# Programs and Algorithms of Numerical Mathematics 15

Dolní Maxov, June 6–11, 2010

## Proceedings of Seminar

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

Institute of Mathematics
Academy of Sciences of the Czech Republic
Prague 2010

ISBN 978-80-85823-57-8
Matematický ústav AV ČR, v.v.i.
Praha 2010

## Contents

# Preface

This book comprises peer-reviewed papers that originated from invited lectures and short communications presented at the 15th seminar Programs and Algorithms of Numerical Mathematics (PANM) held in Dolní Maxov, Czech Republic, June 6–11, 2010.

The seminar was organized by the Institute of Mathematics of the Academy of Sciences of the Czech Republic in cooperation with the Advanced Remedial Technologies and Processes Research Centre at the Technical University of Liberec. It continued the previous seminars on mathematical software and numerical methods held (biannually, with only one exception) in Alšovice, Bratříkov, Janov nad Nisou, Kořenov, Lázně Libverda, Dolní Maxov, and Prague in the period 1983–2008. The objective of this series of seminars is to provide a forum for presenting and discussing advanced topics in numerical analysis, new approaches to mathematical modeling, and single- or multi-processor applications of computational methods.

More than 60 participants from the field took part in the seminar, most of them from Czech universities and from institutes of the Academy of Sciences of the Czech Republic but also from the Netherlands, Slovakia, and the United States. The participation of a significant number of young scientists, PhD students, and also some undergraduate students is an established tradition of the PANM seminar; this year was no exception. We wish to believe that those, who took part in the PANM seminar for the first time, have found the atmosphere of the seminar friendly and working, and will join the PANM community.

The organizing committee consisted of Jan Chleboun, Petr Přikryl, Karel Segeth, Jakub Šístek, and Tomáš Vejchodský. Mrs Hana Bílková kindly helped in preparing manuscripts for print.

All papers have been reproduced directly from materials submitted by the authors. In addition, an attempt has been made to unify the layout of all papers.

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

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

# VECTORIZATION OF BITMAPS BASED ON THE LSQ METHOD[*]

Stanislav Bartoň

**Abstract**

The paper presents the software procedure (using MAPLE 13) intended for a considerable reduction of digital image data set to a more easily treatable extent. The photos taken in high resolution (and corresponding data sets) contain coordinates of thousands of pixels, polygons, vertexes. Presented approach substitutes this polygon by the new one, where a smaller number of vertexes is used. The task is solved by means of adapted least squares method. The presented algorithm enables the reduction of number of vertexes to 5% of its original extent with an acceptable accuracy ± one pixel (i.e. distance between the initial and the final polygon). The procedure can be used for processing of similar types of 2D images and for the acceleration of following computations.

## 1 Introduction

The acquisition and analysis of the visual information represents a powerful tool for interpretation of a large volume of input data. Recently, the origin of computer vision is intimately intertwined with computer history, having been motivated by a wide spectrum of important applications such as robotics, biology, medicine, industry and physics, and also in agricultural and food sciences. Among all different aspects underlying visual information, the shape of the objects certainly plays a special role. The multidisciplinarity of image analysis, with respect to both techniques and applications, has motivated a rich and impressive set of information resources represented e.g. in a book by Costa and Cesar[5].

This paper presents a completely different approach, where input image data are significantly reduced (to 5 % of original extent) by means of MAPLE 13 algorithm without any loss of precision. An example of this is a digital photo of carrot. Reduced data sets can be subsequently used for faster processing. The MAPLE software environment have been successfully used to determine the shape of agricultural products [1], [2], [3], [4].

## 2 Material and methods

### 2.1 Digital photo processing

A sample digital photo of carrot (bought in May 2010 in Kaufland, Jičín) has been used in this study. But any similar object of natural of artificial origin could

[*] The research has been supported by the Grant Agency of the Czech Academy of Sciences under Contract No. IAA201990701.

be used. The photo was taken by a digital camera Panasonic DMC-T27 with the resolution of 10.5 Mpixels. Points creating carrots perimeter were extracted and approximated as a polygon. This process is in detail described in earlier paper [4] and the applied Maple algorithm may be downloaded from author's web page: *www.user.mendelu.cz/barton*

