

$$\mathcal{A} : H_0^1(\Omega) \mapsto H^{-1}(\Omega), \quad \langle \mathcal{A}u, v\rangle = \int_\Omega k\nabla u \cdot \nabla v, \quad u, v \in H_0^1(\Omega), \tag{4}$$

$$\mathcal{L} : H_0^1(\Omega) \mapsto H^{-1}(\Omega), \quad \langle \mathcal{L}u, v\rangle = \int_\Omega \nabla u \cdot \nabla v, \quad u, v \in H_0^1(\Omega), \tag{5}$$

and the function $f \in L^2(\Omega)$ is identified with the associated linear functional $f \in V^{\#}$ defined by

$$\langle f, v\rangle \equiv \int_\Omega fv\,. \tag{6}$$

Discretization via a conforming finite element method, using, for simplicity of exposition, Lagrange elements, leads to the discrete operators

$$\mathcal{A}_h,\, \mathcal{L}_h : V_h \to V_h^{\#}$$

where the finite dimensional subspace V_h is spanned by the piecewise polynomial basis functions $\phi_1, \ldots, \phi_N$ with the local supports

$$\mathcal{T}_i = \mathrm{supp}(\phi_i), \quad i = 1, \ldots, N.$$

The matrix representations $\mathbf{A}$ and $\mathbf{L}$ are defined as

$$[\mathbf{A}]_{ij} = \langle \mathcal{A}_h\phi_j, \phi_i\rangle = \int_\Omega \nabla\phi_i \cdot k\nabla\phi_j, \tag{7}$$

$$[\mathbf{L}]_{ij} = \langle \mathcal{L}_h\phi_j, \phi_i\rangle = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j, \quad i, j = 1, \ldots, N. \tag{8}$$

3 Theoretical results

Our theoretical results show that there exists a one-to-one correspondence, i.e., a pairing, between the individual eigenvalues of $\mathbf{L}^{-1}\mathbf{A}$ and quantities given by the function values of $k(x)$ in relation to the supports of the FE basis functions. The proof does not require that $k(x)$ is continuous. If, moreover, $k(x)$ is constant on a part of the domain Ω that contains fully the supports of one or more basis functions, then the function value of $k(x)$ determines the associated eigenvalue *exactly* and the number of the involved supports bounds from below the multiplicity of the associated eigenvalue. If $k(x)$ is slowly changing over the support of some basis function, then we get a very accurate localization of the associated eigenvalue.

Our approach is based upon the intervals

$$k(\mathcal{T}_j) \equiv [\inf_{x\in\mathcal{T}_j} k(x), \sup_{x\in\mathcal{T}_j} k(x)], \quad j = 1, \ldots, N, \tag{9}$$

where $\mathcal{T}_j = \mathrm{supp}(\phi_j)$.[2] Here we formulate the theoretical results. Theorem 1 localizes the positions of *all* the individual eigenvalues of the matrix $\mathbf{L}^{-1}\mathbf{A}$ by pairing them with the intervals $k(\mathcal{T}_j)$ given in (9). Using the given pairing, Theorem 2 describes the closeness of the eigenvalues to the nodal function values of the scalar function $k(x)$. The proof of Theorem 1 combines perturbation theory for matrices with a classical result from the theory of bipartite graphs.

[2] If $k(x)$ is continuous on $\mathcal{T}_j$, then $k(\mathcal{T}_j)$ coincides with the closure of the range of $k(x)$ over $\mathcal{T}_j$.

Theorem 1 (Pairing the eigenvalues and the intervals $k(\mathcal{T}_j)$, $j = 1, \dots, N$.).
Using the previous notation and assumptions, let $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ *be the eigenvalues of* $\mathbf{L}^{-1}\mathbf{A}$. *Then there exists a (possibly non-unique) permutation* π *such that the eigenvalues of the matrix* $\mathbf{L}^{-1}\mathbf{A}$ *satisfy*

$$\lambda_{\pi(j)} \in k(\mathcal{T}_j), \quad j = 1, \dots, N, \tag{10}$$

where the intervals $k(\mathcal{T}_j)$ *are defined in* (9).

Theorem 2 (Pairing the eigenvales and the nodal values).
Using the notation and assumption of Theorem 1, consider any point $\hat{x}_j$ *such that* $\hat{x}_j \in \mathcal{T}_j$. *Then the associated eigenvalue* $\lambda_{\pi(j)}$ *of the matrix* $\mathbf{L}^{-1}\mathbf{A}$ *satisfies*

$$|\lambda_{\pi(j)} - k(\hat{x}_j)| \leq \sup_{x \in \mathcal{T}_j} |k(x) - k(\hat{x}_j)|, \quad j = 1, \dots, N. \tag{11}$$

If, in addition, $k(x) \in \mathcal{C}^2(\mathcal{T}_j)$, *then*

$$\begin{aligned} |\lambda_{\pi(j)} - k(\hat{x}_j)| &\leq \sup_{x \in \mathcal{T}_j} |k(x) - k(\hat{x}_j)| \\ &\leq \hat{h} \|\nabla k(\hat{x}_j)\| + \tfrac{1}{2}\hat{h}^2 \sup_{x \in \mathcal{T}_j} \|D^2 k(x)\|, \quad j = 1, \dots, N, \end{aligned} \tag{12}$$

where $\hat{h} = \mathrm{diam}(\mathcal{T}_j)$ *and* $D^2 k(x)$ *is the second order derivative of the function* $k(x)$. *In particular,* (11) *and* (12) *hold for any discretization mesh node* $\hat{x}_j$ *such that* $\hat{x}_j \in \mathcal{T}_j$.

Acknowledgement: The work of Tomáš Gergelits and Zdeněk Strakoš has been supported by the Grant Agency of the Czech Republic under the contract No. 17-04150J. The work of Tomáš Gergelits has been also supported by the Charles University, project GA UK No. 172915.

References

[1] B.F. Nielsen, A. Tveito, W. Hackbusch: *Preconditioning by inverting the Laplacian; an analysis of the eigenvalues.* IMA Journal of Numerical Analysis 29, 1 (2009), pp. 24–42.

[2] T. Gergelits, K.-A. Mardal, B.F. Nielsen, Z. Strakoš: *Laplacian preconditioning of elliptic PDEs: Localization of the eigenvalues of the discretized operator.* Submitted to SIAM Journal on Numerical Analysis in September 2018 (after minor revision resubmitted in January 2019).

Parallel solution of Ultrasound Computational Tomography problems

V. Hapla, N. Korta Martiartu, C. Boehm, A. Fichtner

ETH Zürich, Switzerland

Measurements of mechanical waves travelling through a medium can be used to reveal the subsurface and interior structure of unknown objects. This has plentiful applications ranging from medical imaging at millimetre scale to seismic tomography at the planetary scale. However, solving these problems is challenging from both a mathematical and computational perspective, and scalable simulation tools are key to enable scientific progress.

We present an inverse solver for image reconstruction in Ultrasound Computed Tomography (USCT) for early breast cancer detection. USCT is a non-invasive, radiation-free, pressure-free and low-cost technique that uses both transmitted and reflected signals to create images of the soft tissue's acoustic properties. These images are particularly useful for characterizing interior breast tissue and differentiating between benign and malign lesions.

A short time-to-solution, from taking measurements to obtaining the image, is crucial for any medical imaging technique. It must be in the order of minutes to be applicable in practice. In addition, the computational resources in a hospital are limited and should not exceed a dedicated workstation. To meet these requirements, we employ a simplified physical model using ray-tracing and apply time-of-flight tomography to reconstruct the acoustic properties of the breast tissue. This approach leads to a linear least-square problem with a large sparse rectangular matrix. The problem is in general ill-posed, which can be handled by various regularization strategies.

To assemble, regularize and solve this problem, we use MATLAB and Portable, Extensible Toolkit for Scientific Computation (PETSc). PETSc provides all needed ingredients (distributed vectors and sparse matrices, fast parallel assembly and linear algebra routines, and implementations of least-squares methods), is highly portable, and has a permissive open source license (FreeBSD). Therefore, we strive to move all the needed algorithms from MATLAB to PETSc.

Acknowledgement: We gratefully acknowledge support by the Swiss Platform for Advanced Scientific Computing (PASC) and by the Swiss National Supercomputing Centre (CSCS) under project grant d72.

References

[1] N. Ozmen, R. Dapp, M. Zapf, H. Gemmeke, N.V. Ruiter, K.W.A. van Dongen: *Comparing different ultrasound imaging methods for breast cancer detection.* In: IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 62(4), 2015, pp. 637–646. DOI: 10.1109/TUFFC.2014.006707

[2] C. Boehm, N. Korta Martiartu, N. Vinard, I.J. Balic, A. Fichtner: *Time-domain spectral-element ultrasound waveform tomography using a stochastic quasi-Newton method.* Proc. SPIE 10580, Medical Imaging 2018: Ultrasonic Imaging and Tomography, 105800H (6 March 2018). DOI: 10.1117/12.2293299

[3] N.V. Ruiter, M. Zapf, T. Hopp, H. Gemmeke, K.W.A. van Dongen: *USCT data challenge*. Proc. SPIE 10139, Medical Imaging 2017: Ultrasonic Imaging and Tomography, 101391N (13https://www.overleaf.com/project/5c347f05057e554109e244ee March 2017). DOI: 10.1117/12.2272593

[4] N. Korta Martiartu, C. Boehm, V. Hapla, H. Maurer, I.J. Balic, A. Fichtner: *Optimal experimental design for joint reflection-transmission ultrasound imaging of breast tissue: from ray- to wave-based methods*. In preparation.

[5] B.F. Smith et al.: *PETSc Web Page*, `http://www.mcs.anl.gov/petsc`.

Numerical scheme for option pricing under exponential Lévy processes with finite activity

J. Hozman[1], *T. Tichý*[2]

[1] Technical University of Liberec
[2] VŠB - Technical University of Ostrava

1 Introduction

Option pricing is an essential issue of modern theory of financial engineering and goes back to the ideas of Black and Scholes (BS) firstly published in [2]. Nowadays, it is widely accepted that the BS model is not sufficiently accurate in capturing the real world features of security markets (e.g., volatility smile, leverage effect, clustering, heavy tails, leptokurtic feature, etc.), because its idealized assumptions do rarely hold in practice.

Such imperfections lead many researchers to analyze various extensions of the BS model. Among them, we can find a large class of models, which were motivated by empirical observations of large and sudden changes in the underlying asset price resembling jumps, see the pioneering paper [11]. One way, how to mimic such discontinuous paths with jumps, is to utilize Lévy processes, a family covering Brownian motion, pure jumps processes and their combinations; for survey see [4]. The expected number (finite or infinite) of jumps of a certain magnitude per unit time is described by the Lévy measure.

One category of option pricing models under Lévy processes contains models with finite activity. These models were introduced into the mathematical finance in late 1970s, c.f. Merton model [11], as well as studied relatively recently within the Kou model [10]. These models often require knowledge of advanced computational techniques, see, e.g.[1, 9], in order to obtain option price, especially when the payoff function is not the simplest one.

2 Exponential Lévy process and PIDE model

We briefly recall the pricing model from [4]. To price European options written on underlying asset S_t we use a model for the movement of asset prices that permits jumps. Therefore, we consider a process that has discontinuous paths — an exponential Lévy process of the form

$$S_t = S_0 \exp(L_t), \quad L_t = bt + \sigma W_t + Y_t, \quad 0 \le t \le T, \tag{1}$$

where t is the actual time, T the maturity and S_0 the initial price. The Lévy process L_t is a nontrivial combination of a standard Brownian motion W_t and a pure jump process Y_t (e.g. Poisson or compound Poisson process). The parameter $\sigma > 0$ denotes the volatility of underlying asset returns and the value of $b \in \mathbb{R}$ can be expressed using martingale theory as

$$b = r - q - \frac{\sigma^2}{2} - \int_{\mathbb{R}} \left(e^x - 1 - x\mathbb{1}_{|x|\le 1}\right) \nu(\mathrm{d}x), \tag{2}$$

where r, q are the interest and continuous dividend rates, respectively, $\mathbb{1}$ denotes the indicator function of a set and $\nu(\mathrm{d}x)$ stands for a general Lévy measure. If $\int_{\mathbb{R}} \nu(\mathrm{d}x) = \lambda < \infty$, this case exactly means that Lévy process (1) is of a finite activity, in other words, it generates a finite number of jumps within any finite time interval. On the other hand, if the Lévy measure is infinite, we speak of Lévy processes with infinite activity.

The category of finite activity processes is represented by a wide class of jump-diffusion processes, which were introduced into the mathematical finance by Merton [11]. The author considered jumps that are normally distributed with Lévy density

$$\nu(\mathrm{d}x) = \lambda g(x)\,\mathrm{d}x = \lambda \frac{1}{\sqrt{2\pi}\gamma} \exp\left(-\frac{(x-\mu)^2}{2\gamma^2}\right) \mathrm{d}x, \tag{3}$$

where λ, γ and μ are parameters of the model. From relatively new models, let us mention the Kou model, proposed in [10], as a double exponential jump model with Lévy density

$$\nu(\mathrm{d}x) = \lambda g(x)\,\mathrm{d}x = \lambda \left[p\alpha_1 e^{-\alpha_1 x}\mathbb{1}_{x\geq 0} + (1-p)\alpha_2 e^{\alpha_2 x}\mathbb{1}_{x<0}\right] \mathrm{d}x, \tag{4}$$

where $\lambda > 0$ is an intensity of the Poisson process and the rest of parameters takes values $\alpha_1 > 0$, $\alpha_2 > 0$ and $0 < p < 1$.

Next, denote by $V = V(S,t)$ the price of European option that has payoff

$$\max(S-\mathcal{K},0) \quad \text{(call)}, \qquad \max(\mathcal{K}-S,0) \quad \text{(put)}, \tag{5}$$

where $\mathcal{K}$ denotes the specified price at which an option contract can be exercised, usually called the strike price. Similarly to the BS framework, V is priced using an arbitrage-free principle, Itô calculus, elimination of stochastic fluctuations and a construction of a sophisticated portfolio. Following these steps, the fundamental result for advanced option pricing techniques under the Lévy processes characterizes $V(S,t)$ as a solution of a deterministic partial integro-differential equation (PIDE)

$$\begin{aligned}\frac{\partial V}{\partial t}(S,t) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}(S,t) + rS\frac{\partial V}{\partial S}(S,t) - rV(S,t)&\\ + \int_{\mathbb{R}} \left[V(Se^y,t) - V(S,t) - S(e^y-1)\frac{\partial V}{\partial S}(S,t)\right] \nu(\mathrm{d}y) = 0&\end{aligned} \tag{6}$$

for $(S,t) \in (0,\infty)\times(0,T)$ with the terminal condition as the payoff function (5).

Further, it is suitable to change asset values S to the scaled logarithmic ones $x = \ln(S/\mathcal{K})$ and time t to the time to maturity $\hat{t} = T - t$. By this change of variables and using a finite activity of Lévy process we obtain new pricing function $u(x,\hat{t}) = V(\mathcal{K}e^x, T-\hat{t})/\mathcal{K}$ satisfying

$$\frac{\partial u}{\partial \hat{t}} - \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} - \left(r - \frac{\sigma^2}{2} - \lambda\kappa\right)\frac{\partial u}{\partial x} + (r+\lambda)u - \lambda\int_{\mathbb{R}} u(x+y,\hat{t})g(y)\mathrm{d}y = 0 \quad \text{in } \mathbb{R}\times(0,T) \tag{7}$$

where $\kappa = \int_{\mathbb{R}}(e^y-1)g(y)\mathrm{d}y < \infty$. Simultaneously to (7), it is necessary to prescribe the initial condition given by the transformed payoff function (5) as

$$\max(e^x-1,0) \quad \text{(call)}, \qquad \max(1-e^x,0) \quad \text{(put)}. \tag{8}$$

Since, the Cauchy problem (7) with (8) is defined on the unbounded spatial domain $\mathbb{R}$, the asymptotic values of u are consistent with the theoretical European option prices as $S \to 0+$ and $S \to \infty$, see [5], i.e.,

$$\lim_{x\to-\infty} u(x,\hat{t}) = 0, \qquad \lim_{x\to\infty} u(x,\hat{t}) - \left(e^x - e^{-r\hat{t}}\right) = 0,\ \hat{t} > 0, \quad \text{(call)} \tag{9}$$

$$\lim_{x\to-\infty} u(x,\hat{t}) - \left(e^{-r\hat{t}} - e^x\right) = 0, \qquad \lim_{x\to\infty} u(x,\hat{t}) = 0,\ \hat{t} > 0. \quad \text{(put)} \tag{10}$$

3 Numerical approach

The more rigorous approach using PIDE forms the basis of advanced option pricing models. On the other hand, the question, which arises, is how to solve these complex governing equations with a combination of differential and integral terms. Unsurprisingly, a wide class of these pricing equations cannot be solved in closed form, i.e. analytical option pricing formulae are available only for the simple option contracts and/or under very strong limitations on market conditions. Therefore, the numerical methods take a crucial part in financial engineering.

The proposed pricing methodology, based on discontinuous Galerkin (DG) method, is related to numerical solving of (7), which requires localization to a bounded interval Ω. This numerical solution is composed by piecewise polynomial functions on finite element mesh without any requirements on the continuity of the solution between the particular elements, see [12].

Since the pricing equation is defined in the space-time domain, the development of the numerical scheme consists of two consecutive phases — spatial semi-discretization and temporal discretization. Within the first phase, for the time interval $[0,T]$, we construct the solution $u_h = u_h(\hat{t})$ from the space S_h^p of piecewise polynomial (of order p) generally discontinuous functions, defined over the partition $\mathcal{T}_h$ of the domain Ω. Based on capable similar techniques, cf. [7], and with a careful treatment of integral terms, this semi-discrete solution u_h is defined using the variational formulation leading to the system of ODEs

$$\frac{d}{d\hat{t}}(u_h, v_h) + \mathcal{D}_h(u_h, v_h) + \mathcal{I}_h(u_h, v_h) = 0 \quad \forall\, v_h \in S_h^p,\ \forall\, \hat{t} \in (0,T), \tag{11}$$

where $u_h(0)$ is given by (8), $(\cdot,\cdot)$ denotes the inner product in $L^2(\Omega)$ and forms $\mathcal{D}_h(\cdot,\cdot)$ and $\mathcal{I}_h(\cdot,\cdot)$ stand for DG semi-discrete variants of an operator acting on the differential part and integral part of the equation (7), respectively.

The second phase aims to discretize (11) on the time interval $[0,T]$. The proposed numerical scheme should be of a high accuracy with respect to time, have no restrictive condition on the length of the time step and preserve the sparsity of a system of linear algebraic equations resulting from this fully discrete problem. Unlike our previous research [7, 8], we are faced with new challenges here that, due to the simultaneous presence of differential and integral terms and the nonlocal character of $\mathcal{I}_h$, increase the complexity of the option pricing problem.

At first, it is much more essential how the nonlocal integral term is numerically treated. We will follow two possible ways: (*i*) a commonly used direct approximation using the standard quadrature methods, which suffer from high computational demandingness; (*ii*) a relatively modern technique which represents the integral terms as solutions of proper PDEs (firstly formulated in [3]) and leads to the local pseudo-differential formulation, for more details see [9].

In conclusion, we briefly present the numerical experiment on the standard benchmark of the Merton model, performed on the reference data from [3]. We consider European call option with parameter values of practical significance as $T = 0.25$, $\mathcal{K} = 100$, $\sigma = 0.25$, $r = 0.05$, $q = 0.0$, $\lambda = 0.1$, $\mu = -0.90$ and $\gamma = 0.35$. The numerical scheme is implemented in the solver Freefem++ (see [6]) with the time step proportional to a quarter of the day (1/1440) and piecewise linear ($p = 1$) and quadratic ($p = 2$) approximations on the uniformly partitioned (consecutively refined) grids $\mathcal{T}_h$ of the domain $\Omega = (-3, 2)$.

From the practical point of view we evaluate the options at several underlying prices for maturity date and compare these values to the reference and analytical ones, see Table 1. From this pointwise behaviour, one can conclude that the numerical option prices are of higher accuracy as the mesh is finer and are closer to the analytical ones than results in [3].

Table 1: Comparison of the approximate option values with the reference results at three reference underlying prices for the different grid spacing and polynomial orders.

		$p = 1$			$p = 2$	
$\#\mathcal{T}_h$	$S = 90$	$S = 100$	$S = 110$	$S = 90$	$S = 100$	$S = 110$
50	1.77746	6.16174	13.5663	1.84924	6.25116	13.6018
100	1.85463	6.25223	13.6205	1.85770	6.27374	13.6150
200	1.86511	6.27405	13.6255	1.85952	6.27936	13.6181
400	1.86314	6.27944	13.6225	1.86003	6.29071	13.6188
800	1.86034	6.28074	13.6196	1.86019	6.28099	13.6190
1600	1.86038	6.28100	13.6192	1.86024	6.28102	13.6190
ref. val. [3]	1.86030	6.28138	13.6190	1.86030	6.28138	13.6190
anal. val. [11]	1.86025	6.28128	13.6190	1.86025	6.28128	13.6190

Acknowledgement: Both authors were supported through the Czech Science Foundation under project No. 16-09541S. The support is greatly acknowledged.

References

[1] Y. Achdou, O. Pironneau: *Computational Methods for Option Pricing.* SIAM, Philadelphia, 2005.

[2] F. Black, M. Scholes: *The pricing of options and corporate liabilities.* Journal of Political Economy **81**, 1973, pp. 637–659.

[3] P. Carr, A. Mayo: *On the numerical evaluation of option prices in jump diffusion processes.* European Journal of Finance **14**, 2007, pp. 353–372.

[4] R. Cont, P. Tankov: *Financial Modelling with Jump Processes.* CRC, Boca Raton, FL, 2004.

[5] E.G. Haug: *The Complete Guide to Option Pricing Formulas.* McGraw-Hill, 2006.

[6] F. Hecht: *New development in FreeFem++.* Journal of Numerical Mathematics **20**, No. 3-4, 2012, pp. 251–265.

[7] J. Hozman, T. Tichý: *DG method for numerical pricing of multi-asset Asian options — The case of options with floating strike.* Applications of Mathematics **62**(2), 2017, pp. 171–195.

[8] J. Hozman, T. Tichý: *DG framework for pricing European options under one-factor stochastic volatility models.* Journal of Computational and Applied Mathematics **344**, 2018, pp. 585–600.

[9] A. Itkin: *Pricing Derivatives Under Lévy Models.* Birkhäuser, New York, 2017.

[10] S. Kou: *A jump-diffusion model for option pricing.* Management Science **48**, 2002, pp. 1086–1101.

[11] R.C. Merton: *Option pricing when underlying stock returns are discontinuous.* Journal of Financial Economics **3**, 1976, pp. 125–144.

[12] B. Riviére: *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation.* SIAM, Philadelphia, 2008.

The algebraic eigenvalue problem over noncommutative algebras

D. Janovská[1], *G. Opfer*[2]

[1] University of Chemistry and Technology, Prague
[2] University of Hamburg, Germany

1 Introduction

We will consider eigenvalue problems for square matrices with matrix elements from various types of noncommutative algebras in $\mathbb{R}^4$, namely we will study four algebraic systems: quaternions, coquaternions, nectarines, and conectarines. They will be denoted by $\mathcal{A}$ in this paper. In the following table, their names are listed with the multiplication rules for their elements.

Name of algebra	inshort	$\mathbf{i}^2$	j^2	k^2	$\mathbf{i}\mathrm{j}$	jk	$\mathrm{k}\mathbf{i}$
Quaternions	$\mathbb{H}$	-1	-1	-1	k	$\mathbf{i}$	j
Coquaternions	$\mathbb{H}_{\mathrm{coq}}$	-1	1	1	k	$-\mathbf{i}$	j
Nectarines	$\mathbb{H}_{\mathrm{nec}}$	1	-1	1	k	$\mathbf{i}$	$-$j
Conectarines	$\mathbb{H}_{\mathrm{con}}$	1	1	-1	k	$-\mathbf{i}$	$-$j

Eigenvalue problems and other problems like decompositions of matrices in the algebra $\mathbb{H}$, the field of quaternions, are well covered in the literature. The first published was a paper by Louise Wolf, [4, 1936], then, in 1989, a paper on the quaternion QR algorithm, [1], by Bunse-Gerstner, Byers, and Mehrmann appeared, a summary on matrices over $\mathbb{H}$ appeared 1997 by Zhang, [5].

2 Some facts about eigenvalue problem over noncommutative algebras

If all elements of an algebra except the zero element have an inverse, we call the algebra a division algebra. A typical division algebra is the (skew) field of quaternions, abbreviated by $\mathbb{H}$. The algebra of coquaternions, denoted by $\mathbb{H}_{\mathrm{coq}}$, is like $\mathbb{H}$ an algebra in $\mathbb{R}^4$, though, it is not a division algebra.

Definition Let $\mathbf{A} \in \mathcal{A}^{n\times n}$ for some algebra $\mathcal{A}$. If there is an element $\lambda \in \mathcal{A}$ and a column vector $\mathbf{x} \in \mathcal{A}^{n\times 1}$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{x}\lambda, \quad \mathbf{x} \text{ contains an invertible component.} \tag{1}$$

then, λ is called an eigenvalue of $\mathbf{A}$ with respect to the eigenvector $\mathbf{x}$. The pair $(\lambda, \mathbf{x})$ is called an eigenpair of $\mathbf{A}$. The set of all eigenvalues of $\mathbf{A}$ is denoted by $\sigma(\mathbf{A})$.

Lemma Let $\mathcal{A}$ be a noncommutative algebra and λ an eigenvalue of $\mathbf{A}$ with respect to the eigenvector $\mathbf{x}$. Then, for all invertible $h \in \mathcal{A}$ the set $\Lambda := \{h^{-1}\lambda h\}$ consists of eigenvalues of $\mathbf{A}$ with respect to $\mathbf{x}h$.

Proof Multiply equation (1) from the right by h, then

$$\mathbf{A}(\mathbf{x}h) = \mathbf{x}\lambda h = (\mathbf{x}h)(h^{-1}\lambda h),\ \mathbf{x}h \neq \mathbf{0},\ \text{for all invertible } h \in \mathcal{A}. \qquad \square$$

If an algebra contains elements different from the zero element which have no inverse, we will call these elements also singular, and elements which have an inverse nonsingular. Consequently, a square matrix will be called singular if it is not invertible and it will be called nonsingular if it is invertible.

Remark For commutative algebras the Lemma is also valid, however, $\Lambda := \{h^{-1}\lambda h\} = \{\lambda\}$ consists only of one element, in contrast to the noncommutative case in which Λ consists of infinitely many elements.

Definition Let $a \in \mathcal{A}$. The set

$$[a] := \{b : b := h^{-1}ah \text{ for all invertible } h \in \mathcal{A}\}$$

will be called similarity class of a (also called conjugacy class in the literature on algebra). All elements in $[a]$ are called similar. If a and b are similar we also denote this by $a \sim b$.

We can apply the notion of similarity also to the algebra of square matrices over an algebra.

Lemma Let $\mathbf{A}, \mathbf{B} \in \mathcal{A}^{n\times n}$ be two similar matrices. Then $\sigma(\mathbf{A}) = \sigma(\mathbf{B})$, [2].

Proof Similarity implies that there is an invertible matrix $\mathbf{M} \in \mathcal{A}^{n\times n}$ such that $\mathbf{A} = \mathbf{M}^{-1}\mathbf{B}\mathbf{M}$. If $\mathbf{A}\mathbf{x} = \mathbf{x}\lambda$, $\mathbf{x} \neq \mathbf{0}$, then $\mathbf{M}^{-1}\mathbf{B}\mathbf{M}\mathbf{x} = \mathbf{x}\lambda \Rightarrow \mathbf{B}(\mathbf{M}\mathbf{x}) = (\mathbf{M}\mathbf{x})\lambda, \mathbf{M}\mathbf{x} \neq \mathbf{0}$. Thus, λ is also an eigenvalue of $\mathbf{B}$. The same proof applies to $\mathbf{B}\mathbf{y} = \mathbf{y}\lambda$ and it implies that λ is also an eigenvalue of $\mathbf{A}$. $\square$

In order to find all eigenvalues of a given matrix $\mathbf{A} \in \mathcal{A}^{n\times n}$, it is sufficient to find one representative in each similarity class of eigenvalues. The number of eigenvalues of $\mathbf{A}$ will be, correspondingly, defined as the number of distinct similarity classes of $\sigma(\mathbf{A})$.

Theorem Let $\mathcal{A}$ be either a division algebra or a commutative algebra and let $\mathbf{A} \in \mathcal{A}^{n\times n}$ be an upper triangular matrix. Then, the diagonal elements of $\mathbf{A}$ are the eigenvalues of $\mathbf{A}$. The same is true for lower triangular matrices.

Theorem Let $\mathbf{A} \in \mathcal{A}^{n\times n}$ be a given, upper or lower triangular, matrix with matrix entries a_{jk}, $j, k = 1, 2, \ldots, n$. The matrix $\mathbf{A}$ is singular if and only if one of the diagonal elements a_{jj}, $j = 1, 2, \ldots, n$, is singular.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{A}$. We define a column operator $\mathrm{col} : \mathcal{A} \to \mathbb{R}^{N\times 1}$ by

$$\mathrm{col}(\mathbf{x}) := \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \tag{2}$$

where $x_1, x_2, \ldots, x_N$ are components of $\mathbf{x}$.

Let $\mathbf{A} = (a_{jk}) \in \mathcal{A}^{m\times n}$, $j = 1, 2, \ldots, m$, $k = 1, 2, \ldots, n$. Then we define

$$\mathrm{col}(\mathbf{A}) := \begin{bmatrix} \mathrm{col}(a_{11}) \\ \mathrm{col}(a_{21}) \\ \vdots \\ \mathrm{col}(a_{m1}) \\ \mathrm{col}(a_{12}) \\ \mathrm{col}(a_{22}) \\ \vdots \\ \vdots \\ \mathrm{col}(a_{mn}) \end{bmatrix} \in \mathbb{R}^{mnN\times 1}. \tag{3}$$

The eigenvalue problem over an arbitrary algebra can be expressed in the real form

$$(\mathbf{M} - \Lambda)\mathrm{col}(\mathbf{x}) = \mathbf{0}, \tag{4}$$

and there will be a solution $\mathbf{x} \neq 0$ if and only if there is a matrix Λ such that $\mathbf{M} - \Lambda$ is singular. The matrix $\mathbf{M} - \Lambda$ is a triangular block matrix.

3 Computation of Eigenvalues by Newton's technique

For finding the eigenvalues and eigenvectors, we suggest the application of Newton's method. However, since the Jacobi matrix in this case is not square, we end up with an underdetermined system, which we solve by the least squares method. And it turns out that for eigenvalue problems of modest size, this technique works quite well. We specialize the results to the eight algebras in $\mathbb{R}^4$, in particular to coquaternions.

For a general square matrix $\mathbf{A} \in \mathcal{A}^{n\times n}$ we consider the eigenvalue problem (1) for an N dimensional algebra $\mathcal{A}$ in the form

$$\begin{aligned} G_1(\mathbf{x}, \lambda) &:= \mathbf{x}\lambda - \mathbf{A}\mathbf{x}, & (5)\\ G_2(\mathbf{x}) &:= ||\mathrm{col}(\mathbf{x})||^2 - 1 & (6) \end{aligned}$$

and solve

$$G(\mathbf{x}, \lambda) := \left\{ \begin{array}{c} G_1(\mathbf{x}, \lambda) \\ G_2(\mathbf{x}) \end{array} \right\} = \mathbf{0} \tag{7}$$

by Newton' method. The quantity $||\ \ ||^2$ is the square of the standard euclidean norm in $\mathbb{R}^{nN}$. The condition $G_2(\mathbf{x}) = 0$ is a normalization condition for the eigenvectors. It is independent of the algebra $\mathcal{A}$ under investigation. However, it does not imply uniqueness of $\mathbf{x}$. In all algebras the eigenvectors $\mathbf{x}$ and $-\mathbf{x}$ are simultaneous eigenvectors or not. If $z \in \mathcal{A}$ commutes with an eigenvalue λ and $||\mathrm{col}(z)|| = 1$, then $\mathbf{x}$ and $\mathbf{x}z$ are both eigenvectors for the same λ.

Applying the techniques developed in [3], we obtain the derivative of G in the form

$$G'(\mathbf{x}, \lambda)(\mathbf{h}, h_1) = \left\{ \begin{array}{l} \mathbf{h}\lambda + \mathbf{x}h_1 - \mathbf{A}\mathbf{h}, \\ 2\mathrm{col}(\mathbf{x})^{\mathrm{T}}\mathrm{col}(\mathbf{h}), \quad \mathbf{x},\ \mathbf{h} \in \mathcal{A}^{n\times 1}, h_1 \in \mathcal{A}^{1\times 1} \end{array} \right\}. \tag{8}$$

And Newton's technique consists of solving the linear system

$$G'(\mathbf{x}_k, \lambda_k)(\mathbf{h}, h_1) = -G(\mathbf{x}_k, \lambda_k),\ k = 0, 1, \ldots, \tag{9}$$

for $(\mathbf{h}, h_1)$ where the start values $\mathbf{x}_0, \lambda_0$ are given, in principle, arbitrarily. We observe, that the number of real unknowns $(\mathbf{x}, \lambda)$ in equation (7) is $(n+1)N$ whereas the number of real equations is $nN+1$. In the linear Newton equation (9) we have, thus, $(n+1)N$ real unknowns $(\mathbf{h}, h_1)$ and $nN+1$ real equations. Thus, the system is underdetermined.

The left hand side of (9) is linear in $(\mathbf{h}, h_1)$ and, thus, can be expressed by a matrix, the form of which is given in detail in [3]. The result is

$$G'(\mathbf{x}, \lambda)(\mathbf{h}, h_1) = \mathbf{M} \begin{pmatrix} \operatorname{col}(\mathbf{h}) \\ \operatorname{col}(h_1) \end{pmatrix},$$

where $\mathbf{M} \in \mathbb{R}^{(nN+1)\times(n+1)N}$. In order to find $\mathbf{M}$ let e_j be the jth standard unit vector in $\mathbb{R}^{nN+1}$, $j = 1, 2, \ldots, (n+1)N$. Then, the jth column of $\mathbf{M}$, denoted by $\mathbf{M}_j$ is

$$\mathbf{M}_j = \operatorname{col}(G'(x, \lambda)(e_j)),\ j = 1, 2, \ldots, (n+1)N,$$

where $(\mathbf{h}, h_1)$ is combined to one real vector of length $(n+1)N$. As solution of the linear, underdetermined system we use the least squares solution, which in programming systems, like MATLAB is already implemented. If $(\mathbf{h}, h_1)$ is the least squares solution of (9), we put $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}$, $\lambda_{k+1} = \lambda_k + h_1$ and continue with the iteration (9).

The numerical experiments show fast convergence for almost all problems of modest size. For triangular matrices we obtain the expected eigenvalues in the similarity classes of the diagonal elements, and also others.

Acknowledgement: The research was supported by the grant of the University of Chemistry and Technology PIGA 413-17-8644.

References

[1] A. Bunse-Gerstner, R. Byers, V. Mehrmann: *A quaternion QR algorithm*, Numer. Math. 55 (1989), pp. 83–95.

[2] D. Janovská, G. Opfer: *Matrices Over Nondivision Algebras Without Eigenvalues.* Advances in Applied Clifford Algebras 26 (2016), pp. 591–612.

[3] R. Lauterbach, G. Opfer: *The Jacobi matrix for functions in noncommutative algebras*, Adv. Appl. Clifford Algebras, **24** (2014), pp. 1059–1073, Erratum: Adv. Appl. Clifford Algebras, **24** (2014), p. 1075.

[4] L. A. Wolf: *Similarity of matrices in which the elements are real quaternions*, Bull. Am. Math. Soc. 42 (1936), pp. 737–743.

[5] F. Zhang: *Quaternions and matrices of quaternions*, Linear Algebra Appl., **251** (1997), pp. 21–57.

Vibrations of lumped parameter models: Filippov approach

V. Janovský

Faculty of Mathematics and Physics, Charles University in Prague

The aim is to study vibrations of lumped parameter systems, see [7]: the constitutive relations are defined implicitly. In Section 1, we consider a dry friction model. It can be interpreted as Mass-Spring-Dashpot lumped parameter system, see [6]. We explain principles of the Filippov convex method. In Section 2, we consider the oscillator with a unilateral constraint, see [5]. Both models can be efficiently solved as a *Filippov systems* applying the event-driven algorithm [4].

1 A dry friction model

We seek for the displacement of the mass $x = x(t)$ which satisfies the balance of linear momentum

$$x''(t) = \frac{1}{m}\,\left(f(t) - F_s(t) - F_d(t)\right)\,, \tag{1}$$

where $F_s = F_s(t) = k\,x(t)$, $k > 0$, is the spring force and $F_d = F_d(t)$ are *dissipative forces.*

Setting $v = v(t) = x'(t)$

$$\left\{\begin{array}{lcl} v'(t) & = & \dfrac{1}{m}\,\left(f(t) - F_s(t) - F_d(t)\right) \\ F_s'(t) & = & k\,v(t) \end{array}\right. \tag{2}$$

We have to add *constitutive relationship* which is defined via an implicitly defined function $\beta : \mathbb{R}\times\mathbb{R} \mapsto \mathbb{R}$

$$\beta\left(v(t), F_d(t)\right) = 0\ \in \mathbb{R}\,. \tag{3}$$

Following [7], the system (2) & (3) constitutes a system of semi-implicit *differential-algebraic equations* (DAEs).

Consider F_d to be a *Coulomb-type force* (labeled traditionally by F_c). $F_d(t) \equiv F_c(t)$ is implicitly defined as

$$\left\{\begin{array}{lcll} F_c(t) & = & \mathcal{F}\,\mathrm{Sign}\,v(t) & \text{for}\quad v(t) \neq 0 \\ v(t) & = & 0 & \text{for}\quad |F_c(t)| \leq \mathcal{F} \end{array}\right.$$

where $\mathcal{F}$ is a positive constant (the friction coefficient).

Following the ideas of [2], we solve the DAEs formulation via the implicit Euler scheme:

Set $\tau > 0$, the time step.
Define the sequences $\{v^n\}_{n=0}^{\infty}$ and $\{F_s^n\}_{n=0}^{\infty}$, by the recurrence:

Given v^n and F_s^n, set $F_t^{n+1} = \dfrac{m}{\tau}v^n + f^{n+1} - F_s^n$.

```
if $|F_t^{n+1}| \leq \mathcal{F}$, set $F_c^{n+1} = F_t^{n+1}$, $v^{n+1} = 0$

else set
        $\left\{\begin{array}{lcl} F_c^{n+1} & = & \mathcal{F}\,\mathrm{Sign}\,F_t^{n+1} \\ v^{n+1} & = & \left(\frac{m}{\tau} + \tau k\right)^{-1}\left(F_t^{n+1} - F_c^{n+1}\right) \end{array}\right.$
```

```
end
```

$F_s^{n+1} = \tau k v^{n+1}$

As an alternative, we consider the *Filippov method*, see [3]. We apply the ready made package [4]. Instead of the original variables x, v and t we introduce the variables x_1, x_2 and x_3. The package requires to deal with autonomous vector fields. Hence we need to "autonomize" the problem.

We define vector fields $F_1 : \mathbb{R}^3 \to \mathbb{R}^3$ and $F_2 : \mathbb{R}^3 \to \mathbb{R}^3$ as

$$F_1 = \begin{bmatrix} x_2 \\ -\frac{k}{m}x_1 + \frac{1}{m}f(x_3) - \frac{1}{m}\mathcal{F} \\ 1 \end{bmatrix}, \quad F_2 = \begin{bmatrix} x_2 \\ -\frac{k}{m}x_1 + \frac{1}{m}f(x_3) + \frac{1}{m}\mathcal{F} \\ 1 \end{bmatrix}$$

on the sets

$$S_1 = \{x \in \mathbb{R}^3 : H_{12}(x) > 0\} \quad \text{end} \quad S_2 = \{x \in \mathbb{R}^3 : H_{12}(x) < 0\} ,$$

where $H_{12} : \mathbb{R}^3 \to \mathbb{R}$ is the level-set operator

$$H_{12}(x) = x_2 \, .$$

The set

$$\Sigma_{12} = \{x \in \mathbb{R}^3 : H_{12}(x) = 0\}$$

is called the *discontinuity surface.* We define *Filippov system* $x' = F(x)$,

$$x' = \begin{cases} F_1(x) & \text{for} \quad x \in S_1 \\ F_2(x) & \text{for} \quad x \in S_2 \end{cases} \tag{4}$$

This is a short cut for the *differential inclusion*

$$x' \in \begin{cases} F_1(x)\,, & x \in S_1 \\ \overline{\text{co}}(F_1, F_2)\,, & x \in \Sigma_{12} \\ F_2(x)\,, & x \in S_2 \end{cases} \tag{5}$$

where $\overline{\text{co}} = \{z \in \mathbb{R}^3 : z = \lambda F_1 + (1-\lambda)F_2, \; \lambda \in [0,1]\}$ is a convex hull.

The *Filippov convex method* (the idea): We solve ODEs on S_1,S_2 and on Σ_{12}, concatenating smooth trajectories.

The formulation of the *constitutive relationship*:

Given $x \in \mathbb{R}^3$, let $F_c = -kx_1 + f(x_3)$.

```
if $|F_c| \geq \mathcal{F}$, set $F_c := \mathcal{F}\,\text{Sign}\,F_c$
else, set $F_c = -kx_1 + f(x_3)$ .
```

Example 1: $m = 1$, $k = 1$, $f(t) = sin(\omega t)$, $\omega = 1/6$, $\mathcal{F} = 0.4$. The initial condition: $x^{\text{init}} = 4$, $v^{\text{init}} = 0$, the solution time span: $[0, 10 * T]$, $T = 2\pi/\omega$.
The performance of

- The DAEs: Elapsed time = 1995.06 secs, the number of time steps = 376991 (the fixed time step = 0.001)
- The event-driven algorithm [4]: Elapsed time = 24.38 secs, the number of time steps = 151204 (an adaptive time stepping),
  AbsTol: 1.0000 e–006, MaxStep: 0.01, RelTol: 1.0000 e–006.

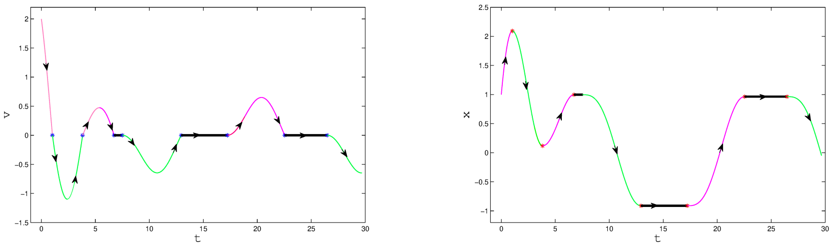


Figure 1: Example of a Filippov's solution. Notation: $v = x_1' = x_2$, $x = x_1$, $t = x_3$. The initial condition: $(1, 2, 0)^{\mathrm{T}}$. The black part of trajectory corresponds to sliding.

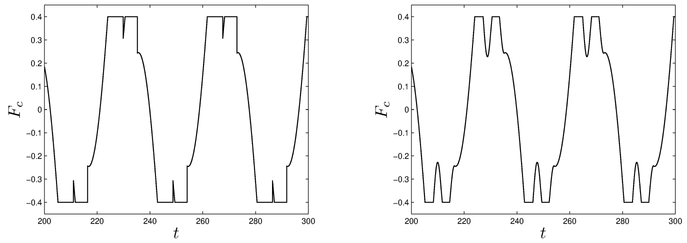


Figure 2: A plot of F_c versus time t (a zoom). On the left: via the DAEs. On the right: via the Filippov method. Mind the details in resolution.

2 Oscillator with a unilateral constraint

The following oscillator was investigated in [5]:

$$x'' + F_s(x) = f(t) \tag{6}$$

where

$$F_s(x) = \begin{cases} Lx & \text{if} \quad x > 0 \quad \ldots \textit{response of the wall} \\ x & \text{if} \quad x < 0 \quad \ldots \textit{restoring force} \end{cases} \tag{7}$$

It is obvious how to convert the above problem to Filippov system $x' = F(x)$ for an autonomous dynamical system. For a motivation, see (4). We apply the solver [4].

As case study we investigate **Example 2**: $f(t) = \cos(\omega t)$, $\omega = 2.7$, $L = 10^7$.
Initial condition: $x(0) = -1$, $x'(0) = -2$, time span: $0 \le t \le 300$.

Selected results are reported in Figure 3 and Figure 4.

References

[1] M. di Bernardo, C.J. Budd, A.R. Champneys, P. Kowalczyk: *Piecewise-smooth Dynamical Systems, Theory and Applications*, Springer, 2008.

[2] S. Darbha, K.B. Nakshatrala, K.R. Rajagopal: *On the vibrations of lumped parameter systems governed by differential-algebraic equations*, Journal of the Franklin Institute 347, 2010, pp.87–101.

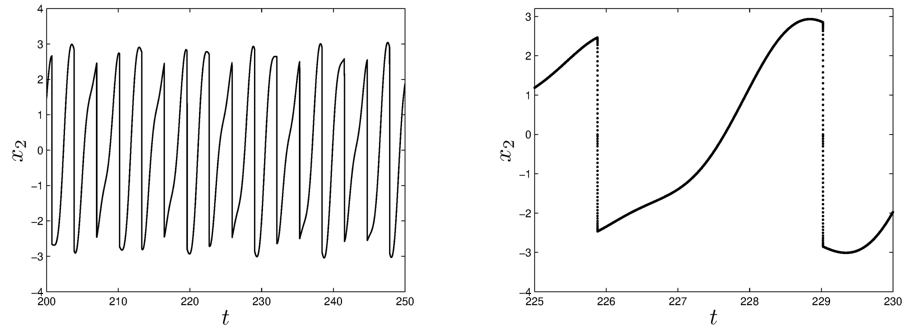


Figure 3: x_2 versus time t as $200 \leq t \leq 250$. On the right: Numerical performance, adaptive time-stepping.

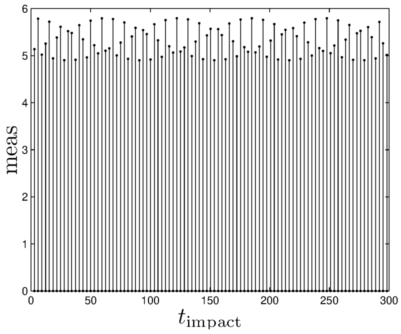


Figure 4: measure versus time t_{impact}, $0 \leq t_{\text{impact}} \leq 300$. Impact points: Define $S = \{0 \leq t \leq 300 \,:\, x_1(t) > 0\}$. Cover S by open intervals $(t_{\min} \leq t \leq t_{\max})$ such that $x_1(t) > 0$ for $t_{\min} < t < t_{\max}$. Set $t_{\text{impact}} \equiv (t_{\max} + t_{\min})/2$ and define meas $\equiv I_{t_{\min}}^{t_{\max}}(F_s)$. The cover consists of 95 intervals.

[3] A.F. Filippov: *Differential Equations with Discontinoous Righthand Sides*, Kluver Academic Publishes, 1988.

[4] P.P. Piiroinen, Y.A. Kuznetsov: *An event-driven method to simulate Filippov systems with accurate computing of sliding motions*, ACM Transactions on Mathematical Software Vol 34, No.3. Article13 (2008).

[5] D. Pražák, K.R. Rajagopal, J. Slavík: *A nonstandard analysis approach to a constrained forced oscillator*, Journal of Logic & Analysis 9:4 (2017) pp. 1–22

[6] D. Pražák, K.R. Rajagopal: *Mechanical oscillators with dampers defined by implicitly constitutive relations*, Comment.Math.Univ.Carolin. 57,1, 2016, pp. 51–61.

[7] K.R. Rajagopal: *A generalized framework for studying vibrations of lumped parameter systems*, Mechanics Research Communications 37 (2010), pp. 463–466.

[8] Z. Yuan, V. Průša, K.R. Rajagopal, A. Srinivasa: *Vibrations of a lumped parameter mass-spring-dashpot system wherein the spring is described by a non-invertible elongation-force constitutive function.* International Journal of Non-Linear Mechanics, 76 (2015), pp. 154-163.

Finite element modelling of elastic wave propagation in heterogeneous media

R. Kolman[1], *S.S. Cho*[2], *J. González*[3], *K.C. Park*[4]

[1] Institute of Thermomechanics of the Czech Academy of Sciences, Prague
[2] Korea Atomic Energy Research Institute, Daejeon, Korea
[3] Universidad de Sevilla, Sevilla, Spain
[4] University of Colorado at Boulder, Boulder, Co, USA

1 Introduction

Nowadays, additive technologies (3D or 4D printing) for manufacturing of mechanical parts of complex systems are modern and developing technology for demanding applications in mechanical and civil engineering, biomechanics or aircraft and space technologies. In additive manufacturing, body shape can be designed of general complex shapes and structures based on topology optimization or smart design knowledge.

There is a prediction for future of additive technologies for improvement of this technology so that a body of complex shape could be manufactured with different material for each region and materials will be mixed in an arbitrary ratio. By this technology, mechanical or electromagnetic properties can be controlled by material distribution and graded and layered materials can be manufactured [1].

Based on this motivation, we are focusing on elastic wave propagation in general heterogeneous media where mechanical properties (density and elastic tensor) are distributed non-uniformly and changed in space. In this work, we numerically study elastic wave propagation in a heterogeneous bar discretized by the finite element method. It is known that the standard finite element method with explicit time integration produces spurious oscillations [3]. In this contribution, we present an explicit scheme based on local time stepping respecting local wave speed and local stability limit for each finite element. The work aim is to suppress the spurious oscillations in wave propagation tasks in heterogeneous bars.

2 Wave equation for a layered bar

We assume a prismatic layered bar of length L with a cross-section A with a spatially piecewise-constant distributed Young's modulus $E(x)$ and mass density $\rho(x)$ along the bar and each material quantity is a function of space as the position of material point x. The elastic behavior is given by the Hook's law $\sigma(x) = E(x)\varepsilon(x) = E(x)\partial u(x)/\partial x$, where ε marks the infinitesimal strain and the normal stress $\sigma(x)$. In one-dimensional linear theory of elastodynamics, the wave speed at the position x is given by formula $c(x) = \sqrt{E(x)/\rho(x)}$ and $\sigma(x)$ can be evaluated as $\sigma(x) = \rho(x)c(x)v(x)$, where $v = \frac{\partial u}{\partial t}$ marks the velocity of a material point at the position x. For more details for elastodynamic theory see [2].

Let us assume a disjunctive partition of domain of interest $\Omega = [0, L] \subset \mathbb{R}$ with $\Omega_i = [x_{i-1}, x_i), i = 1, 2, \ldots, n$, $x_0 = 0, x_n = L$, so that $\bigcup_{i=1}^{n} \Omega_i = \Omega$ and $\Omega_i \bigcap \Omega_{i+1} = \emptyset$. Length of each domain Ω_i is given $L_i = x_i - x_{i-1}$.

The classical equation governing elastic wave propagation in one-dimensional domain Ω_i without the volume body force takes the form, see [2],

$$E_i \frac{\partial^2 u(x,t)}{\partial x^2} = \rho_i \frac{\partial^2 u(x,t)}{\partial t^2} \quad \text{on} \quad \Omega_i \times [0,T], \tag{1}$$

where $x \in \Omega_i$ is the position, $t \in \mathbb{R}^+$ is the time, T it the total time of interest of the wave event, $u(x,t)$ is the displacement field. The displacement $u(x,t)$ on the domain Ω_i is assumed to be differentiable up to the second partial derivatives with respect to independent variables x and t. The governing equation (1) is complemented by the initial and boundary conditions. We assume the bar in the rest at the initial time, i.e. with zero initial displacement and velocity field, $u(x,0) = 0$ and $\dot{u}(x,0) = 0$, resp. at the time $t = 0$. At the interfaces of layers–domains, the compatibility interface conditions are assumed for displacements $u_i(x_i,t) = u_{i+1}(x_i,t)$ and stresses $\sigma_i(x_i,t) = \sigma_{i+1}(x_i,t)$, $i = 1,2,\ldots,n-1$.

3 A local time stepping scheme for an one-dimensional case

Based on pullback interpolation scheme presented in [4], we modify the mentioned scheme including a local time stepping process using local stability limit for each finite element with a different length and wave speed. The presented time stepping process is consisted of following two computational steps as follows:

STEP 1. Pull-back integration with local stepping:

1a) Integration by the central difference scheme with the local (elemental) critical time step size Δt_e^c for each finite element at the time $t^{n+c} = t^n + \Delta t_e^c$

$$\begin{gathered}(\mathbf{u}_{fs}^{n+c})_e = \mathbf{u}_e^n + \Delta t_e^c \dot{\mathbf{u}}_e^n + \frac{1}{2}(\Delta t_e^c)^2 \ddot{\mathbf{u}}_e^n \\ + \text{ application of local Dirichlet boundary conditions}\end{gathered} \tag{2}$$

$$(\ddot{\mathbf{u}}_{fs}^{n+c})_e = \mathbf{M}_e^{\text{-}1} \left[\mathbf{f}_e^{n+c} - \mathbf{K}_e(\mathbf{u}_{fs}^{n+c})_e\right] \tag{3}$$

The elemental critical time step size for the e-the element Δt_e^c is set as $\Delta t_e^c = h_e/c_e$ or $\Delta t_e^c = 2/\omega_{max}^e$, where ω_{max}^e is the maximum eigen-angular velocity for the e-th separate finite element respecting to local Dirichlet boundary conditions. $\mathbf{M}_e$ and $\mathbf{K}_e$ are local mass and stiffness matrices, $\mathbf{f}_e$ is the local vector of external forces.

1b) Pull-back interpolation of local nodal displacement vectors at the time $t^{n+1} = t^n + \Delta t$ with $\alpha = \Delta t/\Delta t_e^c$, $\beta_1(\alpha) = \frac{1}{6}\alpha\big(1 + 3\alpha - \alpha^2\big)$, $\beta_2(\alpha) = \frac{1}{6}\alpha\big(\alpha^2 - 1\big)$

$$\begin{gathered}(\mathbf{u}_{fs}^{n+1})_e = \mathbf{u}_e^n + \Delta t \dot{\mathbf{u}}_e^n + (\Delta t_e^c)^2 \beta_1 \ddot{\mathbf{u}}_e^n + (\Delta t_e^c)^2 \beta_2 (\ddot{\mathbf{u}}_{fs}^{n+c})_e \\ + \text{ application of local Dirichlet boundary conditions}\end{gathered} \tag{4}$$

1c) Assembling of local contributions of displacement vector from Step 1b.

$$\mathbf{u}_{fs}^{n+1} = (\mathbf{L}^{\mathrm{T}}\mathbf{L})^{\text{-}1}\mathbf{L}^{\mathrm{T}}(\mathbf{U}_{fs}^{n+1}) \tag{5}$$

where $\mathbf{L}$ is the assembly Boolean matrix.

STEP 2. Push-forward integration with averaging:

2a) Push-forward predictor of displacement vector at the time $t^{n+1} = t^n + \Delta t$ by the central difference scheme with the time step size Δt.

$$\mathbf{u}_{cd}^{n+1} = \mathbf{u}^n + \Delta t \dot{\mathbf{u}}^n + \frac{1}{2}\Delta t^2 \ddot{\mathbf{u}}^n \tag{6}$$

2b) Averaging of the total displacement vectors at the time $t^{n+1} = t^n + \Delta t$ form Steps 1c and 2a for given $\theta = [0, 1]$.

$$\begin{aligned} \mathbf{u}^{n+1} = \theta \mathbf{u}_{fs}^{n+1} + (1-\theta)\mathbf{u}_{cd}^{n+1} \\ + \text{ application of local Dirichlet boundary conditions} \end{aligned} \tag{7}$$

2c) Evaluation of acceleration and velocity nodal vectors at the time $t^{n+1} = t^n + \Delta t$.

$$\ddot{\mathbf{u}}^{n+1} = \mathbf{M}^{\text{-}1}\left[\mathbf{f}(t^{n+1}) - \mathbf{K}\mathbf{u}^{n+1}\right] \tag{8}$$

$$\begin{aligned} \dot{\mathbf{u}}^{n+1} = \dot{\mathbf{u}}^n + \frac{1}{2}(\ddot{\mathbf{u}}^n + \ddot{\mathbf{u}}^{n+1}) \\ + \text{ application of local Dirichlet boundary conditions} \end{aligned} \tag{9}$$

$\mathbf{M}$ and $\mathbf{K}$ are the global mass and stiffness matrices, $\mathbf{f}$ is the global vector of external forces.

We have manufactured and tested several benchmarks on elastic wave propagation in heterogeneous bars and results were without cardinal spurious oscillations.

4 Conclusion

Based on the tests, we could say that the local time stepping scheme in finite element modelling is able to suppress spurious oscillations in discontinuous wave propagation in general heterogeneous media [5]. Further, we plan to extend the work on multidimensional problems.

Acknowledgement: The work of R. K. was supported by the grant projects with 19-04956S of the Czech Science Foundation (CSF) and, also, the work was supported by the Centre of Excellence for Nonlinear Dynamic Behaviour of Advanced Materials in Engineering CZ.02.1.01/0.0/0.0/15_003/0000493 (Excellent Research Teams) in the framework of Operational Programme Research, Development and Education, within institutional support RVO: 61388998.

References

[1] R. Ebrahimi: *Advances in Functionally Graded Materials and Structures*, InTech, 2016.

[2] K.F. Graff: *Wave Motion in Elastic Solids*, Clarendon Press, 1975.

[3] T.J.R. Hughes: *The Finite Element Method: Linear and Dynamic Finite Element Analysis*, Dover Publications: New York, 2000.

[4] K.C. Park, S.J. Lim, S.J. H. Huh: *A method for computation of discontinuous wave propagation in heterogeneous solids: basic algorithm description and application to one-dimensional problems*, International Journal of Numerical Methods and Engineering 91 (6) 2012, pp. 622–643.

[5] R. Kolman, S.S. Cho, K.C. Park, J.A. González, A. Berezovski, P. Hora, V. Adámek: *A method with local time stepping for discontinuous heterogeneous wave propagation in heterogeneous solids*, To appear International Journal for Numerical Methods in Engineering, 2018.

Selected geotechnical problems solved by the FETI method

J. Kruis, T. Koudelka

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

1 Introduction

Analysis of the earth pressure is still in the center of attention. Experimental research of the earth pressure should use original medium size experimental device (stand) and numerical analysis requires advanced original software equipped with special material models and tools. The experimental research was done in the Institute of Theoretical and Applied Mechanics of Czech Academy of Sciences [1], [2] and [3]. A special experimental equipment, stand (see figure 1), was constructed and it enables to simulate various movements of retaining structures and soil body. There are possible horizontal translation of the wall and rotations of the wall along the top or bottom edge. The deformation and failure processes of the soil as well as both components of the contact stress, i.e. the normal pressure and vertical friction at the rear face of the retaining structure, were monitored and analyzed. Also the displacement of the front retaining wall was monitored automatically and continuously.

The main aim of the numerical analysis is description of the slip surface evolution. In the first approach, the finite element method (FEM) and the Mohr-Coulomb plasticity model (see [4]) were used for description of the response of soil in the experimental device for small angles of the front wall. The results showed that the plasticity model can be easily and sufficiently used. The numerical model determines the slip surface corresponding to the experimentally obtained surfaces. In the case of larger angles, the wedge of soil close to the front wall moves similarly to a rigid body and no additional plastic zones are developed. Only strains localized in the slip surface grow and clear discontinuity is developing.

Description of discontinuity along the slip surface requires special attention. Displacement field is not continuous and the classical formulation of the FEM is not applicable. Various types of contact elements located in the discontinuity are very popular but difficult determination of their material parameters is the main disadvantage. Additional problems are connected with the motion of the wedge which is similar to the rigid body motion. In such a case, the stiffness matrix is split into two parts where one of them is nearly singular. It may lead to collapse of a method of solution of equation systems. Application of a suitable domain decomposition method can lead to efficient description of the discontinuous displacement. The most suitable method is the FETI method which uses the rigid body modes. A special interface condition on the slip surface can be easily prescribed.

2 FETI method for the interface problems

Modification of the FETI method for problems dealing with perfect or imperfect bonds was introduced in reference [5]. The continuity condition between subdomains is replaced by an interface condition which can be expressed in the form

$$\boldsymbol{B}\boldsymbol{d} = \boldsymbol{c} \tag{1}$$

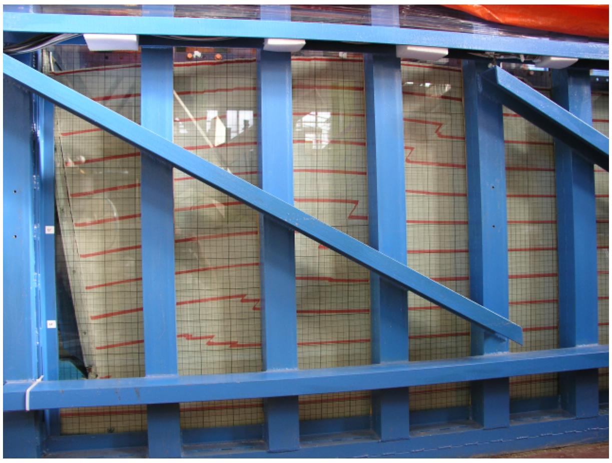


Figure 1: The view on the stand and on the deformed specimen with the evolved slip surface.

where $\boldsymbol{c}$ denotes the vector of differences between two adjacent unknowns defined in the same point on the interface, $\boldsymbol{d}$ is the vector of nodal unknowns and $\boldsymbol{B}$ is the Boolean matrix. In the case of perfect bond, the vector $\boldsymbol{c}$ is the zero vector. The coarse problem of the FETI method with the interface condition has the form

$$\begin{pmatrix} \boldsymbol{B}\boldsymbol{K}^{+}\boldsymbol{B}^{T} & -\boldsymbol{B}\boldsymbol{R} \\ -\boldsymbol{R}^{T}\boldsymbol{B}^{T} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B}\boldsymbol{K}^{+}\boldsymbol{f} - \boldsymbol{c} \\ -\boldsymbol{R}^{T}\boldsymbol{f} \end{pmatrix} \tag{2}$$

where the classical FETI notation is used.

In the case of linear relationship between interface stresses ($\boldsymbol{\lambda}$ represents interface nodal forces which can be transformed into stresses) and the slip ($\boldsymbol{c}$), the following relation can be used

$$\boldsymbol{c} = \boldsymbol{H}\boldsymbol{\lambda} \tag{3}$$

where $\boldsymbol{H}$ denotes the compliance matrix. Generally, the matrix $\boldsymbol{H}$ can depend on attained Lagrange multipliers $\boldsymbol{\lambda}$. Substitution of (3) to the system (2) results in

$$\begin{pmatrix} \boldsymbol{B}\boldsymbol{K}^{+}\boldsymbol{B}^{T} + \boldsymbol{H} & -\boldsymbol{B}\boldsymbol{R} \\ -\boldsymbol{R}^{T}\boldsymbol{B}^{T} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B}\boldsymbol{K}^{+}\boldsymbol{f} \\ -\boldsymbol{R}^{T}\boldsymbol{f} \end{pmatrix} \tag{4}$$

If the perfect bond is taken into account, the compliance matrix $\boldsymbol{H}$ is zero matrix and the classical FETI method is obtained. Otherwise, a nonzero compliance matrix $\boldsymbol{H}$ is added to the coarse problem. In many cases, the matrix $\boldsymbol{H}$ is a diagonal matrix and therefore it causes no difficulty. The modified conjugate gradient method usually used in the FETI approach can solve the coarse problem (4).

3 Numerical simulation

The specimen was created from the dry sand and this material can be modelled by plasticity material models. The general approach to the plasticity models can be found in many references, for example in books [4] and [6]. For the first approach, the rate independent plasticity model with Mohr-Coulomb yield criterion was used.

The specimen had dimensions $1.2 \times 1.0 \times 3.0$ m and it was composed of 11 horizontal layers. The model is depicted in Figure 2. The finite element mesh was generated using the 3D hexahedron elements with linear approximation functions. The friction angle $\phi = 35^o$ and the cohesion $c = 1.8$ kPa were used in the Mohr-Coulomb criterion and they were obtained by the performed shear tests. The top surface of each layer was loaded according to the total compacting energy and after the last compacting load was applied, the front wall movement (rotation about the top edge) was simulated by the increase in the prescribed displacement.

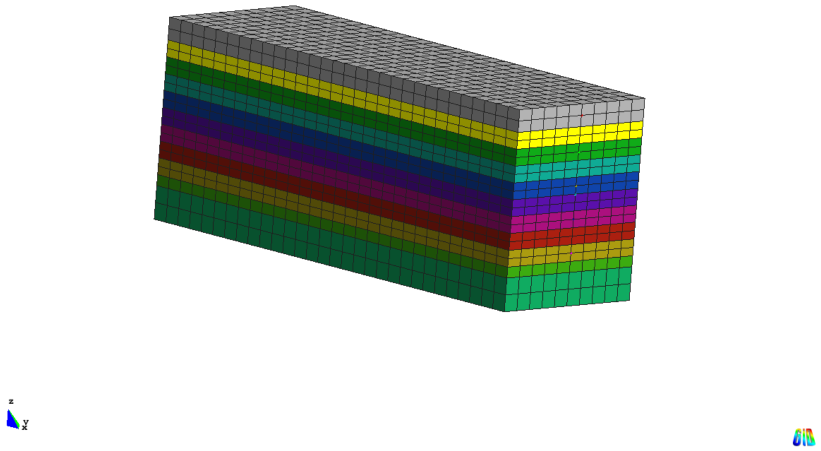


Figure 2: FE mesh of the specimen

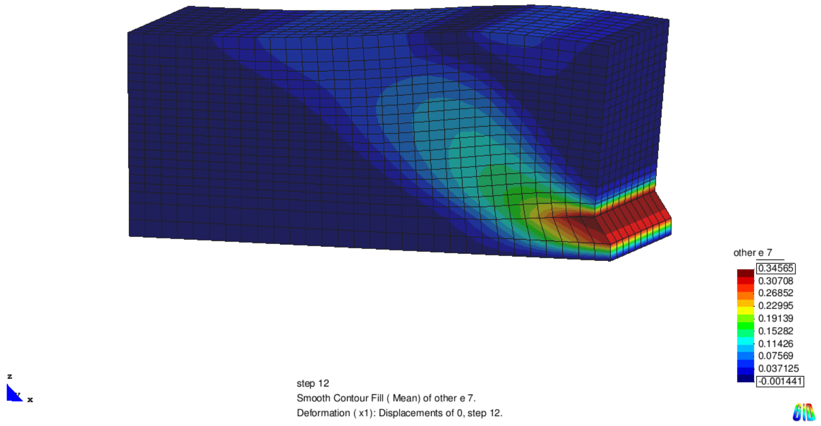


Figure 3: The distribution of consistency parameter for $u = 100$ mm.

Figure 3 represent distribution of the consistency parameter γ, which indicates plastic zones, for displacements $u = 100$ mm of the front wall bottom.

The analysis based on the Mohr-Coulomb plasticity model determined the slip surface. The finite element mesh depicted in Figure 2 was decomposed into submeshes with respect to the slip surface. It means, the slip surface defined subdomain interfaces and the subdomains were split into two groups. First group of subdomains describes the wedge of soil close to the rotated wall which behaves similarly to a rigid body. Second group contains subdomains which described the remaining part of the soil body. Therefore, the system of equations (4) was used for description of the slip.

4 Conclusion

Modified FETI method was used for numerical analysis of earth pressures. The numerical simulation was compared with experimentally obtained data from a stand and very good agreement was achieved. The modified FETI method can be used in various geotechnical problems where failure or slip surfaces occur.

Acknowledgement: This outcome has been achieved with the financial support of the Grant Agency of the Czech Republic, project No. 19-11441S. The financial support is gratefully acknowledged.

References

[1] P. Koudelka: *Numerical Analysis of a Physical Experiment with Retained Mass by GLPT.* In: Proc. RC Geotechnical Engineering in Soft Ground, Tongji University Press, Shanghai, China, 2001, pp. 563–568.

[2] P. Koudelka, T. Koudelka: *Briefly on the Extreme and Intermediate Lateral Pressures on Structures.* In: Gudehus et.al. (ed.): Proc. 12th DEC Geotechnical Engineering, Passau, DGGT, 2001, pp. 343–346.

[3] P. Koudelka, T. Koudelka: *Advanced numerical model based on the theory of General Lateral Pressure.* In: Proceedings of the Fifteenth International Conference on Soil Mechanics and Geotechnical Engineering, A.A. Balkema Publishers, Leiden, Netherlands, 2001, vols 1–3, pp. 1175–1178.

[4] M. Jirásek, Z.P. Bažant: *Inelastic Analysis of Structure.* John Wiley & Sons, Chichester-Toronto, 2001.

[5] J. Kruis, Z. Bittnar: *Reinforcement-matrix interaction modelled by FETI method.* In: U. Langer, M. Discacciati, D. Keyes, O. Widlund, W. Zulehner (ed.): Domain Decomposition Methods in Science and Engineering XVII, Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2007, pp. 567–573.

[6] E.A. de Souza Neto, D. Perić, D.R.J. Owen: *Computational Methods for Plasticity.* John Wiley & Sons, Chichester, 2008.

Deflation spaces for the conjugate gradient method

J. Kružík[1], *D. Horák*[2]

[1] Institute of Geonics of the CAS, Ostrava
[2] VŠB - Technical University of Ostrava

1 Introduction

Many problems in engineering, finance, etc. eventually lead to a systems of linear equations of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{A}$ is n-dimensional symmetric positive definite (SPD) matrix.

The conjugate gradient (CG) algorithm is often the method of choice for the solution of such systems. In order to accelerate the convergence of CG we often need a good preconditioner.

However, there also exists a complementary approach to the preconditioning known as deflation. The deflation utilizes a deflation space that should represent slowly converging components of the solution.

In this work, we discuss the choice of the deflation space and also demonstrate the behaviour of several deflation spaces on various benchmarks.

2 Deflated Conjugate Gradient Method

The deflated conjugate gradient method [1], introduced in [2, 3, 4], works by splitting the solution of Equation (1) into two parts. The first part represents the solution on the deflation space and is directly obtained. The second one is computed by CG iterations that operate only on the $\boldsymbol{A}$-conjugate complement of the deflation space.

Let us define a full rank deflation matrix

$$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_m) \in \mathbb{R}^{n\times m}, m < n$$

and let $\mathcal{W}$ be a subspace spanned by columns of $\boldsymbol{W}$. Then we can denote a projector

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{W}\left(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{Q}\boldsymbol{A}$$

onto an $\boldsymbol{A}$-conjugate complement of $\mathcal{W}$.

Given an arbitrary initial guess $\boldsymbol{x}_{-1}$ and defining the residual $\boldsymbol{r}_{-1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{-1}$ we can choose $\boldsymbol{x}_0$ to be

$$\boldsymbol{x}_0 = \boldsymbol{x}_{-1} + \boldsymbol{W}\left(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\boldsymbol{r}_{-1} = \boldsymbol{x}_{-1} + \boldsymbol{Q}\boldsymbol{r}_{-1}. \tag{2}$$

It is easy to show that $\boldsymbol{x}_0$ is the exact solution of (1) in $\mathcal{W}$ and therefore $\boldsymbol{r}_0$ is orthogonal to $\mathcal{W}$. If we use $\boldsymbol{x}_0$ as the initial guess for CG, we obtain the InitCG method [5] illustrated in Algorithm 2.

```
Algorithm 2: InitCG
Input: $A$, $x_{-1}$, $b$, $W$
$r_{-1} = b - Ax_{-1}$
$x_0 = x_{-1} + Qr_{-1}$
$r_0 = b - Ax_0$
$p_0 = r_0$
for $k = 0, \cdots$:
    $s = Ap_k$
    $\alpha_k = \left(r_k^T r_k\right) / \left(s^T p_k\right)$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k s$
    $\beta_{k+1} = \left(r_{k+1}^T r_{k+1}\right) / \left(r_k^T r_k\right)$
    $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$
Output: $x_k$
```

```
Algorithm 3: DCG
Input: $A$, $x_{-1}$, $b$, $W$
$r_{-1} = b - Ax_{-1}$
$x_0 = x_{-1} + Qr_{-1}$
$r_0 = b - Ax_0$
$p_0 = Pr_0$
for $k = 0, \cdots$:
    $s = Ap_k$
    $\alpha_k = \left(r_k^T r_k\right) / \left(s^T p_k\right)$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k s$
    $\beta_{k+1} = \left(r_{k+1}^T r_{k+1}\right) / \left(r_k^T r_k\right)$
    $p_{k+1} = Pr_{k+1} + \beta_{k+1} p_k$
Output: $x_k$
```

If the columns of $\boldsymbol{W}$ are exact eigenvectors then, in exact arithmetic, the minimization directions $\boldsymbol{p}_k$ are $\boldsymbol{A}$-orthogonal to $\mathcal{W}$ and we achieved the required splitting. However, in the case of general $\boldsymbol{W}$, we need to keep $\boldsymbol{p}_k$ explicitly $\boldsymbol{A}$-orthogonal to $\mathcal{W}$ by projecting components of the deflation space out of the residuals in the construction of $\boldsymbol{p}_k$. This modification leads to the DCG algorithm illustrated in Algorithm 3.

It can be shown [6] that DCG act as CG "preconditioned" by the projector $\boldsymbol{P}$ as the convergence is governed by the spectrum of $\boldsymbol{PA}$ operator.

3 Choice of the Deflation Space

A good choice of deflation space is crucial for making DCG converge quickly. In practice, there were two main deflation spaces.

The first one uses eigenvectors of $\boldsymbol{A}$ as the deflation space. The associated eigenvalues of the eigenvectors belonging to the deflation space are shifted to zero in the spectrum of the DCG operator $\boldsymbol{PA}$. Particularly, eigenvectors belonging to the smallest eigenvalues are used as they slow down the convergence of CG the most. In our experiments, this approach works very well. The problem is how to obtain the eigenvectors.

The second approach is subdomains aggregation. Given a decomposition of the computational domain, each subdomain contributes a single vector into the deflation space. This vector contains ones on the indices of unknowns belonging to the subdomain and zeros otherwise. Such space often approximates a similar space as in the eigenvector approach. We can use, e.g., METIS to obtain the domain decomposition. However, assuming a single computational core owns the whole subdomain then, to utilize the cores appropriately, the subdomains have to be fairly large making the deflation space too coarse to be effective.