### 2.2 Input data file organization

The input file contains three variables. The first one, $\mathbf{O}$ is a list of coordinates of $N$ points describing carrot perimeter, $O_i = [O_{i_1}, O_{i_2}]$, $1 \leq i \leq N$. The second one $\mathbf{P}$ is a list of $n$ vertexes of the polygon approximating carrot perimeter, $P_i = [P_{i_1}, P_{i_2}]$, $1 \leq i \leq 1$. List $\mathbf{\Lambda}$ is a list of $n$ sublists containing coordinates of perimeter points corresponding to sides of the approximating polygon, $\Lambda_i = [P_{j_1}, P_{j_2}]$, $1 \leq j \leq n_i$. For example, $\Lambda_k$ is $k^{th}$ element of the $\mathbf{\Lambda}$ and contains coordinates of the perimeter points corresponding to $k^{th}$ side of the polygon. This side is represented by a $k^{th}$ line segment with endpoints $P_k$ and $P_{k+1}$. All coordinates are in pixels.

### 2.3 Optimization

Each lateral side of the approximating polygon, hereinafter mentioned only as side, is given by the pair of their end points –$P1$ and $P2$. Square of distance of the point $O$ from the line given by points $P1$ and $P2$ can be expressed as a function:

$$S(O, P1, P2) = \frac{(P2_1 P1_2 - P2_1 O_2 - P2_2 P1_1 - P1_2 O_1 + P1_1 O_2 + P2_2 O_1)^2}{(P1_1 - P2_1)^2 + (P1_2 - P2_2)^2} . \quad (1)$$

Sum of squares of distances of points corresponding to $k^{th}$ side is:

$$q_k = \sum_{i=1}^{n_k} S(\Lambda_{i_k}, P_k, P_{k+1}) . \quad (2)$$

Finally sum of squares of all distances is:

$$Q = \sum_{k=1}^{n} q_k = \sum_{k=1}^{n} \left( \sum_{i=1}^{n_k} S(\Lambda_{i_k}, P_k, P_{k+1}) \right) , \quad (3)$$

where $\Lambda_{i_k}$ is $i^{th}$ member of the $k^{th}$ sublist of the list $\mathbf{\Lambda}$ ; in other words, it is $i^{th}$ point of the sublist $\Lambda_k$, corresponding to the $k^{th}$ side. As we can see, $Q = Q(P1, \cdots, Pn) = Q(\mathbf{P})$ is a function only of coordinates of the approximating polygon, because coordinates of perimeter points are constant.

#### 2.3.1 Global optimization – global data shaking

The approximating polygon is a closed curve; each endpoint of it belongs to two sides, $P1 = P_i$, $P2 = P_{i+1}$, for that reason it is not possible to optimize each side separately. We have to minimize $Q$ with respect to $P_{n+1} = P1$, reflecting condition of the closed curve - approximating polygon, the end point of the last side is the first point of the first side. Because $Q$ is a non-linear function of $\mathbf{P}$, it was necessary to use the iteration method. The Gauss-Newton iteration method is one of the most effective tools.

We have to find a new polygon with $h$ vertexes saved in he vector - list $\mathbf{G} = [G_i]$, $1 \leq i \leq h$ minimizing $Q = Q(\mathbf{G})$. Vertexes saved in the list $\mathbf{P}$ may be used as an initial approximation. It is not necessary to put $h = n$, because during the iteration, both endpoints of the side may be so close that it will be possible to substitute them by one point. So we may assume that at the beginning of the iteration $h = n$,, and that later on it may be that $h \leq n$.

Vector $\mathbf{P}$ can be corrected by means of list of small corrections $\mathbf{\Delta P}$ . In this case we can use:

$$Q(\mathbf{P} + \mathbf{\Delta P}) = Q(\mathbf{P}) + J_Q\, \mathbf{\Delta P}\ , \tag{4}$$

where $J_Q$ is the Jacobian matrix and minimizing of (3) is converted into a linear problem of computation of the vector $\mathbf{\Delta P}$. Now we can put $\mathbf{P} = \mathbf{P} + \mathbf{\Delta P}$ and repeat the whole process until the moment when the requested accuracy is reached. The usual condition of accuracy is $||\mathbf{\Delta P}||_2 \leq \epsilon$, where $\epsilon$ is accuracy.

However this approach is divergent, and for that reason unusable.

### 2.3.2 Local optimization – local data shaking

The main idea of this approach is to optimize only the side with the largest distance between perimeter points and corresponding polygon sides. In this case we shall move only with two consequent $P1$ and $P2$ points from the vector $\mathbf{P}$, $P1 = P_i$, $P2 = P_{i+1}$. Index $i$ corresponds to the side with the largest distance from the perimeter points.

We have to remember that we shall move with three sides. These sides have indexes $i-1$, i and $i+1$, and they are given by endpoints $P_{i-1}$, $P_i$, $P_{i+1}$ and $P_{i+2}$, but points $P_{i-1}$ and $P_{i+2}$ are stable, without computed corrections. This approach is based on the same theory as global optimization, but with a reduced volume. Because $P1 = [P1_x, P1_y]$ and $P2 = [P2_x, P2_y]$ vector $\mathbf{P}$ may be organised as $\mathbf{P} = [P1_x, P1_y, P2_x, P2_y]$; organisation of vector of corrections is equivalent.

Non-zero elements of the Jacobian matrix $J_Q$ corresponding to the first iteration step, $i = 4$, are displayed in the Fig. 1.

During the iteration the following cases may occur:

1. The simplest one is a convergence to desired accuracy. In this case there are no changes between points corresponding to sides $i-1, \cdots, i+1$.
2. If correction $\mathbf{P} = \mathbf{P} + \mathbf{\Delta P}$ is introduced, points on the perimeter may be closer to the other side. Points form $i - 1^{th}$ side may move up to $i^{th}$ side, from $i + 1^{th}$ side may move down to $i^{th}$ side. Points from $i^{th}$ side may move down as well as up. This leads to a redistribution of points between sublists $\Lambda_{i-1}$, $\Lambda_i$ and $\Lambda_{i+1}$.
3. If perimeter points are redistributed, the number of points corresponding to one side may be smaller than or equal to 3. In this case these points may

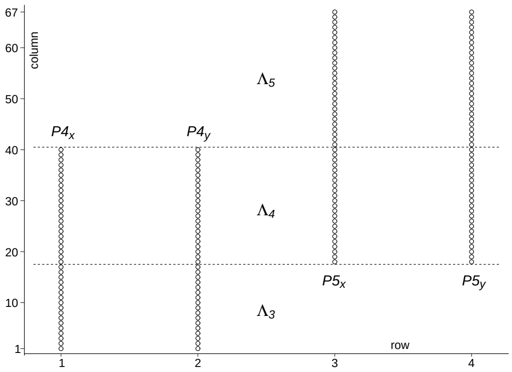


**Fig. 1:** *Non-zero elements of the $J_Q$ corresponding to the first iteration step.*

be distributed between adjacent lines. The number of approximating polygon vertexes drops down by 1, $n = n - 1$. The iteration must be restarted.

4. Lines $i - 1$, $i$ or $i$, $i + 1$ may be parallel. Line $i$ is assumed to be parallel if $S(P_i, P_{i-1}, P_{i+1}) \leq 0.25$, similar condition may be used for lines $i$ and $i + 1$. In this case these lines may be substituted by one with the endpoints $P_{i-1}$, $P_i$, or $P_i$, $P_{i+1}$, and sublists $\Lambda_{i-1}$, $\Lambda_i$ or $\Lambda_i$, $\Lambda_{i+1}$ may be collected. The number of approximating polygon vertexes drops down by 1, $n = n - 1$. The iteration must be restarted.
5. The point is jumping. In the iteration process the point jumps up and down. In this case the greatest difference of the optimized side of polygon is smaller than the second one of all points. The iteration may be finished.

If the optimization is finished, the whole process may be repeated with a new greatest distance until the moment when the same point will be after iteration again the point with the largest distance. In this case we have two variants of continuation:

**Variant 1:** To continue with the point with the second largest distance, later with third etc.

**Variant 2:** To put new polygon vertex into the point with the greatest distance and to split corresponding subvector $L_i$ into two and to restart the whole process of iteration. The number of approximating polygon vertexes rises up by 1, $n = n + 1$.

The usual maximal distance is close to 1 pixel. For that reason it is not necessary to use Variant 2 very often, because the precision of digital photo is ± 1 pixel. This means that Variant 2 is used only from time to time.

## 3 Results

The result is again a polygon with a lower number of vertexes than in initial polygon and with a better approximation of the perimeter of the object. Quality of the approximation may be evaluated in the following ways:

1. By means of the greatest displacement.
2. By means of the average displacement.
3. By means of the number of polygon vertexes.
4. By means of the coefficient of linear correlation. The linear correlation is computed for vectors of distances from the origin of perimeter points and corresponding points on the polygon vertexes.