A new approach based on wavelet compression was suggested in [7]. The basic idea is that given the wavelet scaling coefficients $h_1, \ldots, h_k$ we create a projection onto the scaling subspace

$$\boldsymbol{H}_{1,n} = \begin{pmatrix} h_1 & h_2 & h_3 & \ldots & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_1 & h_2 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{k-1} & h_k & 0 & 0 & 0 & \cdots & h_{k-3} & h_{k-2} \end{pmatrix} \in \mathbb{R}^{\frac{n}{2} \times n}.$$

Then $\boldsymbol{H}_{1,n}\boldsymbol{A}\boldsymbol{H}_{1,n}^T$ contains trends of $\boldsymbol{A}$. Moreover, we can repeat this compression process to use up to m levels of the compression

$$\boldsymbol{H}_{1,n/2^{m-1}} \dots \boldsymbol{H}_{1,n/2}\boldsymbol{H}_{1,n}\boldsymbol{A}\boldsymbol{H}_{1,n}^T\boldsymbol{H}_{1,n/2}^T \dots \boldsymbol{H}_{1,n/2^{m-1}}^T = \boldsymbol{H}_{m,n}\boldsymbol{A}\boldsymbol{H}_{m,n}^T$$

Since $\boldsymbol{H}_{m,n}$ cuts off the high frequencies, we can set $\boldsymbol{W} = \boldsymbol{H}_{m,n}^T$.

The suggested wavelet compression is also used in the algebraic multigrid [8]. Therefore, using the prolongation matrices from multigrid in place of the deflation matrix might work as well. Moreover, the prolongation operators can be chained, as in the wavelet-based deflation, without the use of any smoothers between multigrid levels.

4 Numerical Experiments

In order to evaluate the aforementioned deflation spaces, an efficient, parallel implementation of DCG was created. It is written as a solver for linear systems in PETSc [9] (KSP). Currently, it is part of the PETSc-based, open-source PERMON library [10].

The benchmarks used in the numerical experiments include all 236 SPD matrices from SuiteSparse Matrix collection, 2D Laplace on a rectangular domain with a hole and 3D linear elasticity multi-material cantilever beam discretized with MFEM [11], and 2D Laplace discretized by boundary element method on an L-shaped domain. The results with appropriate discussion can be found in [6]. A comparison of numerical scalability of CG (none) and DCG with 5 and 40 eigenvectors (eig5, eig40) belonging to the smallest eigenvalues, subdomain aggregations (agg), multigrid prolongations (mg), and Haar wavelet (db2) deflation spaces is depicted in Figure 1.

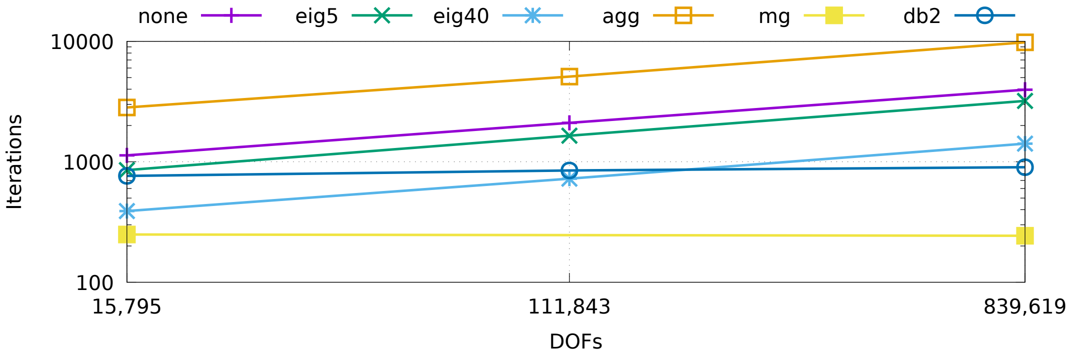


Figure 1: Number of iterations for various deflation spaces on 3D elasticity benchmark.

5 Conclusion

This work demonstrates the usefulness of deflation schemes for Krylov subspace methods. A significant reduction in the number of iterations, as well as time to solution, can be achieved by using appropriate deflation spaces. Using novel wavelet-based deflation or multigrid prolongation operators yields very good results on wide variety benchmarks solved by deflated CG.

Acknowledgement: This work has been supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602; by the Large Infrastructures for Research, Experimental Development and Innovations project “IT4Innovations National Supercomputing Center - LM2015070”; by the grant of SGS No. SP2018/165, VSB - Technical University of Ostrava, Czech Republic and by the grant of the Czech Science Foundation (GACR) project no. GA17-22615S. The work has also been performed under the PRACE-DECI14 project PERMON and the computer resources provided by Edinburgh Parallel Computing Centre (EPCC).

References

[1] Y. Saad, M. Yeung, J. Erhel, F. Guyomarc'h: *A deflated version of the conjugate gradient algorithm.* SIAM Journal on Scientific Computing, vol. 21, no. 5, 2000, pp. 1909–1926. doi: 10.1137/S1064829598339761.

[2] R.A. Nicolaides: *Deflation of conjugate gradients with applications to boundary value problems.* SIAM Journal on Numerical Analysis, vol. 24, no. 2, 1987, pp. 355–365. doi: 10.1137/0724027.

[3] G.I. Marchuk, Y.A. Kuznetsov: *Theory and applications of the generalized conjugate gradient method.* Advances in Mathematics. Supplementary Studies, vol. 10, 1986, pp. 153–167.

[4] Z. Dostál: *Conjugate gradient method with preconditioning by projector.* Int. Journal of Computer Mathematics, vol. 23, no. 3-4, 1988, pp. 315–323. doi:10.1080/00207168808803625.

[5] J. Erhel, F. Guyomarc'h: *An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems.* SIAM Journal on Matrix Analysis and Applications, vol. 21, no. 4, 2000, pp. 1279–1299.

[6] J. Kruzik: *Implementation of the deflated variants of the conjugate gradient method.* Master's thesis, VSB - Technical University of Ostrava, 2018.

[7] J. Kruzik, D. Horak: *Wavelet based deflation of conjugate gradient method.* In: Proceedings of the Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press, Stirlingshire, UK, 2017. doi: 10.4203/ccp.111.9.

[8] W.L. Briggs, V.E. Henson: *Wavelets and multigrid.* In: SIAM Journal on Scientific Computing, vol. 14, no. 2, 1993, pp. 506–510.

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

[10] *PERMON (Parallel, Efficient, Robust, Modular, Object oriented, Numerical) web page.* http://permon.vsb.cz/

[11] *MFEM: Modular finite element methods.* http://mfem.org

On the time growth of the error of the discontinuous Galerkin method

V. Kučera[1], *C.-W. Shu*[2]

[1] Faculty of Mathematics and Physics, Charles University in Prague
[2] Division of Applied Mathematics, Brown University, Providence, USA

1 Introduction

We present an overview of the authors' paper [2] on the time growth of the error of the discontinuous Galerkin (DG) method. In the theory of evolutionary problems, Gronwall's lemma is an often used standard tool which allows one to obtain estimates of some desired quantity. However, Gronwall's lemma leads to the appearance of a factor which grows exponentially with respect to time in the resulting inequality, even for problems where such exponential growth is unnatural. In [2] we analyze the time growth of the error of the DG method applied to a linear nonstationary advection-reaction problem. To circumvent the use of Gronwall's lemma, we introduce a space-time exponential scaling of the error. This allows one to obtain an additional elliptic term from the DG formulation which allows one to avoid the use of Gronwall's lemma. The result is an error estimate which grows exponentially not in time, but in the time particles carried by the flow field spend in the spatial domain. If this is uniformly bounded, one obtains an error estimate of the form $C(h^{p+1/2})$, where C is independent of time.

2 Problem formulation and analysis

Let $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$ be a bounded polygonal (polyhedral) domain with Lipschitz boundary $\partial\Omega$. Let $0 < T \leq +\infty$ and let $Q_T = \Omega \times (0, T)$ be the space-time domain. We consider the following nonstationary advection-reaction equation: We seek $u : Q_T \to \mathbb{R}$ such that

$$\frac{\partial u}{\partial t} + a\cdot\nabla\, u + cu = 0 \quad \text{in } Q_T, \tag{1}$$

along with the initial condition $u(x,0) = u^0(x)$ and boundary condition $u = u_D$ on the inflow boundary $\partial\Omega^- \times (0,T)$. Here $a : \overline{Q_T} \to \mathbb{R}^d$ and $c : \overline{Q_T} \to \mathbb{R}$ are the given advective field and reaction coefficient, respectively. We assume that $c \in C([0,T); L^\infty(\Omega)) \cap L^\infty(Q_T)$ and $a \in C([0,T); W^{1,\infty}(\Omega))$ with $a, \nabla a$ uniformly bounded a.e. in Q_T.

Usually when dealing with problem (1) throughout the numerical literature, one assumes ellipticity of the resulting advection and reaction weak forms, which leads to the requirement

$$c - \tfrac{1}{2}\mathrm{div}a \geq \gamma_0 > 0 \qquad \text{on } Q_T \tag{2}$$

for some constant $\gamma_0 > 0$. The additional ellipticity allows one to obtain estimates of the solution or the error of the chosen numerical method that are uniform in time. The problem is that from the point of view of PDE theory, assumption (2) is entirely artificial.

In order to avoid the artificial condition (2), in [2] we introduced the following space-time exponential scaling. Let $\mu : Q_T \to \mathbb{R}$ be a given function which will be chosen appropriately in the analysis. We write the solution of (1) as

$$u(x,t) = e^{\mu(x,t)}\tilde{u}(x,t). \tag{3}$$

Substituting (3) into (1) gives

$$\frac{\partial \tilde{u}}{\partial t} + a \cdot \nabla \tilde{u} + \Big(\frac{\partial \mu}{\partial t} + a\cdot \nabla \mu + c\Big)\tilde{u} = 0 \tag{4}$$

after dividing by the common positive factor e^{μ}. Problem (4) is an equation for the new unknown $\tilde{u}$. The condition corresponding to (2) now reads: There exists $\mu : Q_T \to \mathbb{R}$ such that

$$\frac{\partial \mu}{\partial t} + a\cdot \nabla \mu + c - \tfrac{1}{2}\mathrm{div} a \geq \gamma_0 > 0 \quad \text{a.e. in } Q_T. \tag{5}$$

We observe that by choosing μ appropriately, one has more room to satisfy (5) even when the original ellipticity condition (2) is not satisfied. We note that simpler versions of the general space-time exponential scaling have been used in the literature, cf. e.g. [1], [3].

2.1 Construction of the scaling function μ

If $c - \frac{1}{2}\mathrm{div} a$ is negative or changes sign frequently, we can use the expression $\mu_t + a \cdot \nabla \mu$ to dominate this term everywhere. If we choose μ_1 such that

$$\frac{\partial \mu_1}{\partial t} + a \cdot \nabla \mu_1 = 1 \quad \text{on } Q_T, \tag{6}$$

then by multiplying μ_1 by a sufficiently large constant, we can satisfy condition (5) for a chosen $\gamma_0 > 0$. To solve equation (6), we define *pathlines* of the flow, i.e. the family of curves $S(t; x_0, t_0)$, each originating at (x_0, t_0), by

$$S(t_0; x_0, t_0) = x_0 \in \overline{\Omega}, \quad \frac{\mathrm{d}S(t; x_0, t_0)}{\mathrm{d}t} = a(S(t; x_0, t_0), t).$$

This means that $S(\cdot; t_0, x_0)$ is the trajectory of a massless particle in the nonstationary flow field a passing through point x_0 at time t_0. Along pathlines equation (6) reads

$$\frac{\mathrm{d}\, \mu_1(S(t; x_0, t_0), t)}{\mathrm{d}t} = \Big(\frac{\partial \mu_1}{\partial t} + a \cdot \nabla \mu_1\Big)(S(t; x_0, t_0), t) = 1,$$

therefore

$$\mu_1(S(t; x_0, t_0), t) = t - t_0. \tag{7}$$

At the origin of the pathline, we have $\mu_1(S(t_0; x_0, t_0), t_0) = 0$ and the value of μ_1 along this pathline is simply the time elapsed since t_0. In the following, we wish to keep μ_1 uniformly bounded. One case when this can occur is when the maximal particle 'life-time' $\widehat{T}$ is finite. By this we mean that the maximal time any massless particle carried by the flow field a spends in Ω, before exiting through the outflow boundary, is bounded by $\widehat{T} < +\infty$.

Under the mentioned assumption in (7) we have $|t - t_0| < \widehat{T}$, hence uniform boundedness of μ on Q_T. In the analysis we need Lipschitz continuity of μ. This can be obtained under the assumption that there are no characteristic boundary points on the inlet boundary. The proof of the following theorem is rather technical, cf. [2]. Since μ_1 is defined very simply along pathlines, which are solutions of ordinary differential equations, the proof follows similar ideas as in the proof of dependence of a solution of an ODE on the initial condition.

Theorem 1 *Let $a \in L^{\infty}(Q_T)$ be continuous with respect to time and Lipschitz continuous with respect to space. Let there exist a constant $a_{\min} > 0$ such that*

$$-a(x, t) \cdot \mathbf{n} \geq a_{\min}$$

for all $x \in \partial\Omega^-, t \in [0, T)$. Let the time any particle carried by the flow field $a(\cdot, \cdot)$ spends in Ω be uniformly bounded by $\widehat{T}$. Then μ_1 defined by (7) on $\overline{\Omega} \times [0, T)$ is uniformly Lipschitz continuous with respect to x and t and satisfies $0 \leq \mu_1 \leq \widehat{T}$.

2.2 Error estimates

Now we introduce the DG discretization of (1). Let $\mathcal{T}_h$ be a triangulation (partition into mutually disjoint simplices) with hanging nodes allowed. For $K \in \mathcal{T}_h$ let $h_K = \operatorname{diam}(K)$, $h = \max_{K\in\mathcal{T}_h} h_K$. For $K \in \mathcal{T}_h$ we set $\partial K^-(t) = \{x \in \partial K; a(x,t)\cdot \mathbf{n}(x) < 0\}$ where $\mathbf{n}(x)$ is the unit outer normal to ∂K. We seek the discrete solution in the space $S_h = \{v_h;\, v_h|_K \in P^p(K), \forall K \in \mathcal{T}_h\}$, where $P^p(K)$ is the set of polynomials on K of degree at most p. For $K \in \mathcal{T}_h$ and $v_h \in S_h$ let v_h^- be the trace of v_h on ∂K from the side of the element adjacent to K, or $v_h^- = 0$ if the face lies on $\partial\Omega$. Finally on ∂K we define the *jump* of v_h as $[v_h] = v_h - v_h^-$, where v_h is the trace from K. We seek $u_h \in C^1([0,T);S_h)$ such that $u_h(0) = u_h^0 \approx u^0$ and

$$\Big(\frac{\partial u_h}{\partial t}, v_h\Big) + b_h(u_h, v_h) + c_h(u_h, v_h) = l_h(v_h), \quad \forall v_h \in S_h. \tag{8}$$

Here b_h, c_h and l_h are the *advection, reaction* and *right-hand side forms*, respectively, defined for u, v piecewise continuous on $\mathcal{T}_h$ in a standard way, cf. [2].

We estimate the DG error $e_h(t) := u(t) - u_h(t) = \eta(t_n) + \xi(t)$, where $\eta(t) = u(t) - \Pi_h u(t)$ and $\xi(t) = \Pi_h u(t) - u_h(t) \in S_h$. Here Π_h is the $L^2(\Omega)$–projection onto S_h. As in (3), we wish to write $\xi = e^{\mu}\tilde{\xi}$. Furthermore, in the weak setting the analogy to dividing the common factor e^{μ} to obtain (4) is setting the test function as $\phi = e^{-\mu}\tilde{\xi} = e^{-2\mu}\xi$ to obtain estimates for $\tilde{\xi}$. However, since $\phi(t) \notin S_h$ this is not possible. The solution is to test by $\Pi_h\phi(t) \in S_h$ and estimate the difference $\Pi_h\phi(t) - \phi(t)$.

Lemma 1 *Let μ be globally bounded and Lipschitz continuous as in Theorem 1. Then there exists C independent of $h, t, \xi, \tilde{\xi}$ such that*

$$\|\Pi_h\phi(t) - \phi(t)\|_{L^2(K)} \le C h_K \max_{x\in K} e^{-\mu(x,t)} \|\tilde{\xi}(t)\|_{L^2(K)},$$
$$\|\Pi_h\phi(t) - \phi(t)\|_{L^2(\partial K)} \le C h_K^{1/2} \max_{x\in K} e^{-\mu(x,t)} \|\tilde{\xi}(t)\|_{L^2(K)}.$$

Now we come to the error analysis. We subtract the equations for u and u_h, set $v_h = \Pi_h\phi(t)$ and rearrange the terms to get the error equation

$$\begin{aligned}\Big(\frac{\partial \xi}{\partial t}, \Pi_h\phi\Big) &+ b_h(\xi,\phi) + b_h(\xi, \Pi_h\phi - \phi) + b_h(\eta, \Pi_h\phi)\\ &+ c_h(\xi,\phi) + c_h(\xi, \Pi_h\phi - \phi) + c_h(\eta, \Pi_h\phi) + \Big(\frac{\partial \eta}{\partial t}, \Pi_h\phi\Big) = 0.\end{aligned} \tag{9}$$

The terms with ϕ are those where the factors e^{μ} and $e^{-\mu}$ cancel out leading to the new reaction terms as in (4). Terms containing $\Pi_h\phi - \phi$ are estimated using Lemma 1 and η is estimated by standard approximation results. Altogether we have the following, cf. [2].

Lemma 2 *Let $\xi = e^{\mu}\tilde{\xi}, \phi = e^{-\mu}\tilde{\xi}$ and let μ be as in Theorem 1. Then*

$$\Big(\frac{\partial \xi}{\partial t}, \Pi_h\phi\Big) + b_h(\xi,\phi) + c_h(\xi,\phi) \ge \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\tilde{\xi}\|^2 + \gamma_0\|\tilde{\xi}\|^2 + \frac{1}{2}\sum_{K\in\mathcal{T}_h} \big\|[\tilde{\xi}]\big\|^2_{a,\partial K^-}$$

where $\|f\|_{a,\partial K^-} = \|\sqrt{|a\cdot \mathbf{n}|}f\|_{L^2(\partial K^-)}$.

Lemma 3 *Let ξ, ϕ and μ be as above. Then*

$$\begin{aligned}&\Big|b_h(\xi, \Pi_h\phi - \phi) + b_h(\eta, \Pi_h\phi) + c_h(\xi, \Pi_h\phi - \phi) + c_h(\eta, \Pi_h\phi) + \Big(\frac{\partial \eta}{\partial t}, \Pi_h\phi\Big)\Big|\\ &\quad \le Ch\|\tilde{\xi}\|^2 + Ch^{2p+1}\big(|u(t)|^2_{H^{p+1}} + |u_t(t)|^2_{H^{p+1}}\big) + \frac{1}{4}\sum_{K\in\mathcal{T}_h}\big\|[\tilde{\xi}]\big\|^2_{a,\partial K^-}.\end{aligned}$$

Now we come to the main theorem of [2] on the error of the DG scheme (8).

Combining Lemmas 2 and 3 gives an estimate of $\tilde{\xi}$. In order to get an estimate of ξ, hence the error e_h, we can write

$$\|\tilde{\xi}(t)\|^2 \geq \min_{Q_T} e^{-2\mu(x,t)}\|\xi(t)\|^2 = e^{-2\max_{Q_T}\mu(x,t)}\|\xi(t)\|^2 \geq e^{-2\widehat{T}}\|\xi(t)\|^2$$

Multiplying the resulting estimate by the factor $e^{-2\widehat{T}}$ and taking the square root gives the exponential factor $e^{\widehat{T}}$ in the resulting estimate instead of the standard Gronwall factor e^{T}.

Theorem 2 *Let the assumptions of Theorem 1 hold. Let the initial condition u_h^0 satisfy $\|u^0 - u_h^0\| \leq Ch^{p+1/2}|u^0|_{H^{p+1}}$. Then there exists a constant C depending on $\widehat{T}$ but independent of h and T such that for h sufficiently small*

$$\begin{aligned}\max_{t\in[0,T]}\|e_h(t)\| + \sqrt{\gamma_0}\|e_h\|_{L^2(Q_T)} + \Big(\frac{1}{2}\int_0^T \sum_{K\in\mathcal{T}_h}\big\|[e_h(\vartheta)]\big\|^2_{a,\partial K^-}\,\mathrm{d}\vartheta\Big)^{1/2}& \\ \leq Ch^{p+1/2}\big(|u^0|_{H^{p+1}} + |u|_{L^2(H^{p+1})} + |u_t|_{L^2(H^{p+1})}\big).&\end{aligned} \tag{10}$$

The interpretation of Theorem 2 is this: If one proceeds in a standard way, the need to use Gronwall's lemma arises. This leads to exponential growth in T. By using exponential scaling we effectively apply Gronwall's lemma along pathlines, which exist only for a finite time $\widehat{T}$, resulting in bounds uniform in T. This can be interpreted as application of Gronwall in the Lagrangian framework, not in the Eulerian. We note that the obtained results would hold if equation (1) were in divergence form with a nonzero divergence of a. This follows from the relation $\mathrm{div}(au) = a\cdot\nabla u + u\,\mathrm{div}a$, which recasts the divergence form into that of (1) with the new reaction coefficient $\tilde{c} = c + \mathrm{div}a$.

Acknowledgement: The work of V. Kučera was supported by the J. William Fulbright Commission in the Czech Republic and research project No. 17-01747S of the Czech Science Foundation. The work of C.-W. Shu was supported by DOE grant DE-FG02-08ER25863 and NSF grants DMS-1418750 and DMS-1719410.

References

[1] B. Ayuso, L.D. Marini: *Discontinuous Galerkin methods for advection-diffusion-reaction problems*, SIAM J. Numer. Anal. **47**:2, 2009, pp. 1391–1420.

[2] V. Kučera, C.-W. Shu: *On the time growth of the error of the DG method for advective problems*, IMA J. Numer. Anal. (to appear), DOI: 10.1093/imanum/dry013, arXiv:1711.09417.

[3] U. Nävert: *A finite element method for convection-diffusion problems*, Ph.D. thesis, Chalmers University of Technology, 1982.

Multiple solutions to steady flow problems involving *do-nothing* or given traction boundary conditions

M. Lanzendörfer[1], *J. Hron*[2]

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

In the practical problems of (computational) fluid dynamics, it is usually important to restrict one's consideration into a bounded domain of interest, despite of the fact that the domain does not cover the entire area of the flow. Certain parts of the domain boundary are then *artificial boundaries*: they allow for an inflow or outflow of the fluid and are not related to any natural physical interface, since they only represent a truncation of the flow extending beyond the considered domain. The choice of boundary conditions to be imposed on artificial boundaries is a question that cannot be answered only on the basis of mathematical, or physical, considerations alone. It should be addressed from a combined viewpoint, which includes also the numerical and modelling considerations.

We focus on one particular aspect of one class of inflow and outflow boundary conditions for the steady Navier–Stokes system

$$\left.\begin{aligned} \operatorname{div}\boldsymbol{v} &= 0 \\ \operatorname{div}(\boldsymbol{v}\otimes\boldsymbol{v}) - \operatorname{div}\boldsymbol{T} &= \boldsymbol{0} \end{aligned}\right\} \text{ in } \Omega, \text{ where} \qquad \begin{aligned} &\boldsymbol{T} = -p\boldsymbol{I} + \boldsymbol{S}, \\ &\boldsymbol{S} = \nu\left(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T\right), \end{aligned}$$

where $\boldsymbol{v}$, p, $\boldsymbol{T}$, $\boldsymbol{S}$ and $\nu > 0$ stand for the velocity, the kinematic pressure, Cauchy stress tensor and its viscous part, and the kinematic viscosity, respectively. In particular, we deal with the problems subject to the *given constant traction* boundary condition on a part of boundary:

$$-\boldsymbol{T}\boldsymbol{n} \equiv p\boldsymbol{n} - \boldsymbol{S}\boldsymbol{n} \equiv p\boldsymbol{n} - \nu\,(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T)\boldsymbol{n} = \boldsymbol{b} \qquad \text{on } \Gamma_{\boldsymbol{b}} \subset \partial\Omega, \tag{1}$$

where $\boldsymbol{n}$ is the unit outer normal vector to the boundary $\partial\Omega$ and $\boldsymbol{b} \equiv \boldsymbol{b}(\boldsymbol{x})$ is a given vector of traction, a common choice being $\boldsymbol{b} = P\boldsymbol{n}$ with some $P \in \mathbb{R}$. Alternatively, we deal with the analogous condition which can be called the *full-gradient-traction* condition

$$p\boldsymbol{n} - \nu\,(\nabla\boldsymbol{v})\boldsymbol{n} \equiv p\boldsymbol{n} - \nu\,\frac{\partial\boldsymbol{v}}{\partial\boldsymbol{n}} = \hat{\boldsymbol{b}} \qquad \text{on } \Gamma_{\hat{\boldsymbol{b}}} \subset \partial\Omega, \tag{2}$$

with the given data $\hat{\boldsymbol{b}} \equiv \hat{\boldsymbol{b}}(\boldsymbol{x})$. This, in the special case $\hat{\boldsymbol{b}} = \hat{P}\boldsymbol{n}$, or $\hat{\boldsymbol{b}} \equiv \boldsymbol{0}$ in particular, represent the so-called *do-nothing* boundary condition well established in practical numerical simulations, see [1].

The conditions (1) and (2) are used frequently in numerical simulations of flows of incompressible fluids, despite of the fact that they do not facilitate the well-posedness of the problem. It is a well known fact that they do not allow for standard energy estimates and, consequently, it has been impossible so far to establish the existence theory except for small data. While the non-uniqueness of steady solutions seems to be expected intuitively, no concrete example of multiple solutions has been given in the literature so far, to the best of our knowledge. Indeed, e.g. as Galdi comments in [2] after proving that for small data there is a unique *small* solution

> "...the question of whether a given solution is unique in the class of all possible weak solutions corresponding to the same data ... is, to date, open, in the case of *do-nothing* conditions."

Despite of this lack of well-posedness theory, the addressed conditions are used in numerical simulations by many researchers on a regular basis, since in many problem settings and flow regimes they are experienced to deliver a unique solution.

The scope of our contribution is to present a set of examples of multiple solutions to the steady flow subject to the conditions that include (1) or (2). We aim to demonstrate one particular simple mechanism behind the non-uniqueness, and to pursue the behaviour of the steady solutions, of the corresponding unsteady flows and of their numerical approximation. We observe multiple solutions for small boundary data, including the case of trivial data where both the trivial and non-trivial solutions can be found. On the other hand, for some instances of large boundary data, the considerations in a simplified setting and our numerical simulations indicate the possibility that no steady solution would exist, this fact being related to the very same mechanism.

We start by studying the isotropic radial planar flow, where the artificial boundaries are considered at the radii $0 < R_1 < R_2$, with the boundary data given there as $\boldsymbol{b} = P_i \boldsymbol{n}$ (or $\hat{\boldsymbol{b}} = P_i \boldsymbol{n}$) at R_i, $i = 1, 2$. Given the difference $P_1 - P_2$, the steady problem then reduces to the task of finding one constant, the flow rate $Q \in \mathbb{R}$, which is observed to be the solution of a quadratic equation. It appears that there is a critical value $P_{\text{crit}} > 0$, such that there are two such steady solutions for $P_1 - P_2 < P_{\text{crit}}$ (which includes the case of trivial boundary data), while for $P_1 - P_2 > P_{\text{crit}}$ there is no (isotropic radial) steady solution.

Taking the advantage of this reduced setting we continue by studying the unsteady problem (with stationary data). The isotropic radial solution is then given by a single function of time, $Q(t)$, found by solving the corresponding ordinary differential equation. We find that the solution either converges asymptotically to one of the steady solutions or it blows up in finite time.

We also report the results of the finite element simulations of the flow. Examining first the aforementioned isotropic radial setting, we observe that a common numerical scheme based on Newton's method can find the both of the two steady solutions, depending on the given initial guess. Examining the unsteady case, we can confirm numerically the behaviour found analytically, including the blow-up of the unsteady solutions in finite time. We then focus on more practical examples: Examining the planar flow in a diverging channel, we obtain numerical results that are qualitatively similar to those in the isotropic radial setting, including finding a non-trivial solution to the trivial data problem with the *do-nothing* boundary conditions. Finally, we provide a numerical example addressing a hemodynamical flow problem: two different solutions to the flow through a bifurcating tube with two outflow sections ended by the *do-nothing* boundaries.

References

[1] J.G. Heywood, R. Rannacher, S. Turek: *Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations*, Int. J. Numer. Meth. Fluids 22(5), 1996, pp. 325–352.

[2] G.P. Galdi: *Mathematical Problems in Classical and Non-Newtonian Fluid Mechanics* in G.P. Galdi, R. Rannacher, A.M. Robertson, S. Turek eds.: *Hemodynamical Flows: Modeling, Analysis and Simulation*, Birkhäuser Verlag, Berlin, 2008, pp 121–274.

[3] M. Lanzendörfer, J. Hron: *On multiple solutions to the steady flow of incompressible fluids subject to do-nothing or constant traction boundary conditions on artificial boundaries*, submitted for publication.

The field of values bounds on ideal GMRES

J. Liesen[1], *P. Tichý*[2]

[1] Technische Universität Berlin, Germany
[2] Charles University in Prague

1 Introduction

Consider a linear algebraic system $Ax = b$ with a nonsingular matrix $A \in \mathbb{F}^{n\times n}$ and a right hand side $b \in \mathbb{F}^n$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$. Given an initial approximation $x_0 \in \mathbb{F}^n$ and the initial residual $r_0 \equiv b - Ax_0$, the GMRES method iteratively constructs approximations x_k such that

$$\|r_k\| = \|b - Ax_k\| = \min_{p\in\pi_k(\mathbb{F})} \|p(A)r_0\|, \quad k = 1, 2, \dots,$$

where $\|v\| \equiv \langle v, v\rangle^{1/2}$ denotes the Euclidean norm on $\mathbb{F}^n$, and $\pi_k(\mathbb{F})$ is the set of polynomials p of degree at most k with coefficients in $\mathbb{F}$, and with $p(0) = 1$.

The convergence analysis of GMRES has been a challenge since the introduction of the algorithm; see [11] or [10, Section 5.7] for surveys of this research area. Here we focus on GMRES convergence bounds that are independent of the initial residual, i.e., for a given A, we consider the worst-case behavior of the method. It is easy to see that for each given A, b and x_0, the kth relative GMRES residual norm satisfies

$$\frac{\|r_k\|}{\|r_0\|} \le \max_{\substack{v\in\mathbb{F}^n\\ \|v\|=1}} \min_{p\in\pi_k(\mathbb{F})} \|p(A)v\|.$$

The expression on the right hand side is called the kth *worst-case GMRES residual norm*. For each given matrix A and iteration step k, this quantity is attainable by the relative GMRES residual norm for some initial residual r_0. Mathematical properties of worst-case GMRES have been studied in [8]; see also [12].

2 Elman's and Starke's bounds

Let $\mathbb{F} = \mathbb{R}$ and let $M \equiv \frac{1}{2}(A + A^T)$ be the symmetric part of A. Assuming that M is positive definite, a widely known result of Elman, stated originally for the relative residual norm of the GCR method in [6, Theorem 5.4 and 5.9], implies that

$$\max_{\substack{v\in\mathbb{R}^n\\ \|v\|=1}} \min_{p\in\pi_k(\mathbb{R})} \|p(A)v\| \le \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^TA)}\right)^{k/2}; \tag{1}$$

see also the paper [5, Theorem 3.3].

Let $\mathcal{F}(A)$ be the field of values of $A \in \mathbb{F}^{n\times n}$, and let $\nu(A)$ be the distance of $\mathcal{F}(A)$ from the origin, i.e.,

$$\mathcal{F}(A) \equiv \{\langle Av, v\rangle : v \in \mathbb{C}^n,\ \|v\| = 1\}, \quad \nu(A) \equiv \min_{z\in\mathcal{F}(A)} |z|.$$

Then the bound (1) can be written as

$$\max_{\substack{v\in\mathbb{R}^n\\ \|v\|=1}} \min_{p\in\pi_k(\mathbb{R})} \|p(A)v\| \le \left(1-\frac{\nu(A)^2}{\|A\|^2}\right)^{k/2}. \tag{2}$$

It can be easily shown (see [1]), that the bound (2) holds for general nonsingular matrices $A \in \mathbb{C}^{n\times n}$, without any assumption on the Hermitian part of A.

Starke proved in [13, Section 2.2] and the subsequent paper [14, Theorem 3.2], that if $A \in \mathbb{R}^{n\times n}$ has a positive definite symmetric part M, then

$$\max_{\substack{v\in\mathbb{R}^n\\ \|v\|=1}} \min_{p\in\pi_k(\mathbb{R})} \|p(A)v\| \le \left(1-\nu(A)\nu(A^{-1})\right)^{k/2}. \tag{3}$$

For a general nonsingular matrix we have

$$\frac{\nu(A)}{\|A\|^2} \le \min_{w\in\mathbb{C}^n\backslash\{0\}} \left|\frac{\langle Aw,w\rangle}{\langle w,w\rangle}\,\frac{\langle w,w\rangle}{\langle Aw,Aw\rangle}\right| = \min_{v\in\mathbb{C}^n\backslash\{0\}} \left|\frac{\langle A^{-1}v,v\rangle}{\langle v,v\rangle}\right| = \nu(A^{-1}),$$

which yields

$$1-\nu(A)\nu(A^{-1}) \le 1-\frac{\nu(A)^2}{\|A\|^2}.$$

Hence, as pointed out by Starke in [13, 14], the bound (3) improves Elman's bound (1). In [4, Corollary 6.2], Eiermann and Ernst proved that the bound (3) holds for any nonsingular matrix $A \in \mathbb{C}^{n\times n}$. In particular, no assumption on the Hermitian part of A is required. Note, however, that the bound (3) provides some information about the convergence of (worst-case) GMRES only when $0 \notin \mathcal{F}(A)$, or, equivalently, $0 \notin \mathcal{F}(A^{-1})$.

In many situations the convergence of GMRES and even of worst-case GMRES is superlinear, and therefore linear bounds like (2) and (3) may significantly overestimate the (worst-case) GMRES residual norms. Nevertheless, such bounds can be very useful in the practical analysis of the GMRES convergence, since they depend only on simple properties of the matrix A, which may be estimated also in complicated applications. For example, Starke used his bound in [13, 14] to analyze the dependence of the convergence of hierarchical basis and multilevel preconditioned GMRES applied to finite element discretized elliptic boundary value problems on the mesh size and the size of the skew-symmetric part of the preconditioned discretized operator. Similarly, Elman's bound was used in the analysis of the GMRES convergence for finite element discretized elliptic boundary value problems that are preconditioned with additive and multiplicative Schwarz methods [2, 3]. Many further such applications exist.

3 Ideal GMRES bound

A straightforward upper bound on the kth worst-case GMRES residual norm is given by the kth *ideal GMRES approximation*, originally introduced in [9],

$$\underbrace{\max_{\substack{v\in\mathbb{F}^n\\ \|v\|=1}} \min_{p\in\pi_k(\mathbb{F})} \|p(A)v\|}_{\text{worst-case GMRES}} \le \underbrace{\min_{p\in\pi_k(\mathbb{F})} \|p(A)\|}_{\text{ideal GMRES}}. \tag{4}$$

As shown by examples in [7, 16] and more recently in [8], there exist matrices A and iteration steps k for which the inequality in (4) can be strict. The example in [16] even shows that the ratio of worst-case and ideal GMRES can be arbitrarily small. A survey of the mathematical relations between the two approximation problems in (4) is given in [15].

4 Results

The main goal of this contribution is to show that the right hand side of the bound (3) also represents an upper bound on the ideal GMRES approximation for general (nonsingular) complex matrices. In other words, the main goal is to show that

$$\min_{p\in\pi_k(\mathbb{F})} \|p(A)\| \leq \big(1-\nu(A)\nu(A^{-1})\big)^{k/2}.$$

This has been stated without proof already in our paper [11, p. 168] and later in the book [10, Section 5.7.3]. In light of the practical relevance of Elman's and Starke's bounds, and of the fact that the inequality in (4) can be strict, we believe that providing a complete proof is important.

We will further discuss some possible improvements of the known bounds based on the field of values. For example, we conjecture that

$$\min_{\|b\|=1} \cos\angle(b, Ab) \;\geq\; \frac{\nu(A)}{r(A)}$$

holds for any square matrix A, where

$$r(A) \equiv \max_{z\in\mathcal{F}(A)} |z|$$

is the *numerical radius* of A satisfying $\frac{1}{2}\|A\| \leq r(A) \leq \|A\|$. If the above mentioned conjecture is true, then Elman's bound can be improved by replacing $\|A\|$ with $r(A)$ in (2).

Finally, assuming that $\mathcal{F}(A)$ is contained in a disk D with center c and radius δ given by

$$c = \frac{\nu(A)+r(A)}{2}, \quad \text{and} \quad \delta = \frac{r(A)-\nu(A)}{2},$$

we show that

$$\min_{p\in\pi_k} \|\, p(A)\,\| \;\leq\; 2\left(\frac{\frac{r(A)}{\nu(A)}-1}{\frac{r(A)}{\nu(A)}+1}\right)^k.$$

This bound can be seen as a generalization of the bound, which is known from the steepest descent method. In particular, if A is Hermitian positive definite, then $r(A)/\nu(A) = \kappa(A)$.

Acknowledgement: This work was partially supported by project 17-04150J of the Grant Agency of the Czech Republic. This work was motivated by a question of Otto Strnad, a student of Zdeněk Strakoš at the Charles University in Prague.

References

[1] B. Beckermann, S.A. Goreinov, E.E. Tyrtyshnikov: *Some remarks on the Elman estimate for GMRES.* SIAM J. Matrix Anal. Appl., 27 (2005), pp. 772–778.

[2] X-C. Cai, O.B. Widlund: *Domain decomposition algorithms for indefinite elliptic problems.* SIAM J. Sci. Statist. Comput., 13 (1992), pp. 243–258.

[3] X-C. Cai, O.B. Widlund: *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems.* SIAM J. Numer. Anal., 30 (1993), pp. 936–952.

[4] M. Eiermann, O.G. Ernst: *Geometric aspects of the theory of Krylov subspace methods.* Acta Numer., 10 (2001), pp. 251–312.

[5] S.C. Eisenstat, H.C. Elman, M.H. Schultz: *Variational iterative methods for nonsymmetric systems of linear equations.* SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[6] H.C. Elman: *Iterative methods for large sparse nonsymmetric systems of linear equations.* PhD thesis, Yale University, New Haven, 1982.

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

[8] V. Faber, J. Liesen, P. Tichý: *Properties of worst-case GMRES.* SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1500–1519.

[9] A. Greenbaum, L.N. Trefethen: *GMRES/CR and Arnoldi/Lanczos as matrix approximation problems.* SIAM J. Sci. Comput., 15 (1994), pp. 359–368. Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992).

[10] J. Liesen, Z. Strakoš: *Krylov subspace methods.* Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2013. Principles and analysis.

[11] J. Liesen, P. Tichý: *Convergence analysis of Krylov subspace methods.* GAMM Mitt. Ges. Angew. Math. Mech., 27 (2004), pp. 153–173 (2005).

[12] J. Liesen, P. Tichý: *The worst-case GMRES for normal matrices.* BIT, 44 (2004), pp. 79–98.

[13] G. Starke: *Iterative Methods and Decomposition-Based Preconditioners for Nonsymmetric Elliptic Boundary Value Problems.* Habilitationsschrift Universität Karlsruhe, 1994.

[14] G. Starke: *Field-of-values analysis of preconditioned iterative methods for nonsymmetric elliptic problems.* Numer. Math., 78 (1997), pp. 103–117.

[15] 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), pp. 453–473.

[16] K-C. Toh: *GMRES vs. ideal GMRES.* SIAM J. Matrix Anal. Appl., 18 (1997), pp. 30–36.

A uniform parallel framework to large-scale simulations of wave-type equations

D. Lukáš

Department of Applied Mathematics, FEECS, VŠB - Technical University of Ostrava

1 Introduction

In many engineering areas such as nondestructive testing of materials and structures investigation of ultrasonic waves is of a great importance. In this paper we shall describe an efficient parallel implementation of a related finite element method. Let us consider a 3-dimensional computational domain Ω with a Lipschitz continuous boundary Γ, the time interval $(0,T)$, and the following initial boundary value problem for a wave-type equation

$$\begin{array}{rcll} \rho\frac{\partial^2 u}{\partial t^2} - \mathcal{L}^*\left(\sigma\left(\mathcal{L}(u)\right)\right) & = & f & \text{in } \Omega\times(0,T), \\ \gamma_{\mathrm{N}}(\sigma(\mathcal{L}(u))) & = & g & \text{on } \Gamma\times(0,T), \\ u & = & 0 & \text{in } \Omega\times\{0\}, \\ \frac{\partial u}{\partial t} & = & 0 & \text{in } \Omega\times\{0\}, \end{array} \tag{1}$$

where $\mathcal{L}$ denotes a linear first-order spatial differential operator, $\mathcal{L}^*$ denotes the related adjoint (with respect to $L^2(\Omega)$) operator, σ represents a linear constitutive law, and γ_{N} is the Neumann trace operator. We are mostly interested in elastodynamics, in which case $\mathcal{L} := \boldsymbol{\nabla}$, $\mathcal{L}^* := \mathbf{div}$ are vectorial operators, ρ stands for the mass density, σ is a tensor representing the linearized Hooke's law, $\gamma_{\mathrm{N}}(\sigma) := \sigma\cdot n$ is vectorial, where n is the outward unit normal vector to Ω. Nonetheless, the framework is common also for acoustics and electromagnetism. In case of acoustics, $\mathcal{L} := \nabla$, $\mathcal{L}^* := \mathrm{div}$ are scalar operators, ρ is the mass density, $\sigma(\phi) := \phi$, and $\gamma_{\mathrm{N}}(\sigma) := \sigma\cdot n$ is the scalar product. In case of electromagnetism, $\mathcal{L} = \mathcal{L}^* := \mathbf{curl}$, $\rho := 1$, $\sigma(\phi) := c^2\phi$, where c denotes the wave speed, and $\gamma_{\mathrm{N}}(\sigma) := \mathbf{curl}(\sigma)\times n$.

Now we pose a weak formulation of (1). To this end we consider the Sobolev space $V := H(\mathcal{L};\Omega) := \left\{v\in L^2(\Omega) : \mathcal{L}(u)\in L^2(\Omega)\right\}$ and its dual V' with respect to the pivot space $L^2(\Omega)$, which is vectorial in case of elastodynamics or electromagnetism. The problem is to find $u\in L^2(0,T;V)$ with $\frac{du}{dt}\in L^2(0,T;L^2(\Omega))$, $\frac{d^2u}{dt^2}\in L^2(0,T;V')$, $u(0)=0$, and $\frac{du}{dt}(0)=0$ such that

$$\left\langle \rho\frac{d^2u}{dt^2}, v\right\rangle_{V'\times V} + (\sigma(\mathcal{L}(u)),\mathcal{L}(v))_{L^2(\Omega)} = (f,v)_{L^2(\Omega)} + (g,v)_{L^2(\Gamma)} \quad \forall v\in V \quad \forall t\in(0,T). \tag{2}$$

We discretize Ω into tetrahedra and approximate V by a conforming finite element method (FEM). The degrees of freedom (DOFs) are scalar or vectorial values at vertices in case of acoustics or elastodynamics and oriented tangential moments along edges in case of electromagnetism. We employ an unconditionaly stable implicit Newmark time stepping scheme, where in each out of m time steps $t_k := k\Delta t$, where $\Delta t := \frac{T}{m}$, the following linear system is solved

$$\left(\mathbf{M} + \frac{(\Delta t)^2}{4}\mathbf{K}\right)\ddot{\mathbf{u}}_{k+1} = \mathbf{b}_k - \mathbf{K}\left(\mathbf{u}_k + \Delta t\dot{\mathbf{u}}_k + \frac{(\Delta t)^2}{4}\ddot{\mathbf{u}}_k\right), \tag{3}$$

where $\mathbf{M}$ and $\mathbf{K}$ are matrices arising from the respective bilinear forms on the left-hand side of (2), $\mathbf{b}_k$ is the discretization of the linear form on the right-hand side of (2), and $\mathbf{u}_k$, $\dot{\mathbf{u}}_k$, $\ddot{\mathbf{u}}_k$ are FEM-coordinate approximations to $u(t_k)$, $\frac{du}{dt}(t_k)$, $\frac{d^2u}{dt^2}(t_k)$. Thanks to the high-frequency nature Δt is very small, hence in the operator the mass matrix $\mathbf{M}$ dominates. We solve (3) by the conjugate gradients (CG) method with the diagonal preconditioning.

2 Parallel implementation by domain decomposition

The domain Ω, and so the collection of tetrahedra, is decomposed into N nonoverlapping subdomains, each of which is associated to exactly one concurrent process. This induces a nonoverlapping distribution of matrices, e.g., $\mathbf{M} = \sum_{i=1}^{N} \mathcal{G}^i(\mathbf{M}_i)$, where $\mathcal{G}^i$ is a local-to-global mapping of DOFs. On the other hand, vectors are distributed with overlaps in the sense that DOFs on interfaces are shared among the adjacent processes. That gives rise to communication.

The CG method is running on each process locally up to three instructions per iteration — one action of the system matrix and two dot-products. The communication proceeds as follows:

- Matrix-vector product $\mathbf{s} := \mathbf{A} \cdot \mathbf{p}$. First local contributions $\widetilde{\mathbf{s}}_i := \mathbf{A}_i \cdot \mathbf{p}_i$ are computed in parallel. The result is correct up to the interface DOFs. They are updated using a nonblocking (asynchronous) communication. Each process successively sends copies of interface restrictions of $\widetilde{\mathbf{s}}_i$ to all of its neighbours. Then the process successively reads similar messages from the neighbours and sums them up leading to the correct vector $\mathbf{s}$. Notice that assembling the diagonal preconditioner proceeds similarly.
- Dot-product $\alpha := \mathbf{s} \cdot \mathbf{p}$. Each process knows a subset (mask) $\mathcal{M}_i$ of its interface DOFs so that this distribution is globally non-overlapping. It applies the mask to the vectors, $\widehat{\mathbf{p}}_i := (\mathbf{p}_i)_{\mathcal{M}_i}$, and computes the local dot-product $\alpha_i := \widehat{\mathbf{s}}_i \cdot \widehat{\mathbf{p}}_i$. The result $\alpha = \sum_{i=1}^{N} \alpha_i$ is gained by the all-to-all communication with a single-valued message per process.

We illustrate the parallel efficiency of our approach in Tab. 1. On the discretization level we combine the domain decomposition with a multigrid approach. First of all, a nontrivial geometry of a piezo-acoustical sensor of an oil level was discretized coarsely ($\approx 10^4$ nodal DOFs). The coarse mesh was decomposed into 24, 48, ..., 192 subdomains (multiples of 24 computational cores per node of our cluster Salomon). On the coarse level all local and global DOF indices were assigned and interfaces were found. Then we applied the uniform refinement, which is free of communication since re-computing the global indices (nodes, edges, faces, tets) is done locally. We employed four refinements leading to 34 milion DOFs and one more leading to 271 milion DOFs. This amount was necessary to capture the wavelength.

Table 1: CPU and memory parallel efficiency of 100 actions of the matrix and the preconditioner.

number of DOFs	Number of cores (1 node = 24 cores)							
	24	48	72	96	120	144	168	192
34 mil.	17 s 1.59 GB	8 s 106 % 0.90 GB	5 s 113 % 0.66 GB	3 s 142 % 0.55 GB	3 s 113 % 0.48 GB	3 s 94 % 0.43 GB	2 s 121 % 0.40 GB	1 s 213 % 0.37 GB
271 mil.					29 s 2.48 GB	25 s 97 % 2.11 GB	22 s 94 % 1.84 GB	20 s 91 % 1.65 GB

Acknowledgement: This work has been supported by the Czech Science Foundation under the project 17-22615S and by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project ‘IT4Innovations National Supercomputing Center – LM2015070’.

High-order 2D boundary element method

D. Lukáš, D. Ulčák

Department of Applied Mathematics, FEECS, VŠB - Technical University of Ostrava

1 Introduction

Probably most popular algorithms for numerical solving of partial differential equations are Finite Element Method (FEM) and Boundary Element Method (BEM). The difference between them is different formulation on the one hand, and matrices arising from them on the other: FEM produces sparse matrices whereas BEM produces dense ones. There are many effective ways to solve systems of linear equations with sparse matrices, so BEM matrices are often *sparsificated* before solving the system. However, vectorisation, i.e. simultaneous processing of more items on one CPU, provides interesting alternative since it does work very well with dense matrices.
In cases when surface/volume ratio of the problem is low enough, BEM matrices have smaller dimension than FEM ones, because BEM deals only with boundary of the domain. We will have a brief look at possibility of further decreasing BEM matrices dimensions via using generally polynomial basis instead of constant/linear one. This work is mostly focused on 2-dimensional case.

2 Gaussian quadrature rules

Let $w \in L^1(a, b)$ be positive function almost everywhere in $\langle a, b, \rangle$ and let $\int_a^b w(x)f(x)\mathrm{d}x < \infty$ hold for every $f \in L^2(a, b)$. Then we call the function w weight function and the bilinear form

$$(f, g)_w = \int\limits_a^b w(x)f(x)g(x)\mathrm{d}x \tag{1}$$

defines inner product on $L^2(a, b)$. Using Gram-Schmidt orthogonalization process on set of monomials $\{1, x, x^2, \dots, x^n\}$, we can get $\{p_{0,w}(x), p_{1,w}(x), p_{2,w}(x) \dots, p_{n,w}(x)\}$ - a set of polynomials orthogonal with respect to $(.,.)_w$. It can be shown that by putting $p_{-1}(x) = 0, p_0(x) = 1$, following recurrence relation holds:

$$p_{n+1}(x) = \left(x - \frac{(xp_n, p_n)_w}{(p_n, p_n)_w}\right) p_n(x) - \frac{(p_n, p_n)_w}{(p_{n-1}, p_{n-1})_w} p_{n-1}(x). \tag{2}$$

If we denote $x_0, \dots, x_n$ roots of polynomial $p_{n+1,w}$, Gaussian quadrature rule of n^{th} degree is defined as follows:

$$\int\limits_a^b w(x)f(x)\mathrm{d}x = \sum_{i=0}^{n} w_i f(x_i), \tag{3}$$

which is exact for every polynomial up to degree $2n + 1$. For analytic non-polynomial functions, quadrature has asymptotically exponential convergence rate. Weights $w_0, \dots, w_n$ can be computed by solving linear system

$$\begin{bmatrix} p_0(x_0) & p_0(x_1) & \dots & p_0(x_k) \\ p_1(x_0) & p_1(x_1) & \dots & p_1(x_k) \\ \vdots & \vdots & \ddots & \dots \\ p_k(x_0) & p_k(x_1) & \dots & p_k(x_k) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_k \end{bmatrix} = \begin{bmatrix} \int_a^b w(x)\mathrm{d}x \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4}$$

In 2d BEM, we will have to deal with integrals containing certain polynomials and a logarithmic singularity. By suitable simplification, it is possible to use $w(x) = -\ln x,\ a = 0,\ b = 1$ for effective computation of single-layer matrix elements. Furthermore, this can be considered also in 3d BEM with semi-analytic approach.

3 High-order BEM

Let us consider following Laplace problem:

$$\Omega \in \mathcal{L}, f \in L^2(\Omega),\ g \in H^{\frac{1}{2}}(\Gamma_D) : \begin{cases} -\Delta u = 0, & u \in \Omega \\ \gamma_D u = g, & x \in \Gamma_D, \end{cases} \tag{5}$$

where γ_D is the Dirichlet trace operator. Employing Green's 3rd identity (and Green function in 2d $G(x,y) = -\frac{1}{2\pi}\ln\|x-y\|$), we can get boundary integral formulation of problem (5) as follows: We need to search for $t \in H^{-\frac{1}{2}}(\Gamma)$ such that

$$\forall v \in H^{-\frac{1}{2}}(\Gamma) :\ \langle v, Vt\rangle = \left\langle v, \left(\frac{1}{2}I + K\right) g \right\rangle \tag{6}$$

holds, considering single-layer potential V, double-layer potential K and identity I:

$$\langle u, Vw\rangle = \int_\Gamma u(x) \int_\Gamma w(y) G(x,y) \mathrm{d}\ell(y) \mathrm{d}\ell(x), \tag{7}$$

$$\langle u, w\rangle = \int_\Gamma u(x) w(x) \mathrm{d}\ell(x), \tag{8}$$

$$\langle u, Kw\rangle = \int_\Gamma u(x) \int_\Gamma w(y) \frac{\mathrm{d}G}{\mathrm{d}n_y} \mathrm{d}\ell(y) \mathrm{d}\ell(x). \tag{9}$$

In 2d, it is necessary that domain Ω lies within unit circle for the sake of uniqueness. However, this can be easily arranged by re-scaling problem.
Now, let us consider discretization of Γ by segments τ_i, their parametrization $x = x_i^1 + t(x_i^2 - x_i^1), t \in \langle 0, 1\rangle$ and basis functions above i^{th} element, i.e.

$$x \in \tau_i :\ \psi_k^{(i)}(x) = L_k(t),\ \ \varphi_0^{(i)}(x) = 1 - t,\ \ \varphi_1^{(i)}(x) = t,\ \ \varphi_k^{(i)}(x) = t(1-t)L_{k-2}(t), \tag{10}$$

where L_k is k^{th} Legendre polynomial on interval $\langle 0, 1\rangle$ and functions are considered as zero for $x \notin \tau_i$. If we use functions $\psi_k^{(i)}$ as piecewise discontinuous basis for approximation of space $H^{-\frac{1}{2}}(\Gamma)$ and also functions $\varphi_k^{(i)}$ as piecewise continuous basis for approximating space $H^{\frac{1}{2}}(\Gamma)$ in Galerkin manner, we finally get system of linear equations

$$\mathbb{V}\mathbf{t} = \left(\frac{1}{2}\mathbb{M} + \mathbb{K}\right)\mathbf{g}, \tag{11}$$

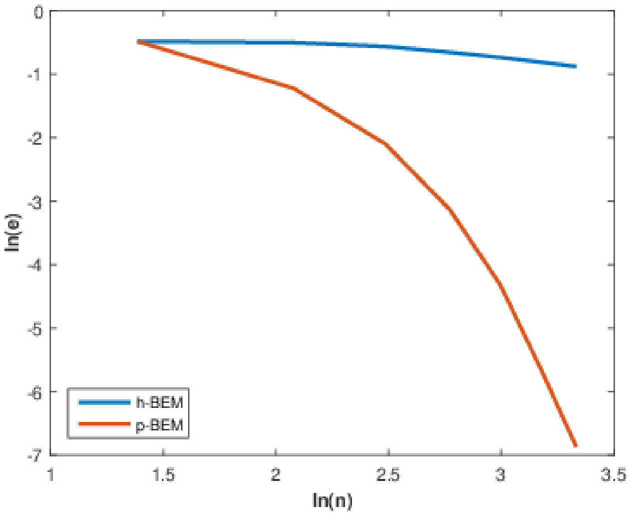


Figure 1: Example of comparison of classical and high-order BEM with respect to dimensions of matrices for Laplace equation on square within unit circle

where $\mathbf{g}$ is projection of Dirichlet boundary condition onto basis $\varphi_k^{(i)}$,

$$\mathbb{V}_{k,l}^{(i,j)} = \int\limits_{\tau_i} \psi_k(x) \int\limits_{\tau_j} \psi_l(y) G(x,y) \mathrm{d}\ell(x) \mathrm{d}\ell(y), \tag{12}$$

and analogously for matrices $\mathbb{K}, \mathbb{M}$.
With particular analytic integration and utilization of Legendre polynomials properties, logarithmic quadrature can be used to evaluate most of the single-layer matrix elements exactly, while non-(sub)diagonal blocks of matrix $\mathbb{M}$ and high-order diagonal blocks of matrix $\mathbb{K}$ are zero, the rest of elements in either matrix can be computed via Gauss-Legendre quadrature. It can be shown ([3]), that for solution regular enough, high-order BEM has exponential convergence rate, see Figure 1.

4 Conclusion

By employing higher-order base functions and Gauss-log quadrature, we achieved another way to decrease dimensions of produced matrices, since increasing order of basis is much more efficient than refining the discretization grid for regular solutions. However, for areas with lower regularity (e.g. L-shape) there should be used more discretization elements rather than polynomials of higher degree. Thus the pairing of classical and high-order BEM could be employed for such domains.

References

[1] M. Costabel: *Boundary integral operators on Lipschitz domains: elementary results.* SIAM journal on Mathematical Analysis, 19(3), 1988, pp. 613–626.

[2] P.J. Davis, P. Rabinowitz: *Methods of numerical integration.* Courier Corporation, 2007.

[3] S.A. Sauter, C. Schwab: *Boundary element methods.* In: Volume 39 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2003, pp. 259–261.

[4] B. von Sydow: *Error estimates for Gaussian quadrature formulae.* In: Numerische Mathematik 29(1), 1977, pp. 59–64.

Solution of Gao beam in contact with a deformable foundation

J. Machalová, H. Netuka

Department of Mathematical Analysis and Applications of Mathematics
Faculty of Science, Palacký University, Olomouc

1 Introduction

The nonlinear beam model (see [2])

$$EI\,w'''' - E\alpha\,(w')^2 w'' + P\mu\,w'' = f \qquad \text{in } (0,\mathrm{L}) \tag{1}$$

together with a deformable elastic foundation will be considered here and we are going to study some contact problems between these two objects. This text follows [5], where details and notations can be found.

The foundation is assumed to be 1-parameter with foundation modulus $k_F > 0$ which is modified as $c_F = (1-\nu^2)\,k_F$. The resulting equation is then as follows

$$EI\,w'''' - E\alpha\,(w')^2 w'' + P\mu\,w'' - c_F\,(\mathrm{g}-w)^+ = f \qquad \text{in } (0,\mathrm{L}). \tag{2}$$

Note that a solution of this equation can be found as a minimum point on V of the functional

$$\begin{aligned}\Pi_T(v) &= \frac{1}{2}\int_0^{\mathrm{L}} EI(v'')^2\mathrm{d}x + \frac{1}{12}\int_0^{\mathrm{L}} E\alpha(v')^4\mathrm{d}x - \frac{1}{2}\int_0^{\mathrm{L}} P\mu(v')^2\mathrm{d}x - \int_0^{\mathrm{L}} fv\,\mathrm{d}x + && (3)\\ &\quad + \frac{1}{2}\int_0^{\mathrm{L}} c_F((\mathrm{g}-v)^+)^2\mathrm{d}x, \qquad v \in V, && (4)\end{aligned}$$

which appears in the variational formulation of the contact problem mentioned in [5]. The space V is determined by the prescribed boundary conditions.

If $P < \overline{P}_{cr}$, where $\overline{P}_{cr}$ is something like critical value for the axial loading, then the problem has exactly one solution. In the following text this will always be assumed.

2 Solving boundary value problems using CVM

Here we concisely present reformulation of the fundamental boundary value problems for Gao beam by means of the Control Variational Method (CVM) to the next optimal control problem

$$\left\{\begin{array}{l}\text{Find } u^* \in U_{ad} = \{u \in L^2((0,\mathrm{L})) : |u(x)| \le C \ \text{ a.e. in } (0,\mathrm{L})\} \text{ such that}\\ J(w(u^*),u^*) = \displaystyle\min_{u\in U_{ad}} J(w(u),u),\\ \text{where } w(u) \text{ solves state problem for control value } u \in U_{ad}.\end{array}\right. \tag{5}$$

More details and equivalence proofs are published in [3] and [4].

(P1) Beam fixed at both ends (Fig.1): $w(0) = w'(0) = 0, w(\mathrm{L}) = w'(\mathrm{L}) = 0$

1st transformation: $v' \to z$

state equation: $EIw'''' = f + u$

$$\text{cost functional: } J_1(w,u) = \frac{1}{2}\int_0^{\mathrm{L}} u(w-\widehat{w})\mathrm{d}x + \frac{1}{12}\int_0^{\mathrm{L}} E\alpha(w')^4\mathrm{d}x - \frac{1}{2}\int_0^{\mathrm{L}} P\mu(w')^2\mathrm{d}x + \\ + \frac{1}{2}\int_0^{\mathrm{L}} c_F\left((\mathrm{g}-w)^+\right)^2\mathrm{d}x, \qquad \widehat{w}:\ EI\widehat{w}'''' = f$$

(P2) Propped cantilever beam (Fig.1): $w(0) = w'(0) = 0, w(\mathrm{L}) = w''(\mathrm{L}) = 0$

1st transformation: $v' \to z$

state equation: $EIw'''' = f + u$

cost functional: $J_2(w,u) = J_1(w,u)$

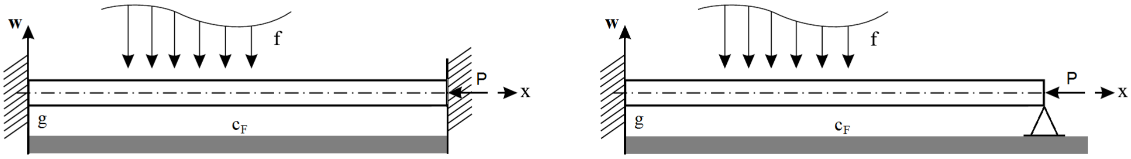


Figure 1: (P1) (P2)

(P3) Cantilever beam (Fig.2): $w(0) = w'(0) = 0,$
$w''(\mathrm{L}) = 0, EIw'''(\mathrm{L}) = \frac{1}{3}E\alpha(w'(\mathrm{L}))^3 - P\mu w'(\mathrm{L})$

2nd transformation: $v' \to z,\ v'' \to z',\ f \to g$

$$\text{state equation: } \begin{cases} -EIz'' = g + u & g:\ g' = -f \\ z(0) = z'(\mathrm{L}) = 0 & \quad g(\mathrm{L}) = 0 \\ w' = z & \\ w(0) = 0 & \end{cases}$$

$$\text{cost functional: } J_3(w,u) = \frac{1}{2}\int_0^{\mathrm{L}} u(w'-\widehat{w}')\mathrm{d}x + \frac{1}{12}\int_0^{\mathrm{L}} E\alpha(w')^4\mathrm{d}x - \frac{1}{2}\int_0^{\mathrm{L}} P\mu(w')^2\mathrm{d}x + \\ + \frac{1}{2}\int_0^{\mathrm{L}} c_F\left((\mathrm{g}-w)^+\right)^2\mathrm{d}x, \qquad \widehat{w} = \widehat{z}:\ EI\widehat{z}'' = g$$

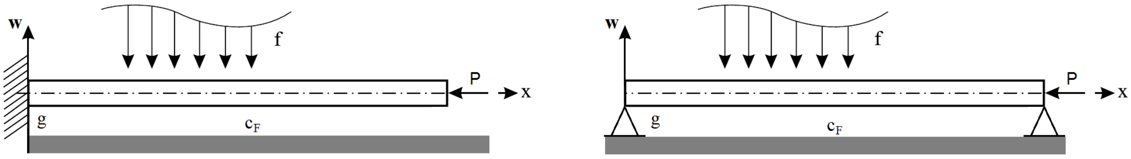


Figure 2: (P3) (P4)

(P4) Simply supported beam (Fig.2): $w(0) = w''(0) = 0, w(\mathrm{L}) = w''(\mathrm{L}) = 0$

3rd transformation: $v' \to z,\ v'' \to y,\ f \to g$

$$\text{state equation: } \begin{cases} -EIw'' = g + u & g:\ -g'' = f \\ w(0) = w(\mathrm{L}) = 0 & \quad g(0) = g(\mathrm{L}) = 0 \end{cases}$$

$$\text{cost functional: } J_4(w,u) = \frac{1}{2}\int_0^{\mathrm{L}} \frac{1}{EI}u^2\mathrm{d}x + \frac{1}{12}\int_0^{\mathrm{L}} E\alpha(w')^4\mathrm{d}x - \frac{1}{2}\int_0^{\mathrm{L}} P\mu(w')^2\mathrm{d}x + \\ + \frac{1}{2}\int_0^{\mathrm{L}} c_F\left((\mathrm{g}-w)^+\right)^2\mathrm{d}x$$

3 Numerical realization and examples

First the transformed function g is computed, if it is necessary. Then the state equation is discretized by the finite element method. Cubic Hermite elements are used for problems (P1) and (P2), while problems (P3) and (P4) need linear elements only. The discretization of state equation leads to the matrix form

$$\mathbf{K}\mathbf{w} = \mathbf{f} + \mathbf{u}\,, \tag{6}$$

whereas cost functional is then represented by a function $\mathbf{F}(\mathbf{u}, \mathbf{w})$. *Nonlinear Conjugate Gradient Method* is applied to minimization of this function in the following form (see [1]):

```
Let $\mathbf{u}^0$ be given.
Compute $\mathbf{w}^0 = S\mathbf{u}^0$ and $\mathbf{d}^0 = -\nabla\mathbf{F}(\mathbf{u}^0, \mathbf{w}^0)$.
Then for $k = 0, 1, \ldots$ (until convergence)
    evaluate $\alpha_k > 0$,
    set $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha_k \mathbf{d}^k$,
    compute $\mathbf{w}^{k+1} = S\mathbf{u}^{k+1}$,
    determine gradient $\mathbf{g}^{k+1} = \nabla\mathbf{F}(\mathbf{u}^{k+1}, \mathbf{w}^{k+1})$,
    compute $\beta_k$ such that
        $\beta_k = \frac{(\mathbf{h}^k)^\top \mathbf{g}^{k+1}}{(\mathbf{d}^k)^\top \mathbf{y}^k}$, where $\mathbf{h}^k = \mathbf{y}^k - 2\mathbf{d}^k \frac{(\mathbf{y}^k)^\top \mathbf{y}^k}{(\mathbf{y}^k)^\top \mathbf{d}^k}$ and $\mathbf{y}^k = \mathbf{g}^{k+1} - \mathbf{g}^k$,
    set $\mathbf{d}^{k+1} = -\mathbf{g}^{k+1} + \beta_k \mathbf{d}^k$.
```

By symbol S we denoted a solution operator of the state problem (6), which is *linear*. Computation of gradient $\nabla\mathbf{F}(\mathbf{u}^{k+1}, \mathbf{w}^{k+1})$ is based on the *adjoint method*, see [6]. Step-size calculations use Wolfe conditions and projection operator on the set U_{ad}.

Next, four examples are presented. Results for the nonlinear Gao beam are compared with results for the classical Euler–Bernoulli beam model. The Gao beam is tougher than the classical one, accordingly in the following figures the upper curves represent the Gao beam and the lower curves the Euler–Bernoulli beam results.

The input data for beams are $E = 21 \cdot 10^4\,\mathrm{MPa}$, $\nu = 0.3$, $\mathrm{h} = 0.1\,\mathrm{m}$, $I = 0.666667 \cdot 10^{-3}\,\mathrm{m}^4$, $\mathrm{L} = 1\,\mathrm{m}$. In the following figures on their left sides there are results for bending and on the right sides for contact problems. The gap between beam and foundation is always $\mathrm{g} = 0.001\,\mathrm{m}$, the foundation modulus is $k_F = 5 \cdot 10^{10}\,\mathrm{N\,m^{-2}}$. Beams fixed at both ends (Fig. 3) have prescribed constant vertical load $q = -10^8\,\mathrm{N\,m^{-1}}$ and axial load $P = -10^8\,\mathrm{N}$. Results for propped cantilever beams with loading $q = -5 \cdot 10^7\,\mathrm{N\,m^{-1}}$ and $P = 10^8\,\mathrm{N}$ are shown in Fig. 4. In Fig. 5 cantilever beam results are presented using $q = -2 \cdot 10^7\,\mathrm{N\,m^{-1}}$ and $P = 10^8\,\mathrm{N}$. The last figure displays simply supported beam results for data $q = -5 \cdot 10^7\,\mathrm{N\,m^{-1}}$ and $P = -10^8\,\mathrm{N}$.

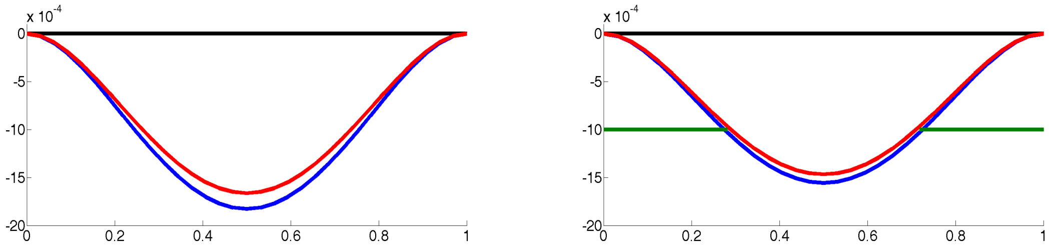


Figure 3: Beams fixed at both ends

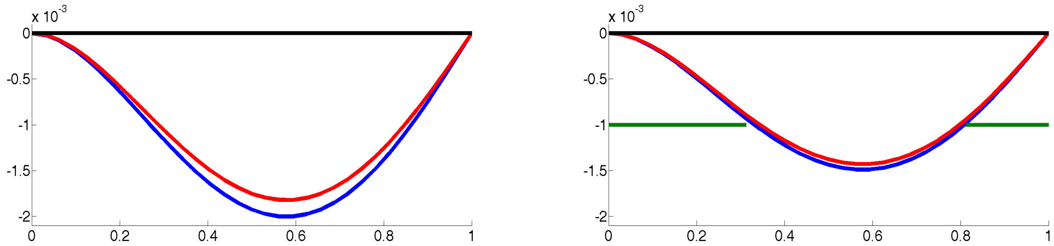


Figure 4: Propped cantilever beams

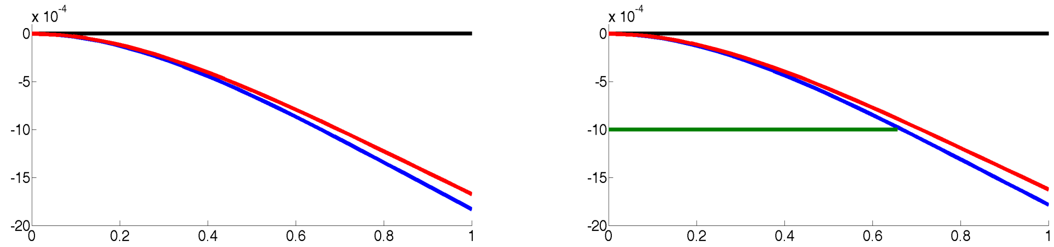


Figure 5: Cantilever beams

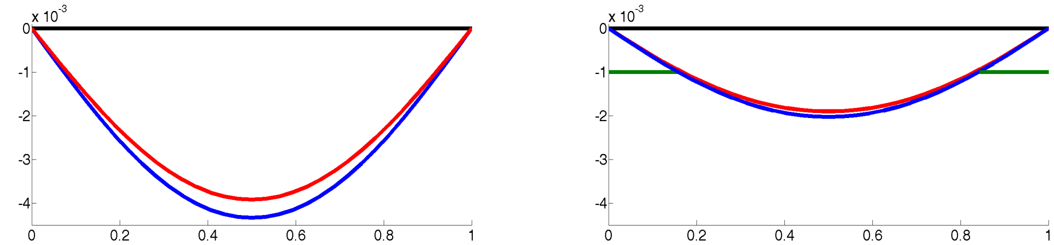


Figure 6: Simply supported beams

References

[1] A. Borzi, V. Schulz: *Computational Optimization of Systems Governed by Partial Differential Equations.* SIAM, Philadelphia, 2012.

[2] D.Y. Gao: *Nonlinear elastic beam theory with application in contact problems and variational approaches.* Mechanics Research Communications, 23 (1), 1996, pp. 11–17.

[3] J. Machalová, H. Netuka: *Control variational method approach to bending and contact problems for Gao beam.* Applications of Mathematics, Vol. 62, No. 6, 2017, pp. 661–677.

[4] J. Machalová, H. Netuka: *Solution of contact problems for Gao beam and elastic foundation.* Mathematics and Mechanics of Solids, Special Issue on Inequality Problems In Contact Mechanics, Vol. 23, Issue 3, 2018, pp. 473–488.

[5] H. Netuka, J. Machalová: *Gao beam: From definition to contact problems.* SNA'19, Ostrava, January 21 - January 25, 2019.

[6] F. Tröltzsch: *Optimal Control of Partial Differential Equations. Theory, Methods and Applications.* AMS, Providence, Rhode Island, 2010.

On the optimal initial conditions for an inverse problem of model parameter estimation – a complementarity principle

C. Matonoha[1], *Š. Papáček*[2]

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

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

1 Introduction

This contribution represents an extension of our earlier studies on the paradigmatic example of the inverse problem of the diffusion parameter estimation from spatio-temporal measurements of fluorescent particle concentration, see [6, 1, 3, 4, 5]. More precisely, we continue to look for an optimal bleaching pattern used in FRAP (Fluorescence Recovery After Photobleaching), being the initial condition of the Fickian diffusion equation maximizing a sensitivity measure. As follows, we define an optimization problem and we show the special feature (so-called complementarity principle) of the optimal binary-valued initial conditions.

2 Problem formulation

We consider the Fickian diffusion problem with a *constant* diffusion coefficient $\delta > 0$ and assume a spatially radially symmetric observation domain, i.e., the data are observed on a cylinder with the fixed radius R and fixed height T. In [5] it is shown how to perform the scaling of the space and time coordinates. Thus, without loss of generality we can assume $R = 1$ and $T = 1$.

In FRAP, the usual governing equation for the radially symmetric spatio-temporal distribution of fluorescent particle concentration $u(r,t)$ is the diffusion equation

$$\frac{\partial u}{\partial t} = \delta \left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} \right), \tag{1}$$

where $r \in (0,1],\ t \in [0,1]$, with the initial and Neumann boundary conditions

$$u(r,0) = u_0(r), \quad \frac{\partial u}{\partial r}(1,t) = 0. \tag{2}$$

The main issue in FRAP and related estimation problems is to find the value of the diffusion coefficient δ from spatio-temporal measurements of the concentration $u(r,t)$, cf. [1, 6].

Each data entry quantifies the variable u at discrete spatio-temporal points (r,t), which are distributed with the constant step-size Δr and the time interval Δt (between two consecutive measurements), i.e.,

$$\begin{array}{llllll} u(r_i, t_j), & i = 0 \dots n, & r_0 = 0, & r_n = 1, & \Delta r = 1/n \\ & j = 0 \dots m, & t_0 = 0, & t_n = 1, & \Delta t = 1/m, \end{array}$$

where i is the spatial index uniquely identifying the pixel position where the value of fluorescence intensity u is measured and j is the time index. The initial condition $u_0(r)$ can be considered as an $(n+1)$-dimensional vector $u_0 \in \mathbb{R}^{n+1}$.