Results are presented in Tab. 1 and Figs. 2 and 3 demonstrating the optimization of the carrot digital photograph. Carrot perimeter creates 1819 vertexes.

| **Parameter** | **Input polygon** | **Optimized polygon** |
|---|---|---|
| Greatest displacement | 1.66 | 1.39 |
| Average displacement | 0.46 ± 0.33 | 0.38 ± 0.27 |
| Vertexes | 61 | 48 |
| Correlation | 0.9999923 | 0.9999942 |
| Data reduction | 3.35% | 2.64% |

**Tab. 1:** *Results of the optimization.*

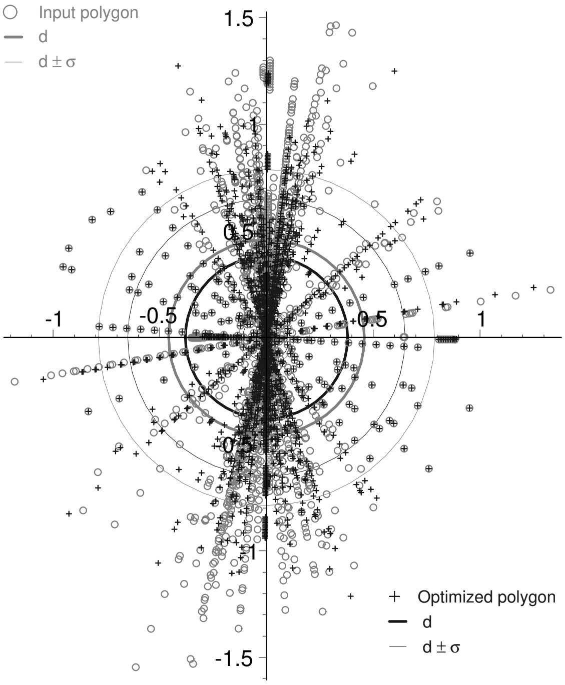


**Fig. 2:** *Vectors of displacements of the initial and optimized polygon. d = mean displacement, σ = quadratic error of the displacement.*

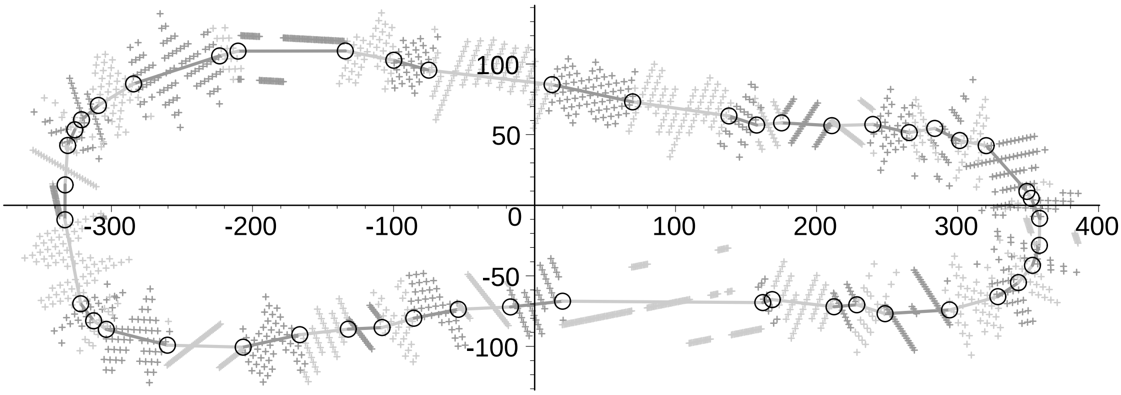


**Fig. 3:** *Visualisation of the optimized polygon. Displacements are 30× enlarged.*

## 4 Conclusions

The proposed procedure is of a general nature and can be used for data reduction for the evaluation of other biological as well as artificial shapes. It can serve as an effective and precise tool for acceleration of the process of computing and for enabling the calculation itself, when using less powerful hardware, e.g. common PC with a computer algebra program and/or in case of data processing using methods of non-linear regression.

## References

[1] Bartoň, S.: Quick algorithms for calculation of coefficients of non-linear and partially continuous functions using the Least Square Method, solution in Maple 6. In: *Proceedings of 8th International Research Conference CO-MAT-TECH 2000*, pp. 79-85. MTF Trnava, STU Bratislava, 2000, ISBN 80-227-1413-5.

[2] Bartoň, S.: Stanovení tvaru zemědělské plodiny. In: *Proceedings of 6. Matematický workshop. Brno*, pp. 1-12. FAST VUT Brno, 2007, ISBN 80-214-2741-8.