Given the data as above, the diffusion coefficient δ can be computed numerically by solving the inverse problem to (1)-(2). If all the parameters $R, T, \Delta r, \Delta_t$ are fixed, the estimation of the true diffusion parameter δ_T can be improved by maximizing the so called sensitivity measure

$$S_{GRS}(u_0) = \delta^2 \sum_{i=0}^{n} \sum_{j=1}^{m} \left[\frac{\partial}{\partial \delta} u(r_i, t_j) \right]^2 , \tag{3}$$

i.e., to consider the *initial condition (bleach)* u_0 in (2) as the experimental design parameter. By optimizing the bleach design, we mean to select the initial conditions in such a way that S_{GRS} is maximized and hence the expected error in δ_T is minimized [5]. As we will show later, the optimal initial condition is binary valued, it has only zero and non-zero components. The non-zero components represent the optimal bleached area.

We used the Crank-Nicolson (CN) scheme (described in the next section) to solve the initial boundary value problem (1)-(2). Then, the sensitivity measure S_{GRS} can be approximated by

$$S_{app}(u_0) = \sum_{j=1}^{m} j^2 \sum_{i=0}^{n} [u(r_i, t_j) - u(r_i, t_{j-1})]^2 = \sum_{j=1}^{m} j^2 \|u_j - u_{j-1}\|^2, \tag{4}$$

see [5], where the vector $u_j \in \mathcal{R}^{n+1}$ is defined in the next section. The optimization problem is formulated as follows

$$u_0^{opt} = \arg \max_{u_0 \in \mathcal{R}^{n+1}} S_{app}(u_0) \quad \text{subject to} \quad 0 \le u_{0i} \le 1, \quad i = 0, \ldots, n. \tag{5}$$

The upper bounds $u_{0i} \le 1$ serve to determine where the initial condition is considered. Note that an arbitrary positive value can be used.

3 Numerical issues of the initial boundary value problem

When computing a numerical solution $u_{i,j} := u(r_i, t_j)$, $i = 0 \ldots n$, $j = 1 \ldots m$, of the IBV problem (1)-(2), the finite difference CN scheme is used. Starting with an initial $u_0 \in \mathcal{R}^{n+1}$ and after some algebraic manipulation we arrive at a linear system with a three-diagonal symmetric positive definite matrix

$$A u_j = B u_{j-1} \tag{6}$$

for $u_j = (u_{0,j}, \ldots, u_{n-1,j})^T \in \mathcal{R}^n$, $j = 1 \ldots m$. The Neumann boundary condition implies the last component $u_{n,j} = u_{n-1,j}$. Here,

$$A = \left(\frac{1}{n\delta} Z + hT \right), \qquad B = \left(\frac{1}{n\delta} Z - hT \right),$$

$$T = \begin{bmatrix} \frac{1}{4} & -s_0 & & & & \\ -s_0 & 1 & -s_1 & & & \\ & -s_1 & 2 & -s_2 & & \\ & & \cdots & \cdots & \cdots & \\ & & & -s_{n-3} & n-2 & -s_{n-2} \\ & & & & -s_{n-2} & n-1-s_{n-1} \end{bmatrix},$$

$$Z = \operatorname{diag}\left(\frac{1}{4}, 1, 2, \ldots, n-2, n-1\right), \qquad h = \frac{n}{m}, \qquad s_k = \frac{2k+1}{4},\ k = 0, \ldots, n-1.$$

The matrix T is positive semidefinite and singular. As

$$s_k + s_{k+1} = k + 1, \quad k = 0, \ldots, n-2,$$

the sum of the off-diagonal elements in T is equal to the diagonal element and thus

$$Te = 0, \quad e = (1, \ldots, 1)^T.$$

The matrix Z is positive definite, so the same is A. Denote

$$C = A^{-1}B.$$

Then for the spectral radius of C we have

$$\varrho(C) \leq 1.$$

Using the above notation we can adjust the function S_{app} as follows. It holds

$$Au_j = Bu_{j-1} \quad \Rightarrow \quad u_j = Cu_{j-1} = C^j u_0, \quad j = 1, \ldots, m.$$

From this we can conclude that

$$u_j - u_{j-1} = C^{j-1}(C - I)u_0$$

and

$$S_{app}(u_0) = \sum_{j=1}^{m} j^2 \|C^{j-1}(C - I)u_0\|^2. \tag{7}$$

The function S_{app} is quadratic and nonnegative. The maximum is achieved at a vertex of the constrained set $0 \leq u_{0i} \leq 1,\ i = 0, \ldots, n$, which is $(n+1)$-dimensional hypercube. Thus, u_0^{opt} is a $\{1, 0\}$-function, see also [3]. The jumps between these values in fact represent the discontinuities in bleached domain leading to more complex optimal bleaching patterns.

Moreover, since

$$Te = 0 \quad \Rightarrow \quad (C - I)e = 0, \tag{8}$$

we have

$$S_{app}(\alpha e) = 0, \quad \alpha \in [0, 1]. \tag{9}$$

Note that (9) holds for an arbitrary α but we consider only a unit hypercube, see (5).

The most important property of the function S_{app} is

$$S_{app}(u_0) = S_{app}(e - u_0). \tag{10}$$

This implies that if u_0^{opt} is an optimal vertex solution to problem (5), then also $e - u_0^{opt}$ is a solution with the same function value. In practice it means that if e.g. the disc is an optimal bleaching pattern, then also its complement, the annulus touching the bleached domain, is an optimal bleaching pattern.

To show (10), denote $v_0 = e - u_0$. Using the same CN scheme with v_0 and using (8), we obtain

$$v_j - v_{j-1} = C^{j-1}(C - I)v_0 = C^{j-1}(C - I)(e - u_0) = -C^{j-1}(C - I)u_0$$

and from (7) $S_{app}(u_0) = S_{app}(v_0)$.

4 Conclusion

In this contribution, the problem of the optimal initial condition for the further identification of a constant diffusion coefficient is formulated. We set up the numerical procedure leading simultaneously to the optimal size and shape of a bleached domain for which the sensitivity measure reaches the maximal value, hence assuring the smallest relative error of the estimated parameter. The optimal initial shapes or bleaching patterns are functions of δ. Optimal shapes are not only disks of various radii (the usual bleach shape used in the FRAP community). For high values of the dimensionless diffusion coefficient, the disc is the optimal shape and for smaller values, shapes with more and more components (i.e., annuli-type shapes) become optimal.

Having prescribed δ and other parameters reflecting the experimental protocol, there exist two corresponding optimal initial conditions (and hence optimal bleach sizes and shapes). These initial conditions satisfy a complementarity principle and thus whichever of them can be used in practice as the optimal bleaching pattern.

Acknowledgement: This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807) and by the MEYS of the Czech Republic - projects “CENAKVA" (No. CZ.1.05/2.1.00/01.0024), “CENAKVA II" (No. LO1205 under the NPU I program) and ‘The CENAKVA Centre Development’ (No. CZ.1.05/2.1.00/19.0380).

References

[1] R. Kaňa, E. Kotabová, M. Lukeš, Š. Papáček, C. Matonoha, L.N. Liu, O. Prášil, C.W. Mullineaux: *Phycobilisome mobility and its role in the regulation of light harvesting in red algae.* Plant Physiology **165(4)** (2014), pp. 1618–1631.

[2] S. Kindermann, Š. Papáček: *On data space selection and data processing for parameter identification in a reaction-diffusion model based on FRAP experiments.* Abstract and Applied Analysis, Article ID 859849 (2015).

[3] S. Kindermann, Š. Papáček: *Optimization of the shape (and topology) of the initial conditions for diffusion parameter identification.*
https://arxiv.org/pdf/1602.03357.pdf (2016)

[4] C. Matonoha, Š. Papáček: *On the optimal initial conditions for an inverse problem of model parameter estimation.* Proceedings of Seminar ”Seminar on Numerical Analysis SNA’17", Ostrava (2017), pp. 72–75.

[5] C. Matonoha, Š. Papáček, S. Kindermann: *Disc vs. Annulus: On the Bleaching Pattern Optimization for FRAP Experiments.* In: Kozubek T. et al. (eds) High Performance Computing in Science and Engineering 2017. Lecture Notes in Computer Science, Springer, Cham **11087** (2018), pp. 160–173.

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

Gao beam: From definition to contact problems

H. Netuka, J. Machalová

Department of Mathematical Analysis and Applications of Mathematics
Faculty of Science, Palacký University, Olomouc

1 Gao beam model

Let us consider an elastic beam subjected to a distributed vertical load $q(x)$ and an external axial force P (which is compressive if $P > 0$) at the right end as shown in Fig.1.

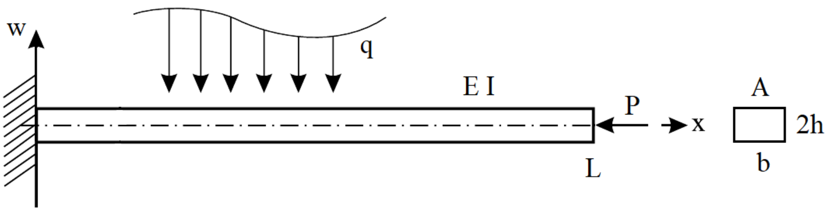


Figure 1: Cantilever beam

It is well known that the original Euler–Bernoulli theory is valid only for infinitesimal strains. This theory can be extended in a straightforward manner to problems involving moderately large displacements provided that the strain remains small by using the von Kármán strains. Based on the Euler–Bernoulli hypothesis (i.e. straight lines normal to the mid-surface remain straight and normal to the mid-surface after deformation) governing equations for a nonlinear isotropic beam model can be expressed as follows, see e.g. [6],

$$(EIw'')'' - (EA[u' + \frac{1}{2}(w')^2]w')' \;=\; q, \tag{1}$$

$$-(EA[u' + \frac{1}{2}(w')^2])' \;=\; 0, \tag{2}$$

where w and u are transverse and horizontal displacements, respectively, E is Young's modulus, A cross-section area, I area moment of inertia, $q(x)$ distributed transverse load (per unit length). The same result one can obtain from the von Kármán nonlinear plate model in one-dimension with

$$\sigma_x \;=\; EA[u' + \frac{1}{2}(w')^2] \tag{3}$$

as the axial stress. From (1) and (2) we get

$$(EIw'')'' - (EA[u' + \frac{1}{2}(w')^2])w'' = q \qquad \text{in } (0, \mathrm{L}), \tag{4}$$

but, as a consequence of (2), this is evidently a linear equation. According to D.Y. Gao, the main reason for this paradox is due to the fact that the stress in the lateral direction was ignored.

Using the following additional assumptions:

- the beam has a uniform cross-section of the rectangular shape, i.e. $I = \frac{2}{3}\mathrm{h}^3\mathrm{b}$, $A = 2\mathrm{hb}$,
- the beam is under moderately large elastic deformations,

- some small terms in the Green–Saint Venant strain tensor are omitted,
- the stresses and the deformations in the y-direction are not neglected,
- constitutive relations from the plane stress problem are used to define the stresses,

a nonlinear beam model was proposed by Gao in the form of the system (see [2])

$$\begin{aligned} EI\,w'''' - (\sigma w')' &= f, && (5)\\ \sigma &= \frac{1}{3}E\alpha\,(w')^2 - (1+\nu)(1-\nu^2)\,P, && (6)\\ u' &= -\frac{1}{2}(w')^2(1+\nu) - \frac{1-\nu^2}{2\mathrm{hb}E}\,P, && (7)\end{aligned}$$

where $f = (1-\nu^2)q$, $\alpha = 3\mathrm{hb}(1-\nu^2)$, ν denotes the Poisson's ratio and σ is the canonical dual stress - in contrast to the previous model (1)-(3) non-constant. Substituting (6) into (5) and using abbreviation $\mu = (1+\nu)(1-\nu^2)$ we finally obtain the equation for deflections

$$EI\,w'''' - E\alpha\,(w')^2 w'' + P\mu\,w'' = f \qquad \text{in } (0,\mathrm{L}) \tag{8}$$

and the corresponding functional of potential energy

$$\Pi_G(v) = \frac{1}{2}\int_0^{\mathrm{L}} EI\,(v'')^2\mathrm{d}x + \frac{1}{12}\int_0^{\mathrm{L}} E\alpha\,(v')^4\mathrm{d}x - \frac{1}{2}\int_0^{\mathrm{L}} P\mu\,(v')^2\mathrm{d}x - \int_0^{\mathrm{L}} fv\,\mathrm{d}x, \tag{9}$$

defined on some kinematically admissible space V. Usual boundary conditions can be applied to this model with only one exception - free end conditions for cantilever beam now take the form $w''(\mathrm{L}) = EI\,w'''(\mathrm{L}) - \frac{1}{3}E\alpha\,(w'(\mathrm{L}))^3 + P\mu w'(\mathrm{L}) = 0.$

The most important properties of Π_G are (see e.g. [3])

a) Π_G is coercive for any P,

b) Π_G is strictly convex if $P < P_{cr}^G$,

where critical value P_{cr}^G could be determined by the convexity condition

$$\Pi_G''(w,v,v) = \int_0^{\mathrm{L}} EI\,(v'')^2\mathrm{d}x - P\int_0^{\mathrm{L}} \mu\,(v')^2\,\mathrm{d}x + \int_0^{\mathrm{L}} E\alpha\,(w')^2(v')^2\,\mathrm{d}x \ge 0 \quad \forall w,v\in V. \tag{10}$$

Let us note that now we do not solve an eigenvalue problem as it is usual for classical Euler–Bernoulli beam. For simplicity, instead of exact value P_{cr}^G the value

$$\overline{P}_{cr} = \min_{v\in V}\frac{\int_0^{\mathrm{L}} EI\,(v'')^2\,\mathrm{d}x}{\int_0^{\mathrm{L}} \mu\,(v')^2\,\mathrm{d}x} = \frac{1}{\mu}P_{cr}^E \le P_{cr}^G, \tag{11}$$

is used in the text below, where P_{cr}^E is the well-known Euler limit load. As a result we get that for any P s.t. $P < \overline{P}_{cr}$ the variational problem

$$\left\{\begin{array}{l}\text{Find } w\in V \text{ such that}\\ \Pi_G(w) = \min\limits_{v\in V}\Pi_G(v),\end{array}\right. \tag{12}$$

which describes beam bending, has exactly one solution.

2 Contact problems

Now let us consider two fundamental contact problems for the Gao beam and a foundation, see Fig. 2. The gap $\mathrm{g} \le 0$ between them is generally a function but, for simplicity, we will consider it here as a given constant.

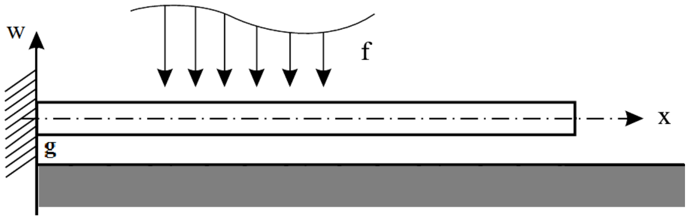


Figure 2: Beam and a foundation

The first problem deals with *undeformable* (or rigid) foundation. Using the non-penetration (or Signorini) condition $w \geq \mathrm{g}$ in $(0, \mathrm{L})$, we arrive at the variational problem

$$\begin{cases} \text{Find } w \in K \text{ such that} \\ \Pi_G(w) \;=\; \min\limits_{v \in K} \Pi_G(v), \end{cases} \tag{13}$$

where $K = \{v \in V : v \geq \mathrm{g} \text{ in } (0, \mathrm{L})\}$ is closed convex subset of space V. Thus, as in the original problem (12), it can be said that for $P < \overline{P}_{cr}$ problem (13) has just one solution. Of course, this problem could be rewritten as a variational inequality.

The second case concerns a *deformable* foundation with $k_F > 0$ as the foundation modulus. It can be modelled using the so-called *normal compliance condition*, see e.g. [1], [4]. The total potential energy Π_T of the whole system is now given by

$$\Pi_T(v) \;=\; \Pi_G(v) + \frac{1}{2}\int_0^{\mathrm{L}} c_F((\mathrm{g} - v)^+)^2 \mathrm{d}x, \qquad v \in V, \tag{14}$$

where $c_F = (1 - \nu^2)k_F$, $v^+(x) = \max\{0, v(x)\}$. This functional has the same properties which have been mentioned above in connection with the functional Π_G. Hence the problem

$$\begin{cases} \text{Find } w \in V \text{ such that} \\ \Pi_T(w) \;=\; \min\limits_{v \in V} \Pi_T(v) \end{cases} \tag{15}$$

has one solution if $P < \overline{P}_{cr}$. This problem, contrary to (13), is not an inequality but has the form of a nonlinear variational equation.

3 Solution using Control Variational Method

The Control Variational Method (CVM) was proposed by D. Tiba and M. Sofonea as a method for analysing and solution of boundary value problems for differential systems. The idea behind CVM consists in transforming the original problem into the new one which is an optimal control problem. Contact problems for cantilever Euler–Bernoulli beam were studied by means of CVM in [1]. Our research generalizes this approach to nonlinear Gao beam which is in addition subjected to axial load and with all kinds of possible boundary conditions.

CVM is realized in the three steps:
1) transformation of the loading function f (if it is considered useful),
2) definition of the state equation,
3) transformation of the potential energy functional into a cost functional J.
These steps result into problem transformation.

The key principle here is to make the state equation linear (optimization then will be easier), the key tool for these purposes are following transformations:

$$\begin{aligned}
&\text{1st transformation: } v' \to z,\\
&\text{2nd transformation: } v' \to z,\ v'' \to z',\ f \to g\text{: } \int_0^{\mathrm{L}} fv\,\mathrm{d}x = \int_0^{\mathrm{L}} gv'\mathrm{d}x \quad \forall v \in V,\\
&\text{3rd transformation: } v' \to z,\ v'' \to y,\ f \to g\text{: } \int_0^{\mathrm{L}} fv\,\mathrm{d}x = -\int_0^{\mathrm{L}} gv''\mathrm{d}x \quad \forall v \in V.
\end{aligned}$$

Applying the three steps using one of the above transformation we arrive at the final optimal control problem. These problems are two-stage problems with the basic stage known as the *state problem.* It is a boundary value problem and the scheme for it has the following form

$$\begin{cases} \text{For given } u \in U_{ad} \text{ find } w := w(u) \text{ such that} \\ w \text{ solves state equation in } (0, \mathrm{L}) \\ \text{together with prescribed boundary conditions,} \end{cases} \tag{16}$$

where the set od admissible controls is given by

$$U_{ad} = \{u \in L^2((0, \mathrm{L})) : |u(x)| \leq C \ \text{ a.e. in } (0, \mathrm{L})\} \tag{17}$$

for some positive constant C (because we do not want to break the beam). The second stage represents an optimization problem

$$\begin{cases} \text{Find } u^* \in U_{ad} \text{ such that} \\ J(w(u^*), u^*) = \min\limits_{u \in U_{ad}} J(w(u), u), \\ \text{where } w(u) \text{ solves (16)} \\ \text{for control value } u \in U_{ad}. \end{cases} \tag{18}$$

Examples of using CVM for solving contact problems including numerical solution are given in [5]. More details related to the transformation process together with existence theorems and proofs of problems equivalence can be found in [3] and [4].

References

[1] M. Barboteu, M. Sofonea, D. Tiba: *The control variational method for beams in contact with deformable obstacles.* ZAMM 92 (1), 2012, pp. 25–40.

[2] D.Y. Gao: *Nonlinear elastic beam theory with application in contact problems and variational approaches.* Mechanics Research Communications, 23 (1), 1996, pp. 11–17.

[3] J. Machalová, H. Netuka: *Control variational method approach to bending and contact problems for Gao beam.* Applications of Mathematics, Vol. 62, No. 6, 2017, pp. 661–677.

[4] J. Machalová, H. Netuka: *Solution of contact problems for Gao beam and elastic foundation.* Mathematics and Mechanics of Solids, Special Issue on Inequality Problems In Contact Mechanics, Vol. 23, Issue 3, 2018, pp. 473–488.

[5] J. Machalová, H. Netuka: *Solution of Gao beam in contact with a deformable foundation.* SNA'19, Ostrava, January 21 - January 25, 2019.

[6] J.N. Reddy: *An Introduction to Nonlinear Finite Element Analysis.* Oxford University Press, Oxford, 2004.

Computing forces on atoms in electronic structure calculations

M. Novák[1], *J. Vackář*[2], *R. Cimrman*[3]

[1] Faculty of Applied Sciences, University of West Bohemia in Pilsen
[2] Institute of Physics of the CAS, Prague
[3] New Technologies Research Centre, University of West Bohemia in Pilsen

1 Introduction

In electronic structure calculations, the total energy of a system of atoms is an important quantity, whose derivatives with respect to movement of atomic centres, also known as the Hellmann-Feynman forces (HFF), present a suitable tool for the material science to determine various material properties "ab-initio". Those forces act on atoms out of the equilibrium positions. By consequence, efficient evaluation of the HFF has many applications such as in seeking stable atomic positions or in molecular dynamics calculations.

According to the Hellmann-Feynman theorem [4], supposing that a fixed discretization basis is used, the forces can be calculated from the gradient of the Hamiltonian (energy operator) H

$$\vec{f_a} = -\nabla e_{\text{TOT}} = -\nabla\left(\psi^+ H\psi\right) = -\psi^+\nabla\left(H\right)\psi\,, \tag{1}$$

where the gradient is considered with respect to the shift of atomic centers, $^+$ denotes Hermitian transpose and ψ is the wave function describing a quantum state. As can be seen in (1), the Hellmann-Feynman theorem states that the wave functions can be "frozen" and the gradient is applied to the Hamiltonian H only.

2 Total energy derivatives

The total energy (including the interaction energy of atomic cores) in the density functional theory is given by (see e.g. [5, 6])

$$e_{\text{TOT}} = \sum_{i=1}^{n} w_i \int \psi_i^+ \frac{1}{2}\nabla^2\psi_i + \int \psi_i^+ V_{\text{EXT}}\psi + \int E_{\text{H}}(\rho) + \int E_{\text{XC}}(\rho) + e_{\text{ION}}\,, \tag{2}$$

where w_i are occupation numbers of ψ_i states, V_{EXT} is the external potential, E_{H} is the electrostatic energy, E_{XC} is the exchange-correlation energy, ρ is the charge density and e_{ION} is the atomic core repulsion energy. In our case V_{EXT} is the sum of pseudopotentials of atomic cores, each of them constituted by a long-range local part and a short-range nonlocal l-dependent part:

$$V_{\text{EXT}} = \sum_a \left(V^a_{\text{LOC}} + \sum_l V^{a,l}_{\text{NL}} P^a_l \right)\,, \tag{3}$$

where P^a_l is a projection operator into a l-subspace, spanned by the spherical harmonics basis $Y_{l,m}$, of the a-th center:

$$P_l = Y_{l,m}Y^+_{l,m}\,, \quad P_l\psi = \sum_m Y_{l,m}\int_{\theta,\varphi} Y_{l,m}\psi \;\mathrm{d}\varphi\;\mathrm{d}\theta\;.$$

Following from (1), the gradient of the total energy contains only the terms with the explicit dependence on atom positions (no implicit dependence through wavefunctions):

$$\nabla_a e_{\text{TOT}} = \int \nabla V^a_{\text{LOC}}\rho + \sum_{l,i} w_i \int \psi_i^+ \nabla \left(V^{a,l}_{\text{NL}} P^a_l\right) \psi_i + \nabla e_{\text{ION}}\,. \tag{4}$$

The most difficult term of equation (4) is the middle one: the nonlocal part of electron-ion interaction (^{a} omitted for brevity):

$$\vec{f}_{\text{NL}} \equiv \psi^+ \nabla_a \left(V_l P_l\right) \psi = \psi^+ \left(\nabla_a V_l\right) P_l \psi + \psi^+ V_l \left(\nabla_a P_l\right) \psi\,, \tag{5}$$

where $\psi^+ \left(\nabla_a V_l\right) P_l \psi$ is the force originating from the shift of the potential that can be evaluated by means of spherical projections relatively easily. On the other hand, $\psi^+ V_l \left(\nabla_a P_l\right) \psi$, which is the change of the charge density in the given l-subspace that occurs due to the shift of the centers of the $l-$projections, presents a significant difficulty in practical evaluation, because by differentiating the projector P_l we obtain the gradient of the spherical harmonic functions

$$\begin{aligned}\psi^+ V_l \left(\nabla_a P_l\right) \psi = \psi^+ V_l \nabla_a \sum_m \left(Y_{l,m} Y^+_{l,m}\right) \psi = \\ \psi^+ V_l \sum_m \left(\left(\nabla_a Y_{l,m}\right) Y^+_{l,m} + Y_{l,m} \left(\nabla_a Y^+_{l,m}\right)\right) \psi\,,\end{aligned}$$

with a singularity (for $l > 0$) at the origin.

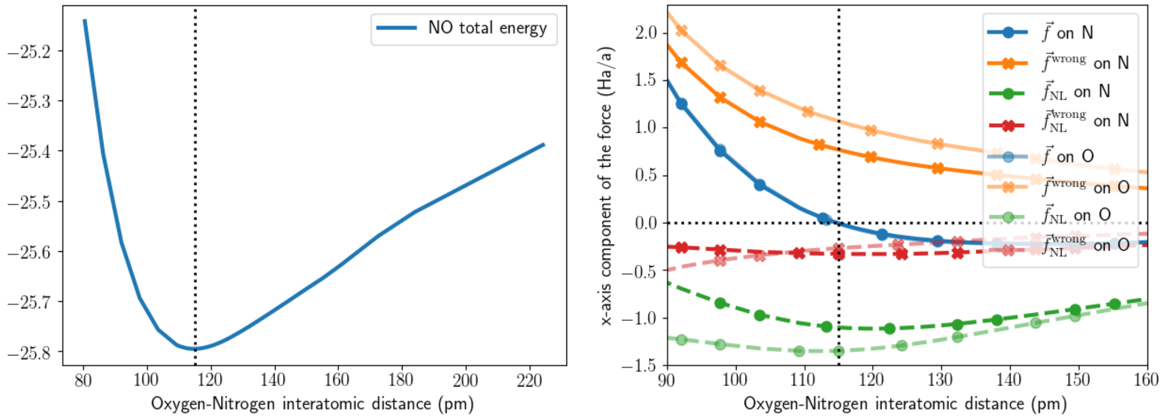
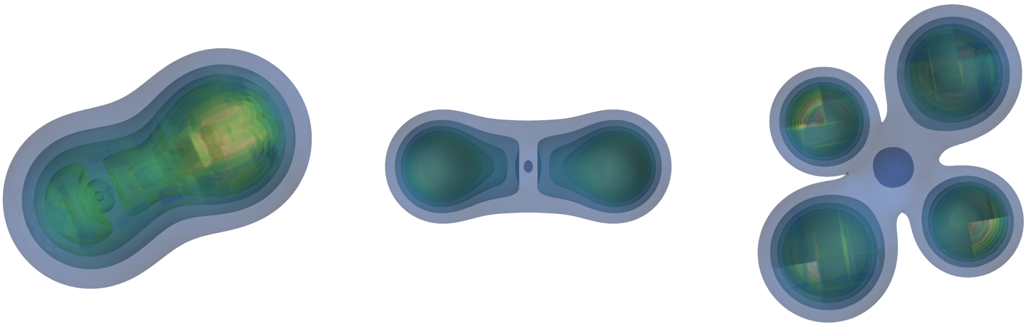


Figure 1: The total energy (left) and the HFF (right) computed for the varying interatomic distance in the NO molecule. The curves marked by $^{\text{wrong}}$ correspond to results without the spherical harmonics gradients.

In literature, this term is treated in various ways: several authors simply neglect it, as in the original paper [5], while in later works, e.g. in Quantum Monte Carlo methods [1], the authors explicitly claim that the term can be neglected. In Fig. 1 we demonstrate, that in our case this term cannot be omitted. In Fig. 1 (left) the dependence of the total energy on the interatomic distance between N and O in the NO molecule is shown, with the equilibrium position marked by the vertical dotted line. The HFF dependence on the interatomic distance as calculated with ($\vec{f}$) or without ($\vec{f}^{\text{wrong}}$) the spherical harmonics gradients are shown in Fig. 1 (right). Besides the total forces, also the nonlocal components $\vec{f}_{\text{NL}}$ are shown. It can be readily seen that only the curves corresponding to $\vec{f}$ are zero in the position of the minimal total energy (marked by the dotted vertical line), and the curves of the forces acting on N and on O coincide exactly.



a NO molecule a CO_2 molecule a CF_4 molecule

Figure 2: The self-consistent charge densities ρ of the test molecules.

For $\vec{f}^{\text{wrong}}$, the forces are not zero in the equilibrium position and moreover the action-reaction principle does not hold (two curves, for N and O, are visible).

In the poster, several approaches to evaluating the term $\vec{f}_{\text{NL}}$, within the density functional theory in combination with nonlocal ab-initio pseudopotentials and the finite-element method as implemented in our new real space code for electronic structure calculations [7, 3, 2], will be analyzed in terms of efficiency and accuracy using test calculations on simple molecules of nitric oxide, carbon dioxide and tetrafluormethane, see Fig. 2.

Acknowledgement: The first author was supported by the Czech Science Foundation, project no. GA17-12925S. The second author acknowledges the support by CEDAMNF project, reg. no. CZ. 02.1.01 co-funded by the ERDF as part of the Ministry of Education, Youth and Sports OP RDE programme. The third author was supported by the Czech Science Foundation, project no. GA17-14840S.

References

[1] A Badinski, R.J. Needs: *Accurate forces in quantum monte carlo calculations with nonlocal pseudopotentials*, Physical Review E **76**, 2007, no. 3, 036707.

[2] R. Cimrman, M. Novák, R. Kolman, M. Tůma, J. Plešek, J. Vackář: *Convergence study of isogeometric analysis based on bézier extraction in electronic structure calculations*, Applied Mathematics and Computation, 2017.

[3] R. Cimrman, M. Novák, R. Kolman, M. Tůma, J. Vackář: *Isogeometric analysis in electronic structure calculations*, Mathematics and Computers in Simulation, 2016.

[4] H. Hellmann: *A new approximation method in the problem of many electrons*, The Journal of Chemical Physics **3**, 1935, no. 1, pp. 61–61.

[5] J. Ihm, A. Zunger, M.L. Cohen: *Momentum-space formalism for the total energy of solids*, Journal of Physics C: Solid State Physics **12**, 1979, no. 21, 4409.

[6] J.E. Pask, P.A. Sterne: *Finite element methods in ab initio electronic structure calculations*, Modelling and Simulation in Materials Science and Engineering **13**, 2005, no. 3, R71.

[7] J. Vackář, O. Čertík, R. Cimrman, M. Novák, O. Šipr, J. Plešek: *Finite element method in density functional theory electronic structure calculations*, Advances in the Theory of Quantum Systems in Chemistry and Physics, Springer, 2012, pp. 199–217.

Analysing no-bias Support Vector Machines formulations without and with regularized Hessians

M. Pecha[1,2], *D. Horák*[1,2]

[1] Department of Applied Mathematics, FEECS, VŠB - Technical University of Ostrava
[2] Institute of Geonics of the CAS, Ostrava

1 Support Vector Machines

The Support Vector Machine (SVM) [1] is a supervised binary classifier, i.e., in training phase of classifier, a classification model is determined from already categorized training samples belonging to Class A (label $+1$) or Class B (label -1). The essential idea of the SVM classifier training is to find the hyperplane that maximizes the margin between the Class A and the Class B samples, i.e. maximal margin hyperplane. The samples contributing to the definition of such hyperplane are called support vectors.

Let us denote the training samples as a set of ordered pairs such that

$$T := \{(\boldsymbol{x}_1, y_1)\,,\ (\boldsymbol{x}_2, y_2)\,,\ \dots, (\boldsymbol{x}_{\boldsymbol{m}}, y_m)\},$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$, n represents number of features, is i-th samples and $y_i \in \{-1, +1\}$ is the label of i-th sample. Let H be the maximal margin hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, where $\boldsymbol{w}$ is its normal vector. For the case of non-linearly separable classes, we introduce hinge-loss function $\xi_i = \max\left[0, 1 - y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)\right]$, which quantifies error between predicted and correct classification of sample $\boldsymbol{x}_i$. If sample $\boldsymbol{x}_i$ is correctly classified, a value of the hinge loss function equals 0. For the case of a sample misclassification, a value of hinge loss function is the distance between hyperplane H and misclassified sample. The problem of finding the hyperplane H can be formulated as a constrained optimization quadratic programming problem in the following primal formulation

$$\operatorname*{arg\,min}_{\boldsymbol{w},\ b,\ \xi_i}\ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\xi_i \quad \text{s.t.} \quad \begin{cases} y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i - b\right) \geq 1 - \xi_i, \\ \xi_i \geq 0, \end{cases} \tag{1}$$

where C is user defined penalty hat penalizes misclassification error. Higher value of C increases the importance of minimising the hinge loss functions ξ_i and the importance of minimising $\|\boldsymbol{w}\|$.

The primal formulation SVM (1) can be modified by exploiting the Lagrange duality. Evaluating the Karush-Kuhn-Tucker conditions, the problem results into the dual formulation with box and equality constraints

$$\operatorname*{arg\,min}_{\boldsymbol{\alpha}}\ \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{Y}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{e} \text{ s.t. } \begin{cases} \boldsymbol{o} \leq \boldsymbol{\alpha} \leq C\boldsymbol{e}, \\ \boldsymbol{B}_e\boldsymbol{\alpha} = 0, \end{cases} \tag{2}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \dots, \boldsymbol{x}_m]$, $\boldsymbol{y} = [y_1,\ y_2,\ \dots, y_m]^T$, $\boldsymbol{Y} := diag(\boldsymbol{y}), \boldsymbol{B}_e := \left[\boldsymbol{y}^T\right]$. The Hessian of (2) is defined as follows $\boldsymbol{H} := \boldsymbol{Y}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{Y}$, which is symmetric positive semi-definite (SPSD) matrix.

Further, we introduce dual to primal reconstruction formulas for the normal vector $\boldsymbol{w} = \boldsymbol{X}\boldsymbol{Y}\boldsymbol{\alpha}$, and the bias $b = \frac{1}{|I^{SV}|}\sum_{i \in I^{SV}}\left(\boldsymbol{x}_i^T\boldsymbol{w} - y_i\right)$, where I^{SV} denotes the support vector index set, i.e. $I^{SV} := \{i \mid 0 < \alpha_i < C,\ i = 1, 2,\ \dots, m\}$.

2 Hessian regularization

In previous dual formulation (2), the Hessian matrix is SPSD. The essential idea of regularization the Hessian in dual is to use square sum of the loss functions ξ_i instead of linear sum in primal so that

$$\underset{\boldsymbol{w},\ b,\ \xi_i}{\arg\min} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \text{ s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, i \in \{1, 2, \ldots, n\}. \tag{3}$$

Then, we derive dual formulation by the Lagrange duality and, evaluating the Karush-Kuhn-Tucker conditions, the primal formulation (3) results into the dual formulation

$$\underset{\boldsymbol{\alpha}}{\arg\min}\ \frac{1}{2}\boldsymbol{\alpha}^T\left(\boldsymbol{H} + C^{-1}\boldsymbol{I}\right)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{e} \text{ s.t. } \begin{cases} \boldsymbol{0} \leq \boldsymbol{\alpha}, \\ \boldsymbol{B_e\alpha} = \boldsymbol{0}. \end{cases} \tag{4}$$

Since the Hessian is regularized by means matrix $C^{-1}\boldsymbol{I}$, it becomes symmetric positive definite (SPD). Mathematically, the associated optimization problem would be more computationally stable than in a case of (2).

3 No-bias data classifications

In high dimensional space, we do not need the bias term in the primal formulation [2], therefore the equality constraints vanished in the dual formulations so that

$$\underset{\boldsymbol{\alpha}}{\arg\min}\ \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{e} \text{ s.t. } \boldsymbol{0} \leq \boldsymbol{\alpha} \leq \boldsymbol{C}, \tag{5}$$

$$\underset{\boldsymbol{\alpha}}{\arg\min}\ \frac{1}{2}\boldsymbol{\alpha}^T\left(\boldsymbol{H} + C^{-1}\boldsymbol{I}\right)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{e} \text{ s.t. } \boldsymbol{0} \leq \boldsymbol{\alpha}, \tag{6}$$

which are called the no-bias formulation. Mathematically, the previous soft-margin problems reduce into solving rotation of the separating hyperplane that best fits the classification problems. However, the no-bias formulations may be a cause of poor performance score of model in some applications. Standard approach for improving the performance score is to append each sample with an additional feature in the following way $\mathbf{x}_i \leftarrow [\mathbf{x}_i, c]$, where $c \in \mathbb{R}^+$. In many classification problems, the value of c is typically set to 1 [2].

4 Numerical experiments

In numerical experiments, we will analyze convergence rates and performance scores of models related to the no-bias formulations (5) and (6) achieved by means PermonSVM classifier on 3 public available datasets, namely Diabetes, Heart, Ionosphere, downloaded from LIBSVM dataset webpages [3]. We will evaluate performances of models on test datasets by means of accuracy and F_1 score. For the MPRGP (Modified Proportioning with Reduced Gradient Projection) [4] algorithm, all initial guess components are set to $0.99 * C$, the relative norm of projected gradient being smaller than $1e-1$ is used as stopping criterion. The expansion step-size is fixed and determined such as $\alpha = 2.0/\|\boldsymbol{H}\|_2$, where $\|\boldsymbol{H}\|_2 = \sqrt{\lambda_{max}\left(\boldsymbol{H}^T\boldsymbol{H}\right)}$. The values of penalty C are chosen from the set $\{1.0,\ 5.0,\ 10.0\}$

Looking at Table 2 and Table 3, the performance scores of models are slightly higher in cases of $l1$-loss for the Diabetes, Heart, and Ionosphere datasets – summarized in Table 4. The theory

Dataset	**#samples**	**#samples+**	**#samples-**	**#features**
Diabetes (training)	514	332	182	8
Diabetes (test)	254	168	86	
Heart (training)	180	84	96	13
Heart (test)	90	36	54	
Ionosphere (training)	235	154	81	34
Ionosphere (test)	116	71	45	

Table 1: Diabetes, Heart, Ionosphere: the training and test descriptions of datasets.

Dataset	C	**Hessian mult.**	**CG steps**	**Exp. steps**	**Accuracy** [%]	F_1
Diabetes	1.0	365	1	181	73.62	0.80
Diabetes	5.0	563	38	261	73.23	0.80
Diabetes	10.0	680	107	285	73.23	0.80
Heart	1.0	130	0	64	80.00	0.74
Heart	5.0	273	69	99	83.33	0.78
Heart	10.0	327	98	112	82.22	0.76
Ionosphere	1.0	262	10	124	81.90	0.87
Ionosphere	5.0	321	47	135	82.76	0.87
Ionosphere	10.0	392	91	148	81.90	0.87

Table 2: No-bias SVM formulation (5): Comparison of the number of iterations, CG steps, expansion steps, and Hessian multiplications obtained after solver converged and evaluating performance scores of models.

Dataset	C	**Hessian mult.**	**CG steps**	**Exp. steps**	**Accuracy** [%]	F_1
Diabetes	1.0	68	3	32	72.83	0.79
Diabetes	5.0	79	8	35	72.44	0.79
Diabetes	10.0	83	10	36	72.83	0.80
Heart	1.0	32	1	15	77.78	0.71
Heart	5.0	36	5	15	77.78	0.71
Heart	10.0	38	7	15	80.00	0.74
Ionosphere	1.0	133	26	53	77.59	0.84
Ionosphere	5.0	204	39	82	75.86	0.83
Ionosphere	10.0	220	49	85	76.72	0.83

Table 3: No-bias SVM formulation (5) (regularized Hessian): Comparison of the number of iterations, CG steps, expansion steps, and Hessian multiplication obtained after solver converged and evaluating performance scores of models.

described by Dostál et al. [5] guarantees the convergence of MPRGP for both SPSD and SPD Hessians. We observe the convergence rate is slower for all tested datasets in case of SPSD Hessian, it corresponds to our remark about computation stability for the problem with regularized Hessian. The maximum value of Hessian multiplication speed up is 8.61, minimum, mean, and median are 1.38, $4,25$, 3.01, respectively. The number of expansion steps is approximately 3 to 5 times higher than CG steps for both no-bias formulations. Standard implementation of the expansion step is more expensive than CG step, because it needs one more Hessian multiplication, therefore reduction of expansion steps is required. Therefore, we work on non-fixed expansion step-size, i.e. an adaptive step-size.

It seems, (5) problem formulation is more robust (catches up outliers during training phase) than

	minimum	**maximum**	**mean**	**median**
accuracy [%]	0.4	6.9	3.15	2.22
F_1	0.00	0.07	0.03	0.03

Table 4: Diabetes, Heart, Ionosphere: Comparing differences of performance scores of model trained by means formulations (5) and (6).

(6), however it produces SPSD Hessian in dual that causes slower convergence rate. Therefore, we start to work on dual formulations arising from primal formulation with general the loss function $\frac{C}{p}\sum_{i=1}^{m}\xi_i^p$. For the exponent lies somewhere between 1 and 2, we assume the regularization of Hessian to be SPD and model should keep robustness of model trained by means formulation (5).

5 Conclusions

In this article, we analyzed no-bias SVM formulation without and with regularized Hessian. We benchmarked our implementation in PermonSVM tool on 3 public available datasets, namely Diabetes, Heart, and Ionosphere. We evaluate performance score of models by means accuracy and F_1 scores. For all tests, the MPRGP algorithm was used as a solver for the QP problem arising from the no-bias dual formulations. We observe, training SVM classifier by means QP formulation with SPSD Hessian produces more robust model however convergence rate is obviously slower than in case of training classifier by means QP formulation with SPD Hessian. Therefor, we derive dual formulations arising from primal formulation with general loss functions. Further, we work on the adaptive expansion step-size to reduce the number of the expansion steps.

Acknowledgement: This paper has been elaborated in the framework of the grant programme ”Support for Science and Research in the Moravia-Silesia Region 2017” (RRC/10/2017), financed from the budget of the Moravian-Silesian Region. Also, this work has been supported by Grant of SGS No. SP2018/165, VŠB - Technical University of Ostrava, Czech Republic.

References

[1] C. Cortes, V. Vapnik: *Support-Vector Networks.* In: Machine Learning, 1995, pp. 273–297.

[2] Ch.-J. Hsieh, K.-W. Chang, Ch.-J. Lin, S.S. Keerthi, S. Sundararajan: *A dual coordinate descent method for large-scale linear SVM.* In: Proceedings of the 25th international conference on Machine learning, 2008, pp. 408–415.

[3] *LIBSVM Data: Classification (Binary Class)*, Available on-line at
`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`

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

[5] Z. Dostál, L. Pospíšil: *Minimizing quadratic functions with semidefinite Hessian subject to bound constraints*, In: Computers & Mathematics with Applications, 2015, pp. 2014–2028.

Domain decomposition methods with Parareal

I. Peterek, L. Foltyn, D. Lukáš

Department of Applied Mathematics, VŠB - Technical University of Ostrava

1 Introduction

Develop an efficient and also robust method for solving space-time problems is an actual but quite difficult task through last years. In a paper the Parareal scheme in combination with some domain decomposition method (DDM) is described, i.e. the Parareal with Schwarz method for solving 1D parabolic problem and the Parareal with non-overlapping DDM which is based on [2, 1] for solving 2D parabolic problem. The Parareal algorithm uses semi-discretization process in which we used a standard finite element method for a spatial domain.

2 Parareal

The Parareal algorithm was introduced in [3]. A core of the Parareal algorithm forms a method of lines where an implicit Euler technique is used to approximate the time derivative.

At the beginning a solution with a coarse time step $dT > 0$ is computed. Then the coarse solution is used as initial conditions for n independent time subintervals $[(i-1) \cdot dT; i \cdot dT]$, $i = 1, \dots, n$, which are solved with a fine time step $dt > 0$, $dt \ll dT$. At the end of a cycle the coarse solution is corrected by obtained fine solution.

A main interest of our work was to make solving of linear systems for fine step more efficient. To do that we chose mentioned DDM techniques.

3 Parareal coupled with Schwarz DDM in 1D

For more efficient computation in case of time-space problems we split our domain in both time and space into smaller subdomains. This can be done by using Parareal combined with some DDM. Basic idea of coupling both methods is to use also overlapping Schwarz method [4] when solving independent problems with fine time step on each subinterval in Parareal.

This approach creates new independent time-space problems with boundary and initial conditions. Problems are solved iteratively on each of n time subintervals. After each Schwarz iteration we update boundary conditions between overlapping subdomains and then solve problems again with new boundary conditions. Fine solution on each time subinterval does not have to fully convergate in one iteration of Parareal therefore we can only use fixed number of iterations for Schwarz method. Solution on coarse mesh is done by Parareal in the same way as if no DDM was used.

3.1 Numerical results

in following experiments we consider 1D heat equation in this form

$$\begin{aligned} c_H \frac{\partial u}{\partial t}(\mathbf{x},t) - \Delta_{\mathbf{x}} u(\mathbf{x},t) &= 1 \quad \forall(\mathbf{x},t) \in (0;1) \times (0;T]\ , \\ u(0,t) &= 0 \quad \forall t \in (0;T]\ , \\ u(1,t) &= 0 \quad \forall t \in (0;T]\ , \\ u(\mathbf{x},0) &= 0 \quad \forall\, \mathbf{x} \in (0;1) \end{aligned} \tag{1}$$

where $c_H > 0$ is heat capacity, we choose $c_H = 25$, and T is the end of time interval $T = 1$.

The aim of experiments is to show how use of Schwarz method affects convergence of Parareal. We investigate error of pure Parareal solution compared to Parareal coupled with Schwarz. The error was computed as euclidean norm on some coarse mesh. For all test we use spatial step $h = \frac{1}{100}$, fine time step $dt = dT/10$. Coarse time step will be variable(number of time subintervals describes also coarse time step $dT = \frac{1}{20} = 20$ subintervals). In Figure 1. comparison of error in given iteration for both approaches is shown. We can see that coupling with Schwarz method does not affect convergence, in fact our coupled method seems to be a little bit better for higer number of subintervals as long as enough Schwarz iterations is used. It is also important to notice that both Parareal and coupled Parareal and Schwarz will be faster to use only if computations are done in parallel.

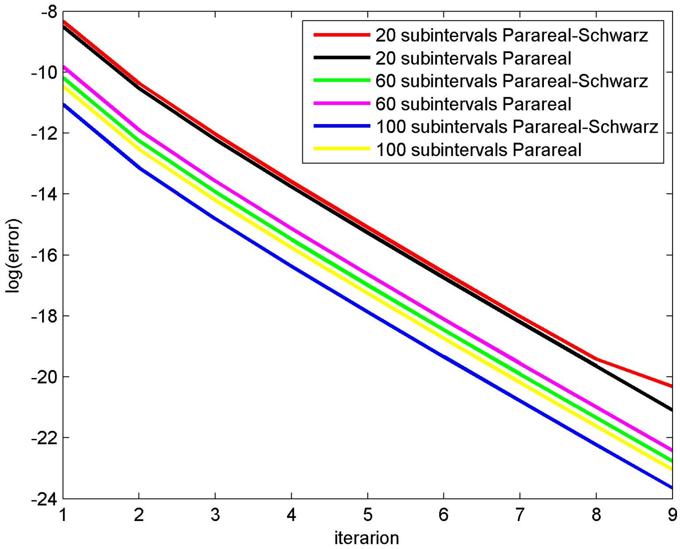


Figure 1: Comparison of pure Parareal solution error in given iteration to Parareal-Schwarz with 6 iterations of Schwarz method for 4 spatial subdomains with 8 elements overlap in each time subinterval.

4 Parareal coupled with DDM in 2D

For 2D spatial domain the non-overlapping DDM based on [2, 1] was used. In short, the spatial domain without Dirichlet boundaries is divided into inner subdomains and a skeleton. The skeleton is further divided to edges and vertices which join individual edges together. Hence, a resulting matrix A is decomposed to a form

$$\mathrm{A} = \left[\begin{array}{cc} \mathrm{I}_{II} & \mathrm{O}_{IG} \\ \mathrm{A}_{GI}\mathrm{A}_{II}^{-1} & \mathrm{I}_{GG} \end{array}\right] \left[\begin{array}{cc} \mathrm{A}_{II} & \mathrm{O}_{IG} \\ \mathrm{O}_{GI} & \mathrm{S} \end{array}\right] \left[\begin{array}{cc} \mathrm{I}_{II} & \mathrm{A}_{II}^{-1}\mathrm{A}_{IG} \\ \mathrm{O}_{GI} & \mathrm{I}_{GG} \end{array}\right] \tag{2}$$

where index I represents a set of nodes of inner subdomains, index G set of nodes of the skeleton, I (O) is identity matrix (zero matrix) and S is the Schur complement.

The Schur complement is further decomposed into a similar form

$$\mathrm{S} = \left[\begin{array}{cc} \mathrm{I}_{EE} & \mathrm{O}_{EV} \\ -\mathrm{R}_{VE} & \mathrm{I}_{VV} \end{array}\right] \left[\begin{array}{cc} \mathrm{S}_{EE} & \tilde{\mathrm{S}}_{EV} \\ \tilde{\mathrm{S}}_{VE} & \tilde{\mathrm{S}}_{VV} \end{array}\right] \left[\begin{array}{cc} \mathrm{I}_{EE} & -\mathrm{R}_{VE}^{\mathrm{T}} \\ \mathrm{O}_{VE} & \mathrm{I}_{VV} \end{array}\right] \tag{3}$$

where index E represents a set of nodes forming edges of the skeleton, index V is a set of vertexes, R_{VE} is an interpolation matrix of standard basis functions to basis functions over the skeleton (coarse mesh).

In the next step, we replace the Schur complement S by its approximation $\hat{\mathrm{S}}$

$$\hat{\mathrm{S}} = \left[\begin{array}{cc} \mathrm{I}_{EE} & \mathrm{O}_{EV} \\ -\mathrm{R}_{VE} & \mathrm{I}_{VV} \end{array}\right] \left[\begin{array}{cc} \overline{\mathrm{S}}_{EE} & \mathrm{O}_{EV} \\ \mathrm{O}_{VE} & \mathrm{A}_{H} \end{array}\right] \left[\begin{array}{cc} \mathrm{I}_{EE} & -\mathrm{R}_{VE}^{\mathrm{T}} \\ \mathrm{O}_{VE} & \mathrm{I}_{VV} \end{array}\right] \tag{4}$$

where $\overline{\mathrm{S}}_{EE}$ is a block diagonal matrix where each block correspond to one edge of the skeleton and A_H is a matrix corresponding to the skeleton (coarse mesh).

4.1 Numerical results

For some numerical experiments we considered a following problem

$$\begin{aligned} c_H \frac{\partial u}{\partial t}(\mathbf{x},t) - \Delta_{\mathbf{x}} u(\mathbf{x},t) &= 0 && \forall(\mathbf{x},t) \in (0;1)^2 \times (0;T]\ , \\ u(\mathbf{x},t) &= 0 && \forall \mathbf{x} \in \Gamma_D = \partial\Omega\ \forall t \in (0;T]\ , \\ u(\mathbf{x},0) &= \sin(\pi x)\sin(\pi y) && \forall \mathbf{x} \in (0;1)^2 \end{aligned} \tag{5}$$

where $c_H > 0$ is heat capacity, in our example $c_H = 1$, and T is the end time, in our example $T = 2$.

During numerical experiments we were focusing on an error between approximate solution of the Parareal algorithm to approximate solution of the implicit Euler method with fine time step dt. It is obvious that a resulting error of the Parareal algorithm to an exact solution will be at the end of the algorithm same as an error of the sequential implicit Euler method to the exact solution so we don't mentioned it in our results. Our goal is to emphasize "how much faster" we obtain the approximate solution of the Parareal technique than it will be done by sequential attempt of fully implicit Euler method. The error was computed as euclidean norm on some coarse mesh which was common for all tests. In a Table 1. below a number of Parareal iteration k, in which we reached a given precision, and number of iterations of conjugate gradients nCG required to obtain approximate solution in each step of fine implicit Euler method is shown. Left side of the Table 1. (before double line) is dedicated to the Parareal method without DDM preconditioning and right side to the Parareal method with DDM preconditioning. The given precision was set up as $\varepsilon = 1 \cdot 10^{-16}$. By h we note the step in spatial domain, dT represents the coarse step and by nSD we note the number of subdomains for DDM preconditioning. The fine time step dt was fixed $dt = \frac{dT}{32}$ for all cases.

To better understanding of the table below we note that if we choose the coarse step $dT = \frac{1}{4}$ it means that we divide the time interval $[0; 2]$ onto 8 independent subdomains so after $k = 8$ iterations is the Parareal algorithm “nearly” equal to sequential implicit Euler method with fine step dt. We said “nearly” because there are some additional calculations which are required to update new initial conditions in the Parareal method.

Table 1: Results without DDM and with DDM.

without DDM				with DDM		
dT	h	k	nCG	nSD	k	nCG
$\frac{1}{4}$	$\frac{1}{8}$	4	13	16	4	7
$\frac{1}{4}$	$\frac{1}{16}$	4	25	64	4	24
$\frac{1}{4}$	$\frac{1}{32}$	5	44	64	5	31
$\frac{1}{4}$	$\frac{1}{32}$	5	44	256	5	27
$\frac{1}{8}$	$\frac{1}{8}$	7	12	16	7	8
$\frac{1}{8}$	$\frac{1}{16}$	8	22	64	8	25
$\frac{1}{8}$	$\frac{1}{32}$	8	40	64	8	30
$\frac{1}{8}$	$\frac{1}{32}$	8	40	256	8	27

5 Conclusion

We have shown possible decomposition of space-time domain into smaller subdomains suitable for parallel computations with coupled Parareal and Schwarz method. Described DDM techniques improve the efficiency of the Parareal algorithm in a way of decreased number of iterations which are necessary to obtain solution of given linear systems. We guess that we could also improve the convergence rate of the Parareal scheme by using another time stepping scheme instead of the implicit Euler scheme. We leave this topic as a future work.

References

[1] D. Lukáš, J. Bouchala, P. Vodstrčil, L. Malý: *2-Dimensional primal domain decomposition theory in detail.* Applications of Mathematics, Jun. 2015, vol. 60, no. 3, pp. 265–283.

[2] J.H. Bramble, J.E. Pasciak, A.H. Schatz: *The Construction of Preconditioners for Elliptic Problems by Substructuring. I.* Mathematics of Computation, Jul. 1986, vol. 47, no. 175, p. 103.

[3] J.L. Lions, Y. Maday, G. Turinici: *Résolution d’EDP par un schéma en temps «pararéel».* Comptes Rendus de l’Académie des Sciences – Series I – Mathematics, Apr. 2001, vol. 332, no. 7, pp. 661–668.

[4] A. Toselli, O. Widlund: *Domain decomposition methods-algorithms and theory.* 2006 Springer Science & Business Media

Stability of network centrality indices

S. Pozza[1], *F. Tudisco*[2]

[1] Charles University in Prague
[2] University of Strathclyde, Glasgow, Scotland

1 Introduction

One of the major goals of network analysis is to identify important components in a network by exploiting the topological structure of connections between its nodes. To this end, recent years have seen the introduction of many new measures of importance of a node or a set of nodes, defined in terms of suitable entries of functions of matrices $f(A)$, for different choices of f and A. However, this approach requires a significant computational effort to address the entries of $f(A)$. This is particularly prohibitive when the network changes frequently and the important components have to be updated.

Let $G = (V, E)$ be a directed network where $V = \{1, \ldots, N\}$ is the finite set of nodes, $E \subseteq V \times V$ is the set of edges. To any network $G = (V, E)$ corresponds an entry-wise nonnegative adjacency matrix A defined by

$$A_{k\ell} = \begin{cases} 1 & \text{if } k, \ell \text{ are starting and ending points of } e \in E, \text{ respectively} \\ 0 & \text{otherwise} \end{cases}.$$

In this work we address the problem of estimating the changes in the entries of $f(A)$ with respect to changes in the edge set E. Intuition suggests that, if the topology of connections in the new network $\widetilde{G} = (V, \widetilde{E})$ is not significantly distorted, relevant components in G maintain their leading role in $\widetilde{G}$. We propose a bound showing that the magnitude of the variation of the entry $f(A)_{k,\ell}$ decays exponentially with the distance in G that separates either k or ℓ from the set of nodes touched by the edges that are perturbed.

The details about this work can be found in [10].

2 Subgraph centrality and communicability indeces

Given two nodes $k, \ell \in V$, a walk in G from k to ℓ is an ordered sequence of edges $\{e_1, \ldots, e_r\} \subseteq E$ such that k is the starting point of e_1, ℓ is the endpoint of e_r and, for any $i = 1, \ldots, r-1$, the endpoint of e_i is the starting point of e_{i+1}. The length of a walk is the number of edges forming the sequence (repetitions are allowed). The length of the shortest walk from k to ℓ is called the (geodesic or shortest-path) distance in G from k to ℓ and it is denoted by $d_G(k, \ell)$. The diameter of G is the longest shortest-path distance between any two nodes. Given a set $S \subseteq V$ and a node $k \in V$, we set

$$d_G(k, S) = \min_{s \in S} d_G(k, s) \quad \text{and} \quad d_G(S, k) = \min_{s \in S} d_G(s, k)\,,$$

with the convention that $d_G(k, k) = 0$ and thus $d_G(k, S) = d_G(S, k) = 0$, for any $k \in S$. Notice that for the sake of simplicity we do not consider networks with weighted edges; however, it is possible to extend all the results we are presenting to such case (see [10]).

In order to identify the most important nodes in a network, one needs a quantitative definition of the importance of a node k or a pair of nodes (k, ℓ). Although these quantities have a long history, dating back to the early 1950s, recent years have seen the introduction of many new centrality scores based on the entries of certain function of matrices [8, 7, 3]. The idea behind such metrics is to measure the relevance of a node, for example, by quantifying the number of subgraphs of G that involve that node.

The powers of the adjacency matrix A can be used to count the number of walks of different lengths in G. More precisely, $(A^n)_{k\ell}$ is the number of n-length walks from k to ℓ. This property can be used to define the centrality (and communicability) indeces as follows. Let $f : \mathbb{C} \to \mathbb{C}$ be such that $f(z) = \sum_{n\geq 0} \theta_n\, z^n$, for any $|z| \leq r$, with $\theta_n > 0$. Assuming r larger than the spectral radius of A, we can define the *matrix function*

$$f(A) = \sum_{n=0}^{\infty} \theta_n\, A^n;$$

see, e.g., [9]. The *f-centrality* of the node $k \in V$ is defined as the quantity $f(A)_{kk}$. Similarly, the *f-communicability* from node k to node ℓ is the quantity $f(A)_{k\ell}$. This idea was firstly introduced by Estrada and Rodriguez-Vasquez in [8], for the particular choice $f(z) = \exp(z)$, and then developed and extended in many subsequent works; see, e.g., [7, 3] and the references therein.

3 Index stability and upper bounds

Assume that the network $G = (V, E)$, with adjacency matrix A, is modified into the network $\widetilde{G} = (V, \widetilde{E})$, where $\widetilde{E} \subseteq E \cup \delta E$ is obtained adding or erasing the edges in δE. Then the adjacency matrix of $\widetilde{G}$ is $\widetilde{A} = A + \delta A$, with δA a sparse perturbation. We are interested in a-priori estimations of the absolute variation of the entries of $f(\widetilde{A})$ with respect to those of $f(A)$. To this end, we have derived bounds for $|f(A)_{k\ell} - f(\widetilde{A})_{k\ell}|$ employing the theory of Faber polynomials. This family of polynomials have been used for the analysis of the decay of the elements of functions of banded non-Hermitian matrices [4, 2]. Given a convex continuum Ω, Faber polynomials are defined by means of a conformal map ϕ (with inverse ψ) which maps the exterior of Ω onto the exterior of the unitary disk $\{z \in \mathbb{C} : |z| \leq 1\}$, and satisfies the conditions $\phi(\infty) = \infty$, and $\lim_{z\to\infty} \phi(z)/z = d > 0$. Finally, to state our main result we introduce the *field of values* (or *numerical range*) of a matrix A, which is the convex and compact subset of $\mathbb{C}$ defined as $\mathcal{F}(A) = \{v^*Av : v \in \mathbb{C}^N,\ \|v\|_2 = 1\}$.

Theorem 1. *Let Ω be a convex continuum containing $\mathcal{F}(A)$ and $\mathcal{F}(\widetilde{A})$, and let ϕ and ψ be conformal maps for Ω as defined above. Moreover, denote with $S = \{s|(s,t) \in \delta E\}$ and $T = \{t|(s,t) \in \delta E\}$ respectively the sets of sources and tips of modified edges δE. Given $\tau > 1$, if f is analytic in the level set $\Omega \cup \{\psi(z)|z \notin \Omega, |z| \leq \tau\}$, then*

$$\left|\left(f(A) - f(\widetilde{A})\right)_{k\ell}\right| \leq \mu_\tau(f) \frac{2}{\pi} \frac{\tau}{\tau - 1} \left(\frac{1}{\tau}\right)^{\delta+2},$$

where $\delta = d_G(k, S) + d_G(T, \ell)$ and $\mu_\tau(f) = \int_{D_\tau} |f(\psi(z))|\, \mathrm{d}z$, with $D_\tau = \{z : |z| = \tau\}$.

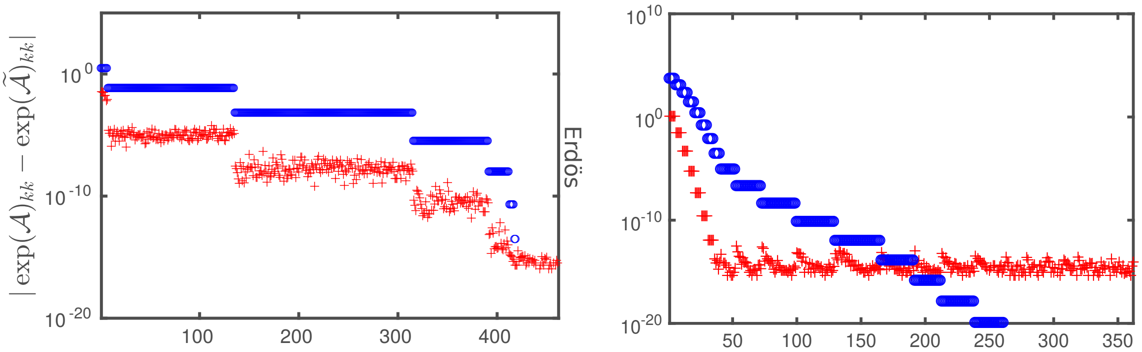


Figure 1: The red crosses are the difference $|\exp(A)_{kk} - \exp\left(\widetilde{A}\right)_{kk}|$ for every node k, whereas the blue dots correspond to the bound obtained specializing Theorem 1 to each case. Left: Erdös collaboration network. Right: London city transportation network.

4 Numerical examples

The first example is an undirected network borrowed from [5] and represents the Erdös collaboration network (Number of nodes: 472). Notice that the diameter is 11, hence it is proportional to the logarithm of the number of nodes. This feature is common to many complex networks and is related to the so called "small-world" phenomenon. We focus here on the analysis of the correlation between the variation of the network centralities and the variation of the distances in G with respect to the set of perturbed edges. For this reason, normalized adjacency matrix A is considered in the first example, so to guarantee the field of values of both the original and the perturbed matrices to be constrained within the unit segment $[-1, 1]$.

The second example is borrowed from a real-world data set representing the London city transportation network [6] (Number of nodes: 369). The undirected network that we consider here is the aggregate version of the original multi-layer network. The nodes correspond to train stations and the existing routes between them are the edges.

In both the examples we have selected, respectively, the 10 and the 5 nodes having smallest centrality $\exp(A)_{kk}$ and we have perturbed the edge topology of the networks by adding all the missing edges among those nodes. Figure 1 represents the variation of network exp-centrality values $|\exp(A)_{kk} - \exp\left(\widetilde{A}\right)_{kk}|$ (red crosses) and the bound obtained specializing Theorem 1 (blue circles). Let us point out that in both the left and right plot of Figure 1 we are relabeling the nodes according with the distance from (and to) the set of modified nodes.

5 Conclusion

When A is modified into $\widetilde{A} = A + \delta A$, the entries of $f(\widetilde{A})$ should in principle be re-computed from scratch and this is can be a very costly operation. Therefore being able to efficiently update the entries of $f(A)$ is a relevant task. When δA has low rank, this problem can be easily addressed via the Sherman-Morrison formula for the special function $f(z) = z^{-1}$. For the case of general functions f, important advances in this direction have been made in [1], simultaneously and independently with respect to the present work.

On a related but different line, using the bounds we have proposed, we are able to predict the magnitude of variation in the f-centralities of G when changes occur in a localized set of edges or, vice-versa, for each node k we can locate a set of nodes whose change in the edge topology affects the score $f(A)_{kk}$ by a small order of magnitude.

Acknowledgement: The work of S.P. has been supported by Charles University Research program No. UNCE/SCI/023, and by INdAM, GNCS (Gruppo Nazionale per il Calcolo Scientifico). The work of F.T. has been supported by the Marie Curie Individual Fellowship MAGNET n. 744014.

References

[1] B. Beckermann, D. Kressner, M. Schweitzer: *Low-rank updates of matrix functions*, SIAM J. Matrix Anal. Appl., 39, 2018, pp. 539–565.

[2] M. Benzi, P. Boito: *Decay properties for functions of matrices over C^*-algebras*, Linear Algebra Appl., 456, 2014, pp. 174–198.

[3] M. Benzi, E. Estrada, C. Klymko: *Ranking hubs and authorities using matrix functions*, Linear Algebra Appl., 438, 2013, pp. 2447–2474.

[4] M. Benzi, N. Razouk: *Decay bounds and O(n) algorithms for approximating functions of sparse matrices*, Electron. Trans. Numer. Anal., 28, 2007, pp. 16–39.

[5] T. Davis, Y. Hu: *University of Florida sparse matrix collection* , ACM Trans. Math. Softw., 38, 2011, pp. 1–25.

[6] M. De Domenico, A. Solé-Ribalta, S. Gómez, A. Arenas: *Navigability of interconnected networks under random failures*, Proc. Natl. Acad. Sci. USA, 111, 2014, pp. 8351–8356.

[7] E. Estrada, D.J. Higham: *Network properties revealed through matrix functions*, SIAM rev., 52, 2010, pp. 696–714.

[8] E. Estrada, J. A. Rodriguez-Velazquez: *Subgraph centrality in complex networks*, Phys. Rev. E, 71, 2005, p. 056103.

[9] N. J. Higham: *Functions of matrices: theory and computation*, SIAM, 2008.

[10] S. Pozza, F. Tudisco: *On the stability of network indices defined by means of matrix functions*, SIAM J. Matrix Anal. Appl., 39, 2018, pp. 1521–1546.

Numerical modeling of flow and mechanics in fractured porous media

J. Stebel, J. Březina

Technical University of Liberec

1 Introduction

Fluids injected under high pressure to boreholes in rock massif can induce mechanical deformation or even seismic waves with undesirable effects. On the other hand, the creation of microscopic fissures is an important effect in hydraulic fracturing, a process to increase the permeability of underground reservoirs.

Our aim is to develop a model that can describe the hydro-mechanical interaction in the presence of fractures and their possible evolution. In this contribution we present a mathematical model of fluid flow and linear elastic response in a porous media containing a fracture. We consider a domain $\Omega \subset R^d$, $d \in \{2,3\}$, consisting of two parts: the so-called matrix Ω_m and the fracture Ω_f (see Fig. 1).

The basic mathematical model of hydro-mechanical interaction is based on the Biot system [2]:

$$\left.\begin{aligned} \partial_t\left(Sp + \alpha\nabla\cdot\boldsymbol{u}\right) - \mathbb{K}\Delta p = g \\ -\nabla\cdot(2\mu\boldsymbol{\varepsilon}[\boldsymbol{u}] + \lambda(\nabla\cdot\boldsymbol{u})\mathbb{I}) + \alpha\nabla p = \boldsymbol{f} \end{aligned}\right\} \text{ in } (0,T)\times\Omega_m. \tag{1}$$

Here the pressure p and the displacement $\boldsymbol{u}$ are the principal unknowns and $\boldsymbol{\varepsilon}[\boldsymbol{u}] := (\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^\top)/2$ is the symmetric part of gradient of $\boldsymbol{u}$. For simplicity, the storativity S, the Biot coefficient α, the hydraulic conductivity tensor $\mathbb{K}$, the Lamé parameters μ, λ, the fluid volumetric source g and the body force $\boldsymbol{f}$ are assumed to be constant in Ω_m and $\mathbb{K}$ symmetric positive definite.

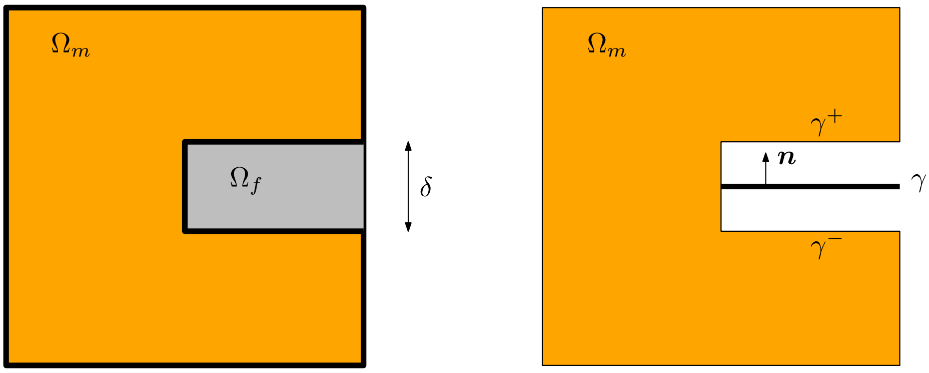


Figure 1: The model domain with full and reduced fracture.

2 Equations in reduced fracture

We assume that the fracture is of the form $\Omega_f := \{\boldsymbol{x} + s\boldsymbol{n};\ \boldsymbol{x}\in\gamma,\ s\in(-\delta/2,\delta,2)\}$, where γ is a $(d-1)$-dimensional manifold and $\boldsymbol{n}$ is the unit normal to γ in a chosen direction. If the width

δ is small compared to the size of Ω then, it is reasonable to replace Ω_f by γ. Indeed, dne can introduce the averaged quantities on γ:

$$P := \frac{1}{\delta}\int_{-\delta/2}^{\delta/2} p(\cdot + s\boldsymbol{n})\,ds, \quad \boldsymbol{U} := \frac{1}{\delta}\int_{-\delta/2}^{\delta/2} \boldsymbol{u}(\cdot + s\boldsymbol{n})\,ds$$

and the approximate tangential and normal gradients on $\gamma^\pm := \gamma \pm \boldsymbol{n}$:

$$\nabla p_{|\gamma^\pm} = (\nabla_\tau p + \nabla_\nu p)_{|\gamma^\pm} \approx \nabla_\tau P \pm \frac{2}{\delta}(p_{|\gamma^\pm} - P)\boldsymbol{n},$$

$$\nabla \boldsymbol{u}_{|\gamma^\pm} = (\nabla_\tau \boldsymbol{u} + \nabla_\nu \boldsymbol{u})_{|\gamma^\pm} \approx \nabla_\tau \boldsymbol{U} \pm \frac{2}{\delta}(\boldsymbol{u}_{|\gamma^\pm} - \boldsymbol{U}) \otimes \boldsymbol{n}.$$

After integrating the Biot system over the width of the fracture and using the above approximations, one obtains the following reduced equations in γ (see [4] for details on the dimension reduction and [3] for error analysis of reduced model of flow):

$$\left.\begin{array}{r} \delta\left\{\partial_t\left(S_f P + \alpha_f \nabla_\tau\cdot\boldsymbol{U}\right) - \nabla_\tau\cdot(\mathbb{K}_f\nabla_\tau P)\right\} + F^+ + F^- - \partial_t G = \delta g_f \\ \delta\left\{-\nabla_\tau\cdot(2\mu_f\boldsymbol{\varepsilon}_\tau[\boldsymbol{U}] + \lambda_f(\nabla_\tau\cdot\boldsymbol{U})\mathbb{I}) + \alpha_f\nabla_\tau P\right\} + \boldsymbol{Q}^+ + \boldsymbol{Q}^- - \nabla_\tau\cdot\mathbb{R} = \delta\boldsymbol{f}_f \end{array}\right\} \text{ in } (0,T)\times\gamma. \quad (2)$$

Here S_f, α_f, $\mathbb{K}_f$, g_f, μ_f, λ_f, $\boldsymbol{f}_f$ are physical constants in the fracture and $\boldsymbol{\varepsilon}_\tau$ stands for the symmetric part of ∇_τ. The fracture can be void ($\mu_f = \lambda_f = 0$) or filled by an elastic material ($\mu_f, \lambda_f > 0$). The terms $F^\pm$, $\boldsymbol{Q}^\pm$ represent fluxes and normal stresses, respectively, acting on both sides of the fracture and G, $\mathbb{R}$ are extra terms arising from the dimension reduction.

The systems (1) and (2) are coupled through the following interface conditions:

$$\mathbb{K}\nabla p\cdot\boldsymbol{n} = \pm F^\pm, \quad (2\mu\boldsymbol{\varepsilon}[\boldsymbol{u}] + \lambda(\nabla\cdot\boldsymbol{u})\mathbb{I})\boldsymbol{n} - \alpha p\boldsymbol{n} = \pm\boldsymbol{Q}^\pm \text{ in } (0,T)\times\gamma^\pm. \quad (3)$$

If the fracture is immersed into the matrix then we assume no interaction in the tangential direction, i.e. homogeneous Neumann conditions on the appropriate part of the relative boundary of γ.

3 Approximation and numerical solution

For convenience of spacial discretization we replace Ω_m by $\Omega\backslash\gamma$. The equations (1) are discretized by equal order discontinuous Galerkin method which helps prevent locking problems in the nearly-incompressible regime and allows easy treatment of internal discontinuities along the fracture. The system (2) is discretized by the standard finite element method. The temporal discretization is done using the implicit Euler method.

We implement the approximate problem using the FEniCS library [1]. We employ a fixed-stress splitting technique [5] to decouple the equations for flow and mechanics and analyze its convergence.

We test the reduced problem in a simplified model of injection of a fluid into the fracture. We use a 2d domain $\Omega = (0,1)^2$ with $\gamma = (\frac{1}{2}, 1) \times \{\frac{1}{2}\}$. The results are depicted in Figure 2.