[3] Bartoň, S.: Three dimensional modelling of the peach in Maple. In: Chleboun J., (Ed)., *Programs and Algorithms of Numerical Mathematics*, pp. 7-14. 1st ed. Praha. Matematický ústav AV ČR, 2008, ISBN 978-80-85823-55-4.

[4] Bartoň, S., Severa, L., and Buchar, J.: New algorithm for biological objects' shape evaluation and data reduction. In: *Acta of Mendel University of Agriculture and Forestry Brno*, vol. 58, No. 1, pp. 13–20. MZLU Brno, 2010, ISSN 1211-8516.

[5] Costa, L.F. and Cesar, R.M.: *Shape classification and analysis theory and practice.* CRC Press, 2009, ISBN 978-0-8493-7929-1.

# HIGH RESOLUTION SCHEMES FOR OPEN CHANNEL FLOW[*]

Marek Brandner, Jiří Egermaier, Hana Kopincová

**Abstract**

One of the commonly used models for river flow modelling is based on the Saint-Venant equations – the system of hyperbolic equations with spatially varying flux function and a source term. We introduce finite volume methods that solve this type of balance laws efficiently and satisfy some important properties at the same time. The properties like consistency, stability and convergence are necessary for the mathematically correct solution. However, the schemes should be also positive semidefinite and preserve steady states to obtain physically relevant solution of the flow problems. These schemes can also be modified to a high order version or for solving flow problems with a friction source term.

## 1 Introduction

One of the most general models for simulating fluid flow is based on the Navier-Stokes equations. This model is suitable for viscous incompressible flow, but it is not directly applicable to open channel flow problems. In this case it is necessary to define some conditions on moving boundary or to use the model with the interaction between water and air layer. In our case it is convenient to use the simpler model based on the Saint-Venant equations. They are the most common choice which describes incompressible open channel flow, where vertical component of the acceleration is neglected. This model can be used for river flow or for problems of coastal areas flow.

## 2 Mathematical model

The one-dimensional Saint-Venant equations have the following form:

$$\begin{aligned} h_t + (hv)_x &= 0, \\ (hv)_t + \left(hv^2 + \frac{1}{2}gh^2\right)_x &= -ghB_x, \end{aligned} \tag{1}$$

where $h = h(x,t)$ is the unknown fluid depth, $v = v(x,t)$ is the unknown horizontal velocity, $B = B(x)$ is the elevation of the bottom surface and $g$ is the gravitational constant. The other source terms (e.g. friction term important for flood modelling) can be added into the system. In the following parts of this paper we use, for

[*] This work has been supported by the Research Plan MSM 4977751301 and by Moravian-Silesian region.

simplicity, the system in the form (1). This system can be simply written in the matrix form

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

The following schemes use the finite volume discretization with the space step $\Delta x$ such that $x_j = j\Delta x, j \in \mathbb{Z}$, and adaptive time step $\Delta t_n$ based on the CFL stability condition.

### 3 Properties of the methods

In addition to important properties like conservation, consistency and stability the numerical schemes should satisfy some other ones.

- Positive semidefiniteness – some of the unknown functions have to be nonnegative from their physical fundament. Therefore it is necessary to use such a scheme that satisfies the nonnegativity of these functions. We suppose $h \geq 0$ in our problem.

- Preserving steady states – the numerical scheme should preserve such steady states, which occur in the exact solution. The steady state means $\mathbf{u}_t = \mathbf{0}$ and therefore $[\mathbf{f}(\mathbf{u}, x)]_x = \boldsymbol{\psi}(\mathbf{u}, x)$. Then the numerical scheme should balance the flux difference and the approximation of the source terms. The presented schemes do not preserve general steady states but only the special one called "rest at lake" ($h + B =$ const., $v = 0$).

- High resolution – we can construct the scheme of the high order of accuracy. However, the high order schemes produce spurious oscillations in the regions with discontinuities in the solution. Therefore our goal is to construct such a scheme which is of high order of accuracy in the area with the smooth solution and first order accurate if there exist jumps in the solution. Moreover, this scheme should contain a small amount of artificial diffusion.

Furthermore, the big advantage of the method is its possibility to use long time steps, especially if we solve large scale problems. From this point of view we can choose between explicit and implicit methods.

Explicit methods are easier to implement and they have low cost per time step, because they need not solve any system of algebraic equations. However, the time step is bounded by the CFL stability condition. Furthermore, they are often inefficient for the solution of the stationary problems.