4 Conclusion

We have derived the reduced model for the Biot poroelasticity in domains containing discrete fractures. The model has been tested using a suitable approximation on a simple problem of

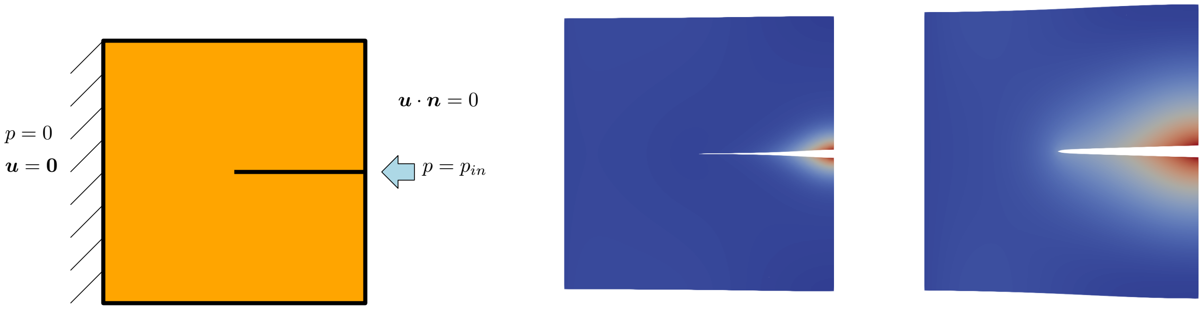

$p = 0$
$\boldsymbol{u} = \boldsymbol{0}$
$\boldsymbol{u} \cdot \boldsymbol{n} = 0$
$p = p_{in}$

Figure 2: Test problem – injection onto fracture. Dirichlet boundary conditions (left); pressure on the deformed mesh (middle: $t = 0.01$, right: $t = 0.1$).

injection into fracture. The numerical results show good agreement with the fully d-dimensional model. Our next aim will be to extend the model to nonlinear mechanics and apply XFEM to treat fracture evolution.

Acknowledgement: This work has been supported by the project "RINGEN - research infrastructure upgrade" No CZ.02.1.01/0.0/0.0/16_013/0001792, co-funded by the EU Operational Programme "Research, Development and Education".

References

[1] M.S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G.N. Wells: *The FEniCS Project Version 1.5.* Archive of Numerical Software, vol. 3, 2015.

[2] M.A. Biot: *General theory of three-dimensional consolidation.* Journal of Applied Physics, **12**(2), 1941, pp. 155–164.

[3] J. Březina, J. Stebel': *Analysis of Model Error for a Continuum-Fracture Model of Porous Media Flow.* In High Performance Computing in Science and Engineering, Lecture Notes in Computer Science, Springer, 2016, pp. 152–160.

[4] V. Martin, J. Jaffré, J.E. Roberts. *Modeling fractures and barriers as interfaces for flow in porous media.* SIAM Journal on Scientific Computing, **26**(5), 2005, pp. 1667–1691.

[5] A. Mikelić, M.F. Wheeler: *Convergence of iterative coupling for coupled flow and geomechanics*, Comput. Geosci., **17**, 2013, pp. 455–461.

Computation of composite strengths by limit analysis and mesh adaptivity

S. Sysala, R. Blaheta, J. Haslinger, A. Kolcun

Institute of Geonics of the CAS, Ostrava

1 Introduction

Limit analysis is one of the main methods for solution of geotechnical and other stability problems. By this method, we determine a limit value ζ^* of the load parameter $\zeta \geq 0$ for a prescribed set of applied external forces. An investigated body collapses beyond the limit value and thus this ultimate load also enables us to describe failure mechanisms. The value ζ^* is defined by a convex optimization (variational) problem which can be formulated either in terms of stress or kinematic fields leading to the static, and kinematic approaches, respectively. We refer to [2, 3, 4] and the references therein.

The aim of this paper is to investigate abilities of this method for computation of compressive strengths of composite materials. To this end, we choose a laboratory prepared sample consisting of a hard coal matrix and a polyurethane (PUR) resin, see see Figure 1. Properties of this coal-PUR composite was studied in [1] by using laboratory experiments, X-ray CT visualization and numerical simulations. The numerical treatment was done there in linear elastic range including a finite element homogenization with voxel grid derived from CT scans of samples.

The rest of the paper is organized as follows. First, we introduce material models for the coal and the PUR resin. Then, we define the kinematic limit analysis problem and briefly describe its numerical solution. Finally, we present results of numerical experiments for two different CT-based 2D geometries.

2 Material models

For purposes of modelling presented in this paper, we use the following simplified assumptions: a) the coal matrix is homogeneous and isotropic and b) the PUR resin has a constant degree of foaming in the composite sample. Further, the Drucker-Prager yield criterion is used for the coal because it enables to distinguish different material behavior in tension and compression. The

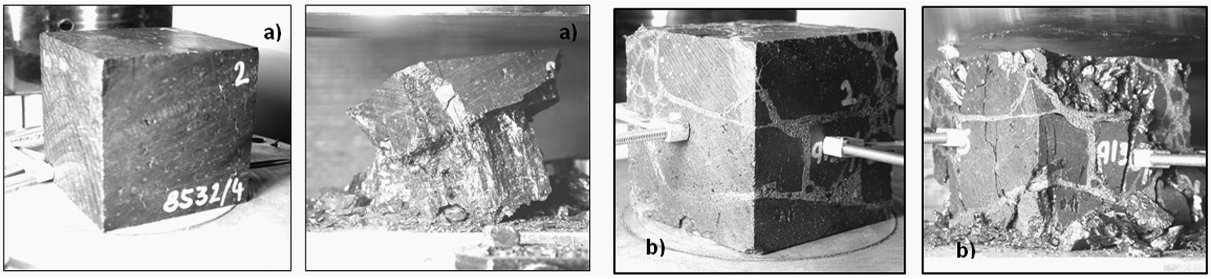


Figure 1: Coal (a) and coal-PUR (b) samples and their failures caused by compressive tests.

corresponding set of admissible stress tensors reads as

$$B_1 = \left\{ \boldsymbol{\tau} \in \mathbb{R}^{3\times3}_{sym} \mid |\boldsymbol{\tau}^D| + \frac{a}{3}\mathrm{tr}\,\boldsymbol{\tau} \leq \gamma \right\}, \quad a := \frac{3\sqrt{2}\tan\phi}{\sqrt{9+12\tan^2\phi}}, \quad \gamma := \frac{3c\sqrt{2}}{\sqrt{9+12\tan^2\phi}}, \tag{1}$$

where $\mathbb{R}^{3\times3}_{sym}$ is a space of second order symmetric tensors, $\boldsymbol{\tau}$ represents the Cauchy stress tensor, $\boldsymbol{\tau}^D$ is the deviatoric part of $\boldsymbol{\tau}$, $|\boldsymbol{\tau}^D|$ stands for the Frobenius norm of $\boldsymbol{\tau}^D$, $\mathrm{tr}\,\boldsymbol{\tau}$ is the trace of $\boldsymbol{\tau}$ and $a, \gamma > 0$ are material parameters computed from the cohesion c and the friction angle ϕ.

The resin is modelled by the von Mises yield criterion. This criterion is chosen since PUR has much more similar properties in tension and compression than the coal. The corresponding admissible stress tensors are defined as follows:

$$B_2 = \left\{ \boldsymbol{\tau} \in \mathbb{R}^{3\times3}_{sym} \mid |\boldsymbol{\tau}^D| \leq Y \right\}, \quad Y := \sqrt{2/3}\sigma_c, \tag{2}$$

where $\sigma_c > 0$ denotes the yield stress which depends on the degree of foaming.

3 Kinematic problem of limit analysis

We consider that the body occupies the domain Ω. The space of admissible kinematic fields defined in Ω is denoted as $\mathbb{V}$. We assume that these fields satisfy homogeneous Dirichlet boundary conditions on selected parts of $\partial\Omega$. Further, $L\colon \mathbb{V} \to R$ denotes the load functional which may consist of volume or surface external forces, and

$$\varepsilon(\boldsymbol{v}) := \frac{1}{2}[\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^\top] \ \text{ in } \Omega, \quad \boldsymbol{v} \in \mathbb{V} \tag{3}$$

represents the linearized strain tensor. Finally, we define the plastic dissipation potential

$$J_\infty(\boldsymbol{v}) = \int_\Omega j_\infty(\varepsilon(\boldsymbol{v}))\,dx, \quad j_\infty(\boldsymbol{e}) := \sup_{\boldsymbol{\tau}\in B} \boldsymbol{\tau} : \boldsymbol{e}, \quad \boldsymbol{e} \in \mathbb{R}^{3\times3}_{sym}, \tag{4}$$

where B is a set of plastically admissible stress tensors depending on a yield criterion and $\boldsymbol{x} \in \Omega$. Notice that the supremum over B need not be finite everywhere in $\mathbb{R}^{3\times3}_{sym}$ and thus the value $+\infty$ of j_∞ is allowed.

The kinematic limit analysis leads to the following problem [2, 3, 4]: find $\zeta^* \geq 0$ such that

$$\zeta^* = \inf_{\substack{\boldsymbol{v}\in\mathcal{K} \\ L(\boldsymbol{v})=1}} J_\infty(\boldsymbol{v}), \quad \mathcal{K} := \{\boldsymbol{v} \in \mathbb{V} \mid J_\infty(\boldsymbol{v}) < +\infty \text{ in } \Omega\}. \tag{5}$$

In general, the minimizer exists in the BD-space and thus may be discontinuous along certain zones in the body which predict the failure.

Next, we specify problem (5) for the coal-PUR composite. To this end, we split Ω into two parts, Ω_1 and Ω_2 representing the coal matrix and the resin, respectively. In Ω_i, we set $B := B_i$, $i = 1, 2$, and find a closed form of the function j_∞ for these two particular cases. The resulting limit analysis problem reads as

$$\zeta^* = \inf_{\substack{\boldsymbol{v}\in\mathcal{K} \\ L(\boldsymbol{v})=1}} \left\{ \frac{\gamma}{a}\int_{\Omega_1} \mathrm{div}\,\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + Y\int_{\Omega_2} |\varepsilon(\boldsymbol{v})|\,\mathrm{d}\boldsymbol{x} \right\}, \tag{6}$$

$$\mathcal{K} := \{\boldsymbol{v} \in \mathbb{V} \mid \ \mathrm{div}\,\boldsymbol{v} \geq a|\varepsilon^D(\boldsymbol{v})| \text{ in } \Omega_1, \ \ \mathrm{div}\,\boldsymbol{v} = 0 \text{ in } \Omega_2\}, \tag{7}$$

where $\operatorname{div} \boldsymbol{v} = \operatorname{tr} \varepsilon(\boldsymbol{v})$ denotes the divergence of $\boldsymbol{v}$. We see that the Drucker-Prager yield criterion leads to a linear functional and conic constraints, while the von Mises yield criterion gives a nonsmooth functional and linear equality constraints.

To solve (5), we use a numerical procedure developed in [2, 3, 4, 5]. First, the following penalization method is applied:

$$\zeta_\alpha = \inf_{\substack{\boldsymbol{v}\in\mathbb{V}\\ L(\boldsymbol{v})=1}} J_\alpha(\boldsymbol{v}), \quad J_\alpha(\boldsymbol{v}) = \int_\Omega j_\alpha(\varepsilon(\boldsymbol{v}))\,dx, \quad j_\alpha(\boldsymbol{e}) = \sup_{\boldsymbol{\tau}\in B}\{\boldsymbol{\tau}:\boldsymbol{e} - \frac{1}{2\alpha}|\boldsymbol{\tau}|^2\}, \quad \boldsymbol{e}\in\mathbb{R}^{3\times3}_{sym}, \tag{8}$$

where $\alpha > 0$ is the penalty parameter. The function j_α is finite-valued (unlike j_∞), convex, smooth and $j_\alpha \to j_\infty$ pointwisely as $\alpha \to +\infty$. It holds that the function $\alpha \mapsto \zeta_\alpha$ is continuous, nondecreasing and $\zeta_\alpha \leq \zeta^*$. Under appropriate assumptions, it is also possible to prove that $\zeta_\alpha \to \zeta^*$ as $\alpha \to +\infty$. Further, we solve the penalized problem by the finite element method, continuation techniques and the semismooth Newton method. To improve accuracy of results, local mesh adaptivity introduced in [5] is also used. In particular, we use P2-elements and 7-point Gauss quadrature for numerical integration. Numerical solution is implemented in Matlab.

4 Numerical experiments

We consider two plane strain problems defined in the same square domain Ω with the size 13 [cm]. These problems follow from two CT-based images, CT1 and CT2, with the resolution 107×107 pixels, see Figures 2 and 3 on the left, respectively. The coal matrix is indicated by dark color. Further, the symmetry boundary conditions are prescribed on the left and bottom sides of Ω. The load functional L is defined by the unit uniaxial compression [MPa] applied on the top of Ω, i.e.,

$$L(\boldsymbol{v}) = -\int_0^{13} v_2(x_1, 13)\,dx_1, \quad \boldsymbol{v} = (v_1, v_2) \in \mathbb{V}, \tag{9}$$

We set $\phi = 20$ [Deg], $c = 7$ [MPa] in Ω_1 and $\sigma_c = 17$ [MPa] in Ω_2. The original mesh follows from the image resolution. Then, three levels of adaptive refinements are used. The finest CT-based meshes have about 125 thousands degrees of freedom and 220 thousands integration points.

The computed strengths (ζ^*) are equal to 19.96 and 19.81 [MPa] for CT1 and CT2, respectively. For visualization of the failure, we use the quantity

$$\frac{\gamma}{a}\operatorname{div} \boldsymbol{v}^* \;\text{ in } \Omega_1, \quad Y|\varepsilon(\boldsymbol{v}^*)| \;\text{ in } \Omega_2, \tag{10}$$

where $\boldsymbol{v}^*$ denotes the numerical minimizer in problem (6). The failure zones for CT1 and CT2 images are depicted in Figures 2 and 3 (middle), respectively. The corresponding kinematic fields $\boldsymbol{v}^*$ with enlarged deformed shapes are shown in Figures 2 and 3 (right).

5 Conclusion

We use limit analysis to compute the compressive strength of composite materials and to visualize failure zones. Numerical examples illustrate abilities of limit analysis. For example, this method in combination with local mesh adaptivity is capable to significantly reduce the number of degrees of freedom because kinematic fields are rigid far from the failure zones.

Acknowledgement: This work has been supported by the Czech Science Foundation, project No. 19-11441S.

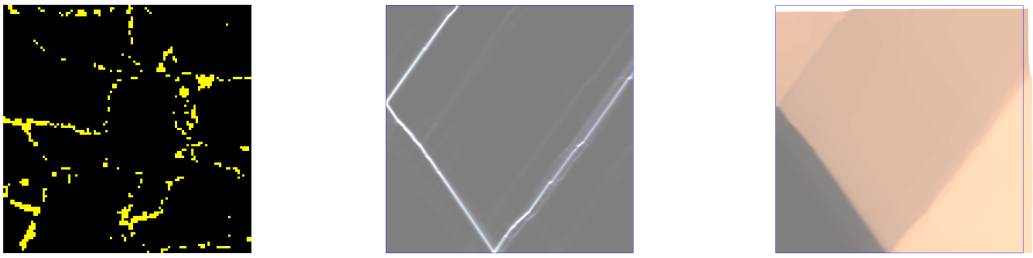


Figure 2: Image CT1 (left), the corresponding failure zones (middle) and kinematic field (right).

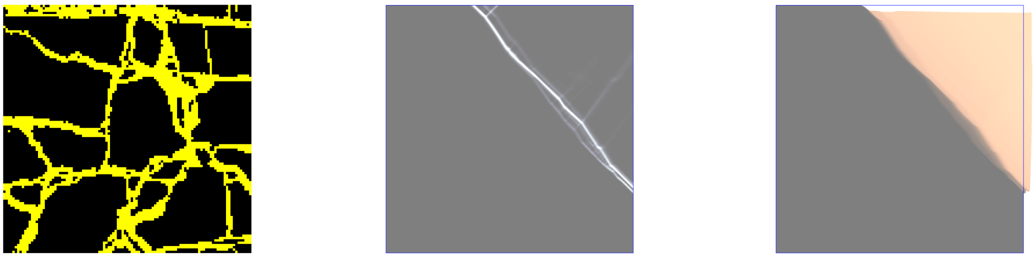


Figure 3: Image CT1 (left), the corresponding failure zones (middle) and kinematic field (right).

References

[1] R. Blaheta, R. Kohut, A. Kolcun, K. Souček, L. Staš, L. Vavro: *Digital image based numerical micromechanics of geocomposites with application to chemical grouting.* International Journal of Rock Mechanics and Mining Sciences, **77**, 2015, pp. 77–88.

[2] J. Haslinger, S. Repin, S. Sysala: *A reliable incremental method of computing the limit load in deformation plasticity based on compliance: Continuous and discrete setting.* Journal of Computational and Applied Mathematics **303**, 2016, pp. 156–170.

[3] J. Haslinger, S. Repin, S. Sysala: *Guaranteed and computable bounds of the limit load for variational problems with linear growth energy functionals.* Applications of Mathematics **61**, 2016, pp. 527–564.

[4] S. Repin. S. Sysala, J. Haslinger: *Computable majorants of the limit load in Hencky's plasticity problems.* Computer and Mathematics with Applications **75**, 2018, pp. 199–217.

[5] S. Sysala, J. Haslinger, S. Repin: *Reliable computation and local mesh adaptivity in limit analysis.* In: J. Chleboun, P. Kůs, P. Přikryl, M. Rozložník, K. Segeth, J. Šístek, T. Vejchodský eds.: Proceedings of the conference Programs and Algorithms of Numerical Mathematics 19, Institute of Mathematics CAS, Prague. To appear in 2019.

A domain decomposition solver for parallel adaptive mesh refinement

J. Šístek[1], *P. Kůs*[2]

[1] Institute of Mathematics, Czech Academy of Sciences, Prague
[2] Max Planck Computing and Data Facility, Max Planck Institute, Garching bei München, Germany

1 Adaptive mesh refinement and domain decomposition

Adaptive mesh refinement is an important part of solving problems with complicated solutions or when a prescribed accuracy needs to be achieved. In this approach, solution is found on a given mesh and its local error is estimated. Regions where the estimated error is high are then refined to improve the accuracy, and the solution is recomputed. This strategy leads to accumulation of degrees of freedom to regions with abrupt changes in the solution, such as boundary or internal layers.

The growing size of the problems, especially in 3D, more and more often requires solving these on a parallel computer. To this end, partitioning of the computational mesh into subdomains is required. If the partitioning is not adjusted to the new meshes in the adaptive process, large imbalances in subdomain sizes quickly emerge, and utilization of the parallel computer becomes inefficient.

A viable way to maintain the subdomain sizes balanced is repartitioning using the space-filling curves, which maintains approximately equal number of elements in each subdomain. Such approach is offered by the *p4est* library [1]. However, this repartitioning strategy typically produces subdomains composed of several disconnected components.

Another important ingredient of simulations based on FEM is the solver for the arising system of linear equations. A good match for a parallel computation is using a domain decomposition method, and a recent member of this family is the multilevel Balancing Domain Decomposition based on Constraints (BDDC) [2, 3]. This method is implemented in our parallel solver *BDDCML*[3] [4].

We have developed a custom implementation of the FEM to study the impact of the special structure of the subdomains created by the *p4est* library on the BDDC solver. Disconnected components of subdomains are detected using subdomain mesh graphs while hanging nodes are incorporated naturally into the computation by the non-overlapping domain decomposition.

2 Numerical results and discussion

The solver has been applied to a number of benchmark Poisson and linear elasticity problems. The problem geometry was a unit cube in all our experiments. These experiments were performed using up to about 1 billion unknowns and 4096 CPUs of the *Salomon* supercomputer at the IT4Innovations supercomputing centre.

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

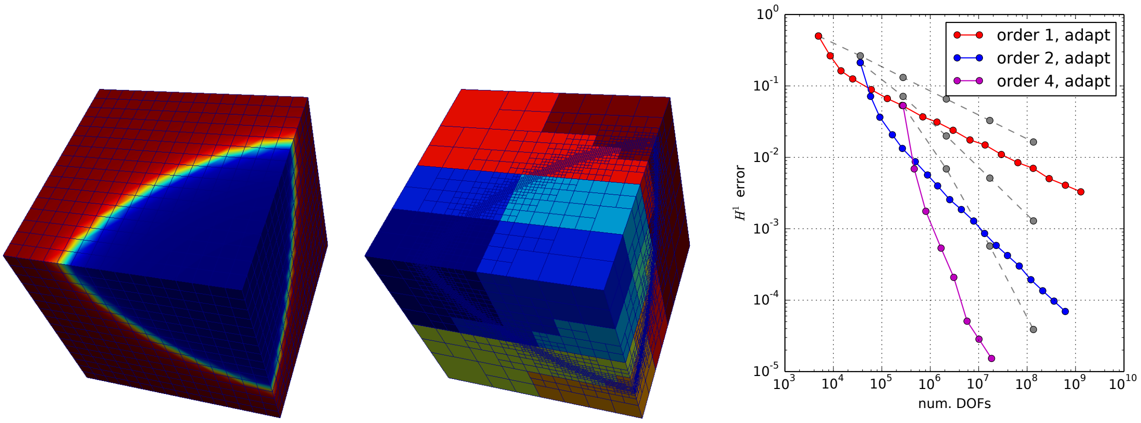


Figure 1: A benchmark for a parallel adaptive computation: visualization of the exact solution (left), adaptively refined mesh partitioned into subdomains (centre), and convergence of the H^1 norm of the error on 2048 subdomains for different polynomial orders (right), adaptive (colour lines) vs. uniform (grey lines) mesh refinements.

In the first set of experiments, we have generated a mesh refined in several predefined regions. The behaviour of the solver was compared to uniformly refined meshes resulting in approximately half of the iterations, yet comparable in time to solution.

The next experiment was running a benchmark adaptive computation, see Fig. 1. This test was performed for the Poisson problem using linear, quadratic, and fourth-order finite elements. The solver has allowed us to verify the convergence rate for problems refined up to 1 billion unknowns using 2048 subdomains (and CPU cores). More details can be found in our paper [5].

Acknowledgement: This research was supported by the Czech Science Foundation through grant 18–09628S, and by the Czech Academy of Sciences through RVO:67985840. Computational time on the *Salomon* supercomputer has been provided by the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), funded by the European Regional Development Fund and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, as well as Czech Ministry of Education, Youth and Sports via the project Large Research, Development and Innovations Infrastructures (LM2011033).

References

[1] C. Burstedde, L. Wilcox, O. Ghattas: *p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees*, SIAM J. Sci. Comput. 33 (3), 2011, pp. 1103–1133.

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

[3] J. Mandel, B. Sousedík, C.R. Dohrmann: *Multispace and multilevel BDDC*, Computing 83 (2-3), 2008, pp. 55–85.

[4] B. Sousedík, J. Šístek, J. Mandel: *Adaptive-Multilevel BDDC and its parallel implementation*, Computing 95 (12), 2013, pp. 1087–1119.

[5] P. Kůs, J. Šístek: *Coupling parallel adaptive mesh refinement with a nonoverlapping domain decomposition solver*, Adv. Eng. Softw. 110, 2017, pp. 34–54.

On polynomial robustness of flux reconstructions

M. Vlasák, Z. Vlasáková

Faculty of Mathematics and Physics, Charles University in Prague

1 Continuous and discrete problem

We assume Poisson equation

$$-\Delta u = f \quad \text{on } \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, $f \in L^2(\Omega)$, $u|_{\partial\Omega} = 0$. The solution $u \in H_0^1(\Omega)$ of problem (1) is approximated as $u_h \in V_h$ by the finite element method (FEM), where $V_h \subset H_0^1(\Omega)$ is the classical finite element space consisting of piecewise polynomial functions up to degree p.

2 A posteriori error estimates

For the sake of simplicity we assume that function f is piecewise polynomial function, otherwise the estimates need to be enhanced by the oscillation term. A posteriori error estimates for problem (1) with a guaranteed upper bound, i.e. all the constants in the upper bound are known and available/computable, are often motivated by the well known Hyper-circle theorem from [4]:

Theorem 1. *Let $u \in H_0^1(\Omega)$ be the exact solution of problem (1) and let $\sigma \in H(\mathrm{div}, \Omega)$ satisfy $-\mathrm{div}\ \sigma = f$. Then*

$$\|\nabla u - \nabla v\|_{L^2(\Omega)}^2 + \|\nabla u - \sigma\|_{L^2(\Omega)}^2 = \|\sigma - \nabla v\|_{L^2(\Omega)}^2 \quad \forall v \in H_0^1(\Omega). \tag{2}$$

Setting $v = u_h$ the FEM solution and assuming that such a σ is available, we get guaranteed upper bound

$$\|\nabla u - \nabla u_h\|_{L^2(\Omega)} \leq \|\sigma - \nabla u_h\|_{L^2(\Omega)}. \tag{3}$$

To investigate the quality of bound (3), which depends on the choice of flux σ, we examine the opposite inequality, i.e. the local efficiency estimate

$$\|\sigma - \nabla u_h\|_{L^2(K)} \leq C\|\nabla u - \nabla u_h\|_{L^2(\omega(K))}, \tag{4}$$

where K is an element of finite element mesh and $\omega(K)$ is a patch of elements surrounding K. It is possible to show for usual choices of flux σ that the constant C from (4) is independent of solutions u, u_h and of the FEM mesh size h. But there are only a few results investigating the dependence of this constant on the polynomial degree p of FEM.

3 Construction of flux σ

There are many approaches to the construction of the flux σ. According to [1], the efficiency constant is independent of the polynomial degree for the construction of σ based on the local mixed FEM. The polynomial robustness of this construction is proved for other underlying methods in [3].

Nevertheless, such a construction is quite complicated and not very comfortable for implementation. Therefore, we suggest following approach. Let $RT_p(K) = P_p(K)^d + xP_p(K)$ be the local Raviart-Thomas space. For the details about approximation spaces to $H(\mathrm{div}, \Omega)$ see [2]. We seek $\sigma|_K \in RT_p(K)$ such that

$$\begin{aligned} \sigma \cdot n &= \langle \nabla u_h \rangle \cdot n \quad \text{on } \partial K \\ (\sigma, w)_{L^2(K)} &= (\nabla u_h, w)_{L^2(K)} \quad \forall w \in P_{p-1}(K)^d, \end{aligned} \tag{5}$$

where n is unit outer normal vector and $\langle \nabla u_h \rangle$ is the mean value. On rectangular meshes the construction of σ can be defined analogically.

4 p-robustness

The goal of the talk is to present the robustness and practical usefulness of flux construction (5). To this end, we present following result:

Theorem 2. *Let $d = 1$. Let $u \in H_0^1(\Omega)$ be the exact solution of problem (1) and u_h be its FEM approximation. Let σ be given by (5). Then the efficiency constant from (4) increases at most as $\sqrt{p}$, i.e.*

$$\|\sigma - \nabla u_h\|_{L^2(K)} \leq C\sqrt{p}\|\nabla u - \nabla u_h\|_{L^2(\omega(K))}, \tag{6}$$

where the constant C from the efficiency estimate (6) is independent of the polynomial degree p.

Acknowledgement: The research of M. Vlasák was supported by the grant 17-01747S of the Czech Science Foundation. M. Vlasák was a junior member of the university centre for mathematical modeling, applied analysis and computational mathematics (MathMAC).

References

[1] D. Braess, V. Pillwein, J. Schöberl: *Equilibrated Residual Error Estimates are p-Robust.* Comput. Methods Appl. Mech. Engrg. 198, 2009, pp. 1189–1197.

[2] D. Boffi, F. Brezzi, M. Fortin: *Mixed Finite Element Methods and Applications.* Springer, Heidelberg, 2013.

[3] A. Ern, M. Vohralík: *Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and mixed discretizations.* Quart. Appl. Math. 5, 1947, pp. 241–269.

[4] W. Prager, J.L. Synge: *Approximations in elasticity based on the concept of function space.* SIAM J. Numer. Anal. 53(2), 2015, pp. 1058–1081.

Acceleration of FFT-based homogenisation by low-rank tensors approximations

J. Vondřejc[1], *D. Liu*[1], *M. Ladecký*[2], *H.G. Matthies*[1]

[1] Technische Universität Braunschweig, Institute of Scientific Computing, Mühlenpfordstrasse 23, 38106 Braunschweig, Germany,

[2] Czech Technical University in Prague, Faculty of Civil Engineering, Thakurova 7/2077
166 29 Prague 6, Czech Republic

1 Introduction

The main task of the homogenization is to homogeneously describe material properties of heterogeneous materials, based on knowledge of microstructure geometry and properties of its phases. Evaluation of the homogenised properties with high accuracy requires a detailed knowledge of materials' microstructure. Unfortunately, this knowledge comes hand in hand with high memory and time requirements of approximate solution to homogenisation problem. To find a solution we use Fourier spectral method. This method, based on the Fast Fourier Transform (FFT), has been introduced in 1994 by Moulinec and Suquet [3] and lately explained by Vondřejc, Zeman and Marek in [2]. The Fourier spectral method has lower time and memory requirements compared to other methods but still needs significant computational resources. Especially, for precisely characterized three-dimensional microstructure, the memory requirements can easily overflow the memory capacity of a typical workstation. In an effort to overcome these issues, we have tried to employ low-rank tensors to our homogenisation scheme. The idea of low-rank tensors, or low-rank approximations is to express large multidimensional tensors by fewer parameters. This compression can lead to a huge reduction of the mentioned computational requirements.

2 Homogenisation problem

We solve a scalar homogenization problem: to find $\boldsymbol{A}_{\mathrm{H}} \in \mathbb{R}^{d\times d}$, $d = 2, 3$, such that for all constant vectors $\boldsymbol{E} \in \mathbb{R}^d$

$$\boldsymbol{A}_{\mathrm{H}}\boldsymbol{E}_i = \frac{1}{|\mathcal{Y}|}\int_{\mathcal{Y}} \boldsymbol{A}(\boldsymbol{x})\left(\boldsymbol{E}_i + \nabla u_i(\boldsymbol{x})\right)\mathrm{dx}, \quad \mathrm{i} = 1,\dots,\mathrm{d} \tag{1}$$

where $u \in V$ is the solution of

$$\nabla\cdot(\boldsymbol{A}(\boldsymbol{x})\nabla u_i(\boldsymbol{x})) = \nabla\cdot(\boldsymbol{A}(\boldsymbol{x})\boldsymbol{E}_i) \quad \text{on} \quad \mathcal{Y}, \quad i = 1,\dots,d \tag{2}$$

satisfying periodic boundary conditions and having zero mean on $\mathcal{Y}$. We consider a rectangular domain $\mathcal{Y}$ and a material data function $\boldsymbol{A}(\boldsymbol{x}) : \mathcal{Y} \to \mathbb{R}^{d\times d}$, which is symmetric and uniformly positive definite in $\mathcal{Y}$. The core of the numerical computation of $\boldsymbol{A}_{\mathrm{H}}$ is then the solution of d problems of the type (2) with $\boldsymbol{E}_i$ chosen consecutively as d particular vectors of some basis of $\mathbb{R}^d$.

3 Fourier-Galerkin method

To find a numerical solution of (2) we use the Fourier-Galerkin method with the FFT algorithm. This method approximates the solution of (2) by complex trigonometric polynomials, also called the Fourier basis. The resulting linear system is

$$\mathcal{F}_{\mathrm{N}}^{-1}\widehat{\nabla}_{\mathrm{N}}^{*}\mathcal{F}_{\mathrm{N}}\widetilde{\mathsf{A}}\mathcal{F}_{\mathrm{N}}^{-1}\widehat{\nabla}_{\mathrm{N}}\mathcal{F}_{\mathrm{N}}\mathbf{u}_i = -\mathcal{F}_{\mathrm{N}}\widehat{\nabla}_{\mathrm{N}}^{*}\mathcal{F}_{\mathrm{N}}^{-1}\widetilde{\mathsf{A}}\mathsf{E}_{\mathrm{i,N}}, \quad i = 1, \ldots, d \tag{3}$$

where $\mathcal{F}_{\mathrm{N}}$ and $\mathcal{F}_{\mathrm{N}}^{-1}$ are Fourier transform and its inverse. The differential operators of gradient $\widehat{\nabla}_{\mathrm{N}}$ and divergence $\widehat{\nabla}_{\mathrm{N}}^{*}$ are naturally applied on trigonometric polynomials in the Fourier space. The tensor $\widetilde{\mathsf{A}} \in \mathbb{R}^{d\times d\times \mathrm{N}\times\mathrm{N}}$ is a block diagonal tensor with components of material matrix $\boldsymbol{A}(\boldsymbol{x})$. Finally $\mathbf{u}_i \in \mathbb{R}^{d\times\mathrm{N}}$ a is tensor of unknown solution.

4 Approximation by low-rank tensors

The Size (memory requirements) of unknown $\mathbf{u}_i$ in the system (3) is N^d and doubles after applying the gradient operator. The exponential dependency on the dimension causes problems with memory. This issue can be overcome by representation of these fields as low-rank tensors. Beside N and d, the size of these tensors depend on rank-r and fortunately a large class of "natural" tensors can be expressed with low rank-r [1]. In following sections, we present chosen format used for the homogenisation problem. We first present the canonical format is suitable for two dimensional tensors. For higher dimensions we chose generalisation of the canonical format named Tucker format. Both of these formats and their properties are described in [1, 4].

4.1 Canonical format

A canonical approximation of a tensor $\mathrm{v} \in \mathbb{K}^{N_1\times\cdots\times N_d}$ ($\mathbb{K}$ is $\mathbb{R}$ or $\mathbb{C}$) is a sum of r rank-1 tensors. The canonical format is only used for tensors with order 2 ($d = 2$) in this case the representation has the form:

$$\mathrm{v} \approx \widetilde{\mathrm{v}} = \sum_{i=1}^{r} c[i]\mathrm{b}^1[i] \otimes \mathrm{b}^2[i]$$

with $\mathrm{b}^{(j)} \in \mathbb{K}^{r\times N_j}$ stores the basis vectors in spatial direction j and ⊗ denotes tensor product. The memory requirement is $d\mathrm{N}r$, which is linearly dependent on the rank r,N and the dimension [1].

4.2 Tucker format

The decomposition of higher order tensors have many variants. The Tucker format representation is linked to the definition of a tensor subspace $\mathcal{V} = \bigotimes_{j=1}^{d} \mathcal{V}^j$ where $\mathcal{V}^j$ is a subspace of $\mathbb{R}^{N_j}$ generated by basis vectors $\{\mathrm{b}^j[i]i = 1, \ldots, r_j\}$ in the spatial direction j. The Tucker format is then a linear combination of tensor products of all possible combinations of basis vectors in different directions, i.e.

$$\mathrm{v} \approx \widetilde{\mathrm{v}} = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \mathrm{c}[i_1, \ldots, i_d] \bigotimes_{j=1}^{d} \mathrm{b}^j[i_j].$$

where the core $\mathrm{c} \in \mathbb{R}^{\boldsymbol{r}}$ is a tensor of order d. The canonical format is then a special form of the Tucker format with a diagonal core. The memory requirement is $d\mathrm{N}r + r^d$, where the size of the core is exponentially dependent on the dimension [1].

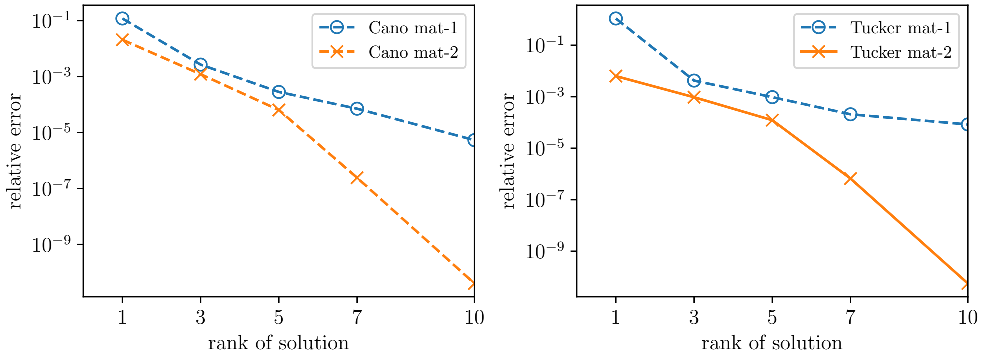


Figure 1: Relative error of $\boldsymbol{A}_{\mathrm{H}}$ low-rank approximation for canonical format in 2D (left) and Tucker format in 3D (right).

5 Numerical experiment and results

The homogenisation scheme (3) equipped with the canonical and Tucker tensors were tested on two microstructures. One with a rigid square inclusion representing a discontinuous material matrix $\boldsymbol{A}_{\square}(\boldsymbol{x})$ and second with a smooth material matrix $\boldsymbol{A}_{S}(\boldsymbol{x})$. For both formats, canonical and Tucker, we set rank $r = \{1, 3, 5, 7, 10\}$ and compute the solutions for different discretization N^d number of points.

Approximation properties of scheme (3) with low-rank tensors is shown in Fig.(1). As an indicator of the approximation quality we chose the relative error of the homogenised material property. The correct homogenised matrix $\boldsymbol{A}_{\mathrm{H}}$ was computed with full tensors and corresponding discretization. We can see, in Fig.(1), that with increasing rank-r we obtain better approximation of full solution. It was also observed that the method converges faster for smooth material.

As mentioned above, the most interesting part is the method converges faster memory consumption during solution. With the memory efficiency we mean the memory consumption compared to the full tensor solution. In Fig.(2) you can see the memory efficiency as a function of the solution rank r. An important observation is that memory savings grows with increasing number of discretization points N.

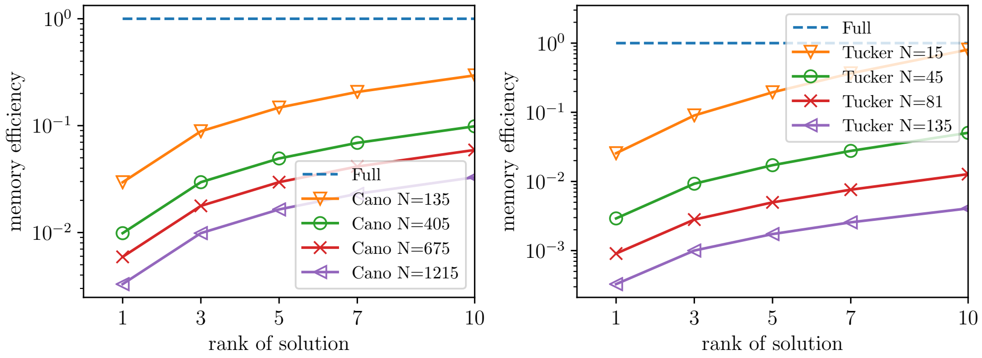


Figure 2: Memory efficiency of low-rank solutions compared to full tensor solution. Canonical tensor for $2D$ on left and Tucker for $3D$.

6 Conclusion

This work is focused on acceleration of the Fourier–Galerkin method using low-rank approximations for problems of numerical homogenisations. The complexity of full computation is based on FFT algorithm which is also very natural for low-rank formats as the d-dimensional the FFT is transformed into the series of one-dimensional FFTs. We consider the canonical format in 2D and Tucker format in 3D. The main result is that low-rank approximations lead to a significant memory and computational reduction. Since the low-rank approximation provides a significant memory reduction it can allow to solve the problems on a finer grid compared to the full solution. It means that with fixed memory demands the low-rank approximation technique can provide better accuracy than full solutions.

Acknowledgement: This work has been supported by the Grant agency of the Czech Republic under the No. 17-04150J.

References

[1] W. Hackbusch: *Tensor Spaces and Numerical Tensor Calculus*, Springer Verlag Berlin Heidelberg New York, 2012.

[2] J. Vondřejc, J. Zeman, I. Marek: *An FFT-based Galerkin method for homogenization of periodic media*, Comput. & Math. with Appl., 2014, pp 156–173.

[3] H. Moulinec, P. Suquet: *A fast numerical method for computing the linear and nonlinear mechanical properties of composites*, Comptes rendus l'Académie des Sci. Série II, Mécanique, Phys. Chim. Astron., 1994, pp. 1417–1423.

[4] I. Oseledets, E. Tyrtyshnikov: *TT-cross approximation for multidimensional arrays*, Linear Algebra Appl.,North-Holland, 2010, pp. 0–88.

Winter school lectures

E. Carson:
High-performance variants of Krylov subspace methods

J. Haslinger:
An introduction to extended finite element methods

M. Plešinger:
On the way from matrix to tensor computations

T. Vejchodský:
Guaranteed eigenvalue bounds for elliptic partial differential operators

High-Performance Variants of Krylov Subspace Methods: I/II

Erin C. Carson
Katedra numerické matematiky, Matematicko-fyzikální fakulta, Univerzita Karlova

SNA '19
January 21-25, 2019

This research was partially supported by OP RDE project No. CZ.02.2.69/0.0/0.0/16_027/0008495

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Lecture Outline

- Parallel computers and performance modeling
  - Architecture trends
- Krylov subspace methods
  - Properties
  - Performance bottlenecks at scale
- High-performance variants of Krylov subspace methods
  - Early approaches
  - Pipelined methods
  - s-step methods
- Practical implementation issues and challenges

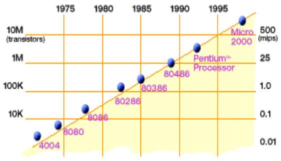
Computational and Data Science at Scale

- Why are we interested in solving larger and larger problems?
- Enables new frontiers in computational science and engineering

  ⇒ Finer-grained simulation, over longer time scales, processing huge amounts of available data

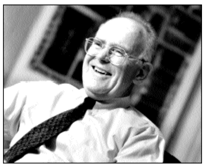
  - Atmosphere, Earth, Environment
  - Physics - applied, nuclear, particle, fusion, photonics
  - Bioscience, Biotechnology, Genetics
  - Chemistry, Molecular Sciences
  - Geology, Seismology
  - Electrical Engineering, Circuit Design, Microelectronics
  - Mechanical Engineering - from prosthetics to spacecraft
- Also industrial and commercial interests
  - "Big Data", databases, data mining
  - Artificial Intelligence (AI)
  - Medical imaging and diagnosis
  - Pharmaceutical design
  - Financial and economic modeling
  - Advanced graphics and virtual reality
  - Oil exploration

Technology Trends: Microprocessor Capacity



2X transistors/Chip Every 1.5 years
“Moores Law”

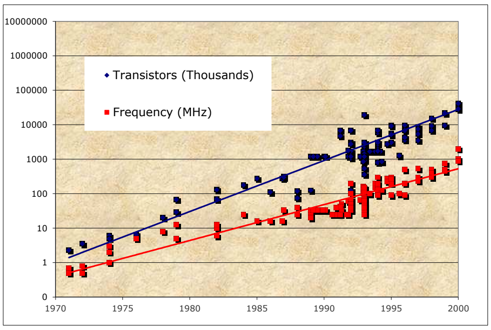
Microprocessors have become smaller, denser, and more powerful.



Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

Slide source: Jack Dongarra

Microprocessor Transistors / Clock (1970-2000)



Slide source: Kathy Yelick

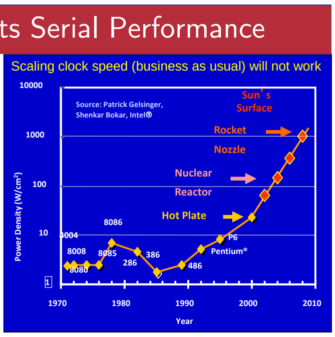
Historical Impact of Device Shrinkage

- What happens when the feature size (transistor size) shrinks by a factor of x?
- Clock rate goes up by x because wires are shorter
  - actually less than x, because of power consumption
- Transistors per unit area goes up by x^2
- Die size has also increased
  - typically another factor of ~ x
- Raw computing power of the chip goes up by ~ x^4 !
  - typically x^3 is devoted to either on-chip
    - parallelism: hidden parallelism such as ILP
    - locality: caches
- So most programs x^3 times faster, without changing them

Slide source: Kathy Yelick

Power Density Limits Serial Performance

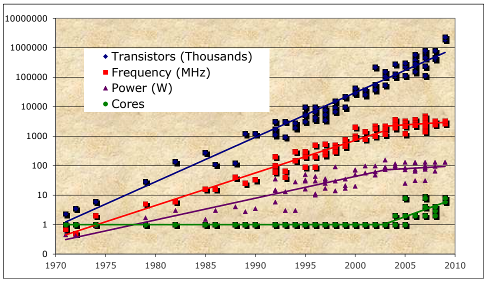
- Concurrent systems are more power efficient
  - Dynamic power is proportional to $V^2 f C$
  - Increasing frequency (f) also increases supply voltage (V) → cubic effect
  - Increasing cores increases capacitance (C) but only linearly
  - Save power by lowering clock speed



- High performance serial processors waste power
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- More transistors, but not faster serial processors

Slide source: Kathy Yelick

Revolution in Processors



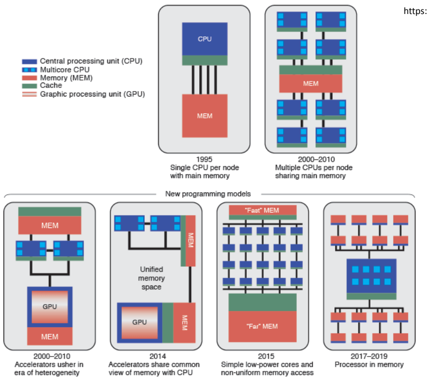
- Chip density is continuing increase ~2x every 2 years
- Clock speed is not
- Number of processor cores may double instead
- Power is under control, no longer growing

Slide source: Kathy Yelick

Parallel Computer Architectures

- Takeaway*: all* programs that need to run faster will have to become parallel programs
- Since mid 2000s - not only are fastest computers parallel, but nearly *all* computers are parallel

Evolution of HPC Nodes



Central processing unit (CPU)
Multicore CPU
Memory (MEM)
Cache
Graphic processing unit (GPU)
CPU
MEM
1995
Single CPU per node with main memory
MEM
2000–2010
Multiple CPUs per node sharing main memory
New programming models
MEM
GPU
MEM
2000–2010
Accelerators usher in era of heterogeneity
MEM
Unified memory space
GPU
MEM
2014
Accelerators share common view of memory with CPU
"Fast" MEM
"Far" MEM
2015
Simple low-power cores and non-uniform memory access
2017–2019
Processor in memory

https://str.llnl.gov/march-2015/still

HPC Architectures Today

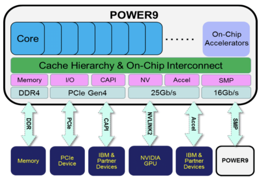
Summit (Oak Ridge National Lab, Tennessee)

- current #1 on the TOP500

HPC Architectures Today

One Processor: 22 SIMD processing cores, on-chip accelerators

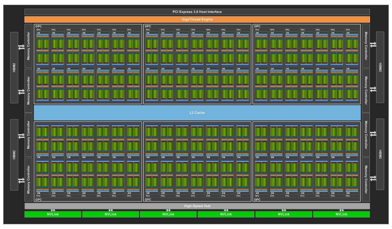
- Each core supports 4 hardware threads
- Each core has separate L1 cache; pairs of cores share L2 and L3 cache



https://www.olcf.ornl.gov/wp-content/uploads/2018/12/summit_workshop_thompto.pdf

HPC Architectures Today

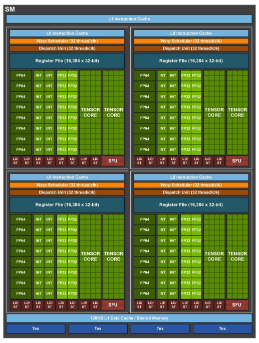
One GPU (NVIDIA V100): 80 streaming multiprocessors (SMs), 16 GB of high-bandwidth memory (HBM2), 6 MB L2 cache shared by SMs



https://www.olcf.ornl.gov/for-users/system-user-guides/summit/summit-user-guide/#nvidia-v100-gpus

HPC Architectures Today

One SM:
32 FP64 (double-precision) cores,
64 FP32 (single-precision) cores,
64 INT32 cores,
8 tensor cores,
128-KB shared memory/L1 cache

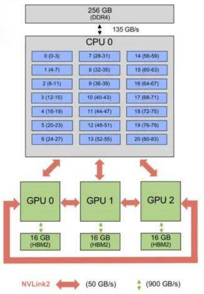


SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
FP64
INT
FP32
TENSOR CORE
LD/ST
SFU
128KB L1 Data Cache / Shared Memory
Tex

https://www.olcf.ornl.gov/for-users/system-user-guides/summit/summit-user-guide/#nvidia-v100-gpus

HPC Architectures Today

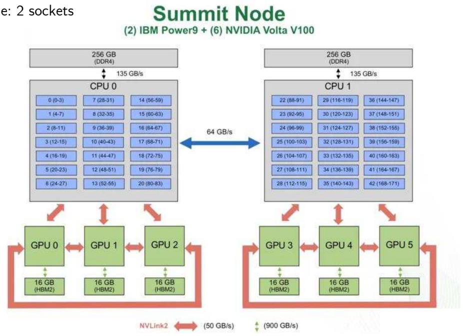
One Socket: 1 CPU, 3 GPUs



https://www.olcf.ornl.gov/for-users/system-user-guides/summit/summit-user-guide

HPC Architectures Today

One Node: 2 sockets



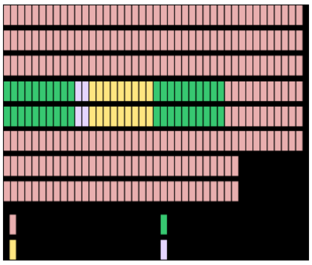
https://www.olcf.ornl.gov/for-users/system-user-guides/summit/summit-user-guide

HPC Architectures Today

One Rack: 18 nodes

- Dual-rail EDR InfiniBand network with non-blocking fat-tree topology
- Node bandwidth of 23 GB/s

HPC Architectures Today



https://en.wikichip.org/wiki/supercomputers/summit

Designing High-Performance Parallel Algorithms

- To design an efficient parallel algorithm, must first model physical costs --- runtime or energy consumption --- of executing a program on a machine
- Tradeoff:
  - More detailed model: more accurate results for a particular machine, but results may not apply to other machines
  - Less detailed model: results applicable to a variety of machines, but may not be accurate for any
    - but abstracting machine details can still give us a general sense of an efficient implementation

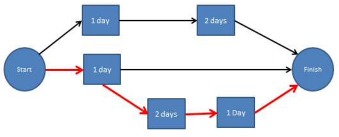
Performance Modeling: Latency-Bandwidth Model

A simplified runtime model:

- Time to perform a floating point operation: γ
- Time to move a message of n words: $\alpha + \beta n$
  - α = latency (seconds), β = 1/bandwidth (seconds/word)

$$\text{Runtime} = \gamma\ (\#\ \text{flops}) + \beta\ (\#\ \text{words}) + \alpha\ (\#\ \text{msgs})$$

#flops,words,msgs are counted along a critical path in the schedule:



Critical Path = 4 Days

Performance Modeling: Latency-Bandwidth Model

γ is per-flop:

- To improve: more parallelism (no longer increase clock frequency)

β is per-word:

- Models bandwidth: maximum amount of data that can be in-flight simultaneously
- To improve: add more ports/wires/etc.

α is per-message and independent of message size

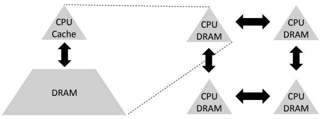
- Models latency: time for data to travel across machine
- Difficult to improve, due to fundamental limits (speed of light, atomic radius,...)

"Bandwidth is money, but latency is physics"

Exascale System Projections

	Today's Systems	Predicted Exascale Systems*
System Peak	10^{16} flops/s	10^{18} flops/s
Node Memory Bandwidth	10^{2} GB/s	10^{3} GB/s
Interconnect Bandwidth	10^{1} GB/s	10^{2} GB/s
Memory Latency	10^{-7} s	$5 \cdot 10^{-8}$ s
Interconnect Latency	10^{-6} s	$5 \cdot 10^{-7}$ s

*Sources: from P. Beckman (ANL), J. Shalf (LBL), and D. Unat (LBL)



Exascale System Projections

	Today's Systems	Predicted Exascale Systems*	Factor Improvement
System Peak	10^{16} flops/s	10^{18} flops/s	100
Node Memory Bandwidth	10^{2} GB/s	10^{3} GB/s	10
Interconnect Bandwidth	10^{1} GB/s	10^{2} GB/s	10
Memory Latency	10^{-7} s	$5 \cdot 10^{-8}$ s	2
Interconnect Latency	10^{-6} s	$5 \cdot 10^{-7}$ s	2

*Sources: from P. Beckman (ANL), J. Shalf (LBL), and D. Unat (LBL)

- Movement of data (communication) is much more expensive than floating point operations (computation), in terms of both **time** and **energy**
- Gaps will only grow larger
- Reducing time spent moving data/waiting for data will be essential for applications at exascale!

Exascale Computing: The Modern Space Race

- "Exascale": 10^{18} floating point operations per second
  - with maximum energy consumption around 20-40 MWatts
- Advancing knowledge, addressing social challenges, improving quality of life, influencing policy, economic competitiveness

Nothing tends so much to the advancement of knowledge as the application of a new instrument.
- Sir Humphry Davy

- Large investment in HPC worldwide

- Technical challenges at all levels

hardware to algorithms to applications

An Exaflop of what?

- When will victory be declared?
  - When a supercomputer reaches exaflop performance on the LINPACK benchmark (TOP500)
    - Solving dense $Ax = b$ using Gaussian elimination with partial pivoting
  - Summit supercomputer has already exceeded exaflop performance for a certain genomics code (https://www.olcf.ornl.gov/2018/06/08/genomics-code-exceeds-exaops-on-summit-supercomputer/)

- Does that mean we are done?
- LINPACK benchmark is typically a compute-bound problem ("BLAS-3")
- Not a good indication of performance for a large number of scientific applications!
  - Lots of remaining work even after exascale performance is achieved
  - Has led to incorporation of other benchmarks into the TOP500 ranking
    - e.g., HPCG: Solving sparse $Ax = b$ iteratively using the conjugate gradient method

Krylov subspace methods

- Linear systems $Ax = b$, eigenvalue problems, singular value problems, least squares, etc.
- Best for: A large & very sparse, stored implicitly, or only approximation needed

- Krylov Subspace Method is a projection process onto the Krylov subspace

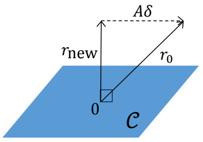
$$\mathcal{K}_i(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \dots, A^{i-1} r_0\}$$

where A is an $N \times N$ matrix and $r_0 = b - Ax_0$ is a length-N vector

- In each iteration,
  - Add a dimension to the Krylov subspace
    - Forms nested sequence of Krylov subspaces

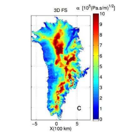
$$\mathcal{K}_1(A, r_0) \subset \mathcal{K}_2(A, r_0) \subset \cdots \subset \mathcal{K}_i(A, r_0)$$

  - Orthogonalize (with respect to some $\mathcal{C}_i$)
  - Select approximate solution $x_i \in x_0 + \mathcal{K}_i(A, r_0)$ using $r_i = b - Ax_i \perp \mathcal{C}_i$

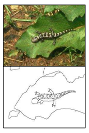


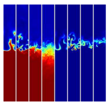
- Ex: Lanczos/Conjugate Gradient (CG), Arnoldi/Generalized Minimum Residual (GMRES), Biconjugate Gradient (BICG), BICGSTAB, GKL, LSQR, etc.

Krylov Subspace Methods in the Wild

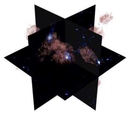

Climate Modeling

Computer Vision

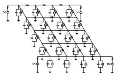


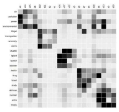

Chemical Engineering

Medical Treatment


Computational Cosmology

Power Grid Modeling




Latent Semantic Analysis

Financial Portfolio Optimization

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_i = \mathcal{K}_i(A, r_0)$

$$r_i \perp \mathcal{K}_i(A, r_0) \quad \Leftrightarrow \quad \|x - x_i\|_A = \min_{z \in x_0 + \mathcal{K}_i(A, r_0)} \|x - z\|_A$$

$$\Rightarrow \quad r_{N+1} = 0$$

Connection with Lanczos

- With $v_1 = r_0 / \|r_0\|$, i iterations of Lanczos produces $N \times i$ matrix $V_i = [v_1, \dots, v_i]$, and $i \times i$ tridiagonal matrix T_i such that

$$AV_i = V_i T_i + \delta_{i+1} v_{i+1} e_i^T, \qquad T_i = V_i^* A V_i$$

- CG approximation x_i is obtained by solving the reduced model

$$T_i y_i = \|r_0\| e_1, \qquad x_i = x_0 + V_i y_i$$

- Connections with orthogonal polynomials, Stieltjes problem of moments, Gauss-Cristoffel quadrature, others (see 2013 book of Liesen and Strakoš)

⇒ CG (and other Krylov subspace methods) are highly nonlinear
  - Good for convergence, bad for ease of finite precision analysis

Implementation of CG

- Standard implementation due to Hestenes and Stiefel (1952) (HSCG)
- Uses three 2-term recurrences for updating x_i, r_i, p_i

```
$r_0 = b - Ax_0$,  $p_0 = r_0$
for $i$ = 1:nmax

    $x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$
    $r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$
    $\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$
    $p_i = r_i + \beta_i p_{i-1}$
end
```

minimizes $\|x - x_i\|_A$ along line
$z(\alpha) = x_{i-1} + \alpha p_{i-1}$

Conjugate Gradient on the World's Fastest Computer

Summit - IBM Power System AC922

➡ current #1 on top500

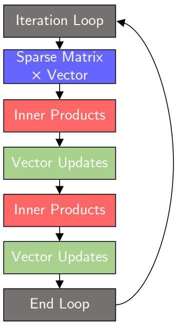
Site:	Oak Ridge National Laboratory
Manufacturer:	IBM
Cores:	2,282,544
Memory:	2,801,664 GB
Processor:	IBM POWER9 22C 3.07GHz
Interconnect:	Dual-rail Mellanox EDR Infiniband
Performance	
Theoretical peak:	187,659 TFlops/s
LINPACK benchmark:	122,300 Tflops/s
HPCG benchmark:	2,926 Tflops/s

LINPACK benchmark
(dense $Ax = b$, direct)
65% efficiency

HPCG benchmark
(sparse $Ax = b$, iterative)
1.5% efficiency

The Conjugate Gradient (CG) Method

```
$r_0 = b - Ax_0, \;\; p_0 = r_0$
for $i$ = 1:nmax
    $\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$
    $x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$
    $r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$
    $\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$
    $p_i = r_i + \beta_i p_{i-1}$
end
```


Iteration Loop

Sparse Matrix × Vector

Inner Products

Vector Updates

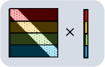
Inner Products

Vector Updates

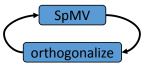
End Loop

Cost Per Iteration

→ Sparse matrix-vector multiplication (SpMV)
- $O(\text{nnz})$ flops
- Must communicate vector entries w/neighboring processors (nearest neighbor MPI collective)



→ Inner products
- $O(N)$ flops
- global synchronization (MPI_Allreduce)
  - all processors must exchange data and wait for *all* communication to finish before proceeding
- Multiple reads/writes to slow memory


SpMV

orthogonalize

Low computation/communication ratio

⇒ Performance is communication-bound

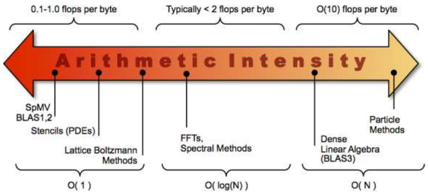


Image source: Sam Williams

Roofline Model Example

Roofline Model (Williams, Waterman, Patterson, 2009)
- Provides estimates of performance for various applications (based on arithmetic intensity) for given machine
- attainable flop/s = min(peak flop/s, peak bandwidth × arithmetic intensity)
- "ceilings" give peak bandwidth or peak flops in absence of possible optimizations

Generally three approaches to improving performance:
- Maximize in-core performance (e.g. get compiler to vectorize)

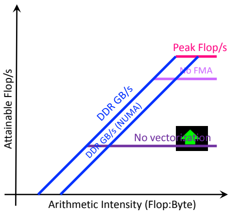


Image source: Sam Williams

Roofline Model Example

Roofline Model (Williams, Waterman, Patterson, 2009)
- Provides estimates of performance for various applications (based on arithmetic intensity) for given machine
- attainable flop/s = min(peak flop/s, peak bandwidth × arithmetic intensity)
- "ceilings" give peak bandwidth or peak flops in absence of possible optimizations

Generally three approaches to improving performance:
- Maximize in-core performance (e.g. get compiler to vectorize)
- Maximize memory bandwidth (e.g. NUMA-aware allocation)

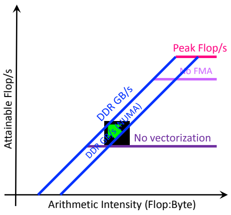


Image source: Sam Williams

Roofline Model Example

Roofline Model (Williams, Waterman, Patterson, 2009)
- Provides estimates of performance for various applications (based on arithmetic intensity) for given machine
- attainable flop/s = min(peak flop/s, peak bandwidth × arithmetic intensity)
- "ceilings" give peak bandwidth or peak flops in absence of possible optimizations

Generally three approaches to improving performance:
- Maximize in-core performance (e.g. get compiler to vectorize)
- Maximize memory bandwidth (e.g. NUMA-aware allocation)
- Minimize data movement (increase AI)

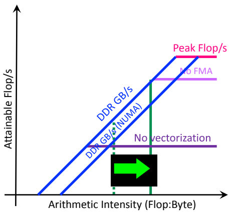


Image source: Sam Williams

Synchronization-reducing variants

Motivated many approaches to reducing synchronization (increasing ratio of computation to communication) in CG:

- Early work: CG with a single synchronization point per iteration
  - 3-term recurrence CG
  - Using modified computation of recurrence coefficients
  - Using auxiliary vectors
- Pipelined Krylov subspace methods
  - Uses modified coefficients and auxiliary vectors to reduce synchronization points to 1 per iteration
  - Modifications also allow decoupling of SpMV and inner products - enables overlapping (MPI non-blocking collectives)
- s-step Krylov subspace methods
  - Compute iterations in blocks of s using a different Krylov subspace basis
  - Enables one synchronization per s iterations

Early approaches to reducing synchronization

- Goal: Reduce the 2 synchronization points per iteration in (HS)CG to 1 synchronization point per iteration
- Compute β_i from α_{i-1} and Ap_{i-1} using relation

$$\|r_i\|^2 = \alpha_{i-1}^2\|Ap_{i-1}\|^2 - \|r_{i-1}\|^2$$

  - Can then also merge the updates of x_i, r_i, and p_i
  - Developed independently by Johnson (1983, 1984), van Rosendale (1983, 1984), Saad (1985)
  - Many other similar approaches

- Could also compute α_{i-1} from β_{i-1}:

$$\alpha_{i-1} = \left(\frac{r_{i-1}^T A r_{i-1}}{r_{i-1}^T r_{i-1}} - \frac{\beta_{i-1}}{\alpha_{i-2}}\right)^{-1}$$

CG with two three-term recurrences (STCG)

- HSCG recurrences can be written as

$$AP_i = R_{i+1}\underline{L}_i, \qquad R_i = P_i U_i$$

we can combine these to obtain a 3-term recurrence for the residuals (STCG):

$$AR_i = R_{i+1}\underline{T}_i, \qquad \underline{T}_i = \underline{L}_i U_i$$

- First developed by Stiefel (1952/53), also Rutishauser (1959) and Hageman and Young (1981)
- Motivated by relation to three-term recurrences for orthogonal polynomials

```
$r_0 = b - Ax_0,\ \ p_0 = r_0,\ \ x_{-1} = x_0,\ \ r_{-1} = r_0,\ \ e_{-1} = 0$
for $i$ = 1:nmax
        $q_{i-1} = \frac{(r_{i-1}, Ar_{i-1})}{(r_{i-1}, r_{i-1})} - e_{i-2}$
        $x_i = x_{i-1} + \frac{1}{q_{i-1}}(r_{i-1} + e_{i-2}(x_{i-1} - x_{i-2}))$
        $r_i = r_{i-1} + \frac{1}{q_{i-1}}(-Ar_{i-1} + e_{i-2}(r_{i-1} - r_{i-2}))$
        $e_{i-1} = q_{i-1}\frac{(r_i, r_i)}{(r_{i-1}, r_{i-1})}$
end
```

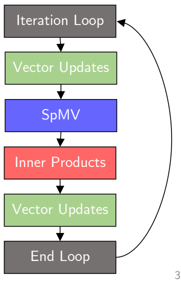
Can be accomplished with a single synchronization point on parallel computers (Strakoš 1985, 1987)

- Similar approach (computing α_i using β_{i-1}) used by D'Azevedo, Eijkhout, Romaine (1992, 1993)

Chronopoulos and Gear's CG (ChG CG)

- Chronopoulos and Gear (1989)
- Looks like HSCG, but very similar to 3-term recurrence CG (STCG)
- Reduces synchronizations/iteration to 1 by changing computation of α_i and using an auxiliary recurrence for Ap_i

```
$r_0 = b - Ax_0,\ p_0 = r_0,$
$s_0 = Ap_0,\ \ \alpha_0 = (r_0, r_0)/(p_0, s_0)$
for $i$ = 1:nmax
        $x_i = x_{i-1} + \alpha_{i-1}p_{i-1}$
        $r_i = r_{i-1} - \alpha_{i-1}s_{i-1}$
        $w_i = Ar_i$
        $\beta_i = \frac{(r_i, r_i)}{(r_{i-1}, r_{i-1})}$
        $\alpha_i = \frac{(r_i, r_i)}{(w_i, r_i) - (\beta_i/\alpha_{i-1})(r_i, r_i)}$
        $p_i = r_i + \beta_i p_{i-1}$
        $s_i = w_i + \beta_i s_{i-1}$
end
```



Pipelined CG (GVCG)

- Pipelined CG of Ghysels and Vanroose (2014)
- Similar to Chronopoulos and Gear approach
  - Uses auxiliary vector $s_i \equiv Ap_i$ and same formula for α_i
- Also uses auxiliary vectors for Ar_i and A^2r_i to remove sequential dependency between SpMV and inner products
  - Allows the use of nonblocking (asynchronous) MPI communication to *overlap* SpMV and inner products
  - Hides the latency of global communications

GVCG (Ghysels and Vanroose 2014)

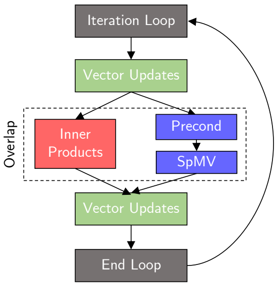
```
$r_0 = b - Ax_0,\ p_0 = r_0$
$s_0 = Ap_0, w_0 = Ar_0, z_0 = Aw_0,$
$\alpha_0 = r_0^T r_0 / p_0^T s_0$
for $i$ = 1:nmax
        $x_i = x_{i-1} + \alpha_{i-1}p_{i-1}$
        $r_i = r_{i-1} - \alpha_{i-1}s_{i-1}$
        $w_i = w_{i-1} - \alpha_{i-1}z_{i-1}$
        $q_i = Aw_i$
        $\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$
        $\alpha_i = \frac{r_i^T r_i}{w_i^T r_i - (\beta_i/\alpha_{i-1}) r_i^T r_i}$
        $p_i = r_i + \beta_i p_{i-1}$
        $s_i = w_i + \beta_i s_{i-1}$
        $z_i = q_i + \beta_i z_{i-1}$
end
```

GVCG (Ghysels and Vanroose 2014)

```
$r_0 = b - Ax_0,\ p_0 = r_0$
[redacted]
$\alpha_0 = r_0^T r_0 / p_0^T s_0$
for $i$ = 1:nmax
        $x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$
        $r_i = r_{i-1} - \alpha_{i-1} s_{i-1}$
        [redacted]
        $q_i = Aw_i$
        $\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$
        $\alpha_i = \frac{r_i^T r_i}{w_i^T r_i - (\beta_i/\alpha_{i-1}) r_i^T r_i}$
        $p_i = r_i + \beta_i p_{i-1}$
        [redacted]
end
```

GVCG (Ghysels and Vanroose 2014)

```
$r_0 = b - Ax_0,\ p_0 = r_0$
$s_0 = Ap_0, w_0 = Ar_0, z_0 = Aw_0,$
$\alpha_0 = r_0^T r_0 / p_0^T s_0$
for $i$ = 1:nmax
        $x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$
        $r_i = r_{i-1} - \alpha_{i-1} s_{i-1}$
        $w_i = w_{i-1} - \alpha_{i-1} z_{i-1}$
        $q_i = Aw_i$
        $\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$
        $\alpha_i = \frac{r_i^T r_i}{w_i^T r_i - (\beta_i/\alpha_{i-1}) r_i^T r_i}$
        $p_i = r_i + \beta_i p_{i-1}$
        $s_i = w_i + \beta_i s_{i-1}$
        $z_i = q_i + \beta_i z_{i-1}$
end
```


Iteration Loop

Vector Updates

Overlap

Inner Products

Precond

SpMV

Vector Updates

End Loop

MPI Non-Blocking Communication

- "Non-blocking" or "asynchronous" collectives available since MPI 3

```
MPI_Iallreduce(...,MPI_Request,...)
// ...other work (SpMV,
preconditioner, etc.)
MPI_Wait(...,MPI_Request)
```

PETSc provides a construct for asynchronous dot-products:

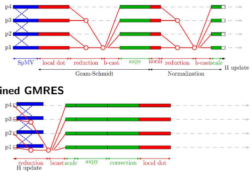
```
VecDotBegin (...,&dot);
PetscCommSplitReductionBegin (comm);
// ...other work
VecDotEnd (...,&dot);
```

call to MPI_Wait

call to MPI_Iallreduce

Classical GMRES

Pipelined GMRES



P. Ghysels, et al. SIAM J. Scientific Computing, 35(1):C48C71, (2013).

Deep Pipelining

- Motivation: want to have perfect overlap of computation of inner products and SpMVs/preconditioner application
- But this depends on the machine, matrix, etc.
- If inner products take much longer than 1 SpMV, do ℓ SpMVs instead
  - ⇒ "deep" pipelined CG with pipeline length ℓ
  - ℓ should be chosen to be the number of SpMV/precond. operations that can be done in the time it takes for one Allreduce
- Deep pipelined GMRES variant
- Deep pipelined CG variant

Available Software

- Implementations in PETSc:
  - KSPPGMRES: pipelined GMRES
  - KSPPIPECG: pipelined CG
  - KSPPIPECR: pipelined CR
  - KSPGROPPCG: Gropp asynchronous variant
  - KSPPIPEBCGS: pipelined BiCGSTAB
  - KSPPIPELCG: deep pipelined CG

Performance of Pipelined CG

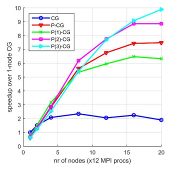


Fig. 5. *Strong scaling experiment on up to 20 nodes (240 processes) for a 5-point stencil 2D Poisson problem with 1.000.000 unknowns. Speedup over single-node classic CG for various pipeline lengths. All methods converged to* $\|r_i\|_2/\|b\|_2 = 1.0e\text{-}5$ *in 1342 iterations.*

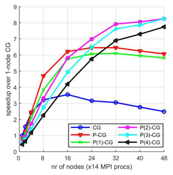


Fig. 6. *Strong scaling experiment on up to 48 nodes (672 processes) for a 5-point stencil 2D Poisson problem with 3.062.500 unknowns. Speedup over single-node classic CG for various pipeline lengths. All methods performed 1500 iterations with* $\|r_i\|_2/\|b\|_2 = 6.3e\text{-}4$.

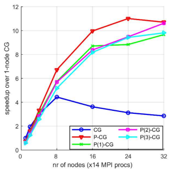


Fig. 7. *Strong scaling experiment on up to 32 nodes (448 processes) for a block Jacobi preconditioned 2D Poisson problem with 3.062.500 unknowns. All methods performed 600 iterations with* $\|r_i\|_2/\|b\|_2 = 1.8e\text{-}4$ *(on 1 node) and* $\|r_i\|_2/\|b\|_2 \leq 9.3e\text{-}4$ *(on 32 nodes).*

20 compute nodes, each with two 6-core Intel Xeon X5660 Nehalem 2:80 GHz processors each (12 cores per node); 4QDR InfiniBand

48 compute nodes, each with two 14-core Intel E5-2680v4, Broadwell generation CPUs; EDR InfiniBand

(Cornelis, Cools, Vanroose, arXiv: 1801.04728, 2018)

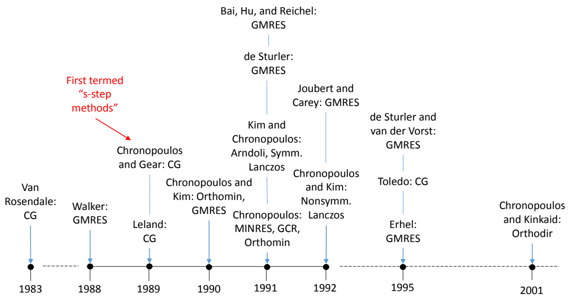
s-step Krylov subspace methods

- Idea: Compute blocks of s iterations at once
  - Compute updates in a different basis
  - Communicate every s iterations instead of every iteration
  - Reduces number of synchronizations per iteration by a factor of s

- An idea rediscovered many times…
- First related work: s-dimensional steepest descent, least squares
  - Khabaza (‘63), Forsythe (‘68), Marchuk and Kuznecov (‘68)
- Flurry of work on s-step Krylov methods in ‘80s/early ‘90s: see, e.g., Van Rosendale (1983); Chronopoulos and Gear (1989)

- Resurgence of interest in recent years due to growing problem sizes; growing relative cost of communication

History of s-step Krylov Subspace Methods


Bai, Hu, and Reichel: GMRES
de Sturler: GMRES
First termed “s-step methods”
Joubert and Carey: GMRES
de Sturler and van der Vorst: GMRES
Kim and Chronopoulos: Arndoli, Symm. Lanczos
Chronopoulos and Gear: CG
Chronopoulos and Kim: Nonsymm. Lanczos
Toledo: CG
Van Rosendale: CG
Walker: GMRES
Chronopoulos and Kim: Orthomin, GMRES
Chronopoulos: MINRES, GCR, Orthomin
Leland: CG
Erhel: GMRES
Chronopoulos and Kinkaid: Orthodir
1983
1988
1989
1990
1991
1992
1995
2001

s-step CG

Key observation: After iteration i, for $j \in \{0,..,s\}$,

$$x_{i+j} - x_i,\ \ r_{i+j},\ \ p_{i+j}\ \ \in\ \ \mathcal{K}_{s+1}(A, p_i) + \mathcal{K}_s(A, r_i)$$

s steps of s-step CG:

Expand solution space s dimensions at once

Compute “basis” matrix $\mathcal{Y}$ such that span$(\mathcal{Y}) = \mathcal{K}_{s+1}(A, p_i) + \mathcal{K}_s(A, r_i)$ according to the recurrence $A\underline{\mathcal{Y}} = \mathcal{Y}\,\mathcal{B}$

Compute inner products between basis vectors in one synchronization

$$\mathcal{G} = \mathcal{Y}^T\mathcal{Y}$$

Compute s iterations of vector updates

Perform s iterations of vector updates by updating coordinates in basis $\mathcal{Y}$:

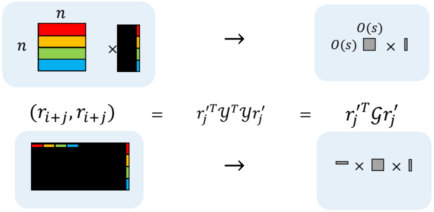
$$x_{i+j} - x_i = \mathcal{Y}x'_j, \qquad r_{i+j} = \mathcal{Y}r'_j, \qquad p_{i+j} = \mathcal{Y}p'_j$$

s-step CG

For s iterations of updates, inner products and SpMVs (in basis $\mathcal{Y}$) can be computed by independently by each processor without communication:

$$Ap_{i+j} \quad = \quad A\mathcal{Y}p'_j \quad = \quad \mathcal{Y}(\mathcal{B}p'_j)$$

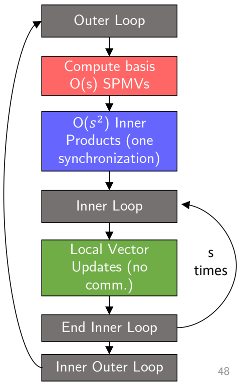
$$(r_{i+j}, r_{i+j}) \quad = \quad r'^{T}_j\mathcal{Y}^T\mathcal{Y}r'_j \quad = \quad r'^{T}_j\mathcal{G}r'_j$$



s-step CG

```
$r_0 = b - Ax_0, p_0 = r_0$
for $k$ = 0:nmax/$s$
        Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}}_k = \mathcal{Y}_k\mathcal{B}_k$ and span$(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$
        $\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$
        $x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$
        for $j = 1{:}\,s$
                $\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$
                $x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$
                $r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$
                $\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$
                $p'_j = r'_j + \beta_{sk+j}p'_{j-1}$
        end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$
end
```


Outer Loop

Compute basis O(s) SPMVs

O(s^2) Inner Products (one synchronization)

Inner Loop

Local Vector Updates (no comm.)