On the other hand, the implicit methods are unconditionally stable or stable over a wide range of the time steps. But they have high cost per time step which is caused by solving the system of algebraic equations. The linear solvers have also problems with convergence as time step increases. Implicit schemes are often insufficiently accurate for transient problems at large time step.

The main idea is to construct an adaptive semi-implicit scheme with advantages of the implicit and explicit methods.

## 4 Semi-implicit upwind method

The following scheme is based on the Roe type scheme described in [1]. The general semi-implicit finite volume scheme for balance laws in the conservative form can be written as

$$\frac{\mathbf{U}_j^{n+1}-\mathbf{U}_j^n}{\Delta t}=-\frac{1}{\Delta x}[(1-\theta)(\mathbf{F}_{j+1/2}^n-\mathbf{F}_{j-1/2}^n)+\theta(\mathbf{F}_{j+1/2}^{n+1}-\mathbf{F}_{j-1/2}^{n+1})]+(1-\theta)\mathbf{\Psi}_j^n+\theta\mathbf{\Psi}_j^{n+1}, \tag{3}$$

where $\mathbf{U}_j^n$ is the approximation of integral average of unknown function $\mathbf{u}(x,t)$ in the cell $\langle x_{j-1/2,j+1/2}\rangle$ at the time $t_n$

$$\mathbf{U}_j^n \approx \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{u}(x,t_n)dx.$$

The numerical fluxes $\mathbf{F}_{j+1/2}^n$ approximate the flux function at the the boundary of neighbouring cells $j$ and $j+1$ and $\mathbf{\Psi}_j^n$ is a suitable approximation of the source term in the cell $\langle x_{j-1/2,j+1/2}\rangle$. The finite volume methods are in detail described in [3].

The parameter $\theta$ takes values from the interval $\langle 0,1\rangle$. For $\theta=0$ the scheme is explicit, for $\theta=1$ it is implicit and for $0<\theta<1$ it is the semi-implicit scheme. The time step of the explicit scheme for hyperbolic problems is bounded by the stability CFL condition. The CFL number can be defined as

$$\text{CFL } = \frac{\Delta t}{\Delta x}\max_{p=1,2}|\lambda^p|,$$

where $\lambda^p$ are approximations of the eigenvalues of the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{u}$. It has been shown [1] that the CFL number for the Roe type semi-implicit scheme satisfies

$$\text{CFL } \leq \frac{1}{1-\theta}$$

in the scalar case. The construction of the numerical fluxes at the time level $t_n$ is based on the approximate Jacobian matrix $\mathbf{A}_{j+1/2}^n \approx \partial\mathbf{f}/\partial\mathbf{u}(x_{j+1/2},t_n)$. The numerical flux has the form:

$$\mathbf{F}_{j+1/2}^n=\frac{1}{2}[\mathbf{f}(\mathbf{U}_j^n)+\mathbf{f}(\mathbf{U}_{j+1}^n)]-\frac{1}{2}|\mathbf{A}_{j+1/2}^n|(\mathbf{U}_{j+1}^n-\mathbf{U}_j^n),$$

where

$$|\mathbf{A}_{j+1/2}^n|=\mathbf{R}_{j+1/2}^n|\mathbf{\Lambda}_{j+1/2}^n|\mathbf{L}_{j+1/2}^n\mathbf{R}_{j+1/2}^n.$$

Here, $|\mathbf{\Lambda}_{j+1/2}^n|=\text{diag}(|\lambda_{j+1/2}^{p,n}|)$, where $\lambda_{j+1/2}^{p,n}$ are eigenvalues of $\mathbf{A}_{j+1/2}^n$, and $\mathbf{R}_{j+1/2}^n$ is the matrix of the right eigenvectors of $\mathbf{A}_{j+1/2}^n$. In the case of the first order scheme the matrix $\mathbf{L}_{j+1/2}^n$ is the identity matrix $\mathbf{I}$, in the case of the flux limited scheme it has the form

$$\mathbf{L}_{j+1/2}^n=\mathbf{I}+\ \text{diag}\left(\varphi(\mathbf{u})\left(1-\min\left\{1,|\lambda_{j+1/2}^{p,n}|\frac{\Delta t}{\Delta x}\right\}\right)\right),$$

where $\varphi(\mathbf{u})$ is some limiter function based on the jumps of the unknown function $\mathbf{u}(x,t)$. It is clear that for CFL $>1$ we also obtain the first order upwind scheme.