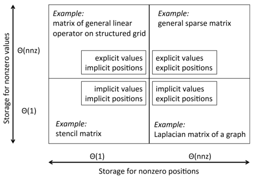
s times

End Inner Loop

Inner Outer Loop

Sparse Matrix Computations

- Sparse Matrix x Vector (SpMV) ($y = Ax$)
  - Very communication-bound; no reuse
  - Lower bound depends on sparsity structure, algorithm used (1D rowwise/colwise, 2D, etc.)
  - Communication cost depends on partition
  - Hypergraph models capture communication dependencies (Catalyurek, Aykanat, 1999)
    - minimize hypergraph cut = minimize words moved


Storage for nonzero values
Θ(nnz)
Θ(1)
Example: matrix of general linear operator on structured grid
explicit values implicit positions
Example: general sparse matrix
explicit values explicit positions
implicit values implicit positions
Example: stencil matrix
implicit values explicit positions
Example: Laplacian matrix of a graph
Θ(1)
Θ(nnz)
Storage for nonzero positions

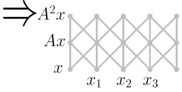
- Repeated SpMVs ($Y = [Ax, A^2x, \dots, A^kx]$)
  - Naive approach: k repeated SpMVs
  - Communication-avoiding approach: "matrix powers kernel"
    - see, e.g., (Demmel, Hoemmen, Mohiyuddin, Yelick, 2008)

SpMV Dependency Graph

$G = (V, E)$ where $V = \{y_0, \dots, y_{n-1}\} \cup \{x_0, \dots, x_{n-1}\}$ and $(y_i, x_j) \in E$ if $A_{ij} \neq 0$

Example: Tridiagonal matrix

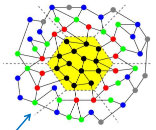
$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{x} \Longrightarrow$$



The Matrix Powers Kernel (Demmel et al., 2007)

Avoids communication:

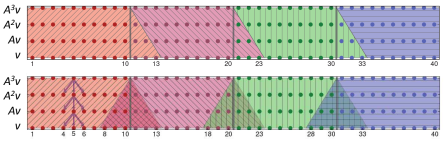
- In serial, by exploiting temporal locality:
  - Reading A, reading vectors
- In parallel, by doing only 1 'expand' phase (instead of s).
- Requires sufficiently low 'surface-to-volume' ratio



black = local elements
red = 1-level dependencies
green = 2-level dependencies
blue = 3-level dependencies

Also works for general graphs!

Tridiagonal Example:

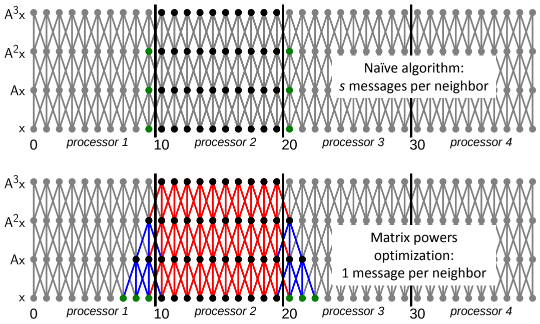


Sequential

Parallel

Parallel Matrix Powers Kernel

Example: tridiagonal matrix, s = 3, n = 40, p = 4



Naïve algorithm:
s messages per neighbor

Matrix powers optimization:
1 message per neighbor

Complexity comparison

Example of parallel (per processor) complexity for s iterations of CG vs. s-step CG for a 2D 9-point stencil:

(Assuming each of p processors owns N/p rows of the matrix and $s \leq \sqrt{N/p}$)

	Flops		Words Moved		Messages	
	SpMV	Orth.	SpMV	Orth.	SpMV	Orth.
Classical CG	$\frac{sN}{p}$	$\frac{sN}{p}$	$s\sqrt{N/p}$	$s \log_2 p$	s	$s \log_2 p$
s-step CG	$\frac{sN}{p}$	$\frac{s^2N}{p}$	$s\sqrt{N/p}$	$s^2 \log_2 p$	1	$\log_2 p$

All values in the table meant in the Big-O sense (i.e., lower order terms and constants not included)

s-step GMRES

Classical GMRES

```
$r_0 = b - Ax_0, v_0 = r_0/\|r_0\|$
for $i = 1: k$
    $w = Av_{i-1}$
    Orthogonalize $w$ against $[v_0, \dots, v_{i-1}]$
    Update vector $v_i$, matrix $H$
end
Use $H, [v_0, \dots, v_k]$ to construct the solution
```

e.g., Modified Gram-Schmidt

s-step GMRES

```
$r_0 = b - Ax_0, v_0 = r_0/\|r_0\|$
for $i = 0: s: k - s$
    Compute $W$ such that span($[v_i, W]$) = $\mathcal{K}_{s+1}(A, v_i)$
    Make $W$ orthogonal against $[v_0, \dots, v_i]$
    Make $W$ orthogonal
    Update $[v_{i+1}, \dots, v_{i+s}]$, matrix $H$
end
Use $H, [v_0, \dots, v_k]$ to construct the solution
```

"matrix powers kernel"

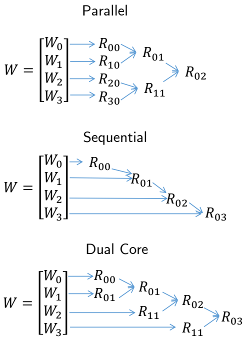
Block Gram-Schmidt

"Tall-Skinny QR"

Tall-Skinny QR (TSQR)

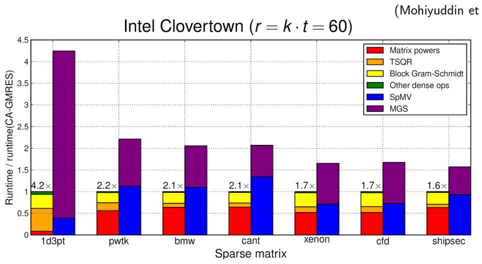
- TSQR: QR factorization of a tall skinny matrix using Householder transformations
- QR decomposition of m x b matrix W, m >> b
  - P processors, block row layout
- **Classic Parallel Algorithm**
  - Compute Householder vector for each column
  - Number of messages ∝ b log P
- **Communication Avoiding Algorithm**
  - Reduction operation, with QR as operator
  - Number of messages ∝ log P

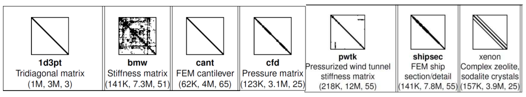
TSQR implementations in Intel MKL library, GNU Scientific Library, ScaLAPACK, Spark



Performance Results

(Mohiyuddin et al, 2009)





Performance and Applications

- Performance studies
  - s-step GMRES on hybrid CPU/GPU arch. (Yamazaki et al., 2014)
  - comparison of s-step and pipelined GMRES (Yamazaki et al., 2017)

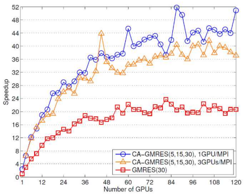


Fig. 6. Parallel Strong Scaling of CA-GMRES and GMRES on 120 distributed GPUs (over GMRES on one GPU), for the G3_Circuit matrix.

- Example applications: s-step BICGSTAB used in
  - combustion, cosmology [Williams, C., et al., IPDPS, 2014]
  - geoscience dynamics [Anciaux-Sedrakian et al., 2016]
  - far-field scattering [Zhang et al., 2016]
  - wafer defect detection [Zhang et al., 2016]

Alternative Approaches

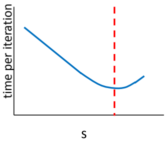
- Enlarged Krylov subspace methods (Grigori, Moufawad, Nataf, 2016)
  - Split vector into t parts based on domain decomposition of A; enlarge Krylov subspace by t dimensions each iteration
    - Faster convergence, more parallelizable
- Combined s-step pipelined methods
  - (ℓ, s)-GMRES (Yamazaki, Hoemmen, Luszczek, Dongarra, 2017)
  - Hybrid approach which combines ideas of s-step and pipelined methods; reduces number of global synchronizations and also overlaps them with other work

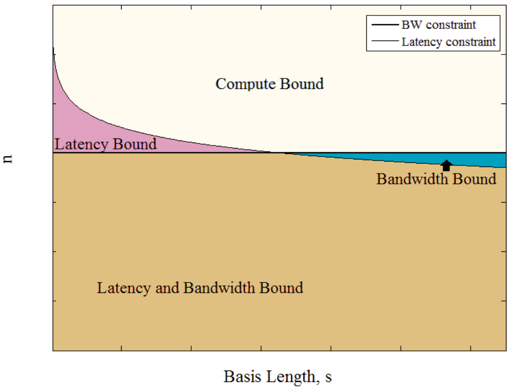
Practical Implementation Challenges

- How to pick parameters? (pipeline depth in pipelined method; s in s-step method)
  - Choice must take into account matrix structure, machine, partition, as well as numerical properties (more on this next time!)
- Preconditioning
  - Must consider overlap in pipelined methods (if enough to overlap with)
  - For s-step, can diminish potential gain from matrix powers kernel if preconditioner is dense (but still win from savings in Allreduce)

Choosing s

- How do we expect communication costs to change as s increases?
- Initially decrease, but at some point, start increasing
  - Point depends on sparsity structure of matrix, partition of matrix, and latency/bandwidth parameters of the machine
- Bandwidth cost can start to dominate
- For s large enough, the extra entries we need go past our neighbors boundaries
  - more messages required -> increased latency cost
- For GMRES, best s for matrix powers may not be best s for TSQR kernel
  - Choice of s requires co-tuning





Lower Bound Tradeoffs for Matrix Powers

- Solomonik, C., Knight, Demmel (2014): Lower bounds on tradeoffs between three basic costs of a parallel algorithm: synchronization, data movement, and computational cost.
- By considering critical path, tradeoffs give lower bounds on the execution time which are dependent on the problem size but independent of the number of processors (assuming homogeneity)
- Theorem: Any parallel execution of an s-dimensional Krylov basis computation for a $(2m+1)^d$-point stencil on a d-dimensional regular mesh requires

$\Omega(m^d b^d s)$ flops, $\Omega(m^d b^{d-1} s)$ words, $\Omega(s/b)$ messages,

for some $b \in \{1, \ldots, s\}$

- Matrix powers kernel attains this lower bound when $n^d/p \geq m^d b^d$ where n^d is # mesh points

Performance Modeling to Estimate Parameters

- Goal: estimate best blocking factor b for matrix powers computation
- Cost model:

Time = $\gamma \times$ flops + $\beta \times$ words moved + $\alpha \times$ # messages

- Choose b to minimize

$$\text{Time} \sim \gamma\, m^d b^d s + \beta m^d b^{d-1} s + \alpha\, s/b$$

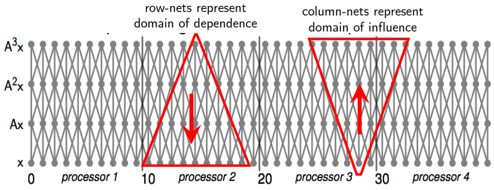
- Latency/BW tradeoff point : $b \sim \dfrac{\alpha^{1/d}}{m\beta^{1/d}}$
- Starting place for parameter selection – to get close to optimal answer, would need more accurate model of time, costs including constants

Matrix Partitioning

- For computing matrix powers (i.e., constructing the basis matrix in s-step methods, we really want to partition the structure of A^s rather than A
  - Analogous to single SpMV, can construct a hypergraph model such that the minimum cut gives a partition with minimum communication volume
- Load balancing
  - The parallel matrix powers kernel involves redundantly computing entries of the vectors on different processors
  - Entries which need to be redundantly computed determined by partition

Hypergraph Partitioning for Matrix Powers

Parallel communication for $V = [x, Ax, A^2x, \ldots, A^s x]$, given a sparse tiling of A **=** Parallel communication for $y = A^s x$, given 1D rowwise layout of A^s (assuming no cancellation and nonzero diagonal)


row-nets represent domain of dependence
column-nets represent domain of influence
A^3x
A^2x
Ax
x
0
processor 1
10
processor 2
20
processor 3
30
processor 4

- "s-level" row- and column-nets encode the structure of A^s
- But expensive to compute (s × Boolean sparse matrix-matrix multiplies)
  - Only worth it if A has particularly irregular sparsity structure (e.g., number of nonzeros per column in A^i grows at various rates) and same matrix will be reused
  - Potential use of randomized algorithms to estimate nnz/column in A^i

Preconditioning for s-step variants

- Preconditioners improve spectrum of system to improve convergence rate
  - E.g., instead of $Ax = b$, solve $M^{-1}Ax = M^{-1}b$, where $M^{-1} \approx A^{-1}$
  - Essential in practice
- In s-step variants, general preconditioning is a challenge
  - Except for very simple cases, ability to exploit temporal locality across iterations is diminished by preconditioning
  - If possible to avoid communication at all, usually necessitates significant modifications to the algorithm
- Tradeoff: speed up convergence, but increase time per iteration due to communication!
  - For each specific app, must evaluate tradeoff between preconditioner quality and sparsity of the system

Preconditioning for s-step KSMs

- Much recent/ongoing work in developing communication-avoiding preconditioned methods
- Many approaches shown to be compatible
  - Diagonal
  - Sparse Approx. Inverse (SPAI) – for s-step BICGSTAB by Mehri (2014)
  - HSS preconditioning (Hoemmen, 2010); for banded matrices (Knight, C., Demmel, 2014); same general technique for any system that can be written as sparse + low-rank
  - Deflation for s-step CG (C., Knight, Demmel, 2014), for s-step GMRES (Yamazaki et al., 2014)
  - CA-ILU(0) – Moufawad and Grigori (2013)
  - Domain decomposition – avoid introducing additional communication by "underlapping" subdomains (Yamazaki et al., 2014)

"Underlapping" Domain Decomposition

(Yamazaki et al., 2014)

- Variant of an additive Schwarz preconditioner, modified to ensure consistent interfaces between the subdomains without additional communication beyond what is required by sparsity structure of A

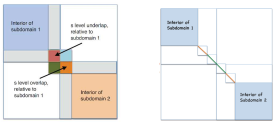
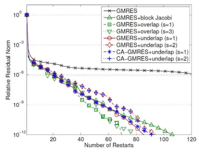
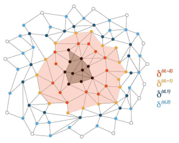


Fig. 8. Matrix Partitioning for the CA Preconditioner for two subdomains. The underlap and the overlap relative to subdomain 1 are shown.



(b) G3_Circuit matrix, with restart = 30.

Fig. 11. Solution Convergence, using Different Domain Decomposition Preconditioners with Local ILU(0)'s on 6 GPUs.



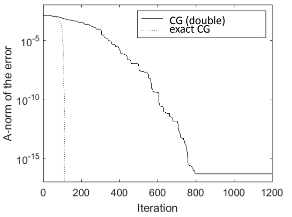
In order to "localize" effects of preconditioner,

- form "interior" by removing s-level "underlap"
- apply "local" preconditioner on "interior"
    - ILU(k), SAI(k), Jacobi, GaussSeidel, etc. on "interior"
- apply diagonal Jacobi on "underlap"

The effects of finite precision

Well-known that roundoff error has two effects:

1. Delay of convergence
    - No longer have exact Krylov subspace
    - Can lose numerical rank deficiency
    - Residuals no longer orthogonal - Minimization of $\|x - x_i\|_A$ no longer exact
2. Loss of attainable accuracy
    - Rounding errors cause true residual $b - Ax_i$ and updated residual r_i deviate!



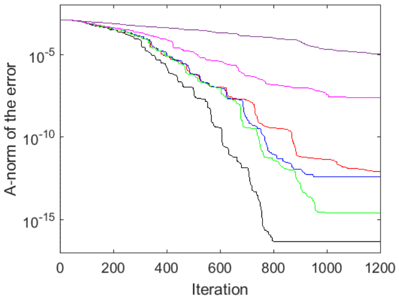
A: bcsstk03 from SuiteSparse,
b: equal components in the eigenbasis of A, $\|b\| = 1$
$N = 112, \kappa(A) \approx 7e6$

Much work on these results for CG; See Meurant and Strakoš (2006) for a thorough summary of early developments in finite precision analysis of Lanczos and CG

Conjugate Gradient method for solving Ax = b
double precision ($\varepsilon = 2^{-53}$)

$$\|x_i - x\|_A = \sqrt{(x_i - x)^T A (x_i - x)}$$

$$\boxed{\begin{aligned} x_i &= x_{i-1} + \alpha_i p_i \\ r_i &= r_{i-1} - \alpha_i A p_i \\ p_i &= r_i + \beta_i p_i \end{aligned}}$$



High-Performance Variants of Krylov Subspace Methods: II/II

Erin C. Carson

Katedra numerické matematiky, Matematicko-fyzikální fakulta, Univerzita Karlova

SNA '19

January 21-25, 2019

This research was partially supported by OP RDE project No. CZ.02.2.69/0.0/0.0/16_027/0008495

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

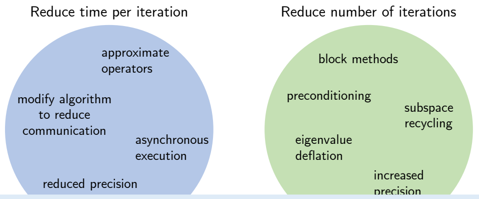
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Review

- Cost of data movement (relative to low computational cost) causes bottlenecks in classical formulations of Krylov subspace methods
- Motivates various approaches
  - Pipelined Krylov subspace methods
    - Add auxiliary recurrences to enable decoupling of inner products and SpMVs; can then be overlapped using non-blocking MPI
    - Effectively hides the cost of synchronization in each iteration
  - s-step Krylov subspace methods
    - Block iterations in groups of s; use block computation of O(s) basis vectors and block orthogonalization
    - Increases temporal locality, allowing asymptotic reduction in number of messages per iteration
  - Many practical implementation details: choosing parameters, preconditioning, etc.
- For certain (e.g., latency-bound) problems, these approaches can reduce the time-per-iteration cost

Improving Performance of Iterative Solvers

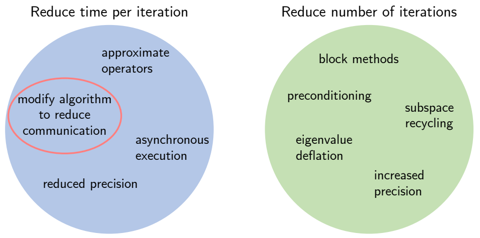
$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$



To minimize runtime, must understand how modifications affect:
1) attainable accuracy 2) convergence rate 3) time per iteration

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$



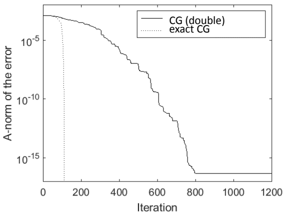
Lecture Outline

- Effects of finite precision in Krylov subspace methods
  - Maximum attainable accuracy
  - Convergence delay
- Existing results for classical Krylov subspace methods
- Results for pipelined and s-step Krylov subspace methods
- Potential remedies for finite precision error in high-performance variants
- Choosing a method in practice
- The future of Krylov subspace methods

The effects of finite precision

Well-known that roundoff error has two effects:

1. Delay of convergence
   - No longer have exact Krylov subspace
   - Can lose numerical rank deficiency
   - Residuals no longer orthogonal - Minimization of $\|x - x_i\|_A$ no longer exact
2. Loss of attainable accuracy
   - Rounding errors cause true residual $b - Ax_i$ and updated residual r_i deviate!



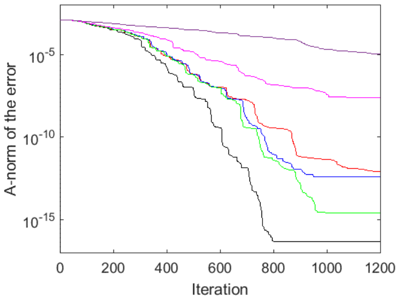
A: bcsstk03 from SuiteSparse,
b: equal components in the eigenbasis of A, $\|b\| = 1$
$N = 112, \kappa(A) \approx 7\text{e}6$

Much work on these results for CG; See Meurant and Strakoš (2006) for a thorough summary of early developments in finite precision analysis of Lanczos and CG

Conjugate Gradient method for solving Ax = b
double precision ($\varepsilon = 2^{-53}$)

$\|x_i - x\|_A = \sqrt{(x_i - x)^T A (x_i - x)}$

$$\boxed{\begin{aligned} x_i &= x_{i-1} + \alpha_i p_i \\ r_i &= r_{i-1} - \alpha_i A p_i \\ p_i &= r_i + \beta_i p_i \end{aligned}}$$



Maximum attainable accuracy

- Accuracy $\|x - \hat{x}_i\|$ generally not computable, *but* $x - \hat{x}_i = A^{-1}(b - A\hat{x}_i)$
- Size of the true residual, $\|b - A\hat{x}_i\|$, used as computable measure of accuracy
- Rounding errors cause the **true residual,** $\boldsymbol{b - A\hat{x}_i}$, and the **updated residual,** $\boldsymbol{\hat{r}_i}$, to deviate

- Writing $b - A\hat{x}_i = \hat{r}_i + b - A\hat{x}_i - \hat{r}_i$,

$$\|b - A\hat{x}_i\| \le \|\hat{r}_i\| + \|b - A\hat{x}_i - \hat{r}_i\|$$

  - As $\|\hat{r}_i\| \to 0$, $\|b - A\hat{x}_i\|$ depends on $\|b - A\hat{x}_i - \hat{r}_i\|$

- Many results on bounding attainable accuracy, e.g.: Greenbaum (1989, 1994, 1997), Sleijpen, van der Vorst and Fokkema (1994), Sleijpen, van der Vorst and Modersitzki (2001), Björck, Elfving and Strakoš (1998) and Gutknecht and Strakoš (2000).

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

and

- Let $f_i \equiv b - A\hat{x}_i - \hat{r}_i$

$$\begin{aligned} f_i &= b - A(\hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \delta x_i) - (\hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \delta r_i) \\ &= f_{i-1} + A\delta x_i + \delta r_i \\ &= f_0 + \textstyle\sum_{m=1}^{i}(A\delta x_m + \delta r_m) \end{aligned}$$

$\|f_i\| \le O(\varepsilon)\sum_{m=0}^{i} N_A\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$ van der Vorst and Ye, 2000

$\|f_i\| \le O(\varepsilon)\|A\|\left(\|x\| + \max_{m=0,\dots,i}\|\hat{x}_m\|\right)$ Greenbaum, 1997

$\|f_i\| \le O(\varepsilon)N_A\||A|\|\|A^{-1}\|\sum_{m=0}^{i}\|\hat{r}_m\|$ Sleijpen and van der Vorst, 1995

Maximum Attainable Accuracy in HPC Variants

- Various synchronization-reducing modifications/variants discussed in Part I
  - Modified recurrence coefficient computation
  - 3-term CG (STCG)
  - Addition of auxiliary recurrences
  - Pipelined CG
  - s-step methods

Modified recurrence coefficient computation

- What is the effect of changing the way the recurrence coefficients (α and β) are computed in HSCG?

- Notice that neither α nor β appear in the bounds on $\|f_i\|$

$$\begin{aligned} f_i &= b - A\hat{x}_i - \hat{r}_i \\ &= b - A(\hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \delta x_i) - (\hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \delta r_i) \end{aligned}$$

- As long as the same $\hat{\alpha}_{i-1}$ is used in updating $\hat{x}_i$ and $\hat{r}_i$,

$$f_i = f_{i-1} + A\delta x_i + \delta r_i$$

  still holds

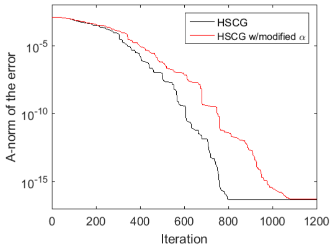
- Rounding errors made in computing $\hat{\alpha}_{i-1}$ do not contribute to the residual gap

- But may change computed $\hat{x}_i$, $\hat{r}_i$, which can affect convergence rate...

Modified recurrence coefficient computation

Example: HSCG with modified formula for α_{i-1}

$$\alpha_{i-1} = \left(\frac{r_{i-1}^T A r_{i-1}}{r_{i-1}^T r_{i-1}} - \frac{\beta_{i-1}}{\alpha_{i-2}}\right)^{-1}$$



Attainable accuracy of STCG

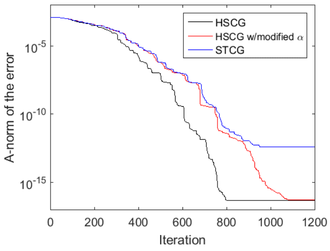
- Analyzed by Gutknecht and Strakoš (2000)
- Attainable accuracy for STCG can be much worse than for HSCG
- Residual gap bounded by sum of local errors PLUS local errors multiplied by factors which depend on

$$\max_{0\le\ell<j\le i}\frac{\|r_j\|^2}{\|r_\ell\|^2}$$

⇒ Large residual oscillations can cause these factors to be large!

⇒ Local errors can be amplified!

STCG



Attainable accuracy of pipelined CG

- What is the effect of adding auxiliary recurrences to the CG method?
- To isolate the effects, we consider a simplified version of a pipelined method
  - Uses same update formulas for α and β as HSCG, but uses additional recurrence for Ap_i

```
$r_0 = b - Ax_0, p_0 = r_0, s_0 = Ap_0$
for $i$ = 1:nmax
        $\alpha_{i-1} = \frac{(r_{i-1},r_{i-1})}{(p_{i-1},s_{i-1})}$
        $x_i = x_{i-1} + \alpha_{i-1}p_{i-1}$
        $r_i = r_{i-1} - \alpha_{i-1}s_{i-1}$
        $\beta_i = \frac{(r_i,r_i)}{(r_{i-1},r_{i-1})}$
        $p_i = r_i + \beta_i p_{i-1}$
        $s_i = Ar_i + \beta_i s_{i-1}$
end
```

Attainable accuracy of simple pipelined CG

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} + \boldsymbol{\delta x_i} \qquad \hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}\,\hat{s}_{i-1} + \boldsymbol{\delta r_i}$$

$$\begin{aligned} f_i &= \hat{r}_i - (b - A\hat{x}_i) \\ &= f_{i-1} - \hat{\alpha}_{i-1}(\hat{s}_{i-1} - A\hat{p}_{i-1}) + \delta r_i + A\delta x_i \\ &= f_0 + \textstyle\sum_{m=1}^{i}(\delta r_m + A\delta x_m) - G_i d_i \end{aligned}$$

where

$$G_i = \hat{S}_i - A\hat{P}_i, \quad d_i = [\hat{\alpha}_0, \dots, \hat{\alpha}_{i-1}]^T$$

Attainable accuracy of simple pipelined CG

$$\|G_i\| \le \frac{O(\varepsilon)}{1 - O(\varepsilon)}\left(\kappa(\hat{U}_i)\|A\|\|\hat{P}_i\| + \|A\|\|\hat{R}_i\|\|\hat{U}_i^{-1}\|\right)$$

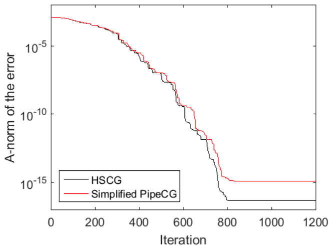
$$\hat{U}_i = \begin{bmatrix} 1 & -\hat{\beta}_1 & 0 & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & 1 & -\hat{\beta}_{i-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \qquad \hat{U}_i^{-1} = \begin{bmatrix} 1 & \hat{\beta}_1 & \dots & \dots & \hat{\beta}_1\hat{\beta}_2\cdots\hat{\beta}_{i-1} \\ 0 & 1 & \hat{\beta}_2 & \dots & \hat{\beta}_2\cdots\hat{\beta}_{i-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & \hat{\beta}_{i-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

$$\beta_\ell\beta_{\ell+1}\cdots\beta_j = \frac{\|r_j\|^2}{\|r_{\ell-1}\|^2}, \qquad \ell < j$$

- Residual oscillations can cause these factors to be large!
- Errors in computed recurrence coefficients can be amplified!

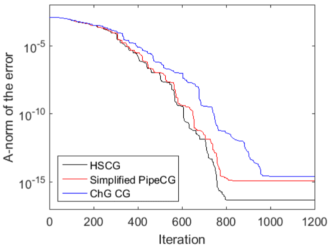
- Very similar to the results for attainable accuracy in the 3-term STCG
- Seemingly innocuous change can cause drastic loss of accuracy

Simple pipelined CG



effect of using auxiliary vector $s_i \equiv Ap_i$

Simple pipelined CG



effect of changing formula for recurrence coefficient α and using auxiliary vector $s_i \equiv Ap_i$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:
$s_i \equiv Ap_i, \quad q_i \equiv M^{-1}Ap_i, \quad u_i \equiv M^{-1}r_i, \quad w_i = AM^{-1}r_i, \quad z_i \equiv AM^{-1}Ap_i$

Computed explicitly: $m_i \equiv M^{-1}w_i \ (\equiv M^{-1}AM^{-1}r_i)$, $v_i = Am_i \ (\equiv AM^{-1}AM^{-1}r_i)$

$$\begin{aligned} \hat{p}_i &= \hat{u}_i + \hat{\beta}_i\hat{p}_{i-1} + \delta_i^p & \hat{x}_{i+1} &= \hat{x}_i + \hat{\alpha}_i\hat{p}_i + \delta_i^x \\ \hat{s}_i &= \widehat{w}_i + \hat{\beta}_i\hat{s}_{i-1} + \delta_i^s & \hat{r}_{i+1} &= \hat{r}_i - \hat{\alpha}_i\hat{s}_i + \delta_i^r \\ \hat{z}_i &= A\widehat{m}_i + \hat{\beta}_i\hat{z}_{i-1} + \delta_i^z & \widehat{w}_{i+1} &= \widehat{w}_i - \hat{\alpha}_i\hat{z}_i + \delta_i^w \\ \hat{q}_i &= \widehat{m}_i + \hat{\beta}_i\hat{q}_{i-1} + \delta_i^q & \hat{u}_{i+1} &= u_i - \hat{\alpha}_i\hat{q}_i + \delta_i^u \end{aligned}$$

$$\begin{aligned} f_{i+1} &= (b - A\hat{x}_{i+1}) - \hat{r}_{i+1} \\ &= f_i - \hat{\alpha}_i\underbrace{(A\hat{p}_i - \hat{s}_i)} - A\delta_i^x - \delta_i^r \end{aligned}$$

$$g_i = \hat{\beta}_i g_{i-1} + \underbrace{(A\hat{u}_{i+1} - \widehat{w}_{i+1})} + A\delta_i^p - \delta_i^s$$

$$h_{i+1} = h_i - \hat{\alpha}_i\underbrace{(A\hat{q}_i - \hat{z}_i)} + A\delta_i^u - \delta_i^w$$

$$j_i = \hat{\beta}_i j_{i-1} + A\delta_i^q - \delta_i^z$$

Attainable Accuracy of Pipelined CG

$$f_{i+1} = f_0 - \sum_{j=0}^{i}\hat{\alpha}_j g_j - \sum_{j=0}^{i}(A\delta_j^x + \delta_j^r)$$

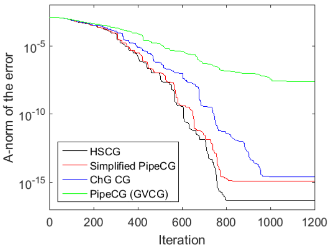
$$g_j = \left(\prod_{k=1}^{j}\hat{\beta}_k\right)g_0 + \sum_{k=1}^{j}\left(\prod_{\ell=k+1}^{j}\hat{\beta}_\ell\right)\boxed{\qquad} + \sum_{k=1}^{j}\left(\prod_{\ell=k+1}^{j}\hat{\beta}_\ell\right)h_k$$

$$h_k = h_0 - \sum_{\ell=0}^{k-1}\hat{\alpha}_\ell j_\ell + \sum_{\ell=0}^{k-1}\boxed{\qquad}$$

$$j_\ell = \left(\prod_{m=1}^{\ell}\hat{\beta}_m\right)j_0 + \sum_{m=1}^{\ell}\left(\prod_{n=m+1}^{\ell}\hat{\beta}_n\right)\boxed{\qquad}$$

Local rounding errors all potentially amplified!

Pipelined CG

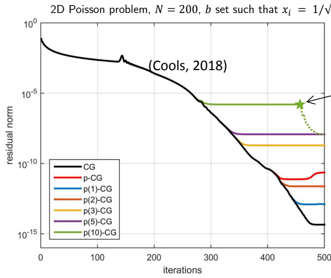


effect of changing formula for recurrence coefficient α and using auxiliary vectors $s_i \equiv Ap_i, w_i \equiv Ar_i, z_i \equiv A^2 r_i$

Effect of Deeper Pipelines

- Deeper pipeline -> effectively adding more auxiliary recurrences
- We expect residual gap to increase with increasing pipeline depth
- Some initial work (Cools, 2018) uses Chebyshev shifts to attempt to stabilize (deep) pipelined CG; but increasing gap is still apparent

2D Poisson problem, $N = 200$, b set such that $x_i \ = \ 1/\sqrt{N}$



(Cools, 2018)

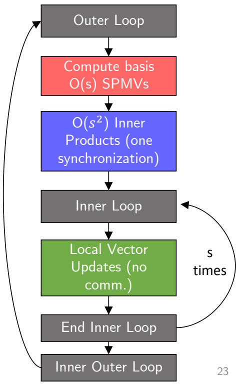
square root breakdown + explicit restart

(Cools, 2018)

s-step CG

```
$r_0 = b - Ax_0, p_0 = r_0$
for $k = 0$:nmax/$s$
    Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}}_k = \mathcal{Y}_k\mathcal{B}_k$ and
        span$(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$
    $\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$
    $x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$
    for $j = 1$: $s$
        $\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$
        $x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$
        $r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$
        $\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$
        $p'_j = r'_j + \beta_{sk+j}p'_{j-1}$
    end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$
end
```



Outer Loop

Compute basis O(s) SPMVs

$O(s^2)$ Inner Products (one synchronization)

Inner Loop

Local Vector Updates (no comm.)

End Inner Loop

Inner Outer Loop

s times

Sources of local roundoff error in s-step CG

Computing the s-step Krylov subspace basis:

$$A\underline{\hat{\mathcal{Y}}}_k = \hat{\mathcal{Y}}_k \mathcal{B}_k + \Delta\mathcal{Y}_k$$

Error in computing s-step basis

Updating coordinate vectors in the inner loop:

$$\hat{x}'_{k,j} = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}$$
$$\hat{r}'_{k,j} = \hat{r}'_{k,j-1} - \mathcal{B}_k\, \hat{q}'_{k,j-1} + \eta_{k,j}$$
$$\text{with} \quad \hat{q}'_{k,j-1} = \mathrm{fl}(\hat{\alpha}_{sk+j-1}\hat{p}'_{k,j-1})$$

Error in updating coefficient vectors

Recovering CG vectors for use in next outer loop:

$$\hat{x}_{sk+j} = \hat{\mathcal{Y}}_k \hat{x}'_{k,j} + \hat{x}_{sk} + \phi_{sk+j}$$
$$\hat{r}_{sk+j} = \hat{\mathcal{Y}}_k \hat{r}'_{k,j} + \psi_{sk+j}$$

Error in basis change

Attainable accuracy of s-step CG

- We can write the gap between the true and updated residuals f in terms of these errors:

$$f_{sk+j} = f_0$$
$$- \sum_{\ell=0}^{k-1} \left[A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s} [A\hat{\mathcal{Y}}_\ell \xi_{\ell,i} + \hat{\mathcal{Y}}_\ell \eta_{\ell,i} - \Delta\mathcal{Y}_\ell \hat{q}'_{\ell,i-1}] \right]$$
$$-A\phi_{sk+j} - \psi_{sk+j} - \sum_{i=1}^{j} [A\hat{\mathcal{Y}}_k \xi_{k,i} + \hat{\mathcal{Y}}_k \eta_{k,i} - \Delta\mathcal{Y}_\ell \hat{q}'_{k,i-1}]$$

- Using standard rounding error results, this allows us to obtain an upper bound on $\|f_{sk+j}\|$.

Attainable accuracy of s-step CG

$f_i \equiv b - A\hat{x}_i - \hat{r}_i$

For CG:

$$\|f_i\| \le \|f_0\| + \varepsilon \sum_{m=1}^{i} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

For s-step CG: $i \equiv sk + j$

$$\|f_{sk+j}\| \le \|f_0\| + \varepsilon c \bar{\Gamma}_k \sum_{m=1}^{sk+j} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

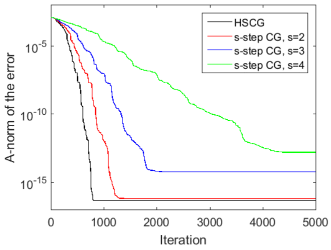
where c is a low-degree polynomial in s, and

$$\bar{\Gamma}_k = \max_{\ell \le k} \Gamma_\ell\,, \quad \text{where} \quad \Gamma_\ell = \|\hat{\mathcal{Y}}_\ell^+\| \cdot \||\hat{\mathcal{Y}}_\ell|\|$$

(see C., 2015)

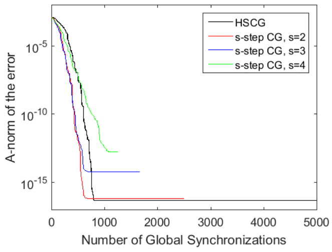
s-step CG

s-step CG with monomial basis ($\mathcal{Y} = [p_i, Ap_i, \ldots, A^s p_i, r_i, Ar_i, \ldots A^{s-1} r_i]$)



Can also use other, more well-conditioned bases to improve convergence rate and accuracy (see, e.g. Philippe and Reichel, 2012).

s-step CG



- Even assuming perfect parallel scalability with s (which is usually not the case due to extra SpMVs and inner products), already at $s = 4$ we are worse than HSCG in terms of number of synchronizations!

"Backwards-like" analysis of Greenbaum

- Anne Greenbaum (1989): finite precision CG with matrix A behaves like exact CG run on a larger matrix $\tilde{A}$ whose eigenvalues lie in tight clusters around the eigenvalues of A
- Based on work of Chris Paige for finite precision Lanczos (1976, 1980):
  - Complete rounding error analysis
  - Computed eigenvalues lie between extreme eigenvalues of A to within a small multiple of machine precision
  - At least one small interval containing an eigenvalue of A is found by the Nth iteration
  - The algorithm behaves as if it used full reorthogonalization until a close eigenvalue approximation is found
  - Loss of orthogonality among basis vectors follows a rigorous pattern and implies that some eigenvalue approximation has converged
- Can we make similar statements for HPC variants?

Roundoff Error in Lanczos vs. s-step Lanczos

Finite precision Lanczos process: (A is $N \times N$ with at most n nonzeros per row)

$$A\hat{V}_m = \hat{V}_m \hat{T}_m + \hat{\beta}_{m+1}\hat{v}_{m+1}e_m^T + \delta\hat{V}_m$$

$$\hat{V}_m = [\hat{v}_1, \dots, \hat{v}_m], \qquad \delta\hat{V}_m = [\delta\hat{v}_1, \dots, \delta\hat{v}_m], \qquad \hat{T}_m = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_2 & & \\ \hat{\beta}_2 & \ddots & \ddots & \\ & \ddots & \ddots & \hat{\beta}_m \\ & & \hat{\beta}_m & \hat{\alpha}_m \end{bmatrix}$$

for $i \in \{1, \dots, m\}$,

$$\|\delta\hat{v}_i\|_2 \le \varepsilon_1 \sigma$$
$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0 \sigma$$
$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$
$$\left|\hat{\beta}_{i+1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_i^2 - \|A\hat{v}_i\|_2^2\right| \le 4i(3\varepsilon_0 + \varepsilon_1)\sigma^2$$

$$\sigma \equiv \|A\|_2$$
$$\theta\sigma \equiv \||A|\|_2$$

Lanczos [Paige, 1976]

$$\varepsilon_0 = O(\varepsilon N)$$
$$\varepsilon_1 = O(\varepsilon n \theta)$$

s-step Lanczos [C., Demmel, 2015]:

$$\varepsilon_0 = O\left(\varepsilon N \boldsymbol{\Gamma}^{\mathbf{2}}\right)$$
$$\varepsilon_1 = O(\varepsilon n \theta \boldsymbol{\Gamma})$$

$$\Gamma = c \cdot \max_{\ell \le k} \|\hat{\mathcal{Y}}_\ell^+\| \, \||\hat{\mathcal{Y}}_\ell|\|$$

The amplification term

- Roundoff errors in s-step variant follow same pattern as classical variant, but amplified by factor of Γ or Γ^2
  - Theoretically confirms empirical observations on importance of basis conditioning (dating back to late '80s)
- Using the definition
$$\Gamma \equiv \Gamma_k = \max_{\ell \le k} \|\mathcal{Y}_\ell^+\| \cdot \||\mathcal{Y}_\ell|\|$$
gives simple, but loose bounds
- What we really need: $\||\mathcal{Y}||y'|\| \le \Gamma\|\mathcal{Y}y'\|$ to hold for the computed basis $\mathcal{Y}$ and coordinate vector y' in every bound.
- Alternate definition of Γ gives tighter bounds; requires light bookkeeping
- Example: for bounds on $\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right|$ and $\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right|$, we can use the definition

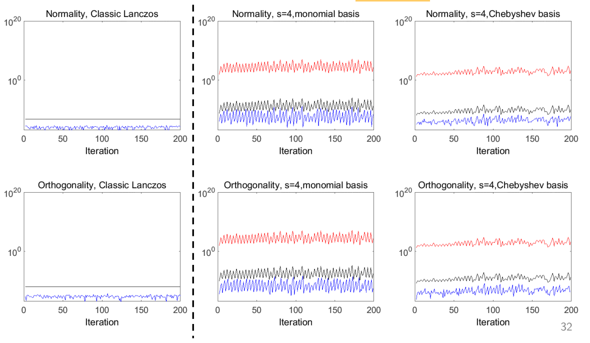
$$\Gamma_{k,j} \equiv \max_{x \in \{\hat{w}'_{k,j}, \hat{u}'_{k,j}, \hat{v}'_{k,j}, \hat{v}'_{k,j-1}\}} \frac{\||\hat{\mathcal{Y}}_k||x|\|}{\|\hat{\mathcal{Y}}_k x\|}$$

Problem: 2D Poisson,
n = 256,
random starting vector

Computed value
Bound
Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$
$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0\sigma$$

$\boldsymbol{s = 4}$

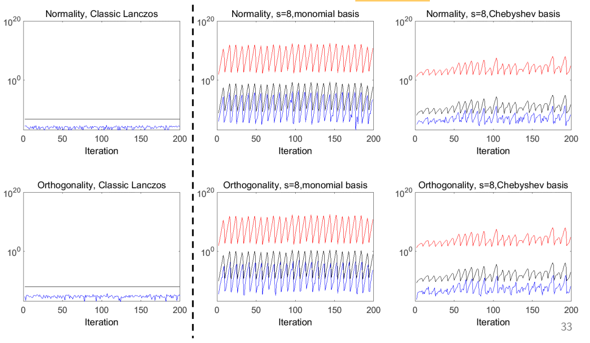


Problem: 2D Poisson,
n = 256,
random starting vector

Computed value
Bound
Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$
$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0\sigma$$

$\boldsymbol{s = 8}$

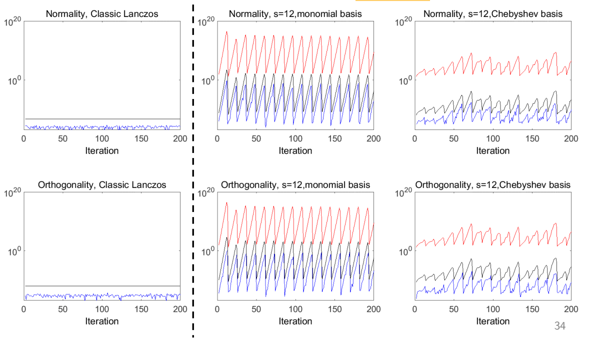


Problem: 2D Poisson,
n = 256,
random starting vector

Computed value
Bound
Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$
$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0\sigma$$

$\boldsymbol{s = 12}$



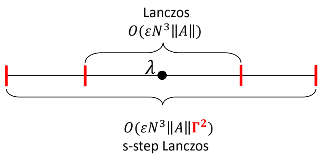
Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

$$\left(\Gamma = c \cdot \max_{\ell \le k} \|\hat{\mathcal{Y}}_\ell^+\| \, \||\hat{\mathcal{Y}}_\ell|\|\right)$$

$$\Gamma \le \left(24\varepsilon(N + 11s + 15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

- Bounds on accuracy of Ritz values depend on Γ^2



Lanczos
$O(\varepsilon N^3\|A\|)$

λ

$O(\varepsilon N^3\|A\|\boldsymbol{\Gamma}^{\mathbf{2}})$
s-step Lanczos

Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

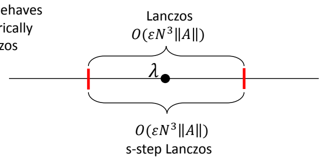
$\left(\Gamma = c \cdot \max_{\ell \le k} \|\hat{\mathcal{Y}}_\ell^+\| \, \||\hat{\mathcal{Y}}_\ell|\|\right)$

$$\Gamma \le \left(24\varepsilon(N + 11s + 15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

- Bounds on accuracy of Ritz values depend on Γ^2

If $\mathbf{\Gamma \approx 1}$:

s-step Lanczos behaves the same numerically as classical Lanczos



Lanczos

$O(\varepsilon N^3\|A\|)$

λ

$O(\varepsilon N^3\|A\|)$

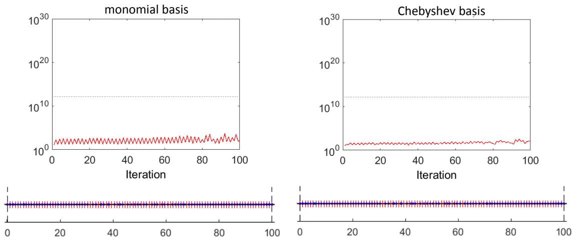
s-step Lanczos

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$s = 2$

Top plots:

🟥	Computed $\Gamma_{k,j}^2$
⋯	$(24(\varepsilon(n + 11s + 15))^{-1}$



Bottom Plots:

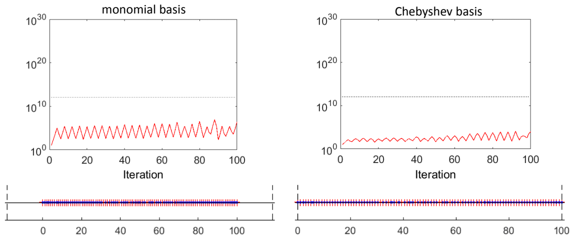
+	True eigenvalues	●	Computed Ritz values
┆	Bounds on range of computed Ritz values		

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$s = 4$

Top plots:

🟥	Computed $\Gamma_{k,j}^2$
⋯	$(24(\varepsilon(n + 11s + 15))^{-1}$



Bottom Plots:

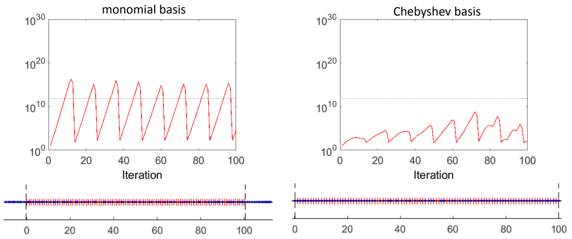
+	True eigenvalues	●	Computed Ritz values
┆	Bounds on range of computed Ritz values		

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$s = 12$

Top plots:

🟥	Computed $\Gamma_{k,j}^2$
⋯	$(24(\varepsilon(n + 11s + 15))^{-1}$

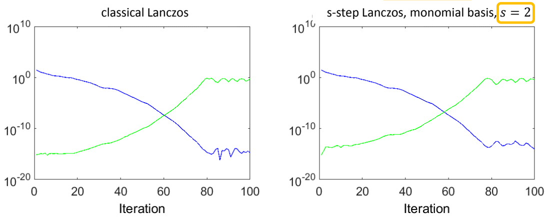


Bottom Plots:

+	True eigenvalues	●	Computed Ritz values
┆	Bounds on range of computed Ritz values		

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \le 7 \times 10^2$



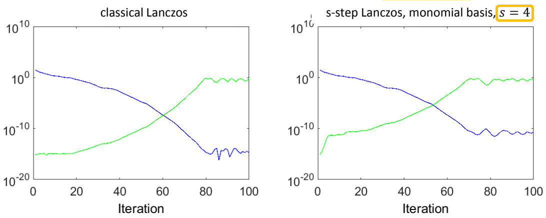
🟩	$\max_i \lvert z_i^{(m)T} \hat{v}_{m+1} \rvert$
🟦	$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Measure of loss of orthogonality

Measure of Ritz value convergence

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \le 3 \times 10^3$



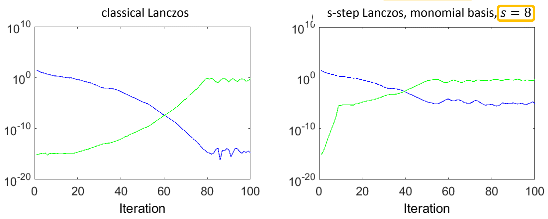
🟩	$\max_i \lvert z_i^{(m)T} \hat{v}_{m+1} \rvert$
🟦	$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Measure of loss of orthogonality

Measure of Ritz value convergence

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \leq 2 \times 10^6$



Measure of Ritz value convergence

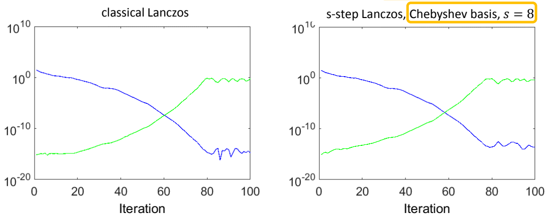
Measure of loss of orthogonality

$\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \leq 2 \times 10^3$



Measure of Ritz value convergence

Measure of loss of orthogonality

$\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Towards understanding convergence delay

- Coefficients α and β (related to entries of T_i) determine distribution functions $\omega^{(i)}(\lambda)$ which approximate distribution function $\omega(\lambda)$ determined by inputs A, b, x_0 in terms of the ith Gauss-Christoffel quadrature
- CG method = matrix formulation of Gauss-Christoffel quadrature (see, e.g., [Liesen & Strakoš, 2013])
- A-norm of CG error for $f(\lambda) = \lambda^{-1}$ given as scaled quadrature error

$$\int \lambda^{-1} d\omega(\lambda) = \sum_{\ell=1}^{i} \omega_\ell^{(i)} \left\{\theta_\ell^{(i)}\right\}^{-1} + \frac{\|x - x_i\|_A^2}{\|r_0\|^2}$$

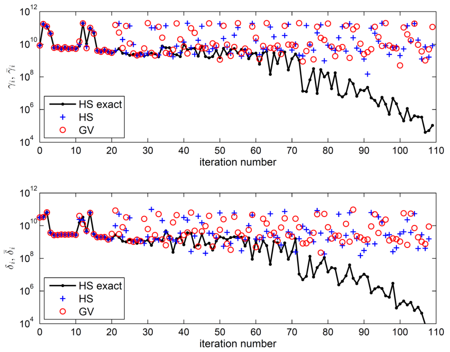
- For particular CG implementation, can the computed $\widehat{\omega}^{(i)}(\lambda)$ be associated with some distribution function $\widehat{\omega}(\lambda)$ related to the distribution function $\omega(\lambda)$, i.e.,

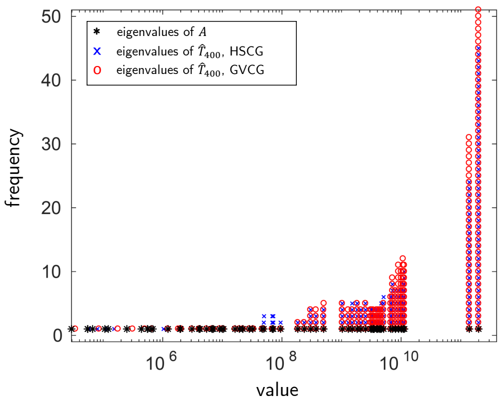
$$\int \lambda^{-1} d\omega(\lambda) \approx \int \lambda^{-1} d\widehat{\omega}(\lambda) = \sum_{\ell=1}^{i} \widehat{\omega}_\ell^{(i)} \left\{\hat{\theta}_\ell^{(i)}\right\}^{-1} + \frac{\|x - \hat{x}_i\|_A^2}{\|r_0\|^2} + F_i$$

where F_i is small relative to error term?

- For classical CG, yes; proved by Greenbaum [1989]
- For pipelined CG and s-step CG, THOROUGH ANALYSIS NEEDED!

Differences in entries γ_i, δ_i in Jacobi matrices T_i in HSCG vs. GVCG (matrix bcsstk03)

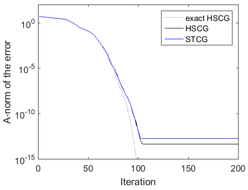


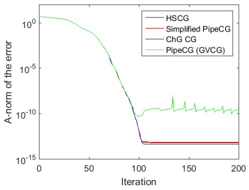


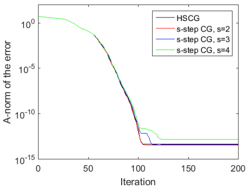
A different problem...

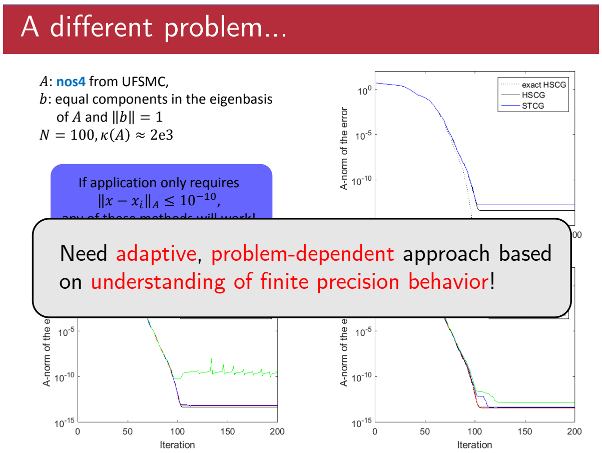
A: nos4 from UFSMC,
b: equal components in the eigenbasis of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2e3$

If application only requires $\|x - x_i\|_A \leq 10^{-10}$, any of these methods will work!







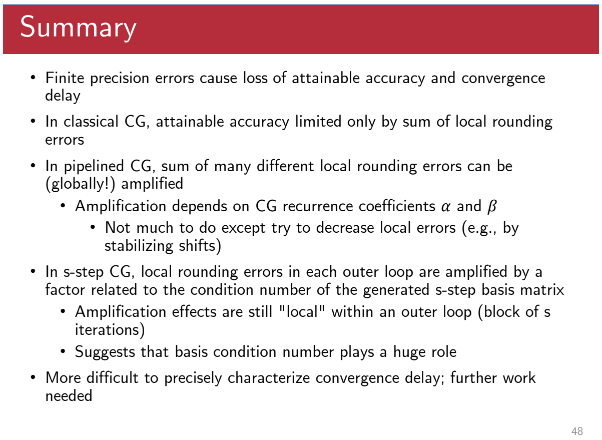


A different problem...

A: **nos4** from UFSMC,
b: equal components in the eigenbasis of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2e3$

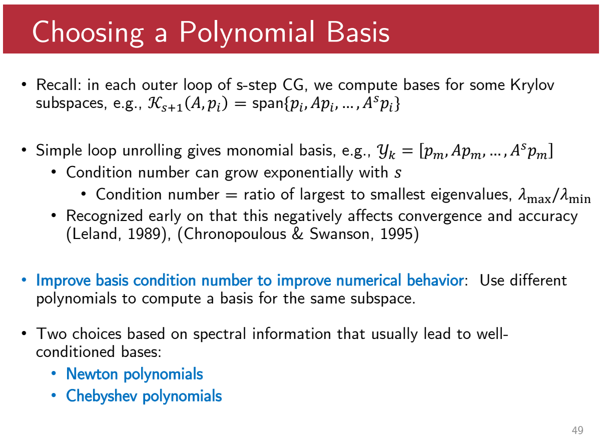
If application only requires $\|x - x_i\|_A \leq 10^{-10}$, any of these methods will work!

Need adaptive, problem-dependent approach based on understanding of finite precision behavior!



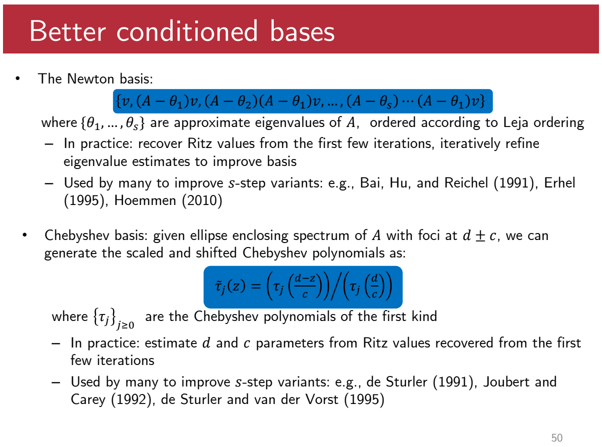
Summary

- Finite precision errors cause loss of attainable accuracy and convergence delay
- In classical CG, attainable accuracy limited only by sum of local rounding errors
- In pipelined CG, sum of many different local rounding errors can be (globally!) amplified
  - Amplification depends on CG recurrence coefficients α and β
    - Not much to do except try to decrease local errors (e.g., by stabilizing shifts)
- In s-step CG, local rounding errors in each outer loop are amplified by a factor related to the condition number of the generated s-step basis matrix
  - Amplification effects are still "local" within an outer loop (block of s iterations)
  - Suggests that basis condition number plays a huge role
- More difficult to precisely characterize convergence delay; further work needed



Choosing a Polynomial Basis

- Recall: in each outer loop of s-step CG, we compute bases for some Krylov subspaces, e.g., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \ldots, A^s p_i\}$
- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \ldots, A^s p_m]$
  - Condition number can grow exponentially with s
    - Condition number = ratio of largest to smallest eigenvalues, $\lambda_{\max}/\lambda_{\min}$
  - Recognized early on that this negatively affects convergence and accuracy (Leland, 1989), (Chronopoulous & Swanson, 1995)
- Improve basis condition number to improve numerical behavior: Use different polynomials to compute a basis for the same subspace.
- Two choices based on spectral information that usually lead to well-conditioned bases:
  - Newton polynomials
  - Chebyshev polynomials



Better conditioned bases

- The Newton basis:

$$\{v, (A-\theta_1)v, (A-\theta_2)(A-\theta_1)v, \ldots, (A-\theta_s)\cdots(A-\theta_1)v\}$$

where $\{\theta_1, \ldots, \theta_s\}$ are approximate eigenvalues of A, ordered according to Leja ordering

– In practice: recover Ritz values from the first few iterations, iteratively refine eigenvalue estimates to improve basis

– Used by many to improve s-step variants: e.g., Bai, Hu, and Reichel (1991), Erhel (1995), Hoemmen (2010)

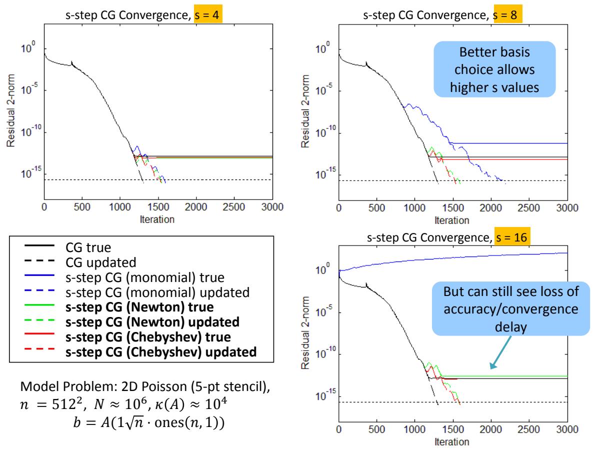
- Chebyshev basis: given ellipse enclosing spectrum of A with foci at $d \pm c$, we can generate the scaled and shifted Chebyshev polynomials as:

$$\tilde{\tau}_j(z) = \left(\tau_j\left(\frac{d-z}{c}\right)\right) \Big/ \left(\tau_j\left(\frac{d}{c}\right)\right)$$

where $\{\tau_j\}_{j\geq 0}$ are the Chebyshev polynomials of the first kind

– In practice: estimate d and c parameters from Ritz values recovered from the first few iterations

– Used by many to improve s-step variants: e.g., de Sturler (1991), Joubert and Carey (1992), de Sturler and van der Vorst (1995)



s-step CG Convergence, s = 4

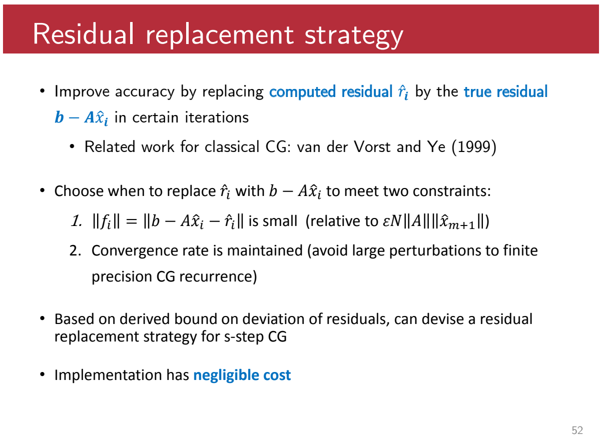
s-step CG Convergence, s = 8

s-step CG Convergence, s = 16

Better basis choice allows higher s values

But can still see loss of accuracy/convergence delay

Model Problem: 2D Poisson (5-pt stencil),
$n = 512^2,\ N \approx 10^6, \kappa(A) \approx 10^4$
$b = A(1\sqrt{n} \cdot \text{ones}(n,1))$



Residual replacement strategy

- Improve accuracy by replacing computed residual $\hat{r}_i$ by the true residual $b - A\hat{x}_i$ in certain iterations
  - Related work for classical CG: van der Vorst and Ye (1999)
- Choose when to replace $\hat{r}_i$ with $b - A\hat{x}_i$ to meet two constraints:
  1. $\|f_i\| = \|b - A\hat{x}_i - \hat{r}_i\|$ is small (relative to $\varepsilon N \|A\| \|\hat{x}_{m+1}\|$)
  2. Convergence rate is maintained (avoid large perturbations to finite precision CG recurrence)
- Based on derived bound on deviation of residuals, can devise a residual replacement strategy for s-step CG
- Implementation has negligible cost

Residual replacement for s-step CG

- Use computable bound for$\|b - A\hat{x}_i - \hat{r}_i\|$ to update d_i, an estimate of error in computing r_i, in each iteration
- Set threshold $\hat{\varepsilon} \approx \sqrt{\varepsilon}$, replace whenever $d_i/\|r_i\|$ reaches threshold

Pseudo-code for residual replacement with group update for s-step CG:

```
if  $d_{i-1} \le \hat{\varepsilon}\|r_{i-1}\|$  and  $d_i > \hat{\varepsilon}\|r_i\|$  and  $d_i > 1.1 d_{init}$
        $z = z + \mathcal{Y}_k\, x'_{k,j} + x_{sk}$
        $x_i = 0$
        $r_i = b - Az$
        $d_{init} = d_i = \varepsilon\big((1 + 2N')\|A\|\|z\| + \|r_i\|\big)$
        $p_i = \mathcal{Y}_k p'_{k,j}$
        break from inner loop and begin new outer loop
end
```

group update of approximate solution

set residual to true residual

A computable bound

- In each iteration, update error estimate d_i $(i \equiv sk + j)$ by:

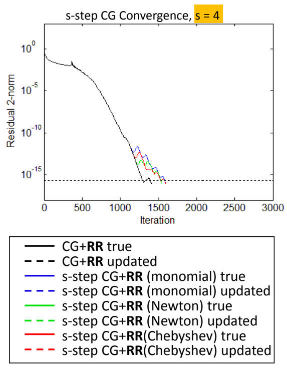
Extra computation all lower order terms, communication only increased by *at most* factor of 2

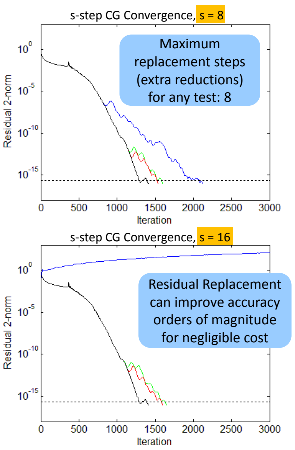
$$d_i \equiv d_{i-1}$$

$$+\varepsilon\Big[(4+N')\Big(\|A\|\,\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,j}|\big\| + \big\||\hat{\mathcal{Y}}_k|\cdot|\mathcal{B}_k|\cdot|\hat{x}'_{k,j}|\big\|\Big) + \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,j}|\big\|\Big]$$

$$+\varepsilon\begin{cases} \|A\|\|\hat{x}_{sk+s}\| + (2+2N')\|A\|\,\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,s}|\big\| + N'\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,s}|\big\|, & j = s \\ 0, & \text{o.w.} \end{cases}$$

where $N' = \max(N, 2s + 1)$.





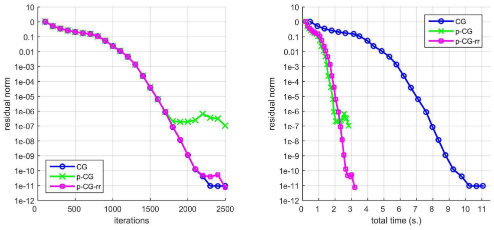
Maximum replacement steps (extra reductions) for any test: 8

Residual Replacement can improve accuracy orders of magnitude for negligible cost

Model Problem: 2D Poisson (5-pt stencil),
$n = 512^2$, $N \approx 10^6$, $\kappa(A) \approx 10^4$
$b = A(1\sqrt{n} \cdot \text{ones}(n, 1))$

Pipelined CG with residual replacement

Similar approach possible for pipelined CG; see (Cools et al., 2018)



20 nodes (two 6-core Intel Xeon X5660 Nehalem 2:80-GHz processors per node), 2D Poisson problem with 1e6 unknowns; in pipelined CG with residual replacement, 39 replacements were performed.

Adaptive s-step CG

- Consider the growth of the relative residual gap caused by errors in outer loop k, which begins with global iteration number m
- We can approximate an upper bound on this quantity by

$$\frac{\|f_{m+s} - f_m\|}{\|A\|\|x\|} \lesssim \varepsilon\left(1 + \kappa(A)\Gamma_k \frac{\max_{j\in\{0,\dots,s\}}\|\hat{r}_{m+j}\|}{\|A\|\|x\|}\right) \qquad f_i \equiv b - A\hat{x}_i - \hat{r}_i$$

- If our application requires relative accuracy ε^*, we must have

$$\Gamma_k \equiv c \cdot \|\hat{\mathcal{Y}}_k^+\|\,\big\||\hat{\mathcal{Y}}_k|\big\| \lesssim \frac{\varepsilon^*}{\varepsilon \max_{j\in\{0,\dots,s\}}\|\hat{r}_{m+j}\|}$$

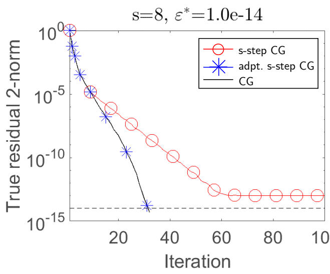
- $\|\hat{r}_i\|$ large → Γ_k must be small; $\|\hat{r}_i\|$ small → Γ_k can grow

⇒ adaptive s-step approach [C., 2018]

- s starts off small, increases at rate depending on $\|\hat{r}_i\|$ and ε^*

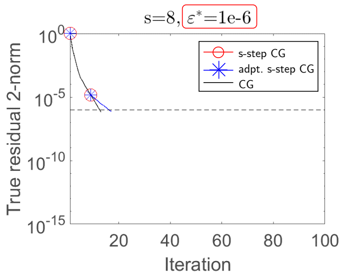
Adaptive s-step CG

mesh3e1 (UFSMC)
$n = 289$
$\kappa(A) \approx 10$
$b_i = 1/\sqrt{N}$



Adaptive s-step CG

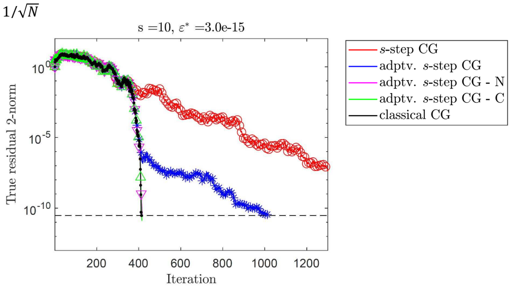
mesh3e1 (UFSMC)
$n = 289$
$\kappa(A) \approx 10$
$b_i = 1/\sqrt{N}$



Extensions to adaptive s-step CG

- Method of Meurant and Tichý (2018) for cheap approximation of extremal Ritz values
  - Uses Cholesky factors of Lanczos tridiagonal T_i, $T_i = L_i L_i^T$
  - Use α and β computed during each iteration to incrementally update estimates of $\|L_i\|_2^2 = \lambda_{max}(T_i) \approx \lambda_{max}(A)$, $\left\|L_i^{-1}\right\|_2^{-2} = \lambda_{min}(T_i) \approx \lambda_{min}(A)$
    - Essentially no extra work, no extra communication
- Can be used in two ways in adaptive algorithm
  1. Incrementally refine estimate of $\kappa(A)$ (used in determining which s to use)
  2. Incrementally refine parameters used to construct Newton or Chebyshev polynomials

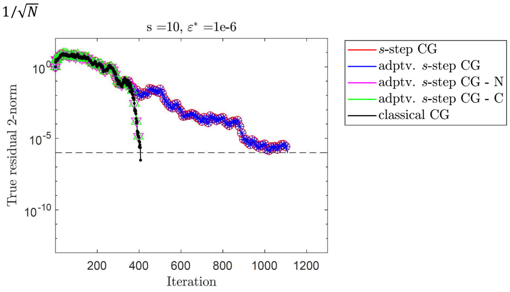
A = 494bus from SuiteSparse
$b_{\mathrm{i}} = 1/\sqrt{N}$



Number of global synchronizations

Fixed s-step	Old adaptive s-step	Improved adaptive s-step w/Newton	Improved adaptive s-step w/Chebyshev	classical CG
-	132	59	53	414

A = 494bus from SuiteSparse
$b_{\mathrm{i}} = 1/\sqrt{N}$

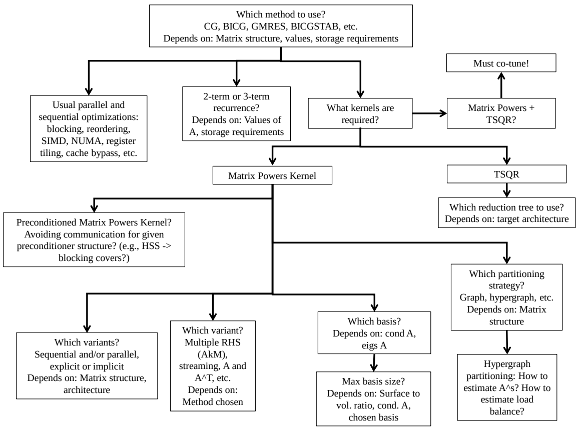


Number of global synchronizations

Fixed s-step	Old adaptive s-step	Improved adaptive s-step w/Newton	Improved adaptive s-step w/Chebyshev	classical CG
111	111	43	43	407

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)
- Krylov solve is communication-bound (particularly latency bound due to global synchronization)
- Extremal eigenvalues are known or easy to estimate
- Accuracy much less than machine epsilon required by the application
- s-step methods
  - The matrix is well-partitioned into domains with low surface-to-volume ratio
  - Simple preconditioning is sufficient/the preconditioner is amenable to communication avoidance
  - The same coefficient matrix (or at least a coefficient matrix with the same nonzero structure) will be reused over multiple solves
  - improvement even for small numbers of nodes (reduces both intra- and inter-processor communication)
- (deep) pipelined methods
  - cost of applying preconditioner + SpMV is less than or the same as a global synchronization
  - improvement only for large numbers of nodes


Which method to use?
CG, BICG, GMRES, BICGSTAB, etc.
Depends on: Matrix structure, values, storage requirements

Usual parallel and sequential optimizations: blocking, reordering, SIMD, NUMA, register tiling, cache bypass, etc.

2-term or 3-term recurrence?
Depends on: Values of A, storage requirements

What kernels are required?

Matrix Powers + TSQR?

Must co-tune!

Matrix Powers Kernel

TSQR

Which reduction tree to use?
Depends on: target architecture

Preconditioned Matrix Powers Kernel?
Avoiding communication for given preconditioner structure? (e.g., HSS -> blocking covers?)

Which partitioning strategy?
Graph, hypergraph, etc.
Depends on: Matrix structure

Hypergraph partitioning: How to estimate A^s? How to estimate load balance?

Which variants?
Sequential and/or parallel, explicit or implicit
Depends on: Matrix structure, architecture

Which variant?
Multiple RHS (AkM), streaming, A and A^T, etc.
Depends on: Method chosen

Which basis?
Depends on: cond A, eigs A

Max basis size?
Depends on: Surface to vol. ratio, cond. A, chosen basis

Looking Forward

- Hybrid methods
  - stationary iterative method + Krylov subspace method
- Fault tolerance
  - MTTF=0 on an exascale machine
  - A problem to be handled at the algorithm level, or...?
- Making use of specialized hardware
  - accelerators, GPUs, etc.
  - multiple precisions?
  - new performance model, new programming model, bigger tuning space

An introduction to extended finite element method

Jaroslav Haslinger

Charles University, Prague, Czech Republic
Institute of Geonics of the Czech Academy of Sciences, Ostrava, Czech Republic

hasling@karlin.mff.cuni.cz

Outline

- Introduction.
- XFEM in crack problems.
- XFEM in multi-phase problems.
- Fictitious domain – XFEM formulation of 2^{nd} order elliptic PDE's.

Standard FEM

V, V' ... a Hilbert space, its dual, respectively
$a\colon V \times V \to \mathbb{R}$ ... a bounded, V-elliptic bilinear form
$f \in V'$

$$\boxed{\text{Find } u \in V : \quad a(u,v) = \langle f, v\rangle \quad \forall v \in V \qquad (\mathcal{P})}$$

$V_h \subset V$, $\dim V_h = n(h)$, $V_h = \{\phi_i\}_{i=1}^{n(h)}$

$$\boxed{\text{Find } u_h \in V_h : \quad a(u_h,v_h) = \langle f, v_h\rangle \quad \forall v_h \in V_h \qquad (\mathcal{P})_h}$$

$u_h = \sum_{i=1}^{n(h)} u_i\phi_i, \quad \boldsymbol{u} \in R^{n(h)}: \quad \boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$

$\boldsymbol{A} = (a_{ij})_{i,j=1}^{n(h)}, \quad a_{ij} = a(\phi_j, \phi_i), \quad \boldsymbol{f} = (f_i)_{i=1}^{n(h)}, \quad f_i = \langle f, \phi_i\rangle$

It holds: $\|u - u_h\|_V \le c \inf_{v_h \in V_h} \|u - v_h\|_V$

- **discontinuity of solutions**
  (crack problems, the evaluation of dislocations, grain boundaries, ...)

  Moës, N., Dolbow, J., Belytschko, T.A.: *A FEM for crank growth without remeshing*. Int. J. Numer. Meth. Eng. 46 (1999), 131–150.

- **discontinuity of gradients of solutions** (multi-phase problems)

  Sukumar, N., Chopp, D.Z., Moës, N., Belytschko, T.A.: *Modeling holes and inclusions by level sets in the extended finite-element method*, Comput. Meth. Appl. Mech. Eng. 190 (2001), 6183–200.

  Topical review paper:

  Belytschko, T.A., Gracie, R., Ventura, G.: *A review of extended /generalized finite element methods for material modeling*, Modelling Simul. Mater. Sci. Eng. 17 (2009): 043001.

- **applications in fluid mechanics**
  (fluid-structure interactions, free surface flows, ...)

Abstract setting of XFEM

$$u_h = \sum_{i=1}^{n(h)} u_i\phi_i + \sum_{j=1}^{m(h)} a_j M_j, \quad u_i, a_j \in \mathbb{R},$$

- $M_j = \psi_j \Psi, \quad \Psi$ ... global enrichment function,

  $\{\psi_j\}_{j=1}^{m(h)}$ ... partition of unity
- Usually: $\psi_j = N_j$ ... Courant basis functions, $j = 1, \dots, m(h)$
- $V_h = \{\phi_i\}_{i=1}^{n(h)} \oplus \{N_j\Psi\}_{j=1}^{m(h)}$

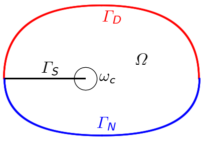
$$\boxed{\text{Find } u_h \in V_h : \quad a(u_h,v_h) = \langle f, v_h\rangle \quad \forall v_h \in V_h \qquad (\mathcal{P})_{h,X}}$$

$$\begin{pmatrix} \boldsymbol{A}_{uu} & \boldsymbol{A}_{ua} \\ \boldsymbol{A}_{ua}^\top & \boldsymbol{A}_{aa} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{a} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_u \\ \boldsymbol{F}_a \end{pmatrix}$$

- $\boldsymbol{A}_{uu} \in \mathbb{R}^{n\times n}, \ \boldsymbol{A}_{aa} \in \mathbb{R}^{m\times m}, \ \boldsymbol{A}_{ua} \in \mathbb{R}^{n\times m}, \quad n := n(h), \ m := m(h)$

Discontinuity of solutions. Crack problems.

a) scalar case in plane



$$\left.\begin{aligned} -\triangle u &= f && \text{in} && \Omega \\ u &= 0 && \text{on} && \Gamma_D \\ \tfrac{\partial u}{\partial n} &= g && \text{on} && \Gamma_N \\ \tfrac{\partial u}{\partial n} &= 0 && \text{on} && \Gamma_S \end{aligned}\right\}$$

$f, g, \bar{\Omega}$ ... sufficiently smooth

$$\boxed{\text{Find } u \in V : \quad (\nabla u, \nabla v)_{0,\Omega} = (f,v)_{0,\Omega} + (g,v)_{0,\Gamma_N} \quad \forall v \in V \qquad (\mathcal{P})}$$

- $V = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D\}$
- $\exists!\ u \in H^{3/2-\epsilon}(\Omega) \cap V$ solving $(\mathcal{P})$, $\epsilon > 0$ is arbitrary
- $u = u_r + u_s$, $u_r \in H^2(\Omega)$ – regular part, $u_s = \chi K \sqrt{r} \sin\theta/2$ – singular part
- χ – cut-off function, K – stress intensity factor, (r,θ) - local polar coord.

b) vector case in plane

$$\left.\begin{aligned} -\operatorname{div}\sigma(\boldsymbol{u}) = \boldsymbol{f} & \quad \text{in} \quad \Omega \\ \boldsymbol{u} = \boldsymbol{0} & \quad \text{on} \quad \Gamma_D \\ \sigma(\boldsymbol{u})\boldsymbol{n} = \boldsymbol{g} & \quad \text{on} \quad \Gamma_N \\ \sigma(\boldsymbol{u})\boldsymbol{n} = \boldsymbol{0} & \quad \text{on} \quad \Gamma_S \end{aligned}\right\}$$

- $\sigma(\boldsymbol{u}) = \lambda(\operatorname{tr}\varepsilon(\boldsymbol{u}))\boldsymbol{I} + 2\mu\varepsilon(\boldsymbol{u}), \quad \varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^\top)$
- λ, μ ... the Lamé coefficients
- $V = \{\boldsymbol{v} \in (H^1(\Omega))^2 \mid \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_D\}$

$$\boxed{\text{Find } \boldsymbol{u} \in V: \quad (\sigma(\boldsymbol{u}), \varepsilon(\boldsymbol{v}))_{0,\Omega} = (\boldsymbol{f}, \boldsymbol{v})_{0,\Omega} + (\boldsymbol{g}, \boldsymbol{v})_{0,\Gamma_N} \quad \forall \boldsymbol{v} \in V \qquad (\mathcal{P})}$$

$$\boldsymbol{u} = \boldsymbol{u}_r + \boldsymbol{u}_s, \quad \boldsymbol{u}_r \in (H^2(\Omega))^2, \quad \boldsymbol{u}_s = \chi\sum_{j=1}^{4}\boldsymbol{c}_j F_j, \quad \boldsymbol{c}_j \in \mathbb{R}^2$$

$$\{F_j\}_{j=1}^4 = \{\sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\cos\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta\}$$

XFEM for crack problems

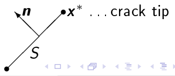
a) scalar case

[Nicaise, S., Renard, Y., Chahine, E.: *Optimal convergence analysis for XFEM*. Int. J. Numer. Meth. Eng. 86 (2011), 528–548]

- $\Omega \subset \mathbb{R}^2$ ... polygonal domain, $\{\mathcal{T}_h\}_{h\to 0_+}$ ... family of regular triang. of $\bar{\Omega}$
- S ... straight line segment
- I ... the set of all node indices of P_1 elements
- $I_H \subset I, \quad i \in I_H \iff \operatorname{supp} N_i$ is completely cut by the crack
- $$\chi \in W^{2,\infty}(\Omega) \text{ ... cut-off function: } \begin{cases} \chi(r) = 1, & r < r_0 \\ 0 < \chi(r) < 1, & r_0 < r < r_1 \\ \chi(r) = 0, & r < r_1 \end{cases}$$

Heaviside type function:

$$H(\boldsymbol{x}) = \left\{\begin{array}{rl} 1, & (\boldsymbol{x} - \boldsymbol{x}^*).\boldsymbol{n} \geq 0 \\ -1, & (\boldsymbol{x} - \boldsymbol{x}^*).\boldsymbol{n} < 0 \end{array}\right\}$$



$$V_h = \{N_i\}_{i\in I} \oplus \{HN_i\}_{i\in I_H} \oplus \{\chi u_s\}, \qquad u_s = \sqrt{r}\sin\tfrac{\theta}{2}$$

$$v_h \in V_h \iff v_h = \sum_{i\in I} a_i N_i + \sum_{i\in I_H} b_i HN_i + K_h\chi u_s, \qquad a_i, b_i, K_h \in \mathbb{R}$$

$$\left.\begin{array}{l} \text{Find } u_h \in V_h \text{ such that} \\ (\nabla u_h, \nabla v_h)_{0,\Omega} = (f, v_h)_{0,\Omega} + (g, v_h)_{0,\Gamma_N} \quad \forall v_h \in V_h \end{array}\right\} \qquad (\mathcal{P})_h$$

b) vector case

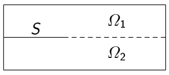
$$V_h = \left\{\boldsymbol{v}_h \mid \boldsymbol{v}_h = \sum_{i\in I}\boldsymbol{a}_i N_i + \sum_{i\in I_H}\boldsymbol{b}_i HN_i + \sum_{j=1}^{4}\boldsymbol{c}_j\chi F_j, \quad \boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_j \in \mathbb{R}^2\right\}$$

Error estimates

$$V_h \subset V \implies \|u - u_h\|_{1,\Omega} \leq c \inf_{v_h\in V_h}\|u - v_h\|_{1,\Omega} \leq c\|u - \Pi_h u\|_{1,\Omega}$$

Π_h ... XFEM interpolation operator

- $H|_{\Omega_k} = (-1)^{k+1}, \ k = 1, 2$
- $u_r = u - \chi u_s, \ u_r^k := u_r|_{\Omega_k} \in H^2(\Omega_k)$
- $\tilde{u}_r^k$ ... extension of u_r^k onto $\bar{\Omega}$
- $\|\tilde{u}_r^k\|_{2,\bar{\Omega}} \leq c\|u_r^k\|_{2,\Omega_k}, \ k = 1, 2$



$$\Pi_h u = \sum_{i\in I} a_i N_i + \sum_{i\in I_H} b_i HN_i + \chi u_s$$

- if $i \in I \setminus I_H$ then $a_i = u_r(x_i)$
- if $i \in I_H$ and $x_i \in \bar{\Omega}_k, \ k = 1, 2$ then

$$\left.\begin{array}{l} a_i = \frac{1}{2}\left(u_r^k(x_i) + \tilde{u}_r^\ell(x_i)\right) \\ b_i = \frac{1}{2}\left(u_r^k(x_i) - \tilde{u}_r^\ell(x_i)\right) H(x_i) \end{array}\right\} \quad \ell \in \{1, 2\}, \ \ell \neq k$$

Lemma: *It holds:*

- $\Pi_h u = \pi_h u_r + \chi u_s$ *on any triangle* K *non-enriched by* H
- $\Pi_h u|_{K\cap\Omega_k} = \pi_h\tilde{u}_r^k|_{K\cap\Omega_k} + \chi u_s|_{K\cap\Omega_k}, \ k = 1, 2$ *on any triangle totally enriched by* H,

where π_h *is the standard Lagrange interpolation operator by P1-elements.*

Approximation properties of Π_h

$K \in \mathcal{T}_h, \ h_K = \operatorname{diam}(K), \ \varrho_K$ ... diameter of the circle inscribed in K

Lemma: *Let* $K \in \mathcal{T}_h$ *be a triangle totally enriched by* H. *Then there exists an absolute constant* $c > 0$ *such that*

$$\|u - \Pi_h u\|_{1,K\cap\Omega_1} \leq ch_K\sigma_K|\tilde{u}_r^1|_{2,K}$$

and

$$\|u - \Pi_h u\|_{1,K\cap\Omega_2} \leq ch_K\sigma_K|\tilde{u}_r^2|_{2,K},$$

where $\sigma_K = h_K/\varrho_K$.

Lemma: *Let* $K \in \mathcal{T}_h$ *be a triangle partially enriched by* H *and denote* $K^* = K \setminus S$. *Then*

$$\|u - \Pi_h u\|_{1,K^*} \leq ch_K\left(|\tilde{u}_r^1|_{2,B(\boldsymbol{x}^*,2h_k)} + |\tilde{u}_r^2|_{2,B(\boldsymbol{x}^*,2h_k)}\right),$$

where $c > 0$ *is an absolute constant.*

Lemma: *Let* $K \in \mathcal{T}_h$ *be a triangle containing the crack tip. Then*

$$\|u - \Pi_h u\|_{1,K^*} \leq ch_K\left(|\tilde{u}_r^1|_{2,B(\boldsymbol{x}^*,h_k)} + |\tilde{u}_r^2|_{2,B(\boldsymbol{x}^*,h_k)}\right).$$

Theorem: *Let* f, g, Ω *be sufficiently smooth such that* $u - u_s \in H^2(\Omega)$. *Then*

$$\|u - u_h\|_{1,\Omega} \leq ch\|u - \chi u_s\|_{2,\Omega},$$

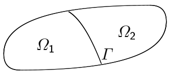
where $u_h \in V_h$ *is the solution of* $(\mathcal{P})_h$, u_s *is the singular part of* u *and* $\chi \in W^{2,\infty}(\Omega)$ *is the cut-off function.*

3. XFEM in multi-phase problems

- Diez, P. Cottereau, R., Zlotnik, S.: *A stable XFEM formulation for multi-phase problems enforcing the accuracy of the fluxes through Lagrange multipliers.* Int. J. Numer. Meth. Eng. 96 (2013) 303–322.
- Moës, N., Cloirec, M., Cartraud, P., Remacle, J.F.: *A computational approach to handle complex microstructure geometries.* Comp. Meth. Appl. Eng. 192 (2003) 3163–3177.

Setting of the problem:

- $\Omega \subset \mathbb{R}^2$, $\bar\Omega = \bar\Omega_1 \cup \bar\Omega_2$, $\Gamma = \bar\Omega_1 \cap \bar\Omega_2$
- $\Omega_1 = \{\boldsymbol{x} \in \Omega \mid \ell(x) > 0\}$
- $\Omega_2 = \{\boldsymbol{x} \in \Omega \mid \ell(x) < 0\}$



$$\left.\begin{array}{rcc} -\mathrm{div}(k\nabla u) = f & \text{in} & \Omega \\ u = 0 & \text{on} & \partial\Omega \end{array}\right\}, \qquad k = \left\{\begin{array}{ccc} k_1 & \text{on} & \Omega_1 \\ k_2 & \text{on} & \Omega_2 \end{array}\right. , \quad k_1 \gg k_2 > 0$$

$$\boxed{\text{Find } u \in H_0^1(\Omega): \quad \int_\Omega k\nabla u.\nabla v\, dx = \int_\Omega fv\, dx \quad \forall v \in H_0^1(\Omega) \qquad (\mathcal{P})}$$

Hidden conditions in $(\mathcal{P})$: $u_i = u|_{\Omega_i}$, $i = 1, 2$

$$\begin{array}{lcl} u_1 = u_2 \text{ on } \Gamma & \Longrightarrow & \frac{\partial u_1}{\partial s} = \frac{\partial u_2}{\partial s} \text{ on } \Gamma \\ k_1\frac{\partial u_1}{\partial n} = k_2\frac{\partial u_2}{\partial n} \text{ on } \Gamma & \Longrightarrow & \text{jump of } \frac{\partial}{\partial n} \text{ across } \Gamma \end{array}$$

XFEM formulation:

- $\Omega \subset \mathbb{R}^2$ ...polygonal domain, $\mathcal{T}_h$ ... triangulation of $\bar\Omega$
- I, I_0 ... the set of indices of the nodes of $\mathcal{T}_h$ in $\bar\Omega$, and $\mathrm{int}\,\Omega$, respectively
- $\{N_i\}_{i\in I}$ ... the Courant basis functions
- $\ell \approx \ell_h = \sum_{i\in I} N_i\ell_i$, ℓ_i ... the nodal values of ℓ
- $\Omega_1 \approx \Omega_{1h} = \{\boldsymbol{x} \in \Omega \mid \ell_h(x) > 0\}$
- $\Omega_2 \approx \Omega_{2h} = \{\boldsymbol{x} \in \Omega \mid \ell_h(x) < 0\}$
- $\Gamma \approx \Gamma_h = \{\boldsymbol{x} \in \Omega \mid \ell_h(x) = 0\}$

Ridge function R:

$$R = \sum_{i\in I} N_i|\ell_i| - \Big|\sum_{i\in I} N_i\ell_i\Big|$$

It holds:

- $R \equiv 0$ in all elements not crossed by the interface Γ. Consequently, the support of R is the set of all elements which will be enriched.
- Let I_a be the set of indices of the nodes of enriched elements. Then

$$R = \sum_{i\in I_a} N_i|\ell_i| - \Big|\sum_{i\in I_a} N_i\ell_i\Big|$$

XFEM space:

$$V_h = \{N_i\}_{i\in I_0} \oplus \{RN_i\}_{i\in I_a\cap I_0}$$

4. Fictitious domain / XFEM formulation of 2^{nd} order elliptic PDE's

[J.H., Y. Renard: *A new fictitious domain approach inspired by the XFEM.* SIAM J. Numer. Anal. 47 (2009) 1474-1499]

$$\left.\begin{array}{rcc} -\triangle u = f & \text{in} & \Omega \\ u = 0 & \text{on} & \Gamma_D \\ \frac{\partial u}{\partial n} = g & \text{on} & \Gamma_N \end{array}\right\}, \quad f \in L^2(\Omega),\ g \in L^2(\Gamma_N)$$

- $V = H^1(\Omega)$, $V_0 = \{v \in V \mid v = 0 \text{ on } \Gamma_D\}$
- $a(u, v) = (\nabla u, \nabla v)_{0,\Omega}$, $\ell(v) = (f, v)_{0,\Omega} + (g, v)_{0,\Gamma_N}$

$$\boxed{\text{Find } u \in V_0: \quad a(u, v) = \ell(v) \quad \forall v \in V_0 \qquad (\mathcal{P})}$$

Mixed formulation of $(\mathcal{P})$: Find $(u, \lambda) \in V \times W$

$$\left.\begin{array}{rcll} a(u, v) + \langle\lambda, v\rangle_{W\times X} & = & \ell(v) & \forall v \in V \\ \langle\mu, u\rangle_{W\times X} & = & 0 & \forall \mu \in W \end{array}\right\}, \qquad (\mathcal{M})$$

where $X = \{w \in L^2(\Gamma_D) \mid \exists v \in V: \ w = v \text{ on } \Gamma_D\}$, $W = X'$

Theorem.

- *Problem* $(\mathcal{M})$ *has a unique solution* (u, λ).
- *In addition,* u *solves* $(\mathcal{P})$ *and* $\lambda = -\frac{\partial u}{\partial n}$ *on* Γ_D.
- $(\mathcal{M})$ *is equivalent to the problem of finding a saddle-point of* $\mathcal{L}$ *on* $V \times W$*, where*

$$\mathcal{L}(v, \mu) = \frac{1}{2}a(v, v) + \langle\mu, v\rangle_{W\times X} - \ell(v).$$

Fictitious domain / XFEM formulation of $(\mathcal{M})$.

- $\hat\Omega \supset \Omega$ ... simple shaped domain, $\{\mathcal{T}_h\}_{h>0}$ ... family of partitions of $\overline{\hat\Omega}$
- $\hat V_h \subset H^1(\hat\Omega)$, $\hat W_h \subset L^2(\hat\Omega)$ ... finite element spaces on $\mathcal{T}_h$:

$$\hat V_h = \{v_h \in C(\overline{\hat\Omega}) \mid\ v_h|_T \in P(T)\ \ \forall T \in \mathcal{T}_h\}, \quad P(T) \supseteq P_k(T),\ k \geq 1 \text{ integer}$$

- Define: $V_h = \hat V_h|_\Omega$, $W_h = \hat W_h|_\Omega$.

Fictitious domain / XFEM formulation of $(\mathcal{M})$**:** Find $(u_h, \lambda_h) \in V_h \times W_h$:

$$\left.\begin{array}{rcll} a(u_h, v_h) + (\lambda_h, v_h)_{0,\Gamma_D} & = & \ell(v_h) & \forall v_h \in V_h \\ (\mu_h, u_h)_{0,\Gamma_D} & = & 0 & \forall \mu_h \in W_h \end{array}\right\}, \qquad (\mathcal{M})_h$$

Assumptions:

(i) $1|_{\Gamma_D} \in W_h$

(ii) $\bar\mu_h \in W_h: \ (\bar\mu_h, v_h)_{0,\Gamma_D} = 0 \ \forall v_h \in V_h \quad \Longrightarrow \quad \bar\mu_h = 0$

$$(i) + (ii) \quad \Longrightarrow \quad (\mathcal{M})_h \text{ has a unique solution}$$

Stabilized formulation of $(\mathcal{M})_h$**:**

Let $R_h\colon V_h \to L^2(\Gamma_D)$ appoximates the normal derivative on Γ_D and

(iii) $h^{1/2}\|R_hv_h\|_{0,\Gamma_D} \leq c\|\nabla v_h\|_{0,\Omega} \quad \forall v_h \in V_h,\ \forall h > 0.$

Define

$$\mathcal{L}_h(v_h, \mu_h) = \mathcal{L}(v_h, \mu_h) - \frac{\gamma}{2}\|\mu_h + R_hv_h\|^2_{0,\Gamma_D}, \quad (v_h, \mu_h) \in V_h \times W_h,$$

where $\gamma := h\gamma_0$, $\gamma_0 > 0$ are given

Stabilized problem $(\mathcal{M})_{st,h}$**:** Find $(u_h, \lambda_h) \in V_h \times W_h$ such that

$$\left.\begin{array}{lcll} a(u_h, v_h) + (\lambda_h, v_h)_{0,\Gamma_D} - \gamma(\lambda_h + R_h u_h, R_h v_h)_{0,\Gamma_D} & = & \ell(v_h) & \forall v_h \in V_h \\ (\mu_h, u_h)_{0,\Gamma_D} - \gamma(\lambda_h + R_h u_h, \mu_h)_{0,\Gamma_D} & = & 0 & \forall \mu_h \in W_h \end{array}\right\}$$

Equivalent form of $(\mathcal{M})_{st,h}$**:** Find $(u_h, \lambda_h) \in V_h \times W_h$ such that

$$\left.\begin{array}{ll} \mathcal{B}_h(u_h, \lambda_h; v_h, \mu_h) = \ell(v_h) & \forall (v_h, \mu_h) \in V_h \times W_h \\ \mathcal{B}_h \colon (V_h \times W_h)^2 \to \mathbb{R} & \end{array}\right\}$$

$$\begin{array}{rcl} \mathcal{B}_h(u_h, \lambda_h; v_h, \mu_h) & = & a(u_h, v_h) + (\lambda_h, v_h)_{0,\Gamma_D} + (\mu_h, u_h)_{0,\Gamma_D} \\ & & -\gamma(\lambda_h + R_h u_h, R_h v_h + \mu_h)_{0,\Gamma_D} \end{array}$$

Inf-sup property of $\mathcal{B}_h$**:**

(iv) Let $P_h \colon L^2(\Gamma_D) \to W_h$ be the L^2-projection on W_h and

$$\|P_h v - v\|_{0,\Gamma_D} \leq c h^{1/2} \|v\|_{1/2,\Gamma_D} \quad \forall v \in H^{1/2}(\Gamma_D),$$

$c > 0$ does not depend on h

Lemma. *Let* (i)*–*(iv) *be satisfied. Then for* $\gamma_0 > 0$ *sufficiently small there exists a constant* $c > 0$ *independent of* h *such that*

$$\sup_{\substack{(z_h,\eta_h) \in V_h \times W_h \\ (z_h,\eta_h) \neq 0}} \frac{\mathcal{B}_h(v_h, \mu_h; z_h, \eta_h)}{|\!|\!|(z_h, \eta_h)|\!|\!|} \geq c |\!|\!|(v_h, \mu_h)|\!|\!| \quad \forall (v_h, \mu_h) \in V_h \times W_h,$$

where

$$|\!|\!|(z_h, \eta_h)|\!|\!|^2 = \|z_h\|^2_{1,\Omega} + h^{-1}\|z_h\|^2_{0,\Gamma_D} + h\|\eta_h\|^2_{0,\Gamma_D}$$

Theorem. *Let* (i)*–*(iv) *be satisfied and* $\gamma_0 > 0$ *be sufficiently small. If* (u, λ) *is a solution to* $(\mathcal{M})$ *and* $\lambda \in L^2(\Gamma_D)$ *then there exists a constant* $c > 0$ *independent of* h *such that*

$$|\!|\!|(u - u_h, \lambda - \lambda_h)|\!|\!| \leq c \inf_{\substack{v_h \in V_h \\ \mu_h \in W_h}} \left(|\!|\!|(u - v_h, \lambda - \mu_h)|\!|\!| + h^{1/2} \left\| R_h v_h - \frac{\partial u}{\partial n} \right\|_{0,\Gamma_D} \right).$$

Acknowledgement: This work has been supported by the Czech Science Foundation, project No. 19-11441S: "Efficient and reliable computational techniques for limit analysis and incremental methods in geotechnical stability"

Thank you for your attention!

On the Way from Matrix to Tensor Computations

Introduction, Basic arithmetics, Tensor decompositions, Hierarchical formats, and Tensor networks

M. Plešinger with a lot of inspiration from collaboration with
I. Hětynková, D. Kressner, C. Tobler, J. Žáková,
with special thanks to B. N. Khoromskij, and many other colleagues ...

martin.plesinger@tul.cz
Department of Mathematics, Technical University of Liberec, Liberec

SNA '19, Ostrava, January 21—25, 2019

Outline of the tutorial

- **Lecture I**
  Introduction to tensors
  Basic terminology and basic manipulation with tensors
  Rank of a tensor
  Tensor arithmetics
- **Lecture II**
  Basic decompositions of a tensor
  Low-rank arithmetics of tensors
  Graph interpretation: Tensor networks & Hierarchical formats
  Arithmetics of hierarchical Tucker
  An example of practical application

[*T. G. Kolda, B. W. Bader*: **Tensor decompositions and applications**, SIAM Review 51(3), pp. 455–500, 2009]

Introduction to tensors

Introduction

The standard tensor definition

A first (and only) definition of a tensor I met at school:

Tensor $\mathcal{T}$ of order k is a k_1-covariant and k_2-contravariant ($k = k_1 + k_2$) multilinear form on linear vector space $\mathscr{V}$ over $\mathbb{R}$,

$$\mathcal{T} : \underbrace{\mathscr{V} \times \mathscr{V} \times \cdots \times \mathscr{V}}_{k_1\text{-times}} \times \underbrace{\mathscr{V}^* \times \mathscr{V}^* \times \cdots \times \mathscr{V}^*}_{k_2\text{-times}} \longrightarrow \mathbb{R}.$$

In this way tensors are used in many branches of mathematics and physics (differential geometry, solid-state physics, continuum mechanics, general relativity, etc.).

It is something like a matrix, but ...

What is a matrix?

Three (distinct) reference frames

A matrix A can be seen as a mapping between linear vector spaces

$$\begin{array}{rcl} A : \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ u & \longmapsto & w = Au, \end{array}$$

as a bilinear form

$$\begin{array}{rcl} A : \mathbb{R}^n \times \mathbb{R}^m & \longrightarrow & \mathbb{R} \\ (u, v) & \longmapsto & f(u, v) = v^{\mathsf{T}} A u, \end{array}$$

and also as an algebraic vector, a member of linear vectors space

$$A \in \mathbb{R}^{m \times n}.$$

What is a matrix?

Transformations of matrices

Let $m = n$ (A is square). We change the basis in $\mathbb{R}^n$ as follows $x = Zx'$, i.e., $x \longmapsto x' = Z^{-1}x$, then

$$\begin{array}{rcl} Au & = & w \\ A(Zu') & = & Zw' \\ \underbrace{(Z^{-1}AZ)}\, u' & = & w' \end{array}, \qquad \begin{array}{rcl} f(u, v) & = & v^{\mathsf{T}} A u \\ f(Zu', Zv') & = & (Zv')^{\mathsf{T}} A (Zu') \\ f'(u', v') & = & v'^{\mathsf{T}} \underbrace{(Z^{\mathsf{T}}AZ)}\, u' \end{array}.$$

We get two different transf's of A, $A \longmapsto Z^{-1}AZ$ (similarity transf.; eigenvalues) and $A \longmapsto Z^{\mathsf{T}}AZ$ (congruence; quadratic forms), resp.

On the other hand, we can study the matrix itself—e.g., decompositions:

$$A = LU, \quad A = LL^{\mathsf{T}}, \quad A = QR, \quad A = XDX^{-1}, \quad A = U\Sigma V^{\mathsf{T}}, \quad \text{etc.}$$

Definition of a tensor

... and its 'justification'

Similarly to matrices, we can observe a tensor from different perspectives: As a (multi)linear mapping(s) between different vector spaces, or form on $\mathscr{V}$ (and its dual $\mathscr{V}^*$).

In many applications, however, we are focused more on the '*interior structure*' of the tensor (e.g., we are looking for some decomposition), than on its interactions with its 'surroundigs'.

Definition. Tensor $\mathcal{T}$ of order k is a k-way *array* of real numbers of the given dimension,

$$\mathcal{T} = (t_{i_1,i_2,\ldots,i_k}) \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k}.$$

Note that $n_i \neq n_j$ for $i \neq j$, in general, thus we do not need to distinguish the co- and contravariant indices.

Why tensors?

- Tensors in this form was introduced in psychometrics and chemometrics while analysis of large multidim. arrays of data
- The goal is to find some structure in the data (big data) that allows to analyze (interpret, understand) the data, and simplifies it in such a way, we can easier manipulate it; c.f. the singular value decomposition (SVD) in the case of matrix.
- The memory consumption while storing the tensor as it is, scales exponentialy with k, so-called "curse of dimensionality",

$$\sim n^k \quad \text{where} \quad n = \max\{n_1, n_2, \ldots, n_k\}.$$

- We want to employ basic linear algebra tools (matrix decompositions, etc.).
- In the optimal case, we would like to find a structure (decomposition) that scales linearly with the tensor order k.

Basic terminology

and basic manipulation

with tensors

Order and shape of tensor

Tensors of small orders

By the order of tensor $\mathcal{T} = (t_{i_1,i_2,\ldots,i_k}) \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k}$ we understood the number of its indices, i.e., the number k. Tensors of small orders have special names, for

- $k = 0$ we call them scalars (and denote by α, β, etc.);
- $k = 1$ we call them vectors (and denote by x, y, etc.);
- $k = 2$ we call them matrices (and denote by A, B, etc.);
- $k \geq 3$ we call them just tensors (and denote by $\mathcal{T}$, $\mathcal{S}$, etc.).

By the dimension, we understood the k-tuple $(n_1, n_2, \ldots, n_k)$. If

- $k = 2$ and $n_1 = n_2$, we call them square matrices;
- $k \geq 3$ and $n_1 = n_2 = \cdots = n_k$, we call them cubic tensors.

Moreover, we denote $N = \prod_{\kappa=1}^{k} n_\kappa = n_1 \cdot n_2 \cdot \cdots \cdot n_k$.

Tensors and subtensors

General subtensors

Our tensor $\mathcal{T}$ is an *ordered* set of numbers $t_{i_1,i_2,\ldots,i_k} \in \mathbb{R}$ with indices

$$i_\kappa \in \{1, 2, \ldots, n_\kappa\} \equiv \mathscr{I}_\kappa, \quad \text{for} \quad \kappa = 1, 2, \ldots, k,$$

or, equivalently, with multiindices

$$(i_1, i_2, \ldots, i_k) \in \mathscr{I}_1 \times \mathscr{I}_2 \times \cdots \times \mathscr{I}_k.$$

Let $\mathscr{I}'_\kappa \subseteq \mathscr{I}_\kappa$. The subarray of $\mathcal{T}$ obtained by employing only the multiindices in the subset $\mathscr{I}'_1 \times \mathscr{I}'_2 \times \cdots \times \mathscr{I}'_k$ is called a subtensor.

There are several kinds of subtensors of particular importance, e.g., so-called fibres, slices, and co-fibres.

Subtensors: Fibres

Rows, columns, tubes, and the others...

Let $\mathcal{T} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k}$, let for some fixed ℓ

$$\mathscr{I}'_\ell = \mathscr{I}_\ell = \{1, 2, \ldots, n_\ell\}, \quad \text{and} \quad \mathscr{I}'_\kappa = \{i_\kappa\} \quad \text{for all} \quad \kappa \neq \ell.$$

The associated subtensor is called the ℓ-mode fibre specified by the $(k–1)$-tuple of indices $(i_1, \ldots, i_{\ell-1}, i_{\ell+1}, \ldots, i_k)$. We denote it

$$\mathcal{T}_{i_1,\ldots,i_{\ell-1},\star,i_{\ell+1},\ldots,i_k} \in \mathbb{R}^{1\times\cdots\times 1\times n_\ell\times 1\times\cdots\times 1},$$

it is isomorphic to an n_ℓ-vector. There is N/n_ℓ of ℓ-mode fibres.

The ℓ-mode fibres, $\ell = 1, 2, \ldots, k$ are for

- $k = 2$ called the columns and rows, respectively;
- $k = 3$ called the columns, rows, and tubes, respectively.

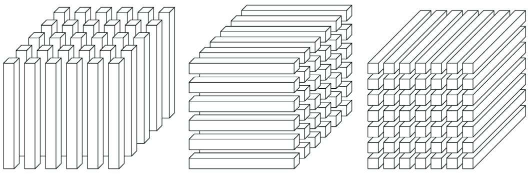
Subtensors: Fibres

Rows, columns, tubes, and the others...

For $k = 3$, the ℓ-mode fibres, $\ell = 1, 2, 3$, i.e.,

$$\mathcal{T}_{\star,i_2,i_3} \in \mathbb{R}^{n_1\times 1\times 1}, \quad \mathcal{T}_{i_1,\star,i_3} \in \mathbb{R}^{1\times n_2\times 1}, \quad \mathcal{T}_{i_1,i_2,\star} \in \mathbb{R}^{1\times 1\times n_3}$$

are called the columns, rows, and tubes, respectively.



Subtensors: Slices

Horizontal, lateral, frontal, and the others...

Let $\mathcal{T} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k}$, let for some fixed τ and ß $(\tau \neq \text{ß})$

$$\mathcal{I}'_\tau = \mathcal{I}_\tau, \quad \mathcal{I}'_{\text{ß}} = \mathcal{I}_{\text{ß}} \quad \text{and} \quad \mathcal{I}'_\kappa = \{i_\kappa\} \quad \text{for all} \quad \kappa \neq \tau \text{ and } \kappa \neq \text{ß}.$$

If $\tau < \text{ß}$, the subtensor is called the $(\tau, \text{ß})$-mode slice given by the $(k–2)$-tuple $(i_1, \ldots, i_{\tau-1}, i_{\tau+1}, \ldots, i_{\text{ß}-1}, i_{\text{ß}+1}, \ldots, i_k)$. We denote it

$$\mathcal{T}_{i_1,\ldots,i_{\tau-1},\star,i_{\tau+1},\ldots,i_{\text{ß}-1},\star,i_{\text{ß}+1},\ldots,i_k} \in \mathbb{R}^{1\times\cdots\times 1\times n_\tau\times 1\times\cdots\times 1\times n_{\text{ß}}\times 1\times\cdots\times 1},$$

it is isomorphic to an n_τ-by-$n_{\text{ß}}$ matrix. There is $N/(n_\tau \cdot n_{\text{ß}})$ of them.

Sometimes, the fibers and slices are considered to be the vectors and matrices. Then we can introduce both, the $(\tau, \text{ß})$- and $(\text{ß}, \tau)$-mode slices. Since they are matrices, they are mutually transposed.

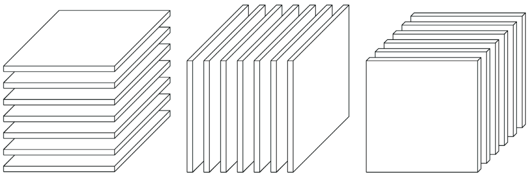
Subtensors: Slices

Horizontal, lateral, frontal, and the others...

For $k = 3$, the $(\tau, \text{ß})$-mode slices, $(\tau, \text{ß}) = (2, 3), (1, 3), (1, 2)$, i.e.,

$$\mathcal{T}_{i_1,\star,\star} \in \mathbb{R}^{1\times n_2\times n_3}, \quad \mathcal{T}_{\star,i_2,\star} \in \mathbb{R}^{n_1\times 1\times n_3}, \quad \mathcal{T}_{\star,\star,i_3} \in \mathbb{R}^{n_1\times n_2\times 1},$$

are called the horizontal, lateral, and frontal, respectively.



Subtensors: Co-fibres

We see that it is easier to identify the type (i.e., horizontal, lateral, frontal) slices of 3-way by the ‘missing index’ than by the pair $(\tau, \text{ß})$ of ‘generating indices’.

Thus we also introduce the ℓ-mode co-fibres such that,

$$\mathcal{I}'_\ell = \{i_\ell\} \quad \text{and} \quad \mathcal{I}'_\kappa = \mathcal{I}_\kappa \quad \text{for all} \quad \kappa \neq \ell,$$

specified by the single index (i_ℓ), denoted

$$\mathcal{T}_{\star,\ldots,\star,i_\ell,\star,\ldots,\star} \in \mathbb{R}^{n_1\times\cdots\times n_{\ell-1}\times 1\times n_{\ell+1}\times\cdots\times n_k}.$$

For $k = 3$, the ℓ-mode co-fibres = the $(\tau, \text{ß})$-mode slices ($\ell \neq \tau$, $\ell \neq \text{ß}$, $\tau < \text{ß}$).

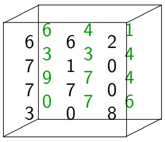
We can continue in a similar manner, but...

Matricization

Unfolding a tensor into a matrix

Collection of all ℓ-mode fibres (handled as vectors) of the given tensor $\mathcal{T}$ into a single matrix $\mathcal{T}^{\{\ell\}} \in \mathbb{R}^{n_\ell\times(N/n_\ell)}$ in the *inverse* lexicographical order is called the ℓ-mode matricization. For

$\mathcal{T} =$



$\in \mathbb{R}^{4\times 3\times 2}$, we get

$$\mathcal{T}^{\{1\}} = [\mathcal{T}_{\star,1,1}, \mathcal{T}_{\star,2,1}, \mathcal{T}_{\star,3,1}, \mathcal{T}_{\star,1,2}, \mathcal{T}_{\star,2,2}, \mathcal{T}_{\star,3,2}] = \left[\begin{array}{ccc|ccc} 6 & 6 & 2 & 6 & 4 & 1 \\ 7 & 1 & 0 & 3 & 3 & 4 \\ 7 & 7 & 0 & 9 & 7 & 4 \\ 3 & 0 & 8 & 0 & 7 & 6 \end{array}\right];$$

$$\mathcal{T}^{\{2\}} = \left[\begin{array}{cccc|cccc} 6 & 7 & 7 & 3 & 6 & 3 & 9 & 0 \\ 6 & 1 & 7 & 0 & 4 & 3 & 7 & 7 \\ 2 & 0 & 0 & 8 & 1 & 4 & 4 & 6 \end{array}\right], \quad \mathcal{T}^{\{3\}} = \left[\begin{array}{cccc|cccc|cccc} 6 & 7 & 7 & 3 & 6 & 1 & 7 & 0 & 2 & 0 & 0 & 8 \\ 6 & 3 & 9 & 0 & 4 & 3 & 7 & 7 & 1 & 4 & 4 & 6 \end{array}\right].$$

Generalized matricization

Unfolding a tensor into a matrix

Let $\mathcal{T}$ be a k-way tensor and

$$\mathcal{R} = \{r_1, r_2, \ldots, r_\mu\}, \quad r_1 < r_2 < \cdots < r_\mu,$$
$$\mathcal{C} = \{c_1, c_2, \ldots, c_\nu\}, \quad c_1 < c_2 < \cdots < c_\nu,$$

such that $\mathcal{R} \cup \mathcal{C} = \{1, 2, \ldots, k\}$ and $\mathcal{R} \cap \mathcal{C} = \varnothing$. Then

$$\mathcal{T}^{\mathcal{R}} = \mathcal{T}^{\{r_1,r_2,\ldots,r_\mu\}} \in \mathbb{R}^{n_R\times n_C}, \qquad n_R = \prod_{i=1}^{\mu} r_i, \quad n_C = \prod_{j=1}^{\nu} c_j.$$

The entry $t_{i_1,i_2,\ldots,i_k}$ of $\mathcal{T}$ is in the matrix $\mathcal{T}^{\mathcal{R}}$ in the row and column specified by multiindices

$$(r_1, r_2, \ldots, r_\mu) \quad \text{and} \quad (c_1, c_2, \ldots, c_\nu), \quad \text{respectively.}$$

Rows and columns are in $\mathcal{T}^{\mathcal{R}}$ sorted in the *inverse* lexicographical order w.r.t. their multiindices.

Generalized matricization

Examples

Clearly, in general

$$(\mathcal{T}^{\mathscr{R}})^{\mathsf{T}} = \mathcal{T}^{\mathscr{C}}.$$

For our $4 \times 3 \times 2$ tensor,

$$\mathcal{T}^{\{1\}} = \left[\begin{array}{ccc|ccc} 6 & 6 & 2 & 6 & 4 & 1 \\ 7 & 1 & 0 & 3 & 3 & 4 \\ 7 & 7 & 0 & 9 & 7 & 4 \\ 3 & 0 & 8 & 0 & 7 & 6 \end{array}\right] = (\mathcal{T}^{\{2,3\}})^{\mathsf{T}},$$

$$\mathcal{T}^{\{2\}} = \left[\begin{array}{cccc|cccc} 6 & 7 & 7 & 3 & 6 & 3 & 9 & 0 \\ 6 & 1 & 7 & 0 & 4 & 3 & 7 & 7 \\ 2 & 0 & 0 & 8 & 1 & 4 & 4 & 6 \end{array}\right] = (\mathcal{T}^{\{1,3\}})^{\mathsf{T}},$$

$$\mathcal{T}^{\{3\}} = \left[\begin{array}{cccc|cccc|cccc} 6 & 7 & 7 & 3 & 6 & 1 & 7 & 0 & 2 & 0 & 0 & 8 \\ 6 & 3 & 9 & 0 & 4 & 3 & 7 & 7 & 1 & 4 & 4 & 6 \end{array}\right] = (\mathcal{T}^{\{1,2\}})^{\mathsf{T}}.$$

But there are two more matricizations...

Generalized matricization

Examples

The last two case for 3-way tensor are for $\mathscr{R} = \{1,2,3\}$ and $\varnothing$,

$$\mathcal{T}^{\{1,2,3\}} = \left[\begin{array}{c} t_{1,1,1} \\ t_{2,1,1} \\ t_{3,1,1} \\ t_{4,1,1} \\ \hline t_{1,2,1} \\ t_{2,2,1} \\ t_{3,2,1} \\ t_{4,2,1} \\ \hline t_{1,3,1} \\ t_{2,3,1} \\ t_{3,3,1} \\ t_{4,3,1} \\ \hline\hline t_{1,1,2} \\ t_{2,1,2} \\ t_{3,1,2} \\ t_{4,1,2} \\ \hline t_{1,2,2} \\ t_{2,2,2} \\ t_{3,2,2} \\ t_{4,2,2} \\ \hline t_{1,3,2} \\ t_{2,3,2} \\ t_{3,3,2} \\ t_{4,3,2} \end{array}\right] = \left[\begin{array}{c} 6 \\ 7 \\ 7 \\ 3 \\ \hline 6 \\ 1 \\ 7 \\ 0 \\ \hline 2 \\ 0 \\ 0 \\ 8 \\ \hline\hline 6 \\ 3 \\ 9 \\ 0 \\ \hline 4 \\ 3 \\ 7 \\ 7 \\ \hline 1 \\ 4 \\ 4 \\ 6 \end{array}\right] = (\mathcal{T}^{\varnothing})^{\mathsf{T}} \equiv \mathrm{vec}(\mathcal{T}).$$

We call this $\uparrow$ the vectorization of a tensor (or matrix).

Generalized matricization

Matricization–vectorization relation

Recall that the ℓ-mode matricization is a matrix that contain the ℓ-mode fibres as columns (particularly sorted).

The rows of ℓ-mode matricization are then vectorizations of ℓ-mode co-fibres.

In our case, columns of $\mathcal{T}^{\{1\}}$ are the 1-mode fibres (columns) of $\mathcal{T}$,

$$\mathcal{T}^{\{1\}} = [\mathcal{T}_{\star,1,1}, \mathcal{T}_{\star,2,1}, \mathcal{T}_{\star,3,1}, \mathcal{T}_{\star,1,2}, \mathcal{T}_{\star,2,2}, \mathcal{T}_{\star,3,2}].$$

and rows of $\mathcal{T}^{\{1\}}$ (i.e., transposed columns of $\mathcal{T}^{\{2,3\}}$) are the transposed vectorizations of the 1-mode co-fibrer (i.e., actually the $(2,3)$-slices (the horizontal slices)) of $\mathcal{T}$.

Note on transposition

The matrix transposition

$$A \in \mathbb{R}^{m \times n} \quad \longmapsto \quad A^{\mathsf{T}} \in \mathbb{R}^{n \times m}$$

exchanges the roles of columns (1-mode) and rows (2-mode fib's).

Tensors can be manipulated in a similar fashion, in general, by an arbitrary permutation of roles of individual fibres. Let

$$\Pi = \begin{pmatrix} 1 & 2 & \cdots & k \\ \pi(1) & \pi(2) & \cdots & \pi(k) \end{pmatrix},$$

then

$$\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k} \quad \longmapsto \quad \mathcal{T}^{\Pi} \in \mathbb{R}^{n_{\pi(1)} \times n_{\pi(2)} \times \cdots \times n_{\pi(k)}},$$

$$(\mathcal{T}^{\Pi})_{i_1,i_2,\dots,i_k} = t_{i_{\pi(1)},i_{\pi(2)},\dots,i_{\pi(k)}}.$$

Norm and scalar product of tensors

We use the simplest available norm

$$\|\mathcal{T}\| = \Big(\textstyle\sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_k=1}^{n_k} |t_{j_1,j_2,\dots,j_k}|^2 \Big)^{\frac{1}{2}} = \big(\mathrm{vec}(\mathcal{T})^{\mathsf{T}} \mathrm{vec}(\mathcal{T}) \big)^{\frac{1}{2}}$$

which directly generalizes the standard

- Euclidean norm of vectors and
- Frobenius norm of matrices.

Moreover, it is induced by the inner product

$$\langle \mathcal{T}, \mathcal{S} \rangle = \textstyle\sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_k=1}^{n_k} s_{j_1,j_2,\dots,j_k} \cdot t_{j_1,j_2,\dots,j_k} = \mathrm{vec}(\mathcal{S})^{\mathsf{T}} \mathrm{vec}(\mathcal{T})$$

which directly generalizes the standard

- Euclidean scalar product of vectors $\langle x, y \rangle = y^{\mathsf{T}} x$ and
- commonly used scalar prod. of matrices $\langle A, B \rangle = \mathrm{trace}(B^{\mathsf{T}} A)$.

Rank of a tensor

Rank of a matrix

Let start gently...

What is the rank of a matrix $A \in \mathbb{R}^{m \times n}$?

- The order of the largerst nonzero minor of A ;-).
- The maximal number of linearly independent columns of A.
- The maximal number of linearly independent rows of A.
- The minimal number of pairs $(x_j, y_j) \in \mathbb{R}^m \times \mathbb{R}^n$, such that
$$A = x_1 y_1^\mathsf{T} + x_2 y_2^\mathsf{T} + \cdots \; = \textstyle\sum_\varrho x_\varrho y_\varrho^\mathsf{T},$$
i.e., the length of the shortest dyadic expansion of A.

Note that the SVD of A serves the shortest dyadic expansion with mutually orthogon(norm)al x_ϱ's and y_ϱ's.

Number of linearly independent fibres...

The ℓ-rank

Since columns and rows are the 1-mode and 2-mode fibres of a matrix, there is a straightforward generalization:

The ℓ-mode rank of the tensor $\mathcal{T}$ is the maximal number of linearly independent ℓ-mode fibres, i.e.,

$$\operatorname{rank}_{\{\ell\}}(\mathcal{T}) \equiv \operatorname{rank}(\mathcal{T}^{\{\ell\}}), \quad \mathcal{T}^{\{\ell\}} \in \mathbb{R}^{n_\ell \times (N/n_\ell)}, \quad N = \textstyle\prod_{\kappa=1}^{k} n_\kappa.$$

Since $\mathcal{T}^{\{\ell\}}$ is a matrix, whose rows are transposed vectorizations of ℓ-mode co-fibres, we get:

the maximal number of linearly independent ℓ-mode fibres
= the maximal number of linearly independent ℓ-mode co-fibres.

Recall that for $k = 2$ (in the matrix case), the 1-mode co-fibres are the 2-mode fibres (rows) and vice versa.

Number of linearly independent fibres...

The vector rank of tensor

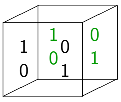
Consequently, for $\ell \neq ß$, there is no direct relation between

$$\operatorname{rank}_{\{\ell\}}(\mathcal{T}) \quad \text{and} \quad \operatorname{rank}_{\{ß\}}(\mathcal{T}).$$

The different-mode ranks may be different. Therefore we introduce the vector rank of the tensor,

$$\overrightarrow{\operatorname{rank}}(\mathcal{T}) \equiv (\operatorname{rank}_{\{1\}}(\mathcal{T}), \operatorname{rank}_{\{1\}}(\mathcal{T}), \ldots, \operatorname{rank}_{\{k\}}(\mathcal{T})).$$

For example

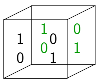
$\mathcal{T} =$ 

$\in \mathbb{R}^{2\times2\times2}$ is of $\overrightarrow{\operatorname{rank}}(\mathcal{T}) = (2, 2, 1)$.

Number of linearly independent fibres...

The vector rank of tensor

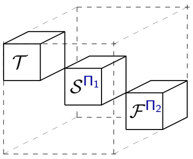
Consider now three of such vectors but of diferent dimensions,

$\mathcal{T} =$ 

$\in \mathbb{R}^{2\times2\times2}$ and similarly $\mathcal{S} \in \mathbb{R}^{3\times3\times3}$, $\mathcal{F} \in \mathbb{R}^{4\times4\times4}$,

i.e., $\overrightarrow{\operatorname{rank}}(\mathcal{T}) = (2,2,1)$, $\overrightarrow{\operatorname{rank}}(\mathcal{S}) = (3,3,1)$, $\overrightarrow{\operatorname{rank}}(\mathcal{F}) = (4,4,1)$.
Their permutations and direct sum (i.e., block-diagonal assembly),

$$\operatorname{diag}_3\left(\mathcal{T}, \mathcal{S}^{\left(\begin{smallmatrix}1&2&3\\3&1&2\end{smallmatrix}\right)}, \mathcal{F}^{\left(\begin{smallmatrix}1&2&3\\2&3&1\end{smallmatrix}\right)}\right) \equiv$$
$$\mathcal{T} \oplus \mathcal{S}^{\left(\begin{smallmatrix}1&2&3\\3&1&2\end{smallmatrix}\right)} \oplus \mathcal{F}^{\left(\begin{smallmatrix}1&2&3\\2&3&1\end{smallmatrix}\right)} =$$



,

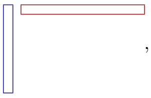
is of vector rank $(2,2,1)+(3,1,3)+(1,4,4) = (6,7,8)$.

Shortest polyadic expansion

Polyadic rank of a tensor

Any matrix A, $r \equiv \operatorname{rank}(A)$ can be written in the dyadic expansion,

$$A = x_1 y_1^\mathsf{T} + x_2 y_2^\mathsf{T} + \cdots \; = \textstyle\sum_{\varrho=1}^{r} x_\varrho y_\varrho^\mathsf{T}, \qquad \text{where}$$

$A_\varrho \equiv x_\varrho {y_\varrho}^\mathsf{T} =$  , $\quad (A_\varrho)_{i,j} = (x_\rho)_i \cdot (y_\rho)_j$

is the rank-one matrix—the outer product of two vectors

This motivates the polyadic expansion of k-way tensor as the sum of rank-one terms—the outer products of k vectors; e.g., for $k = 3$

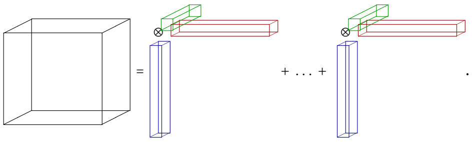
$$\mathcal{T}_\varrho \equiv (x_\varrho, y_\varrho, z_\varrho)_\otimes, \quad \text{where} \quad x_\varrho \in \mathbb{R}^{n_1}, \quad y_\varrho \in \mathbb{R}^{n_2}, \quad z_\varrho \in \mathbb{R}^{n_3},$$

$$(\mathcal{T}_\varrho)_{i_1,i_2,i_3} = (x_\rho)_{i_1} \cdot (y_\rho)_{i_2} \cdot (z_\rho)_{i_3}\,.$$

Shortest polyadic expansion

Polyadic rank of a tensor

Then the polyadic expansion takes form $\mathcal{T} = \sum_\varrho (x_\varrho, y_\varrho, z_\varrho)_\otimes$,



It represents our first kind of tensor decomposition into three matrices $X = [x_1, x_2, \ldots] \in \mathbb{R}^{n_1 \times ?}$, $Y = [y_1, y_2, \ldots] \in \mathbb{R}^{n_2 \times ?}$, $Z = [z_1, z_2, \ldots] \in \mathbb{R}^{n_3 \times ?}$.

This decomposition is intensively studied and it is known under names CanDeComp (Canonic DeComposition), ParaFac (Paralel Factorization), or CP decomposition (CanDeComp–ParaFac).

Shortest polyadic expansion

Polyadic rank of a tensor

In the case of matrices:

- The polyadic expansion can be done in such a way that both $X \in \mathbb{R}^{n \times r}$ and $Y \in \mathbb{R}^{m \times r}$ have orthogon(norm)al columns (via the SVD).
- Rank of A is the minimal number of terms (length of the shortest dyadic exp.).
- The Eckart–Young–Mirsky theorem shows that the difference between A and its approximation obtained by employing only q terms, $q < r = \operatorname{rank}(A)$, i.e., the approximation error has nonzero minimum, equal to $\sigma_r(A)$, the smallest nonzero singular value of A.

What about tensors?

Shortest polyadic expansion

Polyadic rank of a tensor

We can play with the orthogonality by employing QR decomp's of X, Y, Z, etc. It will be *briefly* mentioned later.

The number of rank-one terms is bounded by N, thus there is the minimal number, defining the polyadic rank,

$$\max_{\ell=1,2,\dots,k} \operatorname{rank}_{\{\ell\}}(\mathcal{T}) \le \operatorname{polyrank}(\mathcal{T}) \le \operatorname{nnz}(\mathcal{T}) \le N = n_1 \cdot n_2 \cdot \dots \cdot n_k .$$

This rank, however, is **not** robust. Let

$X = [x', x', x''] \in \mathbb{R}^{n_1 \times 3}$, $Y = [y', y'', y'] \in \mathbb{R}^{n_2 \times 3}$, $Z = [z'', z', z'] \in \mathbb{R}^{n_3 \times 3}$,

and $\operatorname{rank}(X) = \operatorname{rank}(Y) = \operatorname{rank}(Z) = 2$. Consider

$$\mathcal{T} = (x', y', z'')_\otimes + (x', y'', z')_\otimes + (x'', y', z')_\otimes ,$$

$$\mathcal{T}_\varepsilon = \frac{1}{\varepsilon}(x' + \varepsilon x'', y' + \varepsilon y'', z' + \varepsilon z'')_\otimes - \frac{1}{\varepsilon}(x', y', z')_\otimes , \quad \text{then}$$

$$\|\mathcal{T} - \mathcal{T}_\varepsilon\| = \varepsilon \|(x'', y'', z')_\otimes + (x'', y', z'')_\otimes + (x', y'', z'')_\otimes + \varepsilon (x'', y'', z'')_\otimes\| .$$

[*P. Paatero*, J. of Chemometrics 14(3), pp. 285–299, 2000].

Sum of rank-one terms

Another generalization of dyadic expansion

Note that rank-one (rank-at-most-one) terms

$$(x_\varrho, y_\varrho)_\otimes = xy^{\mathsf{T}}, \quad (x_\varrho, y_\varrho, z_\varrho)_\otimes, \quad x_\varrho \in \mathbb{R}^{n_1},\ y_\varrho \in \mathbb{R}^{n_2},\ z_\varrho \in \mathbb{R}^{n_3},$$

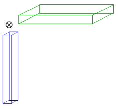
form submanifolds witin $\mathbb{R}^{n_1 \times n_2}$ and $\mathbb{R}^{n_1 \times n_2 \times n_3}$, respectively.

We can take another suitable submanifold and its members consider to be the rank-one terms. For example,

$$\mathcal{T}_\varrho = (x_\varrho, M_\varrho)_\otimes, \quad \text{where} \quad x_\varrho \in \mathbb{R}^{n_1},\ M_\varrho \in \mathbb{R}^{n_2 \times n_3},$$

and $(\mathcal{T}_\varrho)_{i_1,i_2,i_3} = (x_\varrho)_{i_1} \cdot (M_\varrho)_{i_2,i_3}$.

Then rank of $\mathcal{T}$ can be defined as the length of shortest sum

$\mathcal{T} = \sum_\varrho \mathcal{T}_\varrho = \sum_\varrho$  ; this rank $= \operatorname{rank}_{\{1\}}(\mathcal{T}) = \operatorname{rank}(T^{\{1}$

Another example

4-way tensor & the Kronecker product

Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$ and $\mathcal{T} = \sum_\varrho \mathcal{T}_\varrho$, where

$$\mathcal{T}_\varrho \equiv (K_\varrho, M_\varrho)_\otimes \quad \text{such that} \quad (\mathcal{T}_\varrho)_{i_1,i_2,i_3,i_4} = (K_\varrho)_{i_1,i_2} \cdot (M_\varrho)_{i_3,i_4},$$

and $K_\varrho \in \mathbb{R}^{n_1 \times n_2}$, $M_\varrho \in \mathbb{R}^{n_3 \times n_4}$.

The length of the shortest sum can be observed after rearraging to

$$\mathcal{T}^{\{1,2\}} = \sum_\varrho \mathcal{T}_\varrho^{\{1,2\}} = \sum_\varrho \operatorname{vec}(K_\varrho)\big(\operatorname{vec}(M_\varrho)\big)^{\mathsf{T}} \in \mathbb{R}^{(n_1 \cdot n_2) \times (n_3 \cdot n_4)};$$

it is the rank of this matrix, in general $\operatorname{rank}_{\mathscr{R}}(\mathcal{T}) \equiv \operatorname{rank}(\mathcal{T}^{\mathscr{R}})$.

Note another rearranging gives

$$\mathcal{T}^{\{1,3\}} \in \mathbb{R}^{(n_1 \cdot n_3) \times (n_2 \cdot n_4)}, \quad \mathcal{T}^{\{1,3\}} = \sum_{\varrho=1}^{\operatorname{rank}_{\{1,2\}}(\mathcal{T})} M_\varrho \otimes K_\varrho ,$$

where $\otimes$ is the Kronecker product of matrices.

Note on Kronecker product

For matrices, the standard matrix and Kronecker products we have

$$(AB) \otimes (CD) = (A \otimes C)(B \otimes D).$$

Thus, if any two of the following three matrices

$$A, \quad C, \quad E = A \otimes C$$

are invertible, then the third is also invertible.

We can intepret E as the $\{1,3\}$-matricization of a 4-way tensor $\mathcal{E}$, i.e., $E = \mathcal{E}^{\{1,3\}} = A \otimes C$. Then its $\{1,2\}$-matricization takes form

$$\mathcal{E}^{\{1,2\}} = \operatorname{vec}(A)\big(\operatorname{vec}(C)\big)^{\mathsf{T}}$$

All three $\mathcal{E}$, $\mathcal{E}^{\{1,3\}}$, $\mathcal{E}^{\{1,2\}}$ represent the same rank-one object (just differently rearranged) in the given submanifold of 4-way tensors.

But $\mathcal{E}^{\{1,3\}}$ may be invertible whereas $\operatorname{rank}(\mathcal{E}^{\{1,2\}}) = 1$ always.

Final note on ranks

For a given tensor $\mathcal{T}$, we have

- $\operatorname{rank}_{\{\ell\}}(\mathcal{T}) \equiv \operatorname{rank}(\mathcal{T}^{\{\ell\}}) \quad \text{for} \quad \ell = 1, 2, \dots, k,$
- $\overrightarrow{\operatorname{rank}}(\mathcal{T}) \equiv \big(\operatorname{rank}_{\{1\}}(\mathcal{T}), \operatorname{rank}_{\{2\}}(\mathcal{T}), \dots, \operatorname{rank}_{\{k\}}(\mathcal{T})\big),$
- $\operatorname{rank}_{\mathscr{R}}(\mathcal{T}) \equiv \operatorname{rank}(\mathcal{T}^{\mathscr{R}}) \quad \text{for} \quad \mathscr{R} \subseteq \{1, 2, \dots, k\},$
- clearly

  $$\{\operatorname{rank}_{\{\ell\}}(\mathcal{T}),\ \ell = 1, 2, \dots, k\} \subseteq \{\operatorname{rank}_{\mathscr{R}}(\mathcal{T}),\ \mathscr{R} \subseteq \{1, 2, \dots, k\}\},$$
- $\operatorname{polyrank}(\mathcal{T})$:

  $$\max_{\mathscr{R} \subseteq \{1,2,\dots,k\}} \operatorname{rank}_{\mathscr{R}}(\mathcal{T}) \overset{(*)}{\le} \operatorname{polyrank}(\mathcal{T});$$

$(*)$
$$\big((x', y', z'')_\otimes + (x', y'', z')_\otimes + (x'', y', z')_\otimes\big)^{\{1,2\}}$$
$$= \big[(y' \otimes x'), (y'' \otimes x') + (y' \otimes x'')\big]\big[z'', z'\big]^{\mathsf{T}}.$$

Tensor arithmetics

Basic operations

Linear combinations, direct sum, outer product

We already know some basic operations.

- Since tensors of the given fixed dimensions form a linear vector space, we can do componentwisely

$$\alpha\mathcal{T},\quad \mathcal{T}+\mathcal{S},\quad \alpha\mathcal{T}+\beta\mathcal{S},\quad \textstyle\sum_\ell \alpha_\ell \mathcal{T}_\ell\,.$$

- We can do the direct sum of tensors of the same$^{(?!)}$ order k

$$\mathcal{T}\oplus\mathcal{S} = \mathrm{diag}_k(\mathcal{T},\mathcal{S}) \in \mathbb{R}^{(n_1+m_1)\times(n_2+m_2)\times\cdots\times(n_k+m_k)}.$$

- We can do the outer product (a.k.a. tensor or Kronecker p.) of any two (or more) tensors

$$\mathcal{S}\otimes\mathcal{T} = (\mathcal{T},\mathcal{S})_\otimes \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k\times m_1\times m_2\times\cdots\times m_t}$$

$$(\mathcal{S}\otimes\mathcal{T})_{i_1,i_2,\ldots,i_k,j_1,j_2,\ldots,j_t} = (\mathcal{T})_{i_1,i_2,\ldots,i_k}\cdot(\mathcal{S})_{j_1,j_2,\ldots,j_t}$$

$$(\mathcal{S}\otimes\mathcal{T})^{\{i_1,i_2,\ldots,i_k\}} = \mathrm{vec}(\mathcal{T})\left(\mathrm{vec}(\mathcal{S})\right)^{\mathsf{T}}$$

Multiplication: Tensor-matrix (TM) product

The basic structure of TM is the same as for matrices: Sums of products of individual entries of given fibres and col's or rows. Let

$$\mathcal{T}\in\mathbb{R}^{n_1,n_2,\ldots,n_k},\quad S\in\mathbb{R}^{c\times n_\ell},\quad M\in\mathbb{R}^{n_\ell\times d}.$$

The ℓ-mode (pre-/post-)multiplication of tensor by a matrix

$$S\times_\ell\mathcal{T}\in\mathbb{R}^{n_1,\ldots,n_{\ell-1},c,n_{\ell+1},\ldots,n_k},\qquad \mathcal{T}\;{}_\ell{\times}\,M\in\mathbb{R}^{n_1,\ldots,n_{\ell-1},d,n_{\ell+1},\ldots,n_k}$$

is defined as

$$(S\times_\ell\mathcal{T})_{i_1,\ldots,i_{\ell-1},j,i_{\ell+1},\ldots,i_k} \equiv \textstyle\sum_{i_\ell=1}^{n_\ell}(S)_{j,i_\ell}\cdot(\mathcal{T})_{i_1,\ldots,i_{\ell-1},i_\ell,i_{\ell+1},\ldots,i_k},$$
$$(\mathcal{T}\;{}_\ell{\times}\,M)_{i_1,\ldots,i_{\ell-1},j,i_{\ell+1},\ldots,i_k} \equiv \textstyle\sum_{i_\ell=1}^{n_\ell}(\mathcal{T})_{i_1,\ldots,i_{\ell-1},i_\ell,i_{\ell+1},\ldots,i_k}\cdot(M)_{i_\ell,j}.$$

Clearly $\mathcal{T}\;{}_\ell{\times}\,M = M^{\mathsf{T}}\times_\ell\mathcal{T}$, thus we focus on the pre-multiplication. (The so-called Einstein's notation omits the 'sum' signs.)

Multiplication: Tensor-matrix (TM) product

We can see it as MV-product of S with all the ℓ-mode fibres, i.e.,

$$(S\times_\ell\mathcal{T})^{\{\ell\}} = S\mathcal{T}^{\{\ell\}}\in\mathbb{R}^{c\times((\prod_{\kappa=1^k}n_\kappa)/n_\ell)}.$$

Tensor-matrix product is associative in the following two meanings

$$P\times_\ell(S\times_\ell\mathcal{T}) = (PS)\times_\ell\mathcal{T}$$

$$P\times_\tau(S\times_ß\mathcal{T}) = S\times_ß(P\times_\tau\mathcal{T}),\quad\text{for}\quad \tau\neq ß.$$

Multiplication by two matrices in two different modes can be again rearranged by matricization as follows:

$$\left(P\times_\tau(S\times_ß\mathcal{T})\right)^{\{\tau,ß\}} = (S\otimes P)\mathcal{T}^{\{\tau,ß\}}\quad\text{or}\quad(P\otimes S)\mathcal{T}^{\{\tau,ß\}}$$

for $\tau<ß$, or $ß>\tau$, respectively (recall the inverse *lexicographical* ordering of multiindices while matricization).

Linear transformation of a tensor

Employing the associativity while multiplication in different modes, we get for

$$\mathcal{T}\in\mathbb{R}^{n_1,n_2,\ldots,n_k},\quad S_\kappa\in\mathbb{R}^{c_\kappa\times n_\kappa},\quad \kappa=1,2,\ldots,k,$$

$$(S_1,S_2,\ldots,S_k\,|\,\mathcal{T}) \equiv S_1\times_1(S_2\times_2(\cdots(S_k\times_k\mathcal{T})\cdots))\in\mathbb{R}^{c_1,c_2,\ldots,c_k}$$

a general linear transformation of $\mathcal{T}$. In the post-mult. fashion it takes form $(\mathcal{T}\,|\,M_1,M_2,\ldots,M_k)$ for $M_\kappa\in\mathbb{R}^{n_\kappa\times d_\kappa}$.

A single tensor-matrix product can be written as

$$P\times_\ell\mathcal{T} = (I_{n_1},\ldots,I_{n_{\ell-1}},P,I_{n_{\ell+1}},\ldots,I_{n_k}\,|\,\mathcal{T}).$$

Employing vectorization gives

$$\mathrm{vec}\big((S_1,S_2,\ldots,S_k\,|\,\mathcal{T})\big) = (S_k\otimes\cdots\otimes S_2\otimes S_1)\,\mathrm{vec}(\mathcal{T});$$

recall that $\mathrm{vec}(\mathcal{T}) = \mathcal{T}^{\{1,2,\ldots,k\}}$.

Note on tensors of order two

Matrix-matrix product treated as tensor-matrix

First note that $A^{\{1\}}=A$, $A^{\{2\}}=A^{\mathsf{T}}$. Since:

$$\begin{aligned}(S_1\times_1 A)^{\{1\}} &= S_1A^{\{1\}}, &&\text{then}\quad S_1\times_1 A = S_1A,\\ (S_2\times_2 A)^{\{2\}} &= S_2A^{\{2\}}, &&\text{then}\quad S_2\times_2 A = AS_2^{\mathsf{T}},\\ (S_1,S_2\,|\,A) &= S_1\times_1(S_2\times_2 A), &&\text{then}\quad (S_1,S_2\,|\,A) = S_1AS_2^{\mathsf{T}},\end{aligned}$$

for the pre-multiplication and

$$\begin{aligned}A\;{}_1{\times}\,M_1 &= M_1^{\mathsf{T}}\times_1 A, &&\text{then}\quad A\;{}_1{\times}\,M_1 = M_1^{\mathsf{T}}A,\\ A\;{}_2{\times}\,M_2 &= M_2^{\mathsf{T}}\times_2 A, &&\text{then}\quad A\;{}_2{\times}\,M_2 = AM_2,\\ (A\,|\,M_1,M_2) &= (A\;{}_1{\times}\,M_1)\;{}_2{\times}\,M_2, &&\text{then}\quad (A\,|\,M_1,M_2) = M_1^{\mathsf{T}}AM_2,\end{aligned}$$

for the post-mutliplication.

For tensors of order one (vectors): $S_1\times_1 v = S_1v$, $\;v\;{}_1{\times}\,M_1 = M_1^{\mathsf{T}}v$.

Tensor-tensor (TT) product a.k.a. Contraction

Let $\mathcal{T}$ and tF be tensors of orders k and s,

$$\mathcal{T} \in \mathbb{R}^{n_1,n_2,\dots,n_k}, \quad \mathcal{F} \in \mathbb{R}^{m_1,m_2,\dots,m_s}, \quad \text{and} \quad n_\ell = m_{\text{ß}}.$$

Then their $(\ell, \text{ß})$-mode product is a tensor of order $(k + s - 2)$,

$$\mathcal{T} \times_{(\ell,\text{ß})} \mathcal{F} \in \mathbb{R}^{n_1,\dots,n_{\ell-1},n_{\ell+1},\dots,n_k,m_1,\dots,m_{\text{ß}-1},m_{\text{ß}+1},\dots,m_s},$$

where $\left(\mathcal{T} \times_{(\ell,\text{ß})} \mathcal{F}\right)_{i_1,\dots,i_{\ell-1},i_{\ell+1},\dots,i_k,j_1,\dots,j_{\text{ß}-1},j_{\text{ß}+1},\dots,j_s}$

$$= \sum_{\alpha=1}^{n_\ell} (\mathcal{T})_{i_1,\dots,i_{\ell-1},\alpha,i_{\ell+1},\dots,i_k} \cdot (\mathcal{F})_{j_1,\dots,j_{\text{ß}-1},\alpha,j_{\text{ß}+1},\dots,}$$

The other available product is

$$\mathcal{F} \times_{(\text{ß},\ell)} \mathcal{T} = \left(\mathcal{T} \times_{(\ell,\text{ß})} \mathcal{F}\right)^{\Pi}, \quad \text{where} \quad \Pi = \begin{pmatrix} 1 & 2 & \cdots & & & & \cdots & k{+}s{-}2 \\ k & k{+}1 & \cdots & k{+}s{-}2 & 1 & 2 & \cdots & k{-}1 \end{pmatrix}.$$

Alternatively

$$\left(\mathcal{T} \times_{(\ell,\text{ß})} \mathcal{F}\right)^{\{1,2,\dots,k-1\}} = (\mathcal{T}^{\{\ell\}})^{\mathsf{T}} \mathcal{F}^{\{\text{ß}\}},$$
$$\left(\mathcal{F} \times_{(\text{ß},\ell)} \mathcal{T}\right)^{\{1,2,\dots,s-1\}} = (\mathcal{F}^{\{\text{ß}\}})^{\mathsf{T}} \mathcal{T}^{\{\ell\}}.$$

Tensor-tensor (TT) product a.k.a. Contraction

Analogously, we can introduce mutiplication (contraction) in two pairs of indices at once. For

$$\mathcal{T} \in \mathbb{R}^{n_1,n_2,\dots,n_k}, \quad \mathcal{F} \in \mathbb{R}^{m_1,m_2,\dots,m_s}, \quad \text{and} \quad n_\ell = m_{\text{ß}},\ n_\tau = m_\sigma,\ \ell < \tau,$$

we get the $(k + s - 4)$-way tensor

$$\mathcal{T} \times_{((\ell,\tau),(\text{ß},\sigma))} \mathcal{F},$$

with entries (depending on relations between ß and σ) either / or

$$\textstyle\sum_{\alpha\beta} (\mathcal{T})_{i_1,\dots,i_{\ell-1},\alpha,i_{\ell+1},\dots,i_{\tau-1},\beta,i_{\tau+1},\dots,i_k} \cdot (\mathcal{F})_{j_1,\dots,j_{\text{ß}-1},\alpha,j_{\text{ß}+1},\dots,j_{\sigma-1},\beta,j_{\sigma+1},\dots,j_s},$$
$$\textstyle\sum_{\alpha\beta} (\mathcal{T})_{i_1,\dots,i_{\ell-1},\alpha,i_{\ell+1},\dots,i_{\tau-1},\beta,i_{\tau+1},\dots,i_k} \cdot (\mathcal{F})_{j_1,\dots,j_{\sigma-1},\beta,j_{\sigma+1},\dots,j_{\text{ß}-1},\alpha,j_{\text{ß}+1},\dots,j_s}.$$

Again,

$$\left(\mathcal{T} \times_{((\ell,\tau),(\text{ß},\sigma))} \mathcal{F}\right)^{\{1,2,\dots,k-2\}} = (\mathcal{T}^{\{\ell,\tau\}})^{\mathsf{T}} (\mathcal{F}^{\Pi})^{\{\text{ß},\sigma\}},$$

and $\Pi = \mathrm{Id}$ or $\left(\begin{smallmatrix} \cdots & \sigma & \cdots & \text{ß} & \cdots \\ \cdots & \text{ß} & \cdots & \sigma & \cdots \end{smallmatrix}\right)$. Similarly for several pairs of indices.