The construction of the numerical fluxes at the time level $t_{n+1}$ is very similar. We use new values of the unknown function $\mathbf{U}^{n+1}$, but if we do not want to solve a nonlinear system of algebraic equations it is necessary to use a linearization for evaluating the flux function, i.e.

$$\mathbf{f}(\mathbf{U}_j^{n+1}) \approx \mathbf{f}(\mathbf{U}_j^n) + \mathbf{A}_{j+1/2}^n(\mathbf{U}_j^{n+1} - \mathbf{U}_j^n). \tag{4}$$

It remains to define the approximation of the source terms. To preserve the balancing property it is useful to decompose the source term integral in a similar way as the numerical fluxes:

$$\mathbf{\Psi}_j^n = \mathbf{\Psi}_{j+1/2}^{n,-} + \mathbf{\Psi}_{j-1/2}^{n,+},$$

where

$$\mathbf{\Psi}_{j+1/2}^{n,\pm} = \frac{1}{2}(\mathbf{I} \pm \mathbf{A}_{j+1/2}^{-1}|\mathbf{A}_{j+1/2}|)\mathbf{\Psi}_{j+1/2}^n.$$

Then we can construct a block tridiagonal system of the linear equations.

## 5 Semi-implicit central-upwind method

Central-upwind schemes, based on the scheme described in [2], preserve only special steady states, where the spatial derivatives of unknown functions (or their reconstructions) are equal to zero. So we define new unknown function for water level $c = h + B$ (to preserve special steady state “rest at lake”, where $hv = 0$ and $c = h + B =$ const.). Then the system of the Saint-Venant equations can be rewritten in terms of $c$ and momentum $hv$ as

$$\left(\begin{array}{c} c \\ hv \end{array}\right)_t + \left(\begin{array}{c} hv \\ (hv)^2/(c-B) + g(c-B)^2/2 \end{array}\right)_x = \left(\begin{array}{c} 0 \\ -g(c-B)B_x \end{array}\right).$$

The semidiscrete conservative scheme has the following form:

$$\frac{d}{dt}\mathbf{U}_j(t) = -\frac{\mathbf{F}_{j+1/2}(t) - \mathbf{F}_{j-1/2}(t)}{\Delta x} + \mathbf{\Psi}_j(t).$$

Numerical fluxes at the time $t_n$ are defined as (see [2])

$$\mathbf{F}_{j+1/2}^n = \frac{a_{j+1/2}^{n,+}\mathbf{f}(\mathbf{U}_{j+1/2}^{n,-}) - a_{j+1/2}^{n,-}\mathbf{f}(\mathbf{U}_{j+1/2}^{n,+})}{a_{j+1/2}^{n,+} - a_{j+1/2}^{n,-}} + \frac{a_{j+1/2}^{n,+}a_{j+1/2}^{n,-}}{a_{j+1/2}^{n,+} - a_{j+1/2}^{n,-}}\left[\mathbf{U}_{j+1/2}^{n,+} - \mathbf{U}_{j+1/2}^{n,-}\right], \tag{5}$$

where the approximations of the speeds of the local waves are defined as

$$\begin{aligned} a_{j+1/2}^{n,+} &= \max\left\{\lambda^2\left(\mathbf{f}'(\mathbf{U}_{j+1/2}^{n,-})\right), \lambda^2\left(\mathbf{f}'(\mathbf{U}_{j+1/2}^{n,+})\right), 0\right\}, \\ a_{j+1/2}^{n,-} &= \min\left\{\lambda^1\left(\mathbf{f}'(\mathbf{U}_{j+1/2}^{n,-})\right), \lambda^1\left(\mathbf{f}'(\mathbf{U}_{j+1/2}^{n,+})\right), 0\right\}, \end{aligned} \tag{6}$$

and $\mathbf{U}^{n,\pm}_{j+1/2}$ are the left and the right values of some polynomial reconstruction of the unknown function at $x_{j+1/2}$ (in this case $\mathbf{U}^{n,\pm}_{j+1/2} = [C^{n,\pm}_{j+1/2}, (HV)^{n,\pm}_{j+1/2}]^T$). There exist many available reconstructions. We use the following polynomial TVD reconstruction (the symbol $U$ represents the components of the vector $\mathbf{U}$):