MM- and TM-products as TT-products

If matrices treated as tensors

Note that TM and TT have different ordering of indices,

$$S \times_\ell \mathcal{T} = \left(S \times_{(2,\ell)} \mathcal{T}\right)^{\left(\begin{smallmatrix} 1 & 2 & \cdots & \ell & \ell+1 & \cdots \\ \ell & 1 & \cdots & \ell-1 & \ell+1 & \cdots \end{smallmatrix}\right)} = \left(\mathcal{T} \times_{(\ell,2)} S\right)^{\left(\begin{smallmatrix} \cdots & \ell-1 & \ell & \cdots & k-1 & k \\ \cdots & \ell-1 & \ell+1 & \cdots & k & \ell \end{smallmatrix}\right)},$$

$$\mathcal{T} {}_\ell\!\times M = M^{\mathsf{T}} \times_\ell \mathcal{T} = \left(M \times_{(1,\ell)} \mathcal{T}\right)^{\Pi} = \left(\mathcal{T} \times_{(\ell,1)} M\right)^{\Pi}.$$

For MM-products we get

$$\begin{aligned} AB &= A \times_{(2,1)} B = A^{\mathsf{T}} \times_{(1,1)} B = A \times_{(2,2)} B^{\mathsf{T}} = A^{\mathsf{T}} \times_{(1,2)} B^{\mathsf{T}} \\ &= (B \times_{(1,2)} A)^{\Pi} = (B \times_{(1,1)} A^{\mathsf{T}})^{\Pi} = (B^{\mathsf{T}} \times_{(2,2)} A)^{\Pi} = (B^{\mathsf{T}} \times_{(2,1)} A^{\mathsf{T}}) \\ &= (B^{\mathsf{T}} A^{\mathsf{T}}) \end{aligned}$$

where $\Pi = \left(\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right)$. Similarly for

$$A^{\mathsf{T}} B = A \times_{(1,1)} B = \dots,\ AB^{\mathsf{T}} = A \times_{(2,2)} B = \dots,\ A^{\mathsf{T}} B^{\mathsf{T}} = A \times_{(1,2)} B = ..$$

Relation between outer and tensor product

Recall that a vector can be interpreted as a single-column matrix, a matrix as a single-front-slice 3-way tensor, etc.

We formalize that in the form of 'uparrow' operator

$$\begin{aligned} \uparrow: \quad v \in \mathbb{R}^n &\longmapsto v^{\uparrow} \in \mathbb{R}^{n\times 1}, \\ A \in \mathbb{R}^{n\times d} &\longmapsto A^{\uparrow} \in \mathbb{R}^{n\times d\times 1}, \\ \uparrow^2 = \uparrow\uparrow: \quad v \in \mathbb{R}^n &\longmapsto v^{\uparrow\uparrow} \in \mathbb{R}^{n\times 1\times 1}, \end{aligned}$$

etc.

Then for a k-way tensor $\mathcal{T}$ and s-way tensor $\mathcal{F}$ we have

$$(\mathcal{T}, \mathcal{F})_{\otimes} = (\mathcal{T}^{\uparrow}) \times_{(k+1,s+1)} (\mathcal{F}^{\uparrow}).$$

Note again:
The outer product is a.k.a. tensor and Kronecker product.
The tensor (TT) product is a.k.a. contraction.

Basic decompositions of a tensor

Singular value decomposition (SVD)

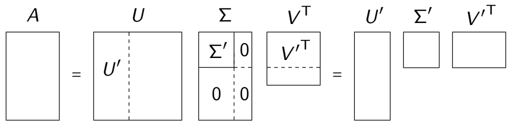
Let start with matrices

Let $A \in \mathbb{R}^{m\times n}$ be a matrix of rank $r = \mathrm{rank}(A)$, then

$$A = U\Sigma V^{\mathsf{T}} = (U, V \,|\, \Sigma) = U'\Sigma' V'^{\mathsf{T}} = (U', V' \,|\, \Sigma')$$

$$\begin{aligned} \text{where} \quad U^{-1} &= U^{\mathsf{T}}, \quad U = [\, U', U'' \,] \in \mathbb{R}^{m\times m}, \quad U' \in \mathbb{R}^{m\times r}, \\ V^{-1} &= V^{\mathsf{T}}, \quad V = [\, V', V'' \,] \in \mathbb{R}^{n\times n}, \quad V' \in \mathbb{R}^{n\times r}, \end{aligned}$$

$$\Sigma = \begin{bmatrix} \Sigma' & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{m\times n}, \qquad \Sigma' = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r\times r},$$
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$



SVDs of ℓ-mode matricizations

Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}$ of $\overrightarrow{\mathrm{rank}}(\mathcal{T}) \equiv (r_1, r_2, \ldots, r_k)$, where

$r_\ell = \mathrm{rank}_{\{\ell\}}(\mathcal{T}) = \mathrm{rank}(\mathcal{T}^{\{\ell\}})$, $\quad \mathcal{T}^{\{\ell\}} \in \mathbb{R}^{n_\ell \times (N/n_\ell)}$, $\quad N = n_1 \cdot n_2 \cdot \cdots \cdot n_k$

Consider then the SVDs

$$\mathcal{T}^{\{\ell\}} = U_\ell \Sigma_\ell V_\ell^\mathsf{T} = U_\ell' \Sigma_\ell' V_\ell'^\mathsf{T}$$

where $U_\ell = [U_\ell', \, U_\ell''] \in \mathbb{R}^{n_\ell \times n_\ell}$, $U_\ell' \in \mathbb{R}^{n_\ell \times r_\ell}$,

$\Sigma_\ell' = \mathrm{diag}(\sigma_{1,\ell}, \sigma_{2,\ell}, \ldots, \sigma_{r_\ell,\ell}) \in \mathbb{R}^{r_\ell \times r_\ell}$, $\quad \sigma_{1,\ell} \geq \sigma_{2,\ell} \geq \cdots \geq \sigma_{r_\ell,\ell} > 0$.

Thus

$$\begin{bmatrix} U_\ell'^\mathsf{T} \mathcal{T}^{\{\ell\}} \\ U_\ell''^\mathsf{T} \mathcal{T}^{\{\ell\}} \end{bmatrix} = U_\ell^\mathsf{T} \mathcal{T}^{\{\ell\}} = \Sigma_\ell V_\ell^\mathsf{T} = \begin{bmatrix} \Sigma_\ell' V_\ell'^\mathsf{T} \\ 0 \end{bmatrix}.$$

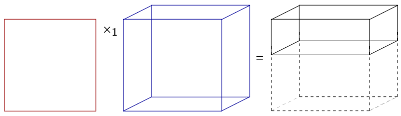
SVDs of ℓ-mode matricizations

Clearly, this is the ℓ-mode product,

$$\left(U_\ell^\mathsf{T} \times_\ell \mathcal{T}\right)^{\{\ell\}} = U_\ell^\mathsf{T} \mathcal{T}^{\{\ell\}} = \Sigma_\ell V_\ell^\mathsf{T} = \begin{bmatrix} \Sigma_\ell' V_\ell'^\mathsf{T} \\ 0 \end{bmatrix} \in \mathbb{R}^{n_\ell \times (N/n_\ell)},$$

$$\text{and} \quad \left(U_\ell'^\mathsf{T} \times_\ell \mathcal{T}\right)^{\{\ell\}} = U_\ell'^\mathsf{T} \mathcal{T}^{\{\ell\}} = \Sigma_\ell' V_\ell'^\mathsf{T} \in \mathbb{R}^{r_\ell \times (N/n_\ell)}.$$

For a three-way tensor and $\ell = 1$:



Note that mutliplication by other $U_ß$s in the other modes ($ß \neq \ell$) does not involve these already made zero co-fibres.

Tucker decompostion a.k.a. high-order SVD (HOSVD)

Finally we get for $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}$ a linear transformation

$$\left(U_1^\mathsf{T}, U_2^\mathsf{T}, \ldots, U_k^\mathsf{T} \,\middle|\, \mathcal{T}\right) = \mathrm{diag}_k(\mathcal{C}_\mathcal{T}, 0) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k},$$

where the subtensor

$$\mathcal{C}_\mathcal{T} = \left(U_1'^\mathsf{T}, U_2'^\mathsf{T}, \ldots, U_k'^\mathsf{T} \,\middle|\, \mathcal{T}\right) \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_k}$$

is called the Tucker core of tensor $\mathcal{T}$. Since U_ℓ's are invertible and orthogonal the first equation can be rearranged to

$$\mathcal{T} = \left(U_1, U_2, \ldots, U_k \,|\, \mathrm{diag}_k(\mathcal{C}_\mathcal{T}, 0)\right) = \left(U_1', U_2', \ldots, U_k' \,|\, \mathcal{C}_\mathcal{T}\right)$$

that is called the Tucker decomposition or HOSVD of tensor $\mathcal{T}$.

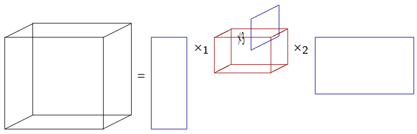
[*L. R. Tucker*, Psychometrika 31(3), pp. 279–311, 1966]

Tucker decompostion a.k.a. high-order SVD (HOSVD)

Thus, for $\mathcal{T}$ with $\overrightarrow{\mathrm{rank}}(r_1, r_2, \ldots, r_k)$ we have decomposition

$$\mathcal{T} = \left(U_1', U_2', \ldots, U_k' \,\middle|\, \mathcal{C}_\mathcal{T}\right), \qquad \mathcal{C}_\mathcal{T} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_k},$$

$$U_\ell' \in \mathbb{R}^{n_\ell \times r_\ell}, \quad U_\ell'^\mathsf{T} U_\ell' = I_{r_\ell}.$$

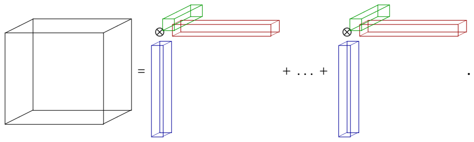


Moreover, the ℓ-mode co-fibres of $\mathcal{C}_\mathcal{T}$ are sorted in a nonincreasing sequence w.r.t. their norms equal to $\sigma_{1,\ell}, \sigma_{2,\ell}, \ldots, \sigma_{r_\ell,\ell}$.

This allows to generalize the Eckart–Young–Mirsky theorem. Compare with the SVD.

Polyadic expansion as the CP decompostion

Recall the polyadic decompostion of $\mathcal{T}$

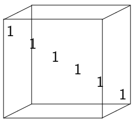


Collecting all the particular vectors into matrices

$$X_1 \in \mathbb{R}^{n_1 \times r}, \quad X_2 \in \mathbb{R}^{n_2 \times r}, \quad \ldots \quad X_k \in \mathbb{R}^{n_k \times r}$$

and using an "identity-like" cubic tensor of order k and dim's r,

$\mathcal{I}_{r,k} =$



$\in \mathbb{R}^{r \times r \times \cdots \times r}$, we get

$$\mathcal{T} = (X_1, X_2, \ldots, X_k \,|\, \mathcal{I}_{r,k}).$$

Comparison of both basic decompositions

Tucker decomposition (HOSVD)

$$\mathcal{T} = (U_1', U_2', \ldots, U_k' \,|\, \mathcal{C}_\mathcal{T})$$

- Matrices U_ℓ' with orthonormal columns (+)
- Different numbers of columns equal to $\mathrm{rank}_{\{\ell\}}(\mathcal{T})$ (±)
- Core of dimensions equal to $\overrightarrow{\mathrm{rank}}(\mathcal{T})$ with the norm "accumulated" in leading principal corner (+)

CP decoposition (CanDeComp, ParaFac)

$$\mathcal{T} = (X_1, X_2, \ldots, X_k \,|\, \mathcal{I}_{r,k})$$

- Matrices X_ℓ may have linearly dependent columns (−)
- The same number of columns equal to $\mathrm{polyrank}(\mathcal{T})$ (±)
- "Core tensor" is cubic with *very simple* structure; so simple it need not be stored (+ + +)

Note that both decompostitions have similar structure—an inner core tensor of (typically?) smaller dimensions than $\mathcal{T}$, surrounded by k matrices, also called leaves (from graph theory).

Low-rank arithmetics of tensors

Let start with matrices. SVD (re)compression

Let $A \in \mathbb{R}^{m\times n}$ be a (low-rank) matrix given in the form of product of two thin matrices $A = XY^\top$, or, in more general case of three

$$A = XSY^\top,\ \ X \in \mathbb{R}^{m\times p},\ m \gg p,\ \ S \in \mathbb{R}^{p\times q},\ \ Y \in \mathbb{R}^{n\times q},\ n \gg q.$$

Our goal is to compute its SVD without evaluating A:
Step 1: Compute economic QR decompositions of *thin* X and Y

$$X = Q_X R_X,\quad Q_X \in \mathbb{R}^{m\times r_X},\quad R_X \in \mathbb{R}^{r_X\times p},\quad r_X = \operatorname{rank}(X),$$
$$Y = Q_Y R_Y,\quad Q_Y \in \mathbb{R}^{n\times r_Y},\quad R_Y \in \mathbb{R}^{r_Y\times q},\quad r_Y = \operatorname{rank}(Y).$$

Thus $A = Q_X W Q_Y{}^\top$ where $W = R_X S R_Y{}^\top \in \mathbb{R}^{r_X\times r_Y}$.
Step 2: Compute the economic SVD of the *small* matrix W

$$W = U'_W \Sigma'_W V'_W{}^\top,\quad U'_W \in \mathbb{R}^{r_X\times r},\ \Sigma'_W \in \mathbb{R}^{r\times r},\ V'_W \in \mathbb{R}^{r_Y\times r}.$$

Thus $A = (Q_X U'_W)\,\Sigma'_W\,(Q_Y V'_W)^\top$.

Sum of two low-rank matrices

Let $A, B \in \mathbb{R}^{m\times n}$ be two low-rank matrices given the form of their economic SVDs,

$$A = U'_A \Sigma'_A V'_A{}^\top,\quad B = U'_B \Sigma'_B V'_B{}^\top,$$

with $r_A = \operatorname{rank}(A)$, $r_B = \operatorname{rank}(B)$.

Then

$$M = \varphi A + \psi B = \underbrace{[\,U'_A,\,U'_B\,]}_{X\in\mathbb{R}^{m\times(r_A+r_B)}} \underbrace{\begin{bmatrix} \varphi\Sigma'_A & 0 \\ 0 & \psi\Sigma'_B \end{bmatrix}}_{S\in\mathbb{R}^{(r_A+r_B)\times(r_A+r_B)}} \underbrace{[\,V'_A,\,V'_B\,]}_{Y\in\mathbb{R}^{n\times(r_A+r_B)}}{}^\top.$$

Compression then serves the economic SVD of M.

Product of low-rank matrix with another matrix

Let $A \in \mathbb{R}^{m\times n}$ be a low-rank matrix given the form of its economic SVD,

$$A = U'_A \Sigma'_A V'_A{}^\top.$$

If also B is a low-rank matrix given similarly, then

$$M = AB = \underbrace{U'_A}_{Q_X} \underbrace{\left(\Sigma'_A (V'_A{}^\top U'_B)\Sigma'_B\right)}_{W\in\mathbb{R}^{r_A\times r_B}} \underbrace{V'_B}_{Q_Y}{}^\top.$$

If B is a general matrix, then

$$M = AB = \underbrace{U'_A}_{Q_X}\ \underbrace{\Sigma'_A}_{R_X S}\ \underbrace{(B^\top V'_A)}_{Y}{}^\top.$$

Compression (which is already partially done) then serves the economic SVD of M.

And similarly for tensors: Compression

Let

$$\mathcal{T} = (X_1, X_2, \dots, X_k \,|\, \mathcal{S}) \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k},\quad \mathcal{S} \in \mathbb{R}^{p_1\times p_2\times\cdots\times p_k},\quad n_\ell \gg p_\ell$$

(e.g. the CP decomp. / polyadic exp., or another similar product).
Step 1: Compute k economic QR decomp's of *thin* $X_\ell = Q_\ell R_\ell$,

$$(X_1, X_2, \dots, X_k \,|\, \mathcal{S}) = \Big(Q_1, Q_2, \dots, Q_k \,\Big|\, \underbrace{(R_1, R_2, \dots, R_k \,|\, \mathcal{S})}_{\mathcal{W}}\Big).$$

Step 2: Compute the Tucker decomposition of *small* tensor $\mathcal{W}$,

$$\mathcal{W} = (U'_{1,\mathcal{W}}, U'_{2,\mathcal{W}}, \dots, U'_{k,\mathcal{W}} \,|\, \mathcal{C}_\mathcal{W}).$$

This gives

$$\mathcal{T} = \big(\underbrace{Q_1 U'_{1,\mathcal{W}}}_{U'_{1,\mathcal{T}}}, \underbrace{Q_2 U'_{2,\mathcal{W}}}_{U'_{2,\mathcal{T}}}, \dots, \underbrace{Q_k U'_{k,\mathcal{W}}}_{U'_{k,\mathcal{T}}} \,|\, \underbrace{\mathcal{C}_\mathcal{W}}_{\mathcal{C}_\mathcal{T}}\big)$$

the Tucker decomposition of large tensor $\mathcal{T}$.

Sum of two tensors

Let $\mathcal{T}, \mathcal{F} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_k}$ in Tucker form

$$\mathcal{T} = (U'_{1,\mathcal{T}}, U'_{2,\mathcal{T}}, \dots, U'_{k,\mathcal{T}} \,|\, \mathcal{C}_\mathcal{T}),\quad \mathcal{F} = (U'_{1,\mathcal{F}}, U'_{2,\mathcal{F}}, \dots, U'_{k,\mathcal{F}} \,|\, \mathcal{C}_\mathcal{F}).$$

Then

$$\mathcal{E} = \varphi\mathcal{T} + \psi\mathcal{F} = \Big(\underbrace{[U'_{1,\mathcal{T}}, U'_{1,\mathcal{F}}]}_{X_1}, \dots, \underbrace{[U'_{k,\mathcal{T}}, U'_{k,\mathcal{F}}]}_{X_k} \,\Big|\, \underbrace{\operatorname{diag}_k(\varphi\mathcal{C}_\mathcal{T}, \psi\mathcal{C}_\mathcal{F})}_{\mathcal{S}}\Big).$$

The compression then yields the Tucker decomposition of $\mathcal{E}$.

Cost: Instead of n^k of sums of two number, we need to do:

- k-times the economic QR decomposition of $n \times r$ matrix;
- k-times the product of $(r^{\times k})$-tensor with $(r \times r)$-matrix;
- one Tucker decompostion of $(r^{\times k})$-tensor;
- k-times the product of $(n \times r)$-matrix with $(r \times r)$-matrix.

(Here $n = \max\{n_1, n_2, \dots, n_k\}$ and $r = \max\{r_1, r_2, \dots, r_k\}$.)

Tensor matrix product

Let $\mathcal{T}, \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}$ in Tucker form

$$\mathcal{T} = (U'_{1,\mathcal{T}}, U'_{2,\mathcal{T}}, \ldots, U'_{k,\mathcal{T}} \,|\, \mathcal{C}_{\mathcal{T}}), \quad \text{and} M \in \mathbb{R}^{m \times n_\ell}$$

Then

$$\mathcal{E} = M \times_\ell \mathcal{T} = (U'_{1,\mathcal{T}}, \ldots, \underbrace{MU'_{\ell,\mathcal{T}}}_{X_\ell}, \ldots, U'_{k,\mathcal{T}} \,|\, \mathcal{C}_{\mathcal{T}}).$$

The compression then yields the Tucker decomposition of $\mathcal{E}$.

Cost: Instead of n^{k-1} of MV products, we need to do:

- r-times the MV product;
- one economic QR decomposition of $n \times r$ matrix;
- one Tucker decompostion of $(r^{\times k})$-tensor;
- one product of $(r^{\times k})$-tensor with $(r \times r)$-matrix;
- k-times the product of $(n \times r)$-matrix with $(r \times r)$-matrix.

Note on norm and scalar product

Recall that

$$\langle \mathcal{T}, \mathcal{F} \rangle = \mathrm{vec}(\mathcal{F})^{\mathsf{T}} \mathrm{vec}(\mathcal{T}), \qquad \|\mathcal{T}\| = (\langle \mathcal{T}, \mathcal{T} \rangle)^{\frac{1}{2}},$$

$$\mathcal{T} = (U'_{1,\mathcal{T}}, U'_{2,\mathcal{T}}, \ldots, U'_{k,\mathcal{T}} \,|\, \mathcal{C}_{\mathcal{T}}),$$

$$\mathrm{vec}(\mathcal{T}) = \left(U'_{k,\mathcal{T}} \otimes \cdots \otimes U'_{2,\mathcal{T}} \otimes U'_{1,\mathcal{T}}\right) \mathrm{vec}(\mathcal{C}_{\mathcal{T}}),$$

and similarly for $\mathcal{F}$. Then $\langle \mathcal{T}, \mathcal{F} \rangle$

$$= \mathrm{vec}(\mathcal{C}_{\mathcal{F}})^{\mathsf{T}} \left(U'_{k,\mathcal{F}} \otimes \ldots \otimes U'_{1,\mathcal{F}}\right)^{\mathsf{T}} \left(U'_{k,\mathcal{T}} \otimes \ldots \otimes U'_{1,\mathcal{T}}\right) \mathrm{vec}(\mathcal{C}_{\mathcal{T}})$$

$$= \mathrm{vec}(\mathcal{C}_{\mathcal{F}})^{\mathsf{T}} \Big(\big(U'_{1,\mathcal{F}}{}^{\mathsf{T}} U'_{k,\mathcal{T}}\big) \otimes \cdots \otimes \big(U'_{1,\mathcal{F}}{}^{\mathsf{T}} U'_{k,\mathcal{T}}\big) \Big) \mathrm{vec}(\mathcal{C}_{\mathcal{T}})$$

$$= \mathrm{vec}(\mathcal{C}_{\mathcal{F}})^{\mathsf{T}} \, \mathrm{vec}\Big(\big(U'_{1,\mathcal{F}}{}^{\mathsf{T}} U'_{k,\mathcal{T}}\big), \ldots, \big(U'_{1,\mathcal{F}}{}^{\mathsf{T}} U'_{k,\mathcal{T}}\big) \Big| \mathcal{C}_{\mathcal{T}} \Big)$$

but also

$$= \mathrm{vec}\Big(\big(U'_{1,\mathcal{T}}{}^{\mathsf{T}} U'_{k,\mathcal{F}}\big), \ldots, \big(U'_{1,\mathcal{T}}{}^{\mathsf{T}} U'_{k,\mathcal{F}}\big) \Big| \mathcal{C}_{\mathcal{F}} \Big)^{\mathsf{T}} \, \mathrm{vec}(\mathcal{C}_{\mathcal{T}})$$

one of the last two lines needs to be evaluated (note that one core may be smaller than the other).

Why to do such complicated arithmetics?

Consider the following problem

$$\mathscr{A}(\mathcal{X}) = \mathcal{B}, \quad \text{where} \quad \mathscr{A} \in \mathscr{L}(\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}, \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k})$$

and $\mathcal{B}$ are given and the goal is to find $\mathcal{X}$.

For example: The Lyapunov operator on $\mathbb{R}^{n \times n}$,

$$\mathscr{A}(X) = AX + XA^{\mathsf{T}}, \quad \mathrm{vec}(\mathscr{A}(X)) = (I \otimes A + A \otimes I)\mathrm{vec}(X).$$

For rank-one rhs $B = bb^{\mathsf{T}}$, $b \neq 0$, the solution X is of full rank with exponentially decaying singular values.

If A is SPD, then also $\mathscr{A}$ is SPD, and then, e.g., the method of conjugate gradients (CG) can be used for solving $\mathscr{A}(\mathcal{X}) = \mathcal{B}$. With an initial guess $\mathcal{X}_0 = (0, 0, \ldots, 0 \,|\, 0)$ and employing the low-rank arithmetics, we get solution in Tucker format.

Cost of CG iteration is changing, it depends on ranks! (Truncation, open pbs.)

A final note on Tucker decomposition

First note that the "Tucker-like" decompositions

$$\mathcal{T} = (U'_1, U'_2, \ldots, U'_k, \mathcal{C}_{\mathcal{T}}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}$$

are not sufficient (from the computational point of view) for handling really large tensors.

Let $\overrightarrow{\mathrm{rank}}(\mathcal{T}) = (r_1, r_2, \ldots, r_k)$, i.e., the Tucker core

$$\mathcal{C}_{\mathcal{T}} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_k} \quad \text{and let} \quad r_1 = r_2 = \cdots = r_k = 2.$$

Then the memory requirement to store $\mathcal{T}$ are roughly

$$\underbrace{k \cdot (n \cdot 2)}_{U'_\ell} + \underbrace{2^k}_{\mathcal{C}_{\mathcal{T}}} \approx 2^k,$$

i.e., for example for $k = 100$ we need to store

$$\approx 2^{100} \approx 1.2677 \cdot 10^{30} \text{ numbers } \approx 9.2234 \cdot 10^{18} \text{ TiB in doubles.}$$

Graph interpretation:

Tensor networks & Hierarchical formats

Tensors & graphs

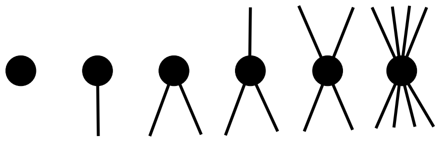
To simplify a bit our notion about tensors, tensor products and tensor decompositions, we employ the graph theory.

Any tensor $\mathcal{T}$ is interpreted as a *graph vertex*, and number of indices of $\mathcal{T}$ as the *degree of the vertex.*

Thus the scalar, vector, matrix, 3-, 4-, and, e.g., 8-way tensors

$$t, \qquad t_i, \qquad t_{i,j}, \qquad t_{i_1,i_2,i_3}, \qquad t_{i_1,\ldots,i_4}, \qquad t_{i_1,\ldots,i_8}$$

are interpreted as

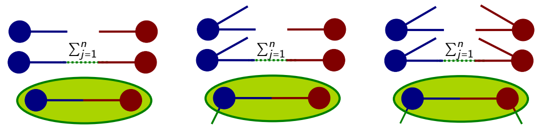


Basic products

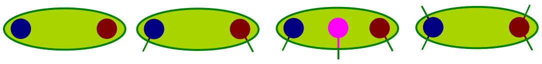
Scalar, MV, and MM-products can be then drawn as follows:

$y \in \mathbb{R}^n$, $x \in \mathbb{R}^n$ $\quad$ $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ $\quad$ $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times d}$

$y^{\mathsf{T}} x = \alpha \in \mathbb{R}$ $\quad$ $Ax = y \in \mathbb{R}^m$ $\quad$ $AB = C \in \mathbb{R}^{m\times d}$

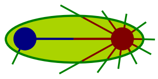


Prod. of scalars, outer prod's. of (two and three) vec's and mat's:



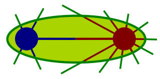
Products involving tensors

- Tensor-matrix product (pre- or post-multiplication)



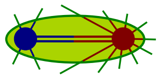
$\sim \qquad \mathcal{W} = M \times_{\ell} \mathcal{T},$

- Tensor-tensor product (contraction)



$\sim \qquad \mathcal{W} = \mathcal{F} \times_{(\text{ß},\ell)} \mathcal{T},$

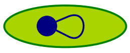
- Tensor-tensor product (contraction) in several pairs of indices at once



$\sim \qquad \mathcal{W} = \mathcal{F} \times_{((\text{ß},\sigma),(\ell,\tau))} \mathcal{T}.$

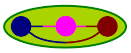
It allows us to be more creative :-)

- A product of matrix $A \in \mathbb{R}^{n\times n}$ with itself?



$\sim \qquad \sum_{i=1}^{n} a_{i,i} = \operatorname{trace}(A)$

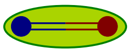
- A circular product of matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times d}$, $C \in \mathbb{R}^{d\times m}$?



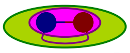
$\sim \qquad \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{\ell=1}^{d} a_{i,j} \cdot b_{j,\ell} \cdot c_{\ell,i}$

- But recall the scalar product of tensors! For matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times n}$ it takes form of both—the *circular product* and *product of a matrix with itself* :-)

$$\langle A, B\rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} b_{i,j} \cdot a_{i,j} = \operatorname{trace}(B^{\mathsf{T}} A)$$



=

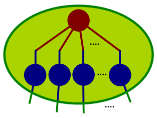


Tucker decomposition

Graph of the Tucker decompostion

$$\mathcal{T} = (U_1', U_2', U_3', \ldots, U_k' \,|\, \mathcal{C}_{\mathcal{T}})$$

takes form

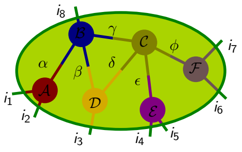


Our goal is to break up the high-order core tensor $\mathcal{C}_{\mathcal{T}}$ to product of several lower-orders tensors. Computationally, we want to replace the core as it is, whos number of entries scales exponentially ($\approx r^k$) with the tensor order k, by a set of tensors, whos number of entries scales linearly or logarithmically with k. How to do it can be easily understood by using graphs.

A general tensor network

By a general tensor network we understand interpretation of a high-order tensor $\mathcal{T}$ as a (prescribed) structured product of a set of lower-order tensors.

The tensor network can be seen as a (de)composition or approximation framework of the tensor $\mathcal{T}$.



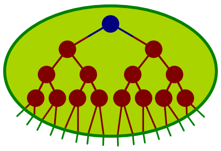
$$t_{i_1,i_2,i_3,i_4,i_5,i_6,i_7,i_8} = \sum_{\alpha,\beta,\gamma,\delta,\epsilon,\phi} a_{i_1,i_2,\alpha} \cdot b_{\alpha,\beta,\gamma,i_8} \cdot c_{\gamma,\delta,\epsilon,\phi} \cdot d_{\beta,i_3,\delta} \cdot e_{i_4,i_5,\epsilon} \cdot f_{\phi,i_6,i_7}$$

$$n^8 \longrightarrow 4n^3 + 2n^4$$

The simples structure for decomposing tensor is a (binary) tree (it avoids computationally complicated circles).

Tree decomposition of the Tucker core

Recall $\mathcal{T} = (U_1', U_2', \ldots, U_k' \,|\, \mathcal{C}_{\mathcal{T}})$. There are two different extremes: **The balanced** (as much as possible) **binary tree**

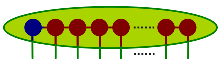


$r^k \longrightarrow (k-2)r^3 + r^2 \approx kr^3$

So-called hierarchical Tucker decompostion (HTD).

[*L. Grasedyck*, SIMAX 31(4), 2010]

The most-unbalanced binary tree



$r^k \longrightarrow (k-2)r^3 + 2r^2 \approx kr^3$

So-called tensor train decompostion (TTD).

[*I. V. Oseledets*, SISC 33(5), 2011]

The blue two-way tensors (matrices) are roots of these binary trees.

How to find the prescribed tree structure?

The root

The root is always a tensor of second order (a matrix). Let, for simplicity, the indices (modes) of the whole core $\mathcal{C} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_k}$ be ordered in such a way that

$$i_1, i_2, \ldots, i_t \qquad \text{and} \qquad i_{t+1}, i_{t+2}, \ldots, i_k$$

correspond to the left and right branches, respectively.

Thus, for HTD and even k, $t = k/2$; for TTD $t = 1$.

Consider the economic SVD of the matricizaton of $\mathcal{C}$

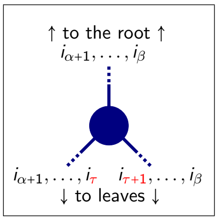
$$\mathcal{C}^{\mathscr{R}} = U'_{\mathscr{R}} \Sigma'_{\mathscr{R}} {V'_{\mathscr{R}}}^{\mathsf{T}}, \qquad \text{where} \qquad \mathscr{R} = \{1, 2, \ldots, t\},$$

- Then the matrix $\Sigma'_{\mathscr{R}}$ is the root of the tree and
- matrices $U'_{\mathscr{R}}$, $V'_{\mathscr{R}} = U'_{\mathscr{C}}$ can be decomposed into left and right branches of the tree, respectively; $\mathscr{C} = \{1, \ldots, k\} \setminus \mathscr{R}$.

How to find the prescribed tree structure?

A single vertex of degree three

Since indices of $\mathcal{C}$ are order properly, any vertex of deg.3 looks like:


↑ to the root ↑
$i_{\alpha+1}, \ldots, i_\beta$
$i_{\alpha+1}, \ldots, i_\tau$
$i_{\tau+1}, \ldots, i_\beta$
↓ to leaves ↓

Let us consider three corresponding matricizations and their economic SVDs:

$$\mathcal{C}^{\{\alpha+1,\ldots,\beta\}} = U'_{\{\alpha+1,\ldots,\beta\}} \Sigma'_{\{\alpha+1,\ldots,\beta\}} {V'_{\{\alpha+1,\ldots,\beta\}}}^{\mathsf{T}},$$
$$\mathcal{C}^{\{\alpha+1,\ldots,\tau\}} = U'_{\{\alpha+1,\ldots,\tau\}} \Sigma'_{\{\alpha+1,\ldots,\tau\}} {V'_{\{\alpha+1,\ldots,\tau\}}}^{\mathsf{T}},$$
$$\mathcal{C}^{\{\tau+1,\ldots,\beta\}} = U'_{\{\tau+1,\ldots,\beta\}} \Sigma'_{\{\tau+1,\ldots,\beta\}} {V'_{\{\tau+1,\ldots,\beta\}}}^{\mathsf{T}}.$$

The key theorem of all tree-form decomp's (HTD, TTD, ...) says:

$$\operatorname{range}\Big(U'_{\{\alpha+1,\ldots,\beta\}}\Big) \subseteq \operatorname{range}\Big(U'_{\{\tau+1,\ldots,\beta\}} \otimes U'_{\{\alpha+1,\ldots,\tau\}}\Big).$$

How to find the prescribed tree structure?

Tensor-tree-decomposition theorem

Theorem:

$$\operatorname{range}\Big(U'_{\{\alpha+1,\ldots,\beta\}}\Big) \subseteq \operatorname{range}\Big(U'_{\{\tau+1,\ldots,\beta\}} \otimes U'_{\{\alpha+1,\ldots,\tau\}}\Big), \quad \alpha < \tau < \beta.$$

Sketch of the proof: Any column of $\mathcal{C}^{\{\cdots\}}$ is a vector $v \in \mathbb{R}^{(\beta-\alpha)}$, that can be reshaped into a matrix $M \in \mathbb{R}^{(\tau-\alpha)\times(\alpha-\beta)}$, $v = \operatorname{vec}(M)$.

Note that columns of M are in $\operatorname{range}(U'_{\{\ldots\}}) = \operatorname{range}(\mathcal{C}^{\{\cdots\}})$ and rows of M in $\operatorname{range}(U'_{\{\ldots\}}) = \operatorname{range}(\mathcal{C}^{\{\cdots\}})$. Thus

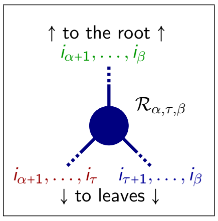
$$\underbrace{M = \mathcal{C}^{\{\ldots\}} {\mathcal{C}^{\{\ldots\}}}^{\dagger} M \qquad \text{and} \qquad M^{\mathsf{T}} = \mathcal{C}^{\{\ldots\}} {\mathcal{C}^{\{\ldots\}}}^{\dagger} M^{\mathsf{T}}}$$
$$M = \mathcal{C}^{\{\ldots\}} \underbrace{{\mathcal{C}^{\{\ldots\}}}^{\dagger} M {{\mathcal{C}^{\{\ldots\}}}^{\dagger}}^{\mathsf{T}}} {\mathcal{C}^{\{\ldots\}}}^{\mathsf{T}}$$

giving $\operatorname{vec}(M) = v = \big(\mathcal{C}^{\{\ldots\}} \otimes \mathcal{C}^{\{\ldots\}}\big)\big({\mathcal{C}^{\{\ldots\}}}^{\dagger} \otimes {\mathcal{C}^{\{\ldots\}}}^{\dagger}\big) v.$ □

How to find the prescribed tree structure?

How to employ the tensor-tree-decomposition theorem?

Denote the three-way tensor $\mathcal{R}_{\alpha,\tau,\beta}$. Since


↑ to the root ↑
$i_{\alpha+1}, \ldots, i_\beta$
$\mathcal{R}_{\alpha,\tau,\beta}$
$i_{\alpha+1}, \ldots, i_\tau$
$i_{\tau+1}, \ldots, i_\beta$
↓ to leaves ↓

$$\operatorname{range}(U'_{\{\ldots\}}) \subseteq \operatorname{range}(U'_{\{\ldots\}} \otimes U'_{\{\ldots\}})$$

There exists a matrix R such that

$$U'_{\{\ldots\}} = (U'_{\{\ldots\}} \otimes U'_{\{\ldots\}})R, \quad R^{\mathsf{T}}R = I$$
$$R \in \mathbb{R}^{(\operatorname{rank}_{\{\ldots\}}(\mathcal{C}) \cdot \operatorname{rank}_{\{\ldots\}}(\mathcal{C})) \times (\operatorname{rank}_{\{\ldots\}}(\mathcal{C}))}$$

It remains to interpret $R = \mathcal{R}^{\{1,2\}}_{\alpha,\tau,\beta}$ so

$$\boxed{\mathcal{R}_{\alpha,\tau,\beta} \in \mathbb{R}^{(\operatorname{rank}_{\{\ldots\}}(\mathcal{C}))\times(\operatorname{rank}_{\{\ldots\}}(\mathcal{C}))\times(\operatorname{rank}_{\{\ldots\}}(\mathcal{C}))}}$$

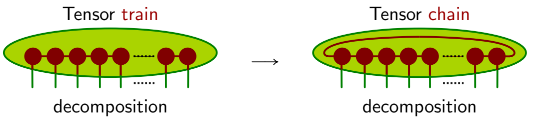
Doing this with all deg.3 vertices yields the HTD with any binary tree (recall the matrices on leaves). The last tensor of order two in TTD is just an identity matrix.

It can be applied on any (not necessarily binary) tree-form decomp.

A few notes on hierarchical / tree-form decompositions

- There is a lot of different ranks of $\mathcal{T}$ in the game (dimensions of cubes).
- To be efficent, these ranks needs to be small.
- To be effective, $\mathcal{T}$ has to be *either* of low rank, *or* well approximable by a such low rank tensor.
- Otherwise we are not able to manage $\mathcal{T}$ in this way.
- Design of the tree should reflect knowledge about the problem.
- Employ symmetries between modes (if there are; $t_{i,j,\ell} = t_{j,i,\ell}$).

Note that there are also cyclic decompositions:

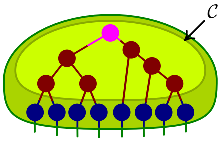


A few notes on hierarchical / tree-form decompositions

Recall that we first did the Tucker decomposition of a tensor and now the tree-form decomposition of the Tucker core.

Both together gives the HTD with structure like:

$$\mathcal{T} = (U'_1, U'_2, \ldots, U'_k \,|\, \mathcal{C}) =$$



Note that in this particular case $\mathscr{R} = \{1, 2, 3, 4\}$,

$$\mathcal{T}^{\mathscr{R}} = \overbrace{\big(U'_4 \otimes U'_3 \otimes U'_2 \otimes U'_1\big)\big(\mathcal{R}^{\{1,2\}}_{2,3,4} \otimes \mathcal{R}^{\{1,2\}}_{0,1,2}\big)\big(\mathcal{R}^{\{1,2\}}_{0,2,4}\big)}^{U'_{\mathscr{R}}} \Sigma'_{\mathscr{R}}$$
$$\Big(\underbrace{\big(U'_8 \otimes U'_7 \otimes U'_6 \otimes U'_5\big)\big(\mathcal{R}^{\{1,2\}}_{6,7,8} \otimes I\big)\big(\mathcal{R}^{\{1,2\}}_{5,6,8} \otimes I\big)\big(\mathcal{R}^{\{1,2\}}_{4,5,8}\big)}_{V'_{\mathscr{R}}}\Big)^{\mathsf{T}}$$

Arithmetics of hierarchical Tucker

Motivation

Recall that we want to solve, e.g.,

$$\mathscr{A}(\mathcal{X}) = \mathcal{B}, \qquad \text{where} \qquad \mathcal{X},\ \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k},$$

where $\mathscr{A}$ is symmetric positive definite (SPD) typically represented by one or more sparse matrices in outer (Kronecker) product, and the low-rank right-hand side $\mathcal{B}$ is given in HTD.

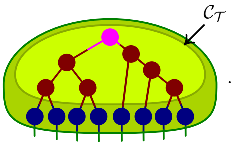
By taking $\mathcal{X}_0 = 0$ and storing it in the same tree structure as $\mathcal{B}$ (e.g., by replacing all numbers by zeros), we can start to search for $\mathcal{X}$ for example by the method of conjugate gradients (CG).

We need to know how to (i) do linear combinations, (ii) TM-product, and (iii) calculate scalar products and norms in HTD.

A sum (a linear combination) of two HTDs

Let $\mathcal{T}$ and $\mathcal{F}$ be of the same order k, of the same dimensions, and with HTDs of the same structure:

$$\mathcal{T} = (U'_{1,\mathcal{T}}, U'_{2,\mathcal{T}}, \ldots, U'_{k,\mathcal{T}} \,|\, \mathcal{C}_{\mathcal{T}}) =$$



.

In the top, there is one root matrix $\Sigma'_{\mathcal{T}}$, in the middle, there is bunch of inner cubes (3-way tensors) $\mathcal{R}_{\alpha,\tau,\beta,\mathcal{T}}$, and in the bottom k leaves matrices $U'_{j,\mathcal{T}}$.

Recall that

$$(\mathcal{R}_{\alpha,\tau,\beta,\mathcal{T}}{}^{\{1,2\}})^{\mathsf{T}} \mathcal{R}^{\{1,2\}}_{\alpha,\tau,\beta,\mathcal{T}} = I = I_{\text{rank}_{\{\alpha+1,\ldots,\beta\}}(\mathcal{T})} \qquad \text{for all } \alpha < \tau < \beta$$

$$U'_{j,\mathcal{T}}{}^{\mathsf{T}} U'_{j,\mathcal{T}} = I = I_{\text{rank}_{\{j\}}(\mathcal{T})} \qquad \text{for } j = 1, 2, \ldots, k.$$

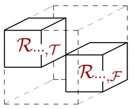
A sum (a linear combination) of two HTDs

A linear combination

$$\mathcal{E} = \varphi\mathcal{T} + \psi\mathcal{F}$$

will be done in several steps: **Step 1:** Concatenation of leaves, block diagonal composition (direct sum) of inner cubes and roots:

$$\left[U'_{j,\mathcal{T}},\ U'_{j,\mathcal{F}} \right],$$


$\mathcal{R}_{\ldots,\mathcal{T}}$
$\mathcal{R}_{\ldots,\mathcal{F}}$

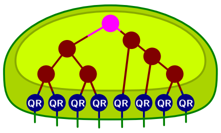
$$,\quad \begin{bmatrix} \varphi\Sigma'_{\mathcal{T}} & 0 \\ 0 & \psi\Sigma'_{\mathcal{F}} \end{bmatrix},$$

gives the sum $\mathcal{E}$ formally in the same HTD structure. However, dimensions of all objects are twice as large and $U'_{\ldots}$'s and $\mathcal{R}^{\{1,2\}}_{\ldots}$'s do not have orthonormal columns.

Step 2: (Re)compression of the sum enforing wanted properties requires plenty of QR's, TM-prod's and one SVD.

A sum (a linear combination) of two HTDs

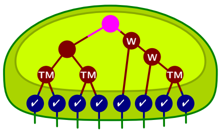
Recompression



e-QR decomp's of leaves matrices; triangular factors go up to cubes

A sum (a linear combination) of two HTDs

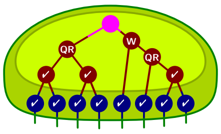
Recompression



Multiplication of cubes by triangular factors (two are waiting)

A sum (a linear combination) of two HTDs

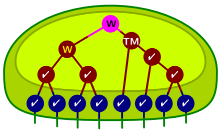
Recompression



$\{1,2\}$-ma'tions & e-QR decomp's of cubes; triangular factors go up

A sum (a linear combination) of two HTDs

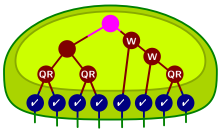
Recompression



Multiplication the last cube by triangular factors (root is waiting)

A sum (a linear combination) of two HTDs

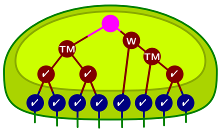
Recompression



$\{1,2\}$-ma'tions & e-QR decomp's of cubes; triangular factors go up

A sum (a linear combination) of two HTDs

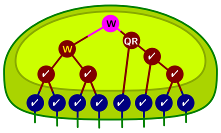
Recompression



Multiplication of cubes by triangular factors (one is waiting)

A sum (a linear combination) of two HTDs

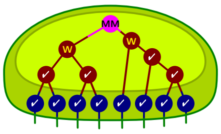
Recompression



$\{1,2\}$-ma'tions & e-QR decomp's of the last cube

A sum (a linear combination) of two HTDs

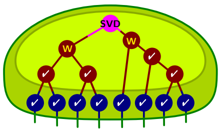
Recompression



Multiplication the root by triangular factors

A sum (a linear combination) of two HTDs

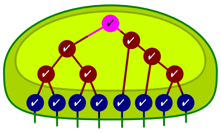
Recompression



e-SVD of the root; we've the root $\Sigma'_{\mathcal{E}}$; U' and V' are going down

A sum (a linear combination) of two HTDs

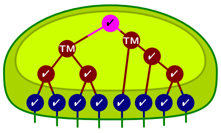
Recompression



Done! ✓ ✓ ✓

A sum (a linear combination) of two HTDs

Recompression



The last two multiplications of cubes.

Tensor-matrix multiplication

Similarly we can do the ℓ-mode tensor-matrix multiplication,

$$\mathcal{E} = M \times_\ell \mathcal{T}.$$

It will be done again in sevaral steps: **Step 1:** Multplication of M with the particular (the ℓth) leaf:

$$\left[MU'_{\ell,\mathcal{T}} \right]$$

that gives the product $\mathcal{E}$ formally in the HTD structure. Similarly as before we can do the:

Step 2: (Re)compression of the product $\mathcal{E}$. Since we multiplied only in one mode, everything is a bit simpler.

Tensor-matrix multiplication

Similarly we can do the ℓ-mode tensor-matrix multiplication,

$$\mathcal{E} = M \times_\ell \mathcal{T}.$$

It will be done again in sevaral steps: **Step 1:** Multplication of M with the particular (the ℓth) leaf:

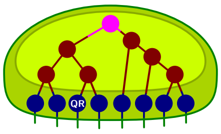
$$\left[MU'_{\ell,\mathcal{T}} \right]$$

that gives the product $\mathcal{E}$ formally in the HTD structure. Similarly as before we can do the:

Step 2: (Re)compression of the product $\mathcal{E}$. Since we multiplied only in one mode, everything is a bit simpler.

Tensor-matrix multiplication (3-mode)

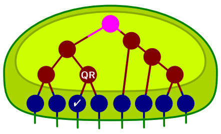
Recompression



e-QR decomp. of the third leaf; triangular factor goes up to cubes

Tensor-matrix multiplication (3-mode)

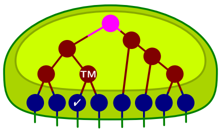
Recompression



$\{1,2\}$-ma'tions & e-QR decomp's of cubes; triangular factors go up

Tensor-matrix multiplication (3-mode)

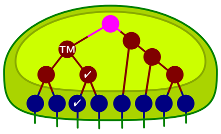
Recompression



Multiplication of cubes by triangular factors (two are waiting)

Tensor-matrix multiplication (3-mode)

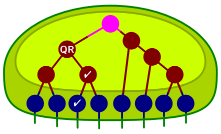
Recompression



Multiplication of cubes by triangular factors (one is waiting)

Tensor-matrix multiplication (3-mode)

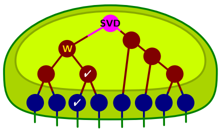
Recompression



$\{1,2\}$-ma'tions & e-QR decomp's of cubes; triangular factors go up

Tensor-matrix multiplication (3-mode)

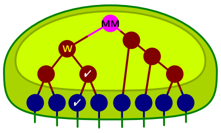
Recompression



e-SVD of the root; we've the root $\Sigma'_{\mathcal{E}}$; U' and V' are going down

Tensor-matrix multiplication (3-mode)

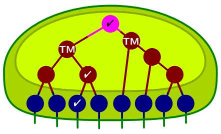
Recompression



Multiplication the root by triangular factors

Tensor-matrix multiplication (3-mode)

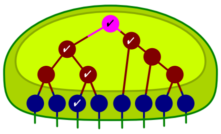
Recompression



The last two multiplications of cubes.

Tensor-matrix multiplication (3-mode)

Recompression



Done! ✓ ✓ ✓

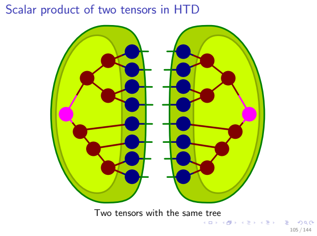
Scalar product of two tensors in HTD

Finally, we present evaluation of the scallar product

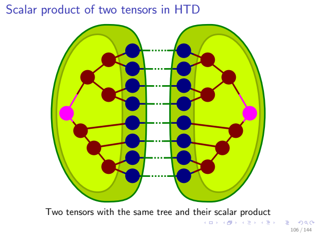
$$\langle \mathcal{T}, \mathcal{F} \rangle$$

of two vectors in HTD with the same trees; and also of the norm

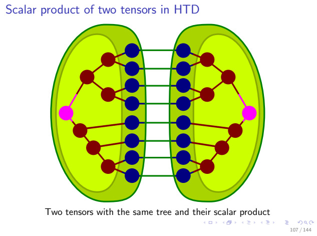
$$\|\mathcal{T}\| = (\langle \mathcal{T}, \mathcal{T} \rangle)^{\frac{1}{2}}.$$


Scalar product of two tensors in HTD

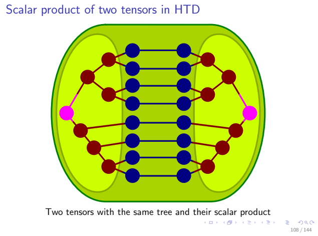
Two tensors with the same tree


Scalar product of two tensors in HTD

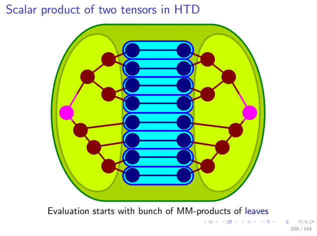
Two tensors with the same tree and their scalar product


Scalar product of two tensors in HTD

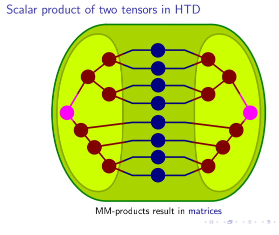
Two tensors with the same tree and their scalar product


Scalar product of two tensors in HTD

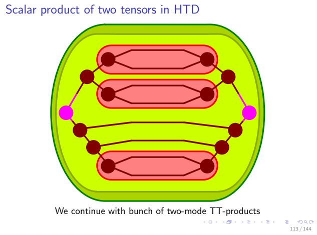
Two tensors with the same tree and their scalar product


Scalar product of two tensors in HTD

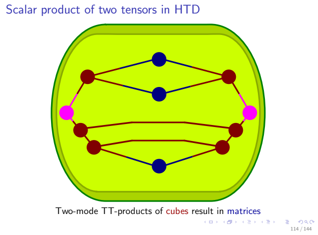
Evaluation starts with bunch of MM-products of leaves


Scalar product of two tensors in HTD

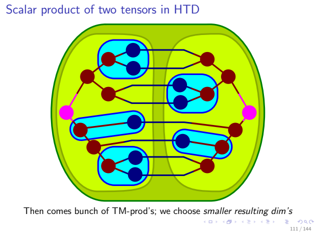
MM-products result in matrices


Scalar product of two tensors in HTD

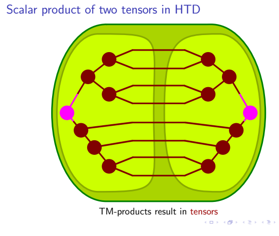
We continue with bunch of two-mode TT-products


Scalar product of two tensors in HTD

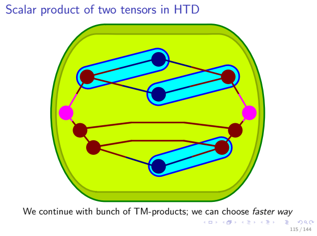
Two-mode TT-products of cubes result in matrices


Scalar product of two tensors in HTD

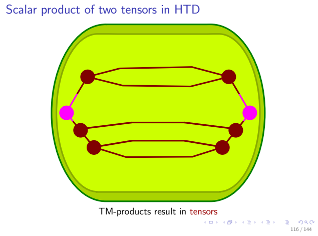
Then comes bunch of TM-prod's; we choose *smaller resulting dim's*


Scalar product of two tensors in HTD

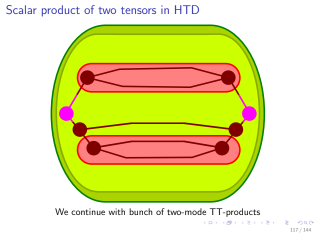
TM-products result in tensors


Scalar product of two tensors in HTD

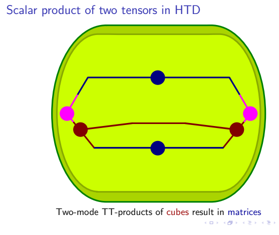
We continue with bunch of TM-products; we can choose *faster way*


Scalar product of two tensors in HTD

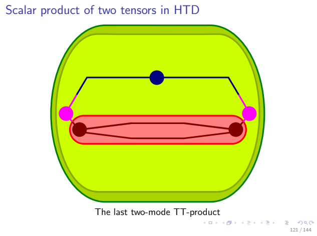
TM-products result in tensors


Scalar product of two tensors in HTD

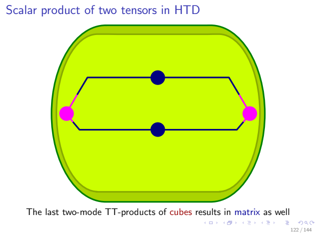
We continue with bunch of two-mode TT-products


Scalar product of two tensors in HTD

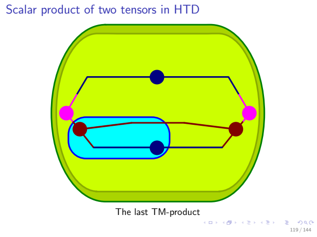
Two-mode TT-products of cubes result in matrices


Scalar product of two tensors in HTD

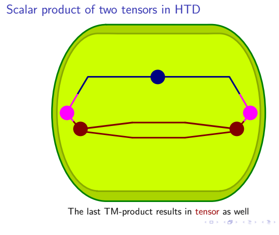
The last two-mode TT-product


Scalar product of two tensors in HTD

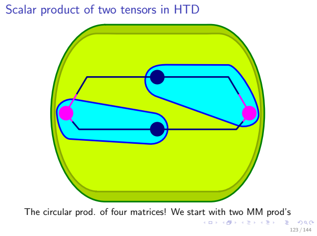
The last two-mode TT-products of cubes results in matrix as well


Scalar product of two tensors in HTD

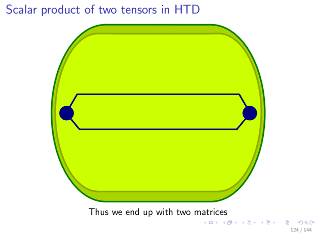
The last TM-product


Scalar product of two tensors in HTD

The last TM-product results in tensor as well

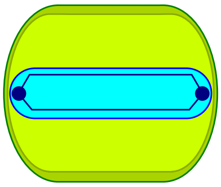

Scalar product of two tensors in HTD

The circular prod. of four matrices! We start with two MM prod's


Scalar product of two tensors in HTD

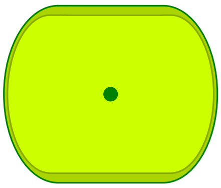
Thus we end up with two matrices

Scalar product of two tensors in HTD



We calculate their scalar product

Scalar product of two tensors in HTD



Done! ✓ ✓ ✓

Final notes on arithmetics of HTDs

For a linear combination and scallar product of two tensors

$$\varphi\mathcal{T} + \psi\mathcal{F}, \quad \langle\mathcal{T}, \mathcal{F}\rangle,$$

$\mathcal{T}$, $\mathcal{F}$ need to be of the same dimensions (and thus also the order).

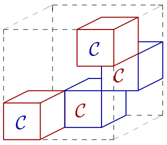
It seems that requirement on the same tree-structure brings a new restriction, but it is possible do that also with tensors with different tree-structures.

However, while doing that with tensors with different *binary* trees, there always appear tensors of higher orders than presented. *Typically* (i.e., if the *root is not* in the game), there appear at least one inner 'cube' of order *four* (no hihger orders are needed$^{(?!)}$).

While summation, it can employ some maximal (or the greates$^{(?!)}$) common sub-tree of both and recalculate the structure of one.

[*Kressner, Tobler*, `htucker`—Matlab toolbox, 2012]
`http://anchp.epfl.ch/htucker`

That's All Volks!



Thank You for Your Attention

Guaranteed eigenvalue bounds for elliptic partial differential operators

Tomáš Vejchodský (vejchod@math.cas.cz)

Institute of Mathematics
Czech Academy of Sciences

Supported by the Neuron Impuls project no. 24/2016

SNA 2019, Ostrava, January 21–25, 2019

Reliable numerical methods

To compute (approximate) solution is not sufficient.
We should provide an information about the error.

Can we provide
a guaranteed upper bound?
$\|u - u_h\| \leq \eta$

Sinking of the Sleipner A offshore platform in 1991, Norway. The failure resulted from inaccurate NASTRAN calculations.

Babuška, Verfürth, Ainsworth, Rannacher, Repin, ...

Eigenvalue problems

Laplace eigenvalue problem

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n \quad \text{in } \Omega \\ u_n &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

Finite element method

- Very flexible (various domains, high order, various problems, ...)
- Converges with optimal speed
- Adaptive mesh refinement
- Nice theory

Guaranteed upper bound

$$? \leq \lambda_n \leq \lambda_{h,n}$$

Can we dream about anything else? Lower bounds!
Guaranteed error bounds on eigenfunctions: $\|u_n - u_{h,n}\| \leq \eta$

Outline

1. Motivation
2. Theory
   - 2.1 Existence
   - 2.2 Min-max principle
   - 2.3 Optimal constants
3. Rayleigh–Ritz (Galerkin) method
4. Lower bounds on eigenvalues
   - 4.1 Weinstein's bound
   - 4.2 Lehmann–Goerisch method
   - 4.3 Interpolation constant based methods
5. Guaranteed bounds on eigenfunctions
6. Literature

2. Theory

2.1 Existence

Abstract formulation

Eigenvalue problem: Find eigenvalue λ_n and eigenfunction $u_n \in V \setminus \{0\}$ such that

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V.$$

- V is a Hilbert space.
- $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are two bilinear forms on V.

Example

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n && \text{in } \Omega \\ u_n &= 0 && \text{on } \partial\Omega \end{aligned}$$

Weak formulation

$$(\nabla u_n, \nabla v) = \lambda_n (u_n, v) \quad \forall v \in V$$

- $V = H_0^1(\Omega)$
- $a(u,v) = (\nabla u, \nabla v)$
- $b(u,v) = (u,v)$ $\qquad (u,v) = \int_\Omega uv \,\mathrm{d}x$

Hilbert–Schmidt theorem

$$Su_n = \mu_n u_n$$

Let

- V be a Hilbert space
- $S : V \to V$ be linear, bounded, compact, self-adjoint operator

Then

- there is (at most) countable sequence of nonzero real eigenvalues of S (repeated according to their multiplicity): $|\mu_1| \geq |\mu_2| \geq |\mu_3| \geq \cdots > 0$, and if the sequence is infinite then $\lim_{n\to\infty} \mu_n = 0$
- eigenfunctions u_n corresponding to these μ_n form a complete orthonormal system in $\overline{\operatorname{range} S}$
- $V = (\ker S) \oplus (\overline{\operatorname{range} S})$

Assumptions

Find $\lambda_n \in \mathbb{R}$, $u_n \in V \setminus \{0\}$: $\quad a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$

- V is a real Hilbert space
- $a(\cdot,\cdot)$ is continuous, bilinear, symmetric, V-elliptic
- $b(\cdot,\cdot)$ is continuous, bilinear, symmetric, positive semidefinite
- $\|v\|_a = a(v,v)^{1/2}$ is the norm induced by $a(\cdot,\cdot)$
- $|v|_b = b(v,v)^{1/2}$ is the seminorm induced by $b(\cdot,\cdot)$
- $|\cdot|_b$ is compact with respect to $\|\cdot\|_a$,
  i.e. from any sequence bounded in $\|\cdot\|_a$, we can extract a subsequence which is Cauchy in $|\cdot|_b$

Existence

Theorem. There exists (at most) countable sequence of eigenvalues

$$0 < \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \to \infty$$

and the corresponding eigenfunctions can be normalized to satisfy

$$b(u_i, u_j) = \delta_{ij} \quad \forall i, j = 1, 2, \dots .$$

Proof

- Solution operator $S : V \to V$: $a(Su, v) = b(u, v) \quad \forall v \in V$
- $a(u_n, v) = \lambda_n \underbrace{b(u_n, v)}_{a(Su_n, v)} \quad \forall v \in V \quad \Leftrightarrow \quad Su_n = \frac{1}{\lambda_n} u_n$
- Exercise: compactness of $|\cdot|_b$ with respect to $\|\cdot\|_a$ is equivalent to compactness of S
- Hilbert–Schmidt theorem: $\mu_1 \ge \mu_2 \ge \mu_3 \ge \cdots > 0$, $\lambda_n = 1/\mu_n$ because $0 < \|u_n\|_a^2 = \lambda_n |u_n|_b^2$.

□ Note

$$\frac{1}{\lambda_i} a(u_i, u_j) = \delta_{ij} \quad \forall i, j = 1, 2, \dots$$

Orthonormal basis of eigenfunctions

Theorem. The space V can be decomposed as

$$V = \mathcal{K} \oplus \mathcal{M},$$

where $\mathcal{K} = \{v \in V : |v|_b = 0\}$ and $\mathcal{M} = \operatorname{span}\{u_1, u_2, \dots\}$.
Moreover,

$$\begin{aligned} a(u, v) &= 0 \quad \forall u \in \mathcal{K},\ \forall v \in \mathcal{M}, \\ b(u, v) &= 0 \quad \forall u \in \mathcal{K},\ \forall v \in V. \qquad (*) \end{aligned}$$

Proof

- $(*)$ follows from $|b(u, v)| \le |u|_b |v|_b = 0$
- Hilbert–Schmidt theorem: $V = (\ker S) \oplus \mathcal{M}$
  Now, $\ker S = \mathcal{K}$, because
  (a) $u \in \mathcal{K} \Rightarrow 0 = b(u, v) = a(Su, v)\ \forall v \in V$
  $\Rightarrow Su = 0 \Rightarrow u \in \ker S$
  (b) $u \in \ker S \Rightarrow 0 = a(Su, u) = b(u, u) = |u|_b^2 \Rightarrow u \in \mathcal{K}$
- Express $v \in \mathcal{M}$ as $v = \sum_{n=1}^{\infty} c_n u_n$ and

$$a(u, v) = \sum_{n=1}^{\infty} c_n a(u, u_n) = \sum_{n=1}^{\infty} c_n \lambda_n b(u, u_n) \overset{(*)}{=} 0.$$

□

Parseval's identities

Theorem. For all $v \in V$, there are unique $v^{\mathcal{K}} \in \mathcal{K}$ and $v^{\mathcal{M}} \in \mathcal{M}$ such that

$$v = v^{\mathcal{K}} + v^{\mathcal{M}}, \quad v^{\mathcal{M}} = \sum_{n=1}^{\infty} c_n u_n, \quad c_n = b(v^{\mathcal{M}}, u_n) = b(v, u_n)$$

$$|v|_b^2 = \sum_{n=1}^{\infty} |b(v, u_n)|^2,$$

$$\|v\|_a^2 = \|v^{\mathcal{K}}\|_a^2 + \|v^{\mathcal{M}}\|_a^2 \quad \text{with } \|v^{\mathcal{M}}\|_a^2 = \sum_{n=1}^{\infty} \lambda_n |b(v, u_n)|^2.$$

Proof

- $v = v^{\mathcal{K}} + v^{\mathcal{M}} = v^{\mathcal{K}} + \sum_{n=1}^{\infty} c_n u_n$
- $|v|_b^2 = b(v, v^{\mathcal{K}} + \sum_{n=1}^{\infty} c_n u_n) = \sum_{n=1}^{\infty} c_n b(v, u_n)$
- $\|v\|_a^2 = \|v^{\mathcal{M}}\|_a^2 + \|v^{\mathcal{K}}\|_a^2$ and $\|v^{\mathcal{M}}\|_a^2 = \sum_{n=1}^{\infty} \lambda_n c_n^2$

□

Example 1: Dirichlet Laplacian

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n && \text{in } \Omega \\ u_n &= 0 && \text{on } \partial\Omega \end{aligned}$$

Weak formulation: Find $\lambda_n \in \mathbb{R}$, $u_n \in H_0^1(\Omega) \setminus \{0\}$:

$$(\nabla u_n, \nabla v) = \lambda_n (I u_n, I v) \quad \forall v \in H_0^1(\Omega),$$

where $I : H_0^1(\Omega) \to L^2(\Omega)$ is the identity operator.

- $V = H_0^1(\Omega)$
- $a(u, v) = (\nabla u, \nabla v)$ ... cont., bilin., sym., V-elliptic
- $b(u, v) = (Iu, Iv)$ ... cont., bilin., sym., pos. def.
- Compactness: I is a compact operator by Rellich theorem.
  Definition: I is compact if from a sequence $\{v_i\} \subset H_0^1(\Omega)$ bounded in $\|\nabla v\|_{L^2(\Omega)} \le C$ we can extract a subsequence such that $\{I v_{i_j}\}$ is Cauchy in $L^2(\Omega)$.

Example 1: Dirichlet Laplacian

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n &&\text{in } \Omega \\ u_n &= 0 &&\text{on } \partial\Omega \end{aligned}$$

Exact solution for an interval $\Omega = (0, L)$

$$\lambda_n = \frac{n^2\pi^2}{L^2}, \quad u_n(x) = \sin\frac{n\pi x}{L}, \quad n = 1, 2, 3, \ldots$$

Easy to verify

$$u_n'(x) = \frac{n\pi}{L}\cos\frac{n\pi x}{L}$$

$$u_n''(x) = -\frac{n^2\pi^2}{L^2}\sin\frac{n\pi x}{L} = -\frac{n^2\pi^2}{L^2}u_n(x)$$

Is it complete?

Exact solution for a square $\Omega = (0, \pi)^2$

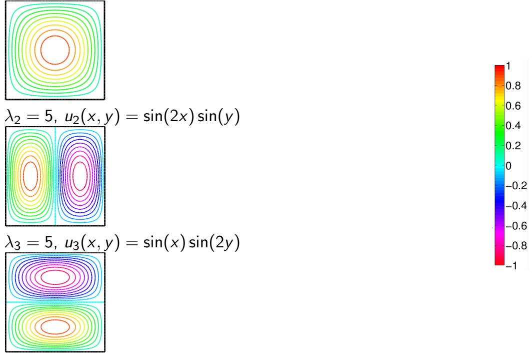
$$\lambda_{k,\ell} = k^2 + \ell^2, \quad u_{k,\ell}(x, y) = \sin(kx)\sin(\ell y), \quad k, \ell = 1, 2, \ldots$$

$$\begin{aligned} \lambda_1 &= 2\ (k = 1,\ \ell = 1) \\ \lambda_2 &= 5\ (k = 2,\ \ell = 1) \\ \lambda_3 &= 5\ (k = 1,\ \ell = 2) \\ \lambda_4 &= 8\ (k = 2,\ \ell = 2) \\ \lambda_5 &= 10\ (k = 3,\ \ell = 1) \end{aligned}$$

$$\begin{aligned} \lambda_6 &= 10\ (k = 1,\ \ell = 3) \\ \lambda_7 &= 13\ (k = 3,\ \ell = 2) \\ \lambda_8 &= 13\ (k = 2,\ \ell = 3) \\ \lambda_9 &= 17\ (k = 4,\ \ell = 1) \\ \lambda_{10} &= 17\ (k = 1,\ \ell = 4) \end{aligned}$$

Example: Square

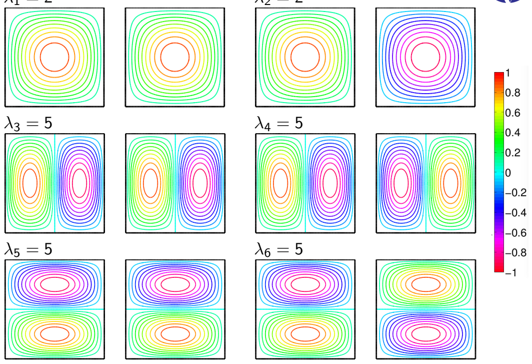
$\lambda_1 = 2,\ u_1(x, y) = \sin(x)\sin(y)$



$\lambda_2 = 5,\ u_2(x, y) = \sin(2x)\sin(y)$

$\lambda_3 = 5,\ u_3(x, y) = \sin(x)\sin(2y)$

Example: Two squares



$\lambda_1 = 2$

$\lambda_2 = 2$

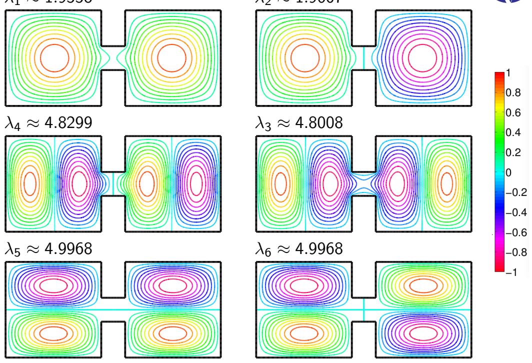
$\lambda_3 = 5$

$\lambda_4 = 5$

$\lambda_5 = 5$

$\lambda_6 = 5$

Example: Dumbbell



$\lambda_1 \approx 1.9558$

$\lambda_2 \approx 1.9607$

$\lambda_4 \approx 4.8299$

$\lambda_3 \approx 4.8008$

$\lambda_5 \approx 4.9968$

$\lambda_6 \approx 4.9968$

2. Theory

2.2 Min-max principle

Minimum principle

Rayleigh quotien: $R(v) = \dfrac{a(v,v)}{b(v,v)} = \dfrac{\|v\|_a^2}{|v|_b^2}$

Theorem. Numbers $0 < \lambda_1 \le \lambda_2 \le \cdots$ and functions $u_1, u_2, \dots \in V \setminus \{0\}$ are eigenpairs of

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

if and only if

$$\lambda_1 = \min_{v \in V,\ |v|_b \neq 0} R(v) \qquad u_1 = \operatorname*{arg\,min}_{v \in V,\ |v|_b \neq 0} R(v),$$
$$\lambda_n = \min_{v \in \mathcal{M}_{n-1}^{\perp}} R(v) \qquad u_n = \operatorname*{arg\,min}_{v \in \mathcal{M}_{n-1}^{\perp}} R(v),$$

where $\mathcal{M}_{n-1} = \operatorname{span}\{u_1, u_2, \dots, u_{n-1}\}$,

$$\begin{aligned}\mathcal{M}_{n-1}^{\perp} &= \{v \in \mathcal{M} : b(v, u_i) = 0,\ \forall i = 1, 2, \dots, n-1\}\\ &= \{v \in V : b(v, u_i) = 0,\ \forall i = 1, 2, \dots, n-1\\ &\qquad\qquad \text{and } |v|_b \neq 0\}.\end{aligned}$$

Minimum principle

Proof. (Including $n = 1$).

$\Rightarrow$ Let $a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$.

Then $u_n \in \mathcal{M}_{n-1}^{\perp}$, $\lambda_n = R(u_n)$, and thus $\inf_{\mathcal{M}_{n-1}^{\perp}} R(v) \le \lambda_n$.

If $v \in \mathcal{M}_{n-1}^{\perp}$ then $v^{\mathcal{K}} = 0$, $c_i = b(v, u_i) = 0$ for $i = 1, \dots, n-1$, and

$$R(v) = \frac{\|v\|_a^2}{|v|_b^2} = \frac{\sum_{i=n}^{\infty} \lambda_i c_i^2}{\sum_{i=n}^{\infty} c_i^2} \ge \lambda_n \frac{\sum_{i=n}^{\infty} c_i^2}{\sum_{i=n}^{\infty} c_i^2} = \lambda_n$$

$\Leftarrow$ The minimum is attained: $\exists u_n \in \mathcal{M}_{n-1}^{\perp} : \quad \lambda_n = R(u_n)$.

Let $t \in \mathbb{R}$, $v \in \mathcal{M}_{n-1}^{\perp}$ and $\varphi(t) = R(u_n + tv)$.

Derivative $\varphi'(0)$ exists and

$$\varphi'(0) = \frac{2}{|u_n|_b} \left(a(u_n, v) - \frac{\|u_n\|_a^2}{|u_n|_b^2} b(u_n, v) \right)$$

Since $\varphi(t)$ has a minimum at $t = 0$, we have $\varphi'(0) = 0$.

If $v = u_i$, $i = 1, 2, \dots, n-1$, then

$$b(u_n, u_i) = 0 \text{ and } a(u_n, u_i) = \lambda_i b(u_n, u_i) = 0.$$

□

(Courant–Fischer–Weyl) Min-max principle

Theorem.

$$\lambda_n = \min_{v \in \mathcal{M}_{n-1}^{\perp}} R(v) = \min_{E \in \mathcal{V}^{(n)}} \max_{v \in E} R(v)$$

where $\mathcal{V}^{(n)}$ is the set of all n-dimensional subspaces of $\mathcal{M}$.

Moreover, the mininum is attaind for $E = \operatorname{span}\{u_1, \dots, u_n\}$.

Proof. (Induction over n.)

$n = 1$: Since $R(\alpha v) = R(v)$ for all $\alpha \neq 0$, we have

$$\min_{E \in \mathcal{V}^{(1)}} \max_{v \in E} R(v) = \min_{v \in \mathcal{M}} R(v) = \min_{v \in V,\ |v|_b \neq 0} R(v)$$

$n > 1$: Let $\widetilde{\mathcal{V}}^{(n)} \subset \mathcal{V}^{(n)}$ be a set of all spaces $\widetilde{E}^z = \operatorname{span}\{u_1, \dots, u_{n-1}, z\}$, where $b(z, u_i) = 0$ for $i = 1, \dots, n-1$.

$$\min_{E \in \mathcal{V}^{(n)}} \max_{v \in E} R(v) \le \min_{\widetilde{E}^z \in \widetilde{\mathcal{V}}^{(n)}} \max_{v \in \widetilde{E}^z} R(v) = \min_{z \in \mathcal{M}_{n-1}^{\perp}} \max_{v \in \widetilde{E}^z} R(v) \overset{(!)}{=} \min_{z \in \mathcal{M}_{n-1}^{\perp}} R(z)$$

To prove (!), let $v \in \widetilde{E}^z$, $|v|_b = |z|_b = 1$. Thus, $v = \alpha z + \sum_{i=1}^{n-1} c_i u_i$, $|v|_b^2 = \alpha^2 + \sum_{i=1}^{n-1} c_i^2 = 1$, and

$$R(v) = \|v\|_a^2 = \alpha^2 \|z\|_a^2 + \sum_{i=1}^{n-1} c_i^2 \|u_i\|_a^2 \le \left(\alpha^2 + \sum_{i=1}^{n-1} c_i^2 \right) \|z\|_a^2 = R(z),$$

because $z \in \mathcal{M}_{i-1}^{\perp}$ for all $i = 1, 2, \dots, n-1$ and $R(u_i) \le R(z)$.

$n > 1$: (cont'd)

Let $E \in \mathcal{V}^{(n)}$.

There exists $z \in E : |z|_b \neq 0$ and $b(z, u_i) = 0$ for $i = 1, 2, \dots, n-1$.

$$\max_{v \in E} R(v) \ge R(z) \ge \min_{z \in \mathcal{M}_{n-1}^{\perp}} R(z)$$

□

Example 2: Neumann Laplacian

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n && \text{in } \Omega \\ \frac{\partial u_n}{\partial \nu} &= 0 && \text{on } \partial\Omega \end{aligned}$$

Weak formulation: Find $\lambda_n \in \mathbb{R}$, $u_n \in H^1(\Omega) \setminus \{0\}$:

$$(\nabla u_n, \nabla v) = \lambda_n (u_n, v) \quad \forall v \in H^1(\Omega)$$

Problem: $u_0 \equiv 1$, $\lambda_0 = 0$
$\Rightarrow$ bilinear form $a(u,v) = (\nabla u, \nabla v)$ is not $H^1(\Omega)$-elliptic.

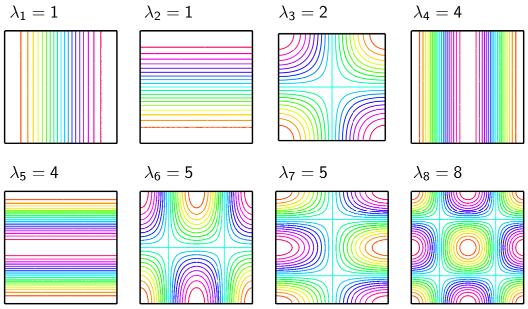
- $V = \{v \in H^1(\Omega) : \int_\Omega v = 0\}$
- $a(u,v) = (\nabla u, \nabla v)$ ... cont., bilin., sym., V-elliptic
- $b(u,v) = (u,v)$ ... cont., bilin., sym., pos. def.
- Compactness: by Rellich theorem.

Exact solution for a square $\Omega = (0,\pi)^2$

$$\lambda_{k,\ell} = k^2 + \ell^2, \quad u_{k,\ell}(x,y) = \cos(kx)\cos(\ell y), \quad k,\ell = 0,1,2,\dots$$

$$\begin{aligned} &\lambda_0 = 0 \ (k=0,\ \ell=0) && \lambda_5 = 4 \ (k=0,\ \ell=2) \\ &\lambda_1 = 1 \ (k=1,\ \ell=0) && \lambda_6 = 5 \ (k=2,\ \ell=1) \\ &\lambda_2 = 1 \ (k=0,\ \ell=1) && \lambda_7 = 5 \ (k=1,\ \ell=2) \\ &\lambda_3 = 2 \ (k=1,\ \ell=1) && \lambda_8 = 8 \ (k=2,\ \ell=2) \\ &\lambda_4 = 4 \ (k=2,\ \ell=0) && \lambda_9 = 9 \ (k=3,\ \ell=0) \end{aligned}$$

Example 2: Neumann Laplacian



Example 3: Steklov eigenvalue problem

$$\begin{aligned} -\Delta u_n + u_n &= 0 && \text{in } \Omega \\ \frac{\partial u_n}{\partial \nu} &= \lambda_n u_n && \text{on } \partial\Omega \end{aligned}$$

Weak formulation: Find $u_n \in H^1(\Omega)$, $\|u_n\|_{L^2(\partial\Omega)} \neq 0$, and $\lambda_n \in \mathbb{R}$:

$$(\nabla u_n, \nabla v) + (u_n, v) = \lambda_n (\gamma u_n, \gamma v)_{\partial\Omega} \quad \forall v \in H^1(\Omega)$$

- $V = H^1(\Omega)$, $V = \mathcal{K} \oplus \mathcal{M}$, $\mathcal{K} = \{v \in H^1(\Omega) : \gamma v = 0 \text{ on } \partial\Omega\}$
  $\mathcal{M} = \{v \in H^1(\Omega) : \gamma v \neq 0 \text{ on } \partial\Omega\}$
- $a(u,v) = (\nabla u, \nabla v) + (u,v)$ ... cont., bilin., sym., V-elliptic
- $b(u,v) = (u,v)_{\partial\Omega}$ ... cont., bilin., sym., pos. semidefinite
- Compactness:
  Trace operator $\gamma : H^1(\Omega) \to L^2(\partial\Omega)$ is compact
  [Kufner, John, Fučík 1997], [Biegert 2009]

Exact solution for a square $\Omega = (-L,L)^2$

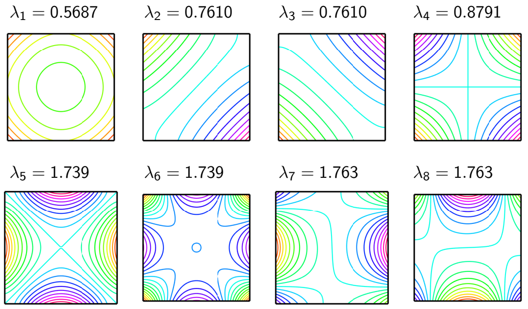
$$\lambda_1 = \frac{\sqrt{2}}{2} \tanh\left(\frac{\sqrt{2}}{2} L\right), \quad u_1(x,y) = \cosh\left(\frac{\sqrt{2}}{2} x\right) \cosh\left(\frac{\sqrt{2}}{2} y\right)$$

$\lambda_2 = ?$

$\lambda_3 = ?$

$$\lambda_4 = \frac{\sqrt{2}}{2} \coth\left(\frac{\sqrt{2}}{2} L\right), \quad u_4(x,y) = \sinh\left(\frac{\sqrt{2}}{2} x\right) \sinh\left(\frac{\sqrt{2}}{2} y\right)$$

Example 3: Steklov eigenvalue problem ($L = \pi/2$)



2. Theory

2.3 Optimal constants

Optimal constants

Abstract eigenvalue problem: Find $\lambda_n \in \mathbb{R}$, $u_n \in V \setminus \{0\}$:

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

Theorem

$$|v|_b \le \lambda_1^{-1/2} \|v\|_a \quad \forall v \in V, \quad \text{with equality for } v = u_1.$$

Proof

Let $v \in V$, $|v|_b \neq 0$.

$$\lambda_1 = \min_{w \in V, |w|_b \neq 0} \frac{\|w\|_a^2}{|w|_b^2} \le \frac{\|v\|_a^2}{|v|_b^2} \quad \Leftrightarrow \quad |v|_b^2 \le \lambda_1^{-1} \|v\|_a^2$$

□

Example 1: Dirichlet Laplacian.

$V = H_0^1(\Omega)$, $\quad \|v\|_a = \|\nabla v\|_{L^2(\Omega)} \quad |v|_b = \|v\|_{L^2(\Omega)}$

Corollary 1. The optimal constant in Friedrichs inequality

$$\|v\|_{L^2(\Omega)} \le C_\mathrm{F} \|\nabla v\|_{L^2(\Omega)} \quad \forall v \in H_0^1(\Omega) \quad \text{is} \quad C_\mathrm{F} = \lambda_1^{-1/2},$$

where λ_1 is the principal eigenvalue of the Dirichlet Laplacian.

- $\Omega = (0, L) \quad \Rightarrow \quad C_\mathrm{F} = \frac{L}{\pi}$
- $\Omega = (0, L_1) \times (0, L_2) \quad \Rightarrow \quad C_F = \frac{1}{\pi} \left(\frac{1}{L_1^2} + \frac{1}{L_2^2} \right)^{-1/2}$

Example 2: Neumann Laplacian.

$V = \{v \in H^1(\Omega) : \int_\Omega v \,\mathrm{d}x = 0\}$, $\quad \|v\|_a = \|\nabla v\|_{L^2(\Omega)}$, $\quad |v|_b = \|v\|_{L^2(\Omega)}$

Corollary 2. The optimal constant in Poincaré inequality

$$\|v\|_{L^2(\Omega)} \le C_\mathrm{P} \|\nabla v\|_{L^2(\Omega)} \quad \forall v \in H^1(\Omega), \int_\Omega v \,\mathrm{d}x = 0, \quad \text{is} \quad C_\mathrm{P} = \lambda_1^{-1/2},$$

where λ_1 is the principal eigenvalue of the Neumann Laplacian.

- $\Omega = (0, L_1) \times (0, L_2) \quad \Rightarrow \quad C_\mathrm{P} = \dfrac{\max\{L_1, L_2\}}{\pi}$

Optimal constants

Abstract eigenvalue problem: Find $\lambda_n \in \mathbb{R}$, $u_n \in V \setminus \{0\}$:

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

Theorem

$$|v|_b \le \lambda_1^{-1/2} \|v\|_a \quad \forall v \in V, \quad \text{with equality for } v = u_1.$$

Example 3: Steklov eigenvalue problem.

$V = H^1(\Omega)$, $\quad \|v\|_a^2 = \|\nabla v\|_{L^2(\Omega)}^2 + \|v\|_{L^2(\Omega)}^2$, $\quad |v|_b = \|v\|_{L^2(\partial\Omega)}$

Corollary 3. The optimal constant in trace inequality

$$\|v\|_{L^2(\partial\Omega)} \le C_\mathrm{T} \|v\|_{H^1(\Omega)} \quad \forall v \in H^1(\Omega) \quad \text{is} \quad C_\mathrm{T} = \lambda_1^{-1/2},$$

where λ_1 is the principal eigenvalue of the Steklov problem.

- $\Omega = (-L, L)^2 \quad \Rightarrow \quad C_\mathrm{T} = \left(\sqrt{2} \coth(\sqrt{2} L/2)\right)^{1/2}$

3. Rayleigh–Ritz (Galerkin) method

Rayleigh–Ritz (Galerkin) method

Eigenvalue problem: Find $\lambda_n \in \mathbb{R}$, $u_n \in V \setminus \{0\}$:

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

Finite dimensional subspace: $V_h \subset V$, $\dim V_h = N < \infty$.

Discrete eigenvalue problem: Find $\lambda_{h,n} \in \mathbb{R}$, $u_{h,n} \in V_h \setminus \{0\}$:

$$a(u_{h,n}, v_h) = \lambda_{h,n} b(u_{h,n}, v_h) \quad \forall v_h \in V_h$$

Properties

Discrete eigenvalue problem: Find $\lambda_{h,n} \in \mathbb{R}$, $u_{h,n} \in V_h \setminus \{0\}$:

$$a(u_{h,n}, v_h) = \lambda_{h,n} b(u_{h,n}, v_h) \quad \forall v_h \in V_h$$

- $0 < \lambda_{h,1} \leq \lambda_{h,2} \leq \cdots \leq \lambda_{h,N}$
- $\dfrac{1}{\lambda_{h,i}} a(u_{h,i}, u_{h,j}) = b(u_{h,i}, u_{h,j}) = \delta_{ij} \quad \forall i,j = 1,2,\dots,N.$
- Minimum principle:

$$\lambda_{h,1} = \min_{v_h \in V_h,\ |v_h|_b \neq 0} R(v_h) \qquad u_{h,1} = \operatorname*{arg\,min}_{v_h \in V_h,\ |v_h|_b \neq 0} R(v_h),$$
$$\lambda_{h,n} = \min_{v_h \in \mathcal{M}^\perp_{h,n-1}} R(v_h) \qquad u_{h,n} = \operatorname*{arg\,min}_{v_h \in \mathcal{M}^\perp_{h,n-1}} R(v_h),$$

where $\mathcal{M}^\perp_{h,n-1} = \{v_h \in V_h : |v_h|_b \neq 0 \text{ and } b(v_h, u_{h,i}) = 0 \quad \forall i = 1,2,\dots,n-1\}$.

Min-max principle:

$$\lambda_{h,n} = \min_{E_h \in \mathcal{V}_h^{(n)}} \max_{v_h \in E_h} R(v_h)$$

where $\mathcal{V}_h^{(n)}$ is the set of all n-dimensional subspaces of V_h.

- Theorem.

$$\lambda_n \leq \lambda_{h,n}, \quad n = 1,2,\dots,N$$

Proof.

$$\mathcal{V}_h^{(n)} \subset \mathcal{V}^{(n)} \quad \Rightarrow \quad \lambda_n = \min_{E \in \mathcal{V}^{(n)}} \max_{v \in E} R(v) \leq \lambda_{h,n} \qquad \square$$

4. Lower bounds on eigenvalues

4.1 Weinstein's bound

Lower bounds – history

Standard (conforming) approach:
Temple (1928), Weinstein (1937), Kato (1949), Lehmann (1949), Goerisch (1985), …

Nonconforming FEM:
Carstensen, Gedicke, Gallistl (2014), Xuefeng LIU (2015), …

Many results: M.G. Armentano, G. Barrenechea, H. Behnke, R.G. Duran, L. Grubišić, Jun Hu, J.R. Kuttler, Y.A. Kuznetsov, Fubiao Lin, Qun Lin, M. Plum, S.I. Repin, V.G. Sigillito, M. Vohralík, Hehu Xie, Yidu Yang, Zhimin Zhang, … *many others*

Recall

Find $\lambda_n \in \mathbb{R}$ and $u_n \in V \setminus \{0\}$ such that

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

- V is a Hilbert space.
- $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are two bilinear forms on V.
- $V = \mathcal{K} \oplus \mathcal{M}$
- $\mathcal{K} = \{v \in V : |v|_b = 0\}$
- $\mathcal{M} = \operatorname{span}\{u_1, u_2, \dots\}$
- $v = v^{\mathcal{K}} + v^{\mathcal{M}}$
- $v^{\mathcal{M}} = \sum_{n=1}^{\infty} c_n u_n, \quad c_n = b(v^{\mathcal{M}}, u_n) = b(v, u_n)$
- $|v|_b^2 = \sum_{n=1}^{\infty} |b(v, u_n)|^2$
- $\|v\|_a^2 = \|v^{\mathcal{K}}\|_a^2 + \|v^{\mathcal{M}}\|_a^2 \quad$ with $\|v^{\mathcal{M}}\|_a^2 = \sum_{n=1}^{\infty} \lambda_n |b(v, u_n)|^2$

Weinstein's bound

Theorem

Let $\lambda_* \in \mathbb{R}$ and $u_* \in V$ with $|u_*|_b \neq 0$ be arbitrary and $w \in V$ be given by

$$a(w, v) = a(u_*, v) - \lambda_* b(u_*, v) \quad \forall v \in V.$$

Then

$$\min_j \frac{|\lambda_j - \lambda_*|^2}{\lambda_j} \leq \frac{\|w\|_a^2}{|u_*|_b^2}.$$

Proof: $\quad w = w^{\mathcal{K}} + w^{\mathcal{M}}$

$$\|w^{\mathcal{M}}\|_a^2 = \sum_{j=1}^{\infty} \lambda_j |b(w, u_j)|^2 = \sum_{j=1}^{\infty} \frac{|a(w, u_j)|^2}{\lambda_j}$$

$$= \sum_{j=1}^{\infty} \frac{|a(u_*, u_j) - \lambda_* b(u_*, u_j)|^2}{\lambda_j} = \sum_{j=1}^{\infty} \frac{|\lambda_j - \lambda_*|^2}{\lambda_j} |b(u_*, u_j)|^2$$

Thus,

$$\|w\|_a^2 \geq \|w^{\mathcal{M}}\|_a^2 \geq \min_j \frac{|\lambda_j - \lambda_*|^2}{\lambda_j} \sum_{j=1}^{\infty} |b(u_*, u_j)|^2 \qquad \square$$

Weinstein's bound

Corollary: Let λ_n has multiplicity m, i.e., $\lambda_{n-1} \neq \lambda_n = \dots = \lambda_{n+m-1} \neq \lambda_{n+m}$. If

$$\sqrt{\lambda_{n-1}\lambda_n} \leq \lambda_* \leq \sqrt{\lambda_n \lambda_{n+m}} \qquad \text{(closeness)}$$

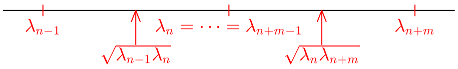
and

$$\|w\|_a \leq \eta$$

then

$$\ell_n \leq \lambda_n,$$

where $\ell_n = \dfrac{1}{4|u_*|_b^2} \left(-\eta + \sqrt{\eta^2 + 4\lambda_* |u_*|_b^2} \right)^2$.


λ_{n-1}
$\lambda_n = \dots = \lambda_{n+m-1}$
λ_{n+m}
$\sqrt{\lambda_{n-1}\lambda_n}$
$\sqrt{\lambda_n\lambda_{n+m}}$

Proof: Clearly,

$$\frac{(\lambda_n - \lambda_*)^2}{\lambda_n} = \min_j \frac{|\lambda_j - \lambda_*|^2}{\lambda_j} \leq \frac{\|w\|_a^2}{|u_*|_b^2} \leq \frac{\eta^2}{|u_*|_b^2}$$

and solve for λ_n. $\square$

Complementary upper bound on the residual

Laplace eigenvalue problem: Find λ_n and $u_n \in H_0^1(\Omega) \setminus \{0\}$:

$$(\nabla u_n, \nabla v) = \lambda_n (u_n, v) \quad \forall v \in H_0^1(\Omega)$$

Definition. Flux $\mathbf{q} \in \mathbf{H}(\operatorname{div}, \Omega)$ is equilibrated if $-\operatorname{div} \mathbf{q} = \lambda_* u_*$.

Theorem. If $\mathbf{q}$ is an equilibrated flux then

$$\|\nabla w\|_0 \leq \eta = \|\nabla u_* - \mathbf{q}\|_0.$$

Proof: Let $v \in H_0^1(\Omega)$, then

$$(\nabla w, \nabla v) = (\nabla u_*, \nabla v) - \lambda_*(u_*, v) - (\operatorname{div} \mathbf{q}, v) - (\mathbf{q}, \nabla v)$$
$$= (\nabla u_* - \mathbf{q}, \nabla v) - (\lambda_* u_* + \operatorname{div} \mathbf{q}, v)$$
$$\leq \|\nabla u_* - \mathbf{q}\|_0 \|\nabla v\|_0 \qquad \square$$

[Neittaanmäki, Repin 2004], [Repin 2008], [Braess, Schöberl, 2008], [Ainsworth, Vejchodský 2011,2014], [Vohralík at al.]

Avoiding equilibration

Shifted eigenvalue problem:

$$\underbrace{(\nabla u_n, \nabla v) + \gamma(u_n, v)}_{a_\gamma(u_n, v)} = (\lambda_n + \gamma)(u_n, v) \quad \forall v \in H_0^1(\Omega)$$

Theorem. Let $\mathbf{q} \in \mathbf{H}(\operatorname{div}, \Omega)$ and $\gamma > 0$. Then

$$\|\nabla w\|_0 \le \|w\|_{a_\gamma} \le \eta, \quad \eta^2 = \|\nabla u_* - \mathbf{q}\|_0^2 + \frac{1}{\gamma}\|\lambda_* u_* + \operatorname{div}\mathbf{q}\|_0^2$$

Proof:

$$\begin{aligned} a_\gamma(w, v) &= (\nabla u_*, \nabla v) - \lambda_*(u_*, v) - (\operatorname{div}\mathbf{q}, v) - (\mathbf{q}, \nabla v) \\ &= (\nabla u_* - \mathbf{q}, \nabla v) - (\lambda_* u_* + \operatorname{div}\mathbf{q}, v) \\ &\le \|\nabla u_* - \mathbf{q}\|_0 \|\nabla v\|_0 + \gamma^{-1/2}\|\lambda_* u_* + \operatorname{div}\mathbf{q}\|_0 \, \gamma^{1/2}\|v\|_0 \\ &\le \left(\|\nabla u_* - \mathbf{q}\|_0^2 + \gamma^{-1}\|\lambda_* u_* + \operatorname{div}\mathbf{q}\|_0^2\right)^{1/2} \left(\|\nabla v\|_0^2 + \gamma\|v\|_0^2\right)^{1/2} \end{aligned}$$

Thus, $\|w\|_{a_\gamma}^2 \le \|\nabla u_* - \mathbf{q}\|_0^2 + \gamma^{-1}\|\lambda_* u_* + \operatorname{div}\mathbf{q}\|_0^2$ □

How to compute $\mathbf{q}$?

Global flux reconstruction: Find $\mathbf{q}_h \in \mathbf{W}_h \subset \mathbf{H}(\operatorname{div}, \Omega)$ minimizing

$$\eta^2 = \|\nabla u_* - \mathbf{q}_h\|_0^2 + \frac{1}{\gamma}\|\lambda_* u_* + \operatorname{div}\mathbf{q}_h\|_0^2$$

FEM space:
$V_h = \{v_h \in V : v_h|_K \in \mathbb{P}^1(K) \; \forall K \in \mathcal{T}_h\}$

FEM approximation:
$u_* = u_{h,n} \in V_h$, $\lambda_* = \lambda_{h,n}$

Raviart–Thomas space:
$\mathbf{RT}_1(K) = [\mathbb{P}^1(K)]^2 \oplus \boldsymbol{x}\mathbb{P}^1(K)$ (local)
$\boldsymbol{W}_h = \{\mathbf{q}_h \in \mathbf{H}(\operatorname{div}, \Omega) : \mathbf{q}_h|_K \in \mathbf{RT}_1(K) \quad \forall K \in \mathcal{T}_h\}$ (global)

Euler–Lagrange equations:

$$(\mathbf{q}_h, \mathbf{w}_h) + \frac{1}{\gamma}(\operatorname{div}\mathbf{q}_h, \operatorname{div}\mathbf{w}_h) = (\nabla u_*, \mathbf{w}_h) - \frac{1}{\gamma}(\lambda_* u_*, \operatorname{div}\mathbf{w}_h) \qquad \forall \mathbf{w}_h \in \boldsymbol{W}_h$$

Equivalent to linear system:

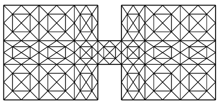
$$M\mathbf{y} = F,$$

where $\mathbf{q}_h = \sum_j y_j \boldsymbol{\psi}_j$, $\quad M_{ij} = (\boldsymbol{\psi}_j, \boldsymbol{\psi}_i) + \frac{1}{\gamma}(\operatorname{div}\boldsymbol{\psi}_j, \operatorname{div}\boldsymbol{\psi}_i)$,

$$F_i = (\nabla u_*, \boldsymbol{\psi}_i) - \frac{1}{\gamma}(\lambda_* u_*, \operatorname{div}\boldsymbol{\psi}_i)$$

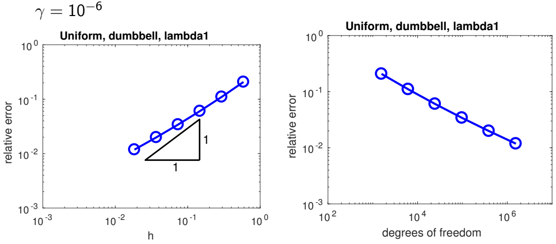
Example: dumbbell

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n \quad && \text{in } \Omega = \text{dumbbell} \\ u_n &= 0 && \text{on } \partial\Omega \end{aligned}$$



Rel. error: $\dfrac{|\lambda_n - \lambda_{h,n}|}{\lambda_n} \le \dfrac{\lambda_{h,n} - \ell_n}{\ell_n}$

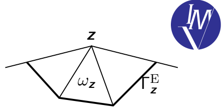
$\gamma = 10^{-6}$



Local flux reconstruction

Flux reconstruction:

$$\mathbf{q}_h = \sum_{\boldsymbol{z} \in \mathcal{N}_h} \mathbf{q}_{\boldsymbol{z}}$$



$\boldsymbol{z}$
$\omega_{\boldsymbol{z}}$
$\Gamma_{\boldsymbol{z}}^{\mathrm{E}}$

Local problems: Find $\mathbf{q}_{\boldsymbol{z}} \in \mathbf{W}_{\boldsymbol{z}}$ minimizing

$$\|\varphi_{\boldsymbol{z}} \nabla u_* - \mathbf{q}_{\boldsymbol{z}}\|_{L^2(\omega_{\boldsymbol{z}})}^2 + \frac{1}{\gamma}\|\lambda_* \varphi_{\boldsymbol{z}} u_* + \operatorname{div}\mathbf{q}_{\boldsymbol{z}}\|_{L^2(\omega_{\boldsymbol{z}})}^2$$

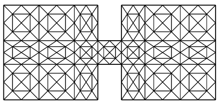
Euler–Lagrange equations:

$$(\mathbf{q}_{\boldsymbol{z}}, \mathbf{w}_h)_{\omega_{\boldsymbol{z}}} + \frac{1}{\gamma}(\operatorname{div}\mathbf{q}_{\boldsymbol{z}}, \operatorname{div}\mathbf{w}_h)_{\omega_{\boldsymbol{z}}} = (\varphi_{\boldsymbol{z}} \nabla u_*, \mathbf{w}_h)_{\omega_{\boldsymbol{z}}} - \frac{1}{\gamma}(\lambda_* \varphi_{\boldsymbol{z}} u_*, \operatorname{div}\mathbf{w}_h)_{\omega_{\boldsymbol{z}}} \qquad \forall \mathbf{w}_h \in \boldsymbol{W}_{\boldsymbol{z}}$$

Patch of elements: $\omega_{\boldsymbol{z}} = \bigcup\{K \in \mathcal{T}_h : \boldsymbol{z} \in K\}$
Partition of unity: $\sum_{\boldsymbol{z} \in \mathcal{N}_h} \varphi_{\boldsymbol{z}} = 1$
$\boldsymbol{W}_{\boldsymbol{z}} = \{\mathbf{q} \in \mathbf{H}(\operatorname{div}, \omega_{\boldsymbol{z}}) : \mathbf{q}|_K \in \mathbf{RT}_1(K) \; \forall K \subset \omega_{\boldsymbol{z}}, \; \mathbf{q} \cdot \boldsymbol{n}_{\boldsymbol{z}} = 0 \text{ on } \Gamma_{\boldsymbol{z}}^{\mathrm{E}}\}$

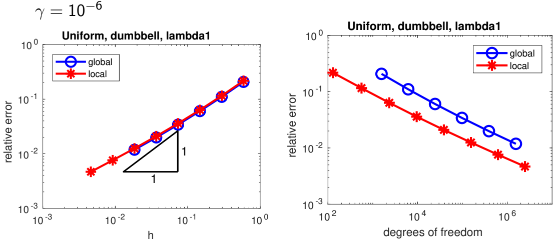
Example: dumbbell

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n && \text{in } \Omega = \text{dumbbell} \\ u_n &= 0 && \text{on } \partial\Omega \end{aligned}$$



Rel. error: $\dfrac{|\lambda_n - \lambda_{h,n}|}{\lambda_n} \le \dfrac{\lambda_{h,n} - \ell_n}{\ell_n}$

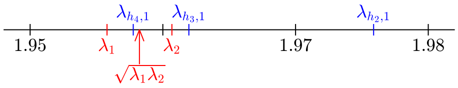
$\gamma = 10^{-6}$



Closeness assumption for dumbbell

$$\sqrt{\lambda_{n-1}\lambda_n} \le \lambda_* \le \sqrt{\lambda_n \lambda_{n+m}} \quad \Rightarrow \quad \ell_n \le \lambda_n$$

Exact eigenvalues: $\lambda_1 = 1.955793794588$, $\lambda_2 = 1.960683031595$



h	ℓ_1	$\lambda_{h,1}$	closeness
$h_1 = 1.1781$	1.6618	2.0228	no
$h_2 = 0.5890$	1.7711	1.9759	no
$h_3 = 0.2945$	1.8449	1.9620	no
$h_4 = 0.1473$	1.8899	1.9578	yes
$h_5 = 0.0736$	1.9163	1.9565	yes
$h_6 = 0.0368$	1.9319	1.9560	yes
$h_7 = 0.0184$	1.9411	1.9559	yes

Weinstein's bound – summary

- easy to use
- it is a generalization of Bauer–Fike estimates for matrices
- good for general symmetric elliptic problems
- sub-optimal speed of convergence
- a priori information on spectrum needed for guaranteed lower bounds

4. Lower bounds on eigenvalues

4.2 Lehmann–Goerisch method

Lehmann–Goerisch method

General setting:

Find $\lambda_n \in \mathbb{R}$ and $u_n \in V \setminus \{0\}$ such that

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

Lehmann method

Theorem

Let $\tilde{\lambda}_N < \rho \le \lambda_{N+1}$

- $\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_N \in V$ be linearly independent
- $A_{0,ij} = a(\tilde{u}_i, \tilde{u}_j)$
- $A_{1,ij} = b(\tilde{u}_i, \tilde{u}_j)$
- $w_i \in V: \quad a(w_i, v) = b(\tilde{u}_i, v) \quad \forall v \in V$
  $A_{2,ij} = a(w_i, w_j)$
- $\mu_1 \le \mu_2 \le \dots \le \mu_N: \quad (\rho A_1 - A_0)\boldsymbol{x} = \mu(A_0 - 2\rho A_1 + \rho^2 A_2)\boldsymbol{x}$

Then $0 < \mu_1$ and

$$\rho - \frac{\rho}{1 + \mu_n} \le \lambda_n, \quad n = 1, 2, \dots, N.$$

[Lehmann 1949, 1950], [Goerisch, Haunhorst 1985]

Lehmann–Goerisch method

Theorem

Let $\tilde{\lambda}_N < \rho \le \lambda_{N+1}$

- $\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_N \in V$ be linearly independent
- $A_{0,ij} = a(\tilde{u}_i, \tilde{u}_j)$
- $A_{1,ij} = b(\tilde{u}_i, \tilde{u}_j)$
- X … vector space
  $\mathcal{B}$ … positive semidefinite symmetric bilinear form on X
  $T: V \to X$ … linear operator:
  (a) $\mathcal{B}(Tu, Tv) = a(u, v) \quad \forall u, v \in V$
  (b) $\hat{\mathbf{w}}_i \in X: \quad \mathcal{B}(\hat{\mathbf{w}}_i, Tv) = b(\tilde{u}_i, v) \quad \forall v \in V$
  (c) $\hat{A}_{2,ij} = \mathcal{B}(\hat{\mathbf{w}}_i, \hat{\mathbf{w}}_j)$
- $\hat{\mu}_1 \le \hat{\mu}_2 \le \dots \le \hat{\mu}_N: \quad (\rho A_1 - A_0)\hat{\boldsymbol{x}} = \hat{\mu}(A_0 - 2\rho A_1 + \rho^2 \hat{A}_2)\hat{\boldsymbol{x}}$

Then $0 < \hat{\mu}_1$ and

$$\rho - \frac{\rho}{1 + \hat{\mu}_n} \le \lambda_n, \quad n = 1, 2, \dots, N.$$

[Lehmann 1949, 1950], [Goerisch, Haunhorst 1985]

Proof: Lehmann $\Rightarrow$ Goerisch

It suffices to show that $\hat{A}_2 - A_2$ is positive semidefinite, because

$\Rightarrow$ $0 < \hat{\mu}_i \le \mu_i$ for all $i = 1, 2, \dots, N$,

$\Rightarrow$ $\rho - \frac{\rho}{1 + \hat{\mu}_n} \le \rho - \frac{\rho}{1 + \mu_n} \le \lambda_n$.

To show that $\hat{A}_2 - A_2$ is positive semidefinite:
Let $\boldsymbol{x} \in \mathbb{R}^N$, $\tilde{u} = \sum_{i=1}^N x_i \tilde{u}_i$, $w = \sum_{i=1}^N x_i w_i$, $\hat{\mathbf{w}} = \sum_{i=1}^N x_i \hat{\mathbf{w}}_i$, and

$$0 \le \mathcal{B}(\hat{\mathbf{w}} - Tw, \hat{\mathbf{w}} - Tw) = \mathcal{B}(\hat{\mathbf{w}}, \hat{\mathbf{w}}) - 2\underbrace{\mathcal{B}(\hat{\mathbf{w}}, Tw)}_{\substack{\overset{(b)}{=} b(\tilde{u}, w) \\ = a(w, w)}} + \underbrace{\mathcal{B}(Tw, Tw)}_{\overset{(a)}{=} a(w, w)}.$$

Thus,

$$0 \le \mathcal{B}(\hat{\mathbf{w}}, \hat{\mathbf{w}}) - a(w, w) \overset{(c)}{=} \boldsymbol{x}^T(\hat{A}_2 - A_2)\boldsymbol{x}.$$

□

Application to Laplace eigenvalue problem

$$(\nabla u_i, \nabla v) + \gamma(u_i, v) = (\lambda_i + \gamma)(u_i, v) \quad \forall v \in H^1_0(\Omega),\ \Omega \subset \mathbb{R}^2$$

Setting

- $V = H^1_0(\Omega)$, $a(u, v) = (\nabla u, \nabla v) + \gamma(u, v)$, $b(u, v) = (u, v)$
- $X = \left[L^2(\Omega)\right]^3$
- $\mathcal{B}(\hat{\mathbf{u}}, \hat{\mathbf{v}}) = (\hat{u}_1, \hat{v}_1) + (\hat{u}_2, \hat{v}_2) + \gamma(\hat{u}_3, \hat{v}_3)$
- $Tu = \begin{pmatrix} \nabla u \\ u \end{pmatrix}$

Facts

(a) $\mathcal{B}(Tu, Tv) = a(u, v)$

(b) $\mathcal{B}(\hat{\mathbf{w}}_i, Tv) = b(\tilde{u}_i, v) \Leftarrow \hat{\mathbf{w}}_i = \begin{pmatrix} \boldsymbol{\sigma}_i \\ \hat{w}_{i,3} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\sigma}_i \\ \frac{1}{\gamma}(\tilde{u}_i + \operatorname{div} \boldsymbol{\sigma}_i) \end{pmatrix}$

$\boldsymbol{\sigma}_i \in \mathbf{H}(\operatorname{div}, \Omega)$

$$\begin{aligned} (\boldsymbol{\sigma}_i, \nabla v) + \gamma(\hat{w}_{i,3}, v) &= (\tilde{u}_i, v) \quad \forall v \in V \\ -(\operatorname{div} \boldsymbol{\sigma}_i, v) + \gamma(\hat{w}_{i,3}, v) &= (\tilde{u}_i, v) \quad \forall v \in V \\ \hat{w}_{i,3} &= \frac{1}{\gamma}(\tilde{u}_i + \operatorname{div} \boldsymbol{\sigma}_i) \end{aligned}$$

(c) $\hat{A}_{2,ij} = \mathcal{B}(\hat{\mathbf{w}}_i, \hat{\mathbf{w}}_j) \;\Leftrightarrow\; \hat{A}_{2,ij} = (\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) + \frac{1}{\gamma}(\tilde{u}_i + \operatorname{div} \boldsymbol{\sigma}_i, \tilde{u}_j + \operatorname{div} \boldsymbol{\sigma}_j)$

Application to Laplace eigenvalue problem

Theorem (Lehmann–Goerisch)

Let $\tilde{\lambda}_N + \gamma < \rho \leq \lambda_{N+1} + \gamma$, $\gamma > 0$

- $\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_N \in V$ be linearly independent
- $A_{0,ij} = (\nabla \tilde{u}_i, \nabla \tilde{u}_j) + \gamma(\tilde{u}_i, \tilde{u}_j)$
- $A_{1,ij} = (\tilde{u}_i, \tilde{u}_j)$
- $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \dots, \boldsymbol{\sigma}_N \in \mathbf{H}(\operatorname{div}, \Omega)$ be arbitrary
  $\hat{A}_{2,ij} = (\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) + \frac{1}{\gamma}(\tilde{u}_i + \operatorname{div} \boldsymbol{\sigma}_i, \tilde{u}_j + \operatorname{div} \boldsymbol{\sigma}_j)$
- $\hat{\mu}_1 \leq \hat{\mu}_2 \leq \dots \leq \hat{\mu}_N: \quad (\rho A_1 - A_0)\hat{\mathbf{x}} = \hat{\mu}(A_0 - 2\rho A_1 + \rho^2 \hat{A}_2)\hat{\mathbf{x}}$

Then $0 < \hat{\mu}_1$ and

$$\ell_n = \rho - \gamma - \frac{\rho}{1 + \hat{\mu}_n} \leq \lambda_n, \quad n = 1, 2, \dots, N$$

[Behnke, Mertins, Plum, Wieners 2000]

How to find good $\hat{\mathbf{w}}_i$?

Observation: Let $\tilde{u}_i \approx u_i$ and $\tilde{\lambda}_i \approx \lambda_i$.

$\Rightarrow$ $a(w_i, v) = b(\tilde{u}_i, v) \approx \frac{1}{\tilde{\lambda}_i} a(\tilde{u}_i, v) \quad \forall v \in V$

$\Rightarrow$ $w_i \approx \frac{1}{\tilde{\lambda}_i} \tilde{u}_i$

Need

$\Rightarrow$ $\hat{A}_2 \approx A_2$

$\Rightarrow$ $\mathcal{B}(\hat{\mathbf{w}}_i, \hat{\mathbf{w}}_j) \approx a(w_i, w_j) \overset{(a)}{=} \mathcal{B}(Tw_i, Tw_j)$

$\Rightarrow$ $\hat{\mathbf{w}}_i \approx Tw_i \approx \frac{1}{\tilde{\lambda}_i} T\tilde{u}_i$

Natural idea

make $|\frac{1}{\tilde{\lambda}_i} T\tilde{u}_i - \hat{\mathbf{w}}_i|^2_{\mathcal{B}}$ small

For Laplacian: Find $\boldsymbol{\sigma}_{h,i} \in \mathbf{H}(\operatorname{div}, \Omega)$ that

makes $\left\| \frac{\nabla u_{h,i}}{\lambda_{h,i} + \gamma} - \boldsymbol{\sigma}_{h,i} \right\|^2_0 + \frac{1}{\gamma} \left\| \frac{\lambda_{h,i} u_{h,i}}{\lambda_{h,i} + \gamma} + \operatorname{div} \boldsymbol{\sigma}_{h,i} \right\|^2_0$ small

Choice of $\boldsymbol{\sigma}_i$ – global

Global minimization:

Find $\boldsymbol{\sigma}_{h,i} \in \boldsymbol{W}_h \subset \mathbf{H}(\operatorname{div}, \Omega)$, $i = 1, 2, \dots, N$, that minimizes

$$\left\| \frac{\nabla u_{h,i}}{\lambda_{h,i} + \gamma} - \boldsymbol{\sigma}_{h,i} \right\|^2_0 + \frac{1}{\gamma} \left\| \frac{\lambda_{h,i} u_{h,i}}{\lambda_{h,i} + \gamma} + \operatorname{div} \boldsymbol{\sigma}_{h,i} \right\|^2_0$$

Euler–Lagrange equations:

$$(\boldsymbol{\sigma}_{h,i}, \mathbf{w}_h) + \frac{1}{\gamma}(\operatorname{div} \boldsymbol{\sigma}_{h,i}, \operatorname{div} \mathbf{w}_h) = \left(\frac{\nabla u_{h,i}}{\lambda_{h,i} + \gamma}, \mathbf{w}_h \right) - \frac{1}{\gamma} \left(\frac{\lambda_{h,i} u_{h,i}}{\lambda_{h,i} + \gamma}, \operatorname{div} \mathbf{w}_h \right)$$

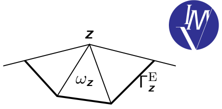
$$\forall \mathbf{w}_h \in \boldsymbol{W}_h$$

$\boldsymbol{W}_h = \{ \boldsymbol{\sigma}_h \in \mathbf{H}(\operatorname{div}, \Omega) : \boldsymbol{\sigma}_h|_K \in \mathbf{RT}_1(K) \quad \forall K \in \mathcal{T}_h \}$

Choice of $\boldsymbol{\sigma}_i$ – local

Flux reconstruction:

$$\boldsymbol{\sigma}_{h,i} = \sum_{\boldsymbol{z}\in\mathcal{N}_h} \boldsymbol{\sigma}_{\boldsymbol{z},i}$$


$\boldsymbol{z}$
$\omega_{\boldsymbol{z}}$
$\Gamma_{\boldsymbol{z}}^{\mathrm{E}}$

Local problems: Find $\boldsymbol{\sigma}_{\boldsymbol{z},i} \in \boldsymbol{W}_{\boldsymbol{z}}$, $i = 1, 2, \ldots, N$ minimizing

$$\left\| \varphi_{\boldsymbol{z}} \frac{\nabla u_{h,i}}{\lambda_{h,i} + \gamma} - \boldsymbol{\sigma}_{\boldsymbol{z},i} \right\|_{0,\omega_{\boldsymbol{z}}}^2 + \frac{1}{\gamma} \left\| \frac{\lambda_{h,i}\varphi_{\boldsymbol{z}} u_{h,i}}{\lambda_{h,i} + \gamma} + \operatorname{div} \boldsymbol{\sigma}_{\boldsymbol{z},i} \right\|_{0,\omega_{\boldsymbol{z}}}^2$$

Euler–Lagrange equations:

$$(\boldsymbol{\sigma}_{\boldsymbol{z},i}, \mathbf{w}_h)_{\omega_{\boldsymbol{z}}} + \frac{1}{\gamma} (\operatorname{div} \boldsymbol{\sigma}_{\boldsymbol{z},i}, \operatorname{div} \mathbf{w}_h)_{\omega_{\boldsymbol{z}}}$$
$$= \left(\varphi_{\boldsymbol{z}} \frac{\nabla u_{h,i}}{\lambda_{h,i} + \gamma}, \mathbf{w}_h \right)_{\omega_{\boldsymbol{z}}} - \frac{1}{\gamma} \left(\frac{\varphi_{\boldsymbol{z}} \lambda_{h,i} u_{h,i}}{\lambda_{h,i} + \gamma}, \operatorname{div} \mathbf{w}_h \right)_{\omega_{\boldsymbol{z}}} \quad \forall \mathbf{w}_h \in \boldsymbol{W}_{\boldsymbol{z}}$$

Patch of elements: $\omega_{\boldsymbol{z}} = \bigcup\{K \in \mathcal{T}_h : \boldsymbol{z} \in K\}$
Partition of unity: $\sum_{\boldsymbol{z}\in\mathcal{N}_h} \varphi_{\boldsymbol{z}} = 1$
$\boldsymbol{W}_{\boldsymbol{z}} = \{\boldsymbol{\sigma} \in \mathbf{H}(\operatorname{div}, \omega_{\boldsymbol{z}}) : \boldsymbol{\sigma}|_K \in \mathbf{RT}_1(K) \ \forall K \subset \omega_{\boldsymbol{z}},\ \boldsymbol{\sigma} \cdot \boldsymbol{n}_{\boldsymbol{z}} = 0 \text{ on } \Gamma_{\boldsymbol{z}}^{\mathrm{E}}\}$

Comparison of flux reconstructions

Weinstein: Find $\mathbf{q}_{h,i} \in \mathbf{W}_h$ minimizing

$$\|\nabla u_{h,i} - \mathbf{q}_{h,i}\|_0^2 + \frac{1}{\gamma} \|\lambda_{h,i} u_{h,i} + \operatorname{div} \mathbf{q}_{h,i}\|_0^2$$

Lehmann–Goerisch: Find $\boldsymbol{\sigma}_{h,i} \in \boldsymbol{W}_h$ minimizing

$$\left\| \frac{\nabla u_{h,i}}{\lambda_{h,i} + \gamma} - \boldsymbol{\sigma}_{h,i} \right\|_0^2 + \frac{1}{\gamma} \left\| \frac{\lambda_{h,i} u_{h,i}}{\lambda_{h,i} + \gamma} + \operatorname{div} \boldsymbol{\sigma}_{h,i} \right\|_0^2$$

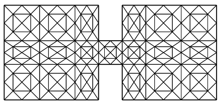
Thus,

$$\boldsymbol{\sigma}_{h,i} = \frac{\mathbf{q}_{h,i}}{\lambda_{h,i} + \gamma}$$

[Vejchodský 2018]

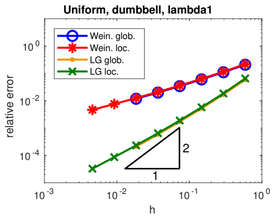
Example: dumbbell

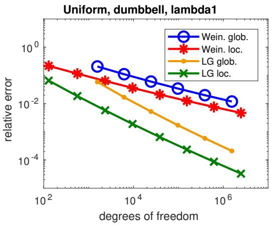
$$-\Delta u_n = \lambda_n u_n \quad \text{in } \Omega = \text{dumbbell}$$
$$u_n = 0 \quad \text{on } \partial\Omega$$



Rel. error: $\dfrac{|\lambda_n - \lambda_{h,n}|}{\lambda_n} \le \dfrac{\lambda_{h,n} - \ell_n}{\ell_n}$

$\gamma = 10^{-6}$





How to get the a priori lower bound ρ?

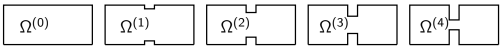
Monotonicity principle: If $V \subset \widetilde{V}$ then $\mathcal{V}^{(n)} \subset \widetilde{\mathcal{V}}^{(n)}$ and

$$\tilde{\lambda}_n = \min_{E\in\widetilde{\mathcal{V}}^{(n)}} \max_{v\in E} R(v) \le \min_{E\in\mathcal{V}^{(n)}} \max_{v\in E} R(v) = \lambda_n$$

Example 1.
$\Omega \subset \widetilde{\Omega} \quad \Rightarrow \quad H_0^1(\Omega) \subset H_0^1(\widetilde{\Omega}) \quad \Rightarrow \quad \tilde{\lambda}_n \le \lambda_n$

Example 2.
$H_0^1(\Omega) \subset H^1(\Omega) \quad \Rightarrow \quad \lambda_n^{\mathrm{Neumann}} \le \lambda_n^{\mathrm{Dirichlet}}$

Homotopy



Analytically:	$\rho = 12.16$	$\rho = 11.39$	$\rho = 10.77$	$\rho = 9.988$
$12.16 \le \lambda_{17}^{(0)}$	$\ell_{15} \doteq 11.39$	$\ell_{13} \doteq 10.77$	$\ell_{11} \doteq 9.988$	

[Plum 1990, 1991]

Adaptive mesh refinement

Recall the residual

$$w \in V: \quad (\nabla w, \nabla v) = (\nabla u_{h,i}, \nabla v) - \lambda_{h,i}(u_{h,i}, v) \quad \forall v \in V$$

Recall theorem:

$$\|\nabla w\|_0 \le \eta, \quad \text{where } \eta^2 = \|\nabla u_{h,i} - \mathbf{q}_{h,i}\|^2_{L^2(\Omega)} + \frac{1}{\gamma}\|\lambda_{h,i} u_{h,i} + \operatorname{div} \mathbf{q}_{h,i}\|^2_{L^2(\Omega)}$$

Local error indicators for mesh refinement:

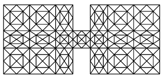
$$\eta_K^2 = \|\nabla u_{h,i} - \mathbf{q}_{h,i}\|^2_{L^2(K)} + \frac{1}{\gamma}\|\lambda_{h,i} u_{h,i} + \operatorname{div} \mathbf{q}_{h,i}\|^2_{L^2(K)}$$

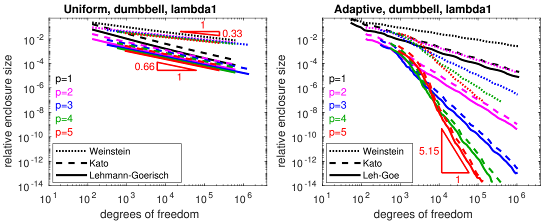
Note: Good for both Weinstein and Lehmann–Goerisch method:

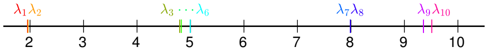
$$\boldsymbol{\sigma}_{h,i} = \frac{\mathbf{q}_{h,i}}{\lambda_{h,i} + \gamma}$$

Example: dumbbell

$$\begin{aligned} -\Delta u_i &= \lambda_i u_i && \text{in } \Omega \\ u_i &= 0 && \text{on } \partial\Omega \end{aligned}$$







Computed bounds ($p = 5$, adaptive):

$$1.9557937945883 \le \lambda_1 \le 1.9557937945884$$
$$1.9606830315950 \le \lambda_2 \le 1.9606830315951$$
$$4.800761124039 \le \lambda_3 \le 4.800761124045$$
$$4.829895254505 \le \lambda_4 \le 4.829895254510$$
$$4.996837097289 \le \lambda_5 \le 4.996837097290$$
$$4.9968509041015 \le \lambda_6 \le 4.9968509041016$$
$$7.986967292128 \le \lambda_7 \le 7.986967292138$$
$$7.987034306816 \le \lambda_8 \le 7.987034306827$$

Lehmann–Goerisch method – summary

- optimal speed of convergence
- implementation based on standard FEM
- adaptivity for free
- naturally generalize to higher orders
- good for a wide class of problems
- an a priori lower bound on some eigenvalue is needed

4. Lower bounds on eigenvalues

4.3 Interpolation constant based methods

[Carstensen, Gallistl, Gedicke 2014], [Liu 2015]

Nonconforming approximation

Eigenvalue problem: Find λ_n and $u_n \in V \setminus \{0\}$ such that

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V$$

Finite dimensional space: $\dim V_h = N < \infty$, but it can be $V_h \not\subset V$.
Discrete eigenvalue problem: Find $\lambda_{h,n} \in \mathbb{R}$, $u_{h,n} \in V_h \setminus \{0\}$:

$$a(u_{h,n}, v_h) = \lambda_{h,n} b(u_{h,n}, v_h) \quad \forall v_h \in V_h$$

Definition:
$V(h) = V + V_h = \{v + v_h : v \in V,\ v_h \in V_h\}$

Extensions of bilinear forms:
$a_h, b_h : V(h) \times V(h) \to \mathbb{R}$
$a_h(u, v) = a(u, v)$ and $b_h(u, v) = b(u, v) \quad \forall u, v \in V$
$a_h(\cdot, \cdot)$ is symmetric and $V(h)$-elliptic
$b_h(\cdot, \cdot)$ is symmetric and positive semidefinite on $V(h)$
Notation: $a = a_h$ and $b = b_h$

Lemmas

Lemma 1 (Discrete Friedrichs inequality).

$$|v_h|_b \le \lambda_{h,1}^{-1/2} \|v_h\|_a \quad \forall v_h \in V_h$$

Proof. $\lambda_{h,1} = \min_{w_h \in V_h} \dfrac{\|w_h\|_a^2}{|w_h|_b^2} \le \dfrac{\|v_h\|_a^2}{|v_h|_b^2}$ □

Elliptic projection: $P_h : V(h) \to V_h$

$$a(u - P_h u, v_h) = 0 \quad \forall v_h \in V_h$$

Lemma 2.

$$\|u\|_a^2 = \|P_h u\|_a^2 + \|u - P_h u\|_a^2$$

Proof.
$\|u - P_h u\|_a^2 = \|u\|_a^2 - 2a(u, P_h u) + \|P_h u\|_a^2$
$a(u, P_h u) = a(P_h u, P_h u) = \|P_h u\|_a^2$ □

Lower bound

Theorem. Let $|u - P_h u|_b \le C_h \|u - P_h u\|_a$. Then

$$\frac{\lambda_{h,n}}{1 + \lambda_{h,n} C_h^2} \le \lambda_n, \quad n = 1, 2, \dots, N.$$

Proof (for λ_1 only). Let $v \in V$.

$$\begin{aligned}
|v|_b &\le |P_h v|_b + |v - P_h v|_b \\
&\le \lambda_{h,1}^{-1/2} \|P_h v\|_a + C_h \|v - P_h v\|_a \\
&\le \left(\lambda_{h,1}^{-1} + C_h^2\right)^{1/2} \left(\|P_h v\|_a^2 + \|v - P_h v\|_a^2\right)^{1/2} \\
&= \left(\frac{1 + \lambda_{h,1} C_h^2}{\lambda_{h,1}}\right)^{1/2} \|v\|_a
\end{aligned}$$

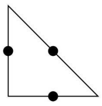
$$\lambda_1 = \min_{v \in V} \frac{\|v\|_a^2}{|v|_b^2} \ge \frac{\lambda_{h,1}}{1 + \lambda_{h,1} C_h^2}$$

□

Crouzeix–Raviart (CR) elements

Laplace eigenvalue problem: Find $\lambda_n \in \mathbb{R}$, $u_n \in H_0^1(\Omega) \setminus \{0\}$:

$$(\nabla u_n, \nabla v) = \lambda_n (u_n, v) \quad \forall v \in H_0^1(\Omega)$$



CR space: $v_h \in V_h^{\mathrm{CR}}$ if

- $v_h|_K \in \mathbb{P}^1(K)$
- v_h is continuous at midpoints of interior edges
- $v_h = 0$ at midpoints of boundary edges

CR eigenvalue problem: Find $\lambda_{h,i}^{\mathrm{CR}} \in \mathbb{R}$, $u_{h,i}^{\mathrm{CR}} \in V_h^{\mathrm{CR}} \setminus \{0\}$:

$$(\nabla u_{h,i}^{\mathrm{CR}}, \nabla v_h) = \lambda_{h,i}^{\mathrm{CR}} (u_{h,i}^{\mathrm{CR}}, v_h) \quad \forall v_h \in V_h^{\mathrm{CR}}.$$

Crouzeix–Raviart interpolation

Let e_i, $i = 1, 2, 3$, be edges of triangle K.

Definition: $\Pi_h : H^1(K) \to \mathbb{P}^1(K)$ such that

$$\int_{e_i} u - \Pi_h u \, \mathrm{d}s = 0 \quad \forall i = 1, 2, 3.$$

Note: If m_i is a midpoint of e_i then $\Pi_h u(m_i) = \frac{1}{|e_i|} \int_{e_i} u \, \mathrm{d}s$.

Lemma. $\Pi_h = P_h$

Proof.

Let $u \in H^1(\Omega) \oplus V_h^{\mathrm{CR}}$ and $v_h \in V_h^{\mathrm{CR}}$.

$$a(u - \Pi_h u, v_h) = \sum_{K \in \mathcal{T}_h} \int_K \nabla(u - \Pi_h u) \cdot \nabla v_h$$

$$= \sum_{K \in \mathcal{T}_h} \left(\sum_{i=1}^{3} \int_{e_i} (u - \Pi_h u) \underbrace{\frac{\partial v_h}{\partial \boldsymbol{n}}}_{=\text{const.}} \, \mathrm{d}s - \int_K (u - \Pi_h u) \underbrace{\Delta v_h}_{=0} \, \mathrm{d}x \right) = 0$$

□

The value of C_h

Interpolation error estimate:

$$\|u - \Pi_h u\|_{L^2(\Omega)} \leq C_h \|\nabla u - \nabla \Pi_h u\|_{L^2(\Omega)}$$

Local interpolation error estimate:

$$\|u - \Pi_h u\|_{L^2(K)} \leq C_h(K) \|\nabla u - \nabla \Pi_h u\|_{L^2(K)}$$

Lemma.

$$C_h \leq \max_{K \in \mathcal{T}_h} C_h(K)$$

Proof.

$$\|u - \Pi_h u\|_{L^2(\Omega)}^2 = \sum_{K \in \mathcal{T}_h} \|u - \Pi_h u\|_{L^2(K)}^2 \leq \sum_{K \in \mathcal{T}_h} C_h^2(K) \|\nabla u - \nabla \Pi_h u\|_{L^2(K)}^2$$

$$\leq \max_{K \in \mathcal{T}_h} C_h^2(K) \|\nabla u - \nabla \Pi_h u\|_{L^2(\Omega)}^2$$

□

Explicit estimates of C_h

Interval

- $C_h = h/\pi$

Triangle

- $C_h = 0.4396h$ [Carstensen, Gedicke 2014]
- $C_h = 0.2983h$ [Carstensen, Gallistl 2014]
- $C_h = 0.1893h$ [Liu 2015]

Tetrahedron

- $C_h = 0.3804h$ [Liu 2015]

Explicit estimate of C_h for an interval

Setting: $\Omega = (\alpha, \beta)$, $V = H_0^1(\alpha, \beta)$,
$a(u, v) = \int_\alpha^\beta u' v' \, \mathrm{d}x$, $b(u, v) = \int_\alpha^\beta uv \, \mathrm{d}x$

Partition: $\alpha = z_0 < z_1 < \cdots < z_N = \beta$

Elements: $K_i = [z_{i-1}, z_i]$, $i = 1, 2, \ldots, N$,
$h_i = z_i - z_{i-1}$, $h = \max_{i=1,\ldots,N} h_i$

CR space: $V_h = \{v \in H_0^1(\alpha, \beta) : v|_{K_i} \in \mathbb{P}^1(K_i),\ i = 1, 2, \ldots, N\}$

Interpolation: $\Pi_h : H_0^1(\alpha, \beta) \to V_h$
$(\Pi_h u)(x_i) = u(x_i)$, $i = 0, \ldots, N$

Lemma.

$$\|u - \Pi_h u\|_{L^2(\Omega)} \leq \frac{h}{\pi} \|u' - (\Pi_h u)'\|_{L^2(\Omega)}$$

Proof.

$$\min_{v \in H^1(K_i)} R(v - \Pi_h v) = \min_{w \in H_0^1(K_i)} R(w) = R\left(\sin \frac{\pi(x - z_i)}{h_i}\right) = \pi^2 / h_i^2$$

□

Upper bound

Interpolation to continuous functions: $\mathcal{I} : V_h^{\mathrm{CR}} \to \widetilde{V}_h \subset H^1(\Omega)$
Examples:

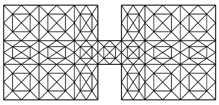
- Oswald quasi-interpolation [Oswald 1994]
- Interpolation to refined mesh [Carstensen, Merdon 2013]

Upper bound

- $\mathcal{T}_h^*$ is the red refinement of $\mathcal{T}_h$
- $u_{h,i}^* = \mathcal{I}_{\mathrm{CM}} \tilde{u}_{h,i}^{\mathrm{CR}}$ for $i = 1, 2, \dots, m$
- $\boldsymbol{S}, \boldsymbol{Q} \in \mathbb{R}^{m \times m}$ with entries $\boldsymbol{S}_{j,k} = (\nabla u_{h,j}^*, \nabla u_{h,k}^*)$ and $\boldsymbol{Q}_{j,k} = (u_{h,j}^*, u_{h,k}^*)$
- $\boldsymbol{S}\mathbf{y}_i = \Lambda_i^* \boldsymbol{Q}\mathbf{y}_i, \quad i = 1, 2, \dots, m$
- $\Lambda_1^* \leq \Lambda_2^* \leq \dots \leq \Lambda_m^*$
- $\lambda_i \leq \Lambda_i^* \quad$ for $i = 1, 2, \dots, m$

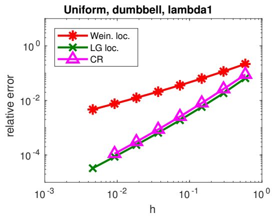
Example: dumbbell

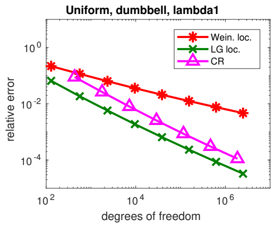
$$\begin{aligned} -\Delta u_n &= \lambda_n u_n && \text{in } \Omega = \text{dumbbell} \\ u_n &= 0 && \text{on } \partial\Omega \end{aligned}$$



Rel. error: $\dfrac{|\lambda_n - \lambda_{h,n}|}{\lambda_n} \leq \dfrac{\lambda_{h,n} - \ell_n}{\ell_n}$

$\gamma = 10^{-6}$





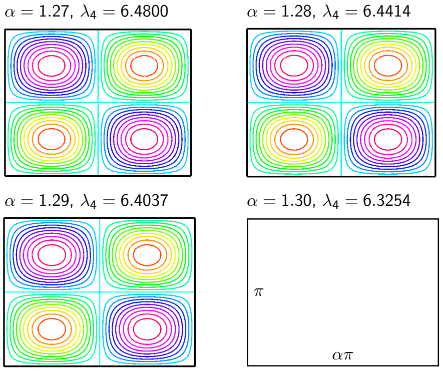
Interpolation constant based method – summary

- no a priori information needed
- optimal speed of convergence
- easy to implement
- interpolation constant known in special cases only
- adaptivity is not for free
- higher order variant is not available

5. Guaranteed bounds on eigenfunctions

[work in progress, collaboration with X. Liu]

Laplace eigenvalue problem in a rectangle


$\alpha = 1.27,\ \lambda_4 = 6.4800$

$\alpha = 1.28,\ \lambda_4 = 6.4414$

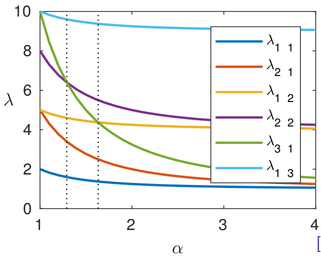
$\alpha = 1.29,\ \lambda_4 = 6.4037$

$\alpha = 1.30,\ \lambda_4 = 6.3254$

Laplace eigenvalue problem in a rectangle

$$
\begin{aligned}
-\Delta u_n &= \lambda_n u_n && \text{in } \Omega = (0,\alpha\pi)\times(0,\pi)\\
u_n &= 0 && \text{on } \partial\Omega
\end{aligned}
$$

Dependence of eigenvalues on α



$$\lambda_{k,m} = \frac{k^2}{\alpha^2} + m^2$$

$$u_{k,m} = \sin\frac{kx}{\alpha}\sin(my)$$

$$\alpha^* = \sqrt{5/3} \approx 1.2910$$

[Trefethen, Betcke 2006]

Error bounds on eigenfunctions

Problem

- Eigenfunctions may be ill-posed $\Rightarrow$ spaces of eigenfunctions
- Directed distance of spaces $\delta(E, E_h)$ [Meyer 2000]

Assume

- $\lambda_n, \lambda_{n+1}, \dots, \lambda_N$ (cluster)
- $E = \operatorname{span}\{u_n, u_{n+1}, \dots, u_N\}$ (space of eigenfunctions)
- $E_h = \operatorname{span}\{u_{h,n}, u_{h,n+1}, \dots, u_{h,N}\}$ (its approximation)
- $\ell_i \le \lambda_i \le \lambda_{h,i}$ (two sided bounds on eigenvalues)

$\Rightarrow$

Compute an upper bound on $\delta(E, E_h)$

Directed distance of spaces

Definition

Let E and E_h be two subspaces of a Hilbert space V then

$$\delta(E, E_h) = \max_{\substack{v\in E\\ \|v\|=1}} \min_{v_h \in E_h} \|v - v_h\|$$

Properties

- if $\dim E = \dim E_h$ then $\delta(E, E_h) = \delta(E_h, E)$
- $\delta^2(E, E_h) = 1 - \min_{\substack{v\in E\\ \|v\|=1}} \max_{\substack{v_h\in E_h\\ \|v_h\|=1}} |(v, v_h)|^2$

Example

Let $E = \operatorname{span}\{u\}$ and $E_h = \operatorname{span}\{u_h\}$ then

$$\delta^2(E, E_h) = 1 - \frac{|(u, u_h)|^2}{\|u\|^2\|u_h\|^2} = 1 - \cos^2\alpha = \sin^2\alpha$$

$$\|u - u_h\|^2 = \|u\|^2 + \|u_h\|^2 - 2\|u\|\|u_h\|\sqrt{1 - \delta^2(E, E_h)}$$

Lehmann-like estimate of eigenfunctions

Eigenvalue problem:

Find $\lambda_n > 0$ and $u_n \in V \setminus \{0\}$ such that

$$a(u_n, v) = \lambda_n b(u_n, v) \quad \forall v \in V.$$

Consider

- $E = \operatorname{span}\{u_n, u_{n+1}, \dots, u_N\}, \quad b(u_i, u_j) = \delta_{ij}, \quad a(u_i, u_j) = \lambda_i \delta_{ij}$
- $E_h = \operatorname{span}\{u_{h,n}, u_{h,n+1}, \dots, u_{h,N}\}$

Goal

Upper bound on $\delta(E, E_h) = \max\limits_{\substack{v \in E \\ \|v\|_a = 1}} \min\limits_{v_h \in E_h} \|v - v_h\|_a$

Lehmann–Goerisch-like estimate of eigenfunctions

Theorem

Let $\lambda_{n-1} \le \xi < \lambda_n, \quad \lambda_N < \rho \le \lambda_{N+1}, \quad \theta \ge \max\limits_{i=n,\dots,N} \left(\dfrac{\xi + \rho}{\lambda_i} - \dfrac{\xi\rho}{\lambda_i^2} \right)$

- $u_{h,n}, u_{h,n+1}, \dots, u_{h,N} \in V$ be linearly independent
- $A_{0,ij} = a(u_{h,i}, u_{h,j})$
- $A_{1,ij} = b(u_{h,i}, u_{h,j})$
- $w_i \in V: \quad a(w_i, v) = b(u_{h,i}, v) \quad \forall v \in V$
  $A_{2,ij} = a(w_i, w_j)$
- $\mu_{\min}$ be the smallest eigenvalue of $[(\xi + \rho)A_1 - \xi\rho A_2]\,\boldsymbol{x} = \mu A_0 \boldsymbol{x}$

Then

$$\delta^2(E, E_h) \le \frac{\theta - \mu_{\min}}{\theta - 1}$$

Lehmann–Goerisch-like estimate of eigenfunctions

Theorem

Let $\lambda_{n-1} \le \xi < \lambda_n, \quad \lambda_N < \rho \le \lambda_{N+1}, \quad \theta \ge \max\limits_{i=n,\dots,N} \left(\dfrac{\xi + \rho}{\lambda_i} - \dfrac{\xi\rho}{\lambda_i^2} \right)$

- $u_{h,n}, u_{h,n+1}, \dots, u_{h,N} \in V$ be linearly independent
- $A_{0,ij} = a(u_{h,i}, u_{h,j})$
- $A_{1,ij} = b(u_{h,i}, u_{h,j})$
- X ... vector space
  $\mathcal{B}$ ... positive semidefinite symmetric bilinear form on X
  $T : V \to X$ ... linear operator:
  (a) $\mathcal{B}(Tu, Tv) = a(u, v) \quad \forall u, v \in V$
  (b) $\hat{\mathbf{w}}_i \in X: \quad \mathcal{B}(\hat{\mathbf{w}}_i, Tv) = b(\tilde{u}_i, v) \quad \forall v \in V$
  (c) $\hat{A}_{2,ij} = \mathcal{B}(\hat{\mathbf{w}}_i, \hat{\mathbf{w}}_j)$
- $\hat{\mu}_{\min}$ be the smallest eigenvalue of $\left[(\xi + \rho)A_1 - \xi\rho \hat{A}_2\right] \boldsymbol{x} = \hat{\mu} A_0 \boldsymbol{x}$

Then

$$\delta^2(E, E_h) \le \frac{\theta - \hat{\mu}_{\min}}{\theta - 1}$$

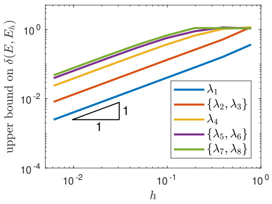
Example

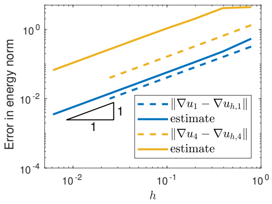
Laplace eigenvalue problem in a square

$$\begin{aligned} -\Delta u_n &= \lambda_n u_n && \text{in } \Omega = (0,\pi)^2 \\ u_n &= 0 && \text{on } \partial\Omega \end{aligned}$$

Exact eigenvalues

$\lambda_1 = 2, \quad \lambda_2 = \lambda_3 = 5, \quad \lambda_4 = 8, \quad \lambda_5 = \lambda_6 = 10, \quad \lambda_7 = \lambda_8 = 13$





Literature (very incomplete)

Books and chapters

- I. Babuška, J.E. Osborn, *Eigenvalue problems*, in: Handbook of Numerical Analysis, Vol. II, North-Holland, Amsterdam, 1991, pp. 641–787.
- D. Boffi, *Finite element approximation of eigenvalue problems*, Acta Numer. 19 (2010) 1–120.
- D. Braess, *Finite Elemente. Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*, Springer 1992, 5 editions. (*Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics.* Cambridge University Press, Cambridge, 1997, 3 editions.)
- S. Brenner, R. Scott, *The mathematical theory of finite element methods*, Springer 1994, 3 editions.
- T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1976.

Papers on conforming approaches

- G. Temple, *The theory of Rayleighs principle as applied to continuous systems*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 119 (2) (1928) 276–293.
- A. Weinstein, *Étude des Spectres des quations aux Dérivées Partielles de la Théorie des Plaques élastiques*, in: Mem. Sci. Math., vol. 88, Gauthier-Villars, Paris, 1937, p. 63.
- T. Kato, *On the upper and lower bounds of eigenvalues*, J. Phys. Soc. Japan 4 (1949) 334–339.
- N.J. Lehmann, *Beiträge zur numerischen Lösung linearer Eigenwertprobleme. I and II*, ZAMM Z. Angew. Math. Mech. 29 (1949) 341–356 and 30 (1950) 1–16.
- F. Goerisch, H. Haunhorst, *Eigenwertschranken für Eigenwertaufgaben mit partiellen Differentialgleichungen*, ZAMM Z. Angew. Math. Mech. 65 (3) (1985) 129–135.

Literature (very incomplete)

Papers on interpolation constant based method:

- C. Carstensen, J. Gedicke, *Guaranteed lower bounds for eigenvalues*, Math. Comp. 83 (290) (2014) 2605–2629.
- C. Carstensen, D. Gallistl, *Guaranteed lower eigenvalue bounds for the biharmonic equation*, Numer. Math. 126 (1) (2014) 33–51.
- X. Liu, S. Oishi, *Verified eigenvalue evaluation for the Laplacian over polygonal domains of arbitrary shape*, SIAM J. Numer. Anal. 51 (3) (2013) 1634–1654.
- X. Liu, *A framework of verified eigenvalue bounds for self-adjoint differential operators*, Appl. Math. Comput. 267 (2015) 341–355.

My contributions

- I. Šebestová, T. Vejchodský, *Two-sided bounds for eigenvalues of differential operators with applications to Friedrichs, Poincaré, trace, and similar constants*, SIAM J. Numer. Anal. 52 (2014), no. 1, 308–329.
- T. Vejchodský, *Flux reconstructions in the Lehmann–Goerisch method for lower bounds on eigenvalues*, J. Comput. Appl. Math. 340 (2018) 676–690.
- T. Vejchodský, *Three methods for two-sided bounds of eigenvalues–A comparison*, Numer. Methods Partial Differ. Equations 34 (2018) 1188–1208.

Acknowledgement

My research of eigenvalue problems is supported by
the Neuron Impuls project no. 24/2016

Neuron BENEVOLENT FUND FOR SUPPORT OF SCIENCE

Appendix: Guaranteed computations

Weinstein bound:

- λ_*, u_*, $\mathbf{q}$ can be arbitrary
- $\eta^2 = \|\nabla u_* - \mathbf{q}\|_0^2 + \frac{1}{\gamma}\|\lambda_* u_* + \operatorname{div}\mathbf{q}\|_0^2$ must be evaluated exactly $(*)$

Lehmann–Goerisch method:

- $\tilde{u}_i$, $\boldsymbol{\sigma}_i$ can be arbitrary
- $(A_0 - \rho A_1)\hat{\boldsymbol{x}} = \hat{\mu}(A_0 - 2\rho A_1 + \rho^2 \hat{A}_2)\hat{\boldsymbol{x}}$ must be solved exactly $(*)$

Interpolation constant based method:

- $\lambda_{h,i}^{\mathrm{CR}}$ must be computed exactly $(*)$

$(*)$ or bounded by interval arithmetic!

Title:	Seminar on Numerical Analysis & Winter School Proceedings of the conference SNA'19 Ostrava January 21 – 25, 2019
Editors:	Radim Blaheta, Jiří Starý, Dagmar Sysalová
Published by:	Institute of Geonics of the CAS, Ostrava
Printed by:	Tiskárna Kleinwächter, Frýdek-Místek

First edition

Ostrava, 2019

80 copies

ISBN 978-80-86407-73-9