$$\begin{aligned} U^{n,+}_{j+1/2} &= U^n_{j+1} - \left(1 - \min\left\{1, \lambda^{\max}_{j+1/2}\tfrac{\Delta t}{\Delta x}\right\}\right)\tfrac{\Delta x}{2}(U_x)^n_{j+1}, \\ U^{n,-}_{j+1/2} &= U^n_j + \left(1 - \min\left\{1, \lambda^{\max}_{j+1/2}\tfrac{\Delta t}{\Delta x}\right\}\right)\tfrac{\Delta x}{2}(U_x)^n_j, \end{aligned} \tag{7}$$

where $\lambda^{\max}_{j+1/2} = \max\limits_{p=1,2}|\lambda^{p,n}_{j+1/2}|$ and the symbol $(U_x)^n_j$ stands for

$$(U_x)^n_j = \begin{cases} (U_x)^n_{j,L} & \text{if } |(U_x)^n_{j,L}| \le |(U_x)^n_{j,R}| \text{ and } (U_x)^n_{j,L}\cdot(U_x)^n_{j,R} > 0, \\ (U_x)^n_{j,R} & \text{if } |(U_x)^n_{j,L}| > |(U_x)^n_{j,R}| \text{ and } (U_x)^n_{j,L}\cdot(U_x)^n_{j,R} > 0, \\ 0 & \text{if } (U_x)^n_{j,L}\cdot(U_x)^n_{j,R} \le 0, \end{cases} \tag{8}$$

where

$$(U_x)^n_{j,L} = \frac{U^n_j - U^n_{j-1}}{\Delta x}, \qquad (U_x)^n_{j,R} = \frac{U^n_{j+1} - U^n_j}{\Delta x}.$$

To preserve the special steady state “rest at lake” it is also necessary to choose approximation of the source term which is equal to the numerical flux difference. This difference can be expressed as

$$\begin{aligned} -\frac{F^{n,(2)}_{j+1/2} - F^{n,(2)}_{j-1/2}}{\Delta x} &= -\frac{1}{2\Delta x}g\left(\left(C^n_{j+1/2} - B(x_{j+1/2})\right)^2 - \left(C^n_{j-1/2} - B(x_{j-1/2})\right)^2\right) \\ &= g\frac{B(x_{j+1/2}) - B(x_{j-1/2})}{\Delta x}\cdot\frac{C^n_{j+1/2} - B(x_{j+1/2}) + C^n_{j-1/2} - B(x_{j-1/2})}{2}. \end{aligned}$$

Therefore the consistent discretization of the source terms has the form

$$\Psi^{n,(2)}_j = -g\frac{B(x_{j+1/2}) - B(x_{j-1/2})}{\Delta x}\cdot\frac{\left(C^{n,-}_{j+1/2} - B(x_{j+1/2})\right) + \left(C^{n,+}_{j-1/2} - B(x_{j-1/2})\right)}{2}. \tag{9}$$

Now we are ready to construct the semi-implicit central-upwind scheme based on the same ideas as the semi-implicit upwind scheme described before. This scheme has the form (3) and the numerical fluxes at the time level $t_{n+1}$ are defined as follows:

$$\mathbf{F}^{n+1}_{j+1/2} = \frac{a^{n,+}_{j+1/2}\mathbf{f}(\mathbf{U}^{n+1,-}_{j+1/2}) - a^{n,-}_{j+1/2}\mathbf{f}(\mathbf{U}^{n+1,+}_{j+1/2})}{a^{n,+}_{j+1/2} - a^{n,-}_{j+1/2}} + \frac{a^{n,+}_{j+1/2}a^{n,-}_{j+1/2}}{a^{n,+}_{j+1/2} - a^{n,-}_{j+1/2}}\left[\mathbf{U}^{n+1,+}_{j+1/2} - \mathbf{U}^{n+1,-}_{j+1/2}\right]. \tag{10}$$

We can see that the approximations of the maximum speeds of the local wave are the same as the approximations at the time level $t_n$. The reconstruction of the unknown functions is based on the (7) again. However, if we use (7) (especially choice of the differences (8)) for the values at the time level $t_{n+1}$ by the same way as for the values at the time level $t_n$ then (3) is the nonlinear system of algebraic equations. Therefore we define the components of $(\mathbf{U}_x)_j^{n+1}$ as

$$(U_x)_j^{n+1} = \begin{cases} (U_x)_{j,L}^{n+1} & \text{if } |(U_x)_{j,L}^n| \leq |(U_x)_{j,R}^n| \text{ and } (U_x)_{j,L}^n \cdot (U_x)_{j,R}^n > 0, \\ (U_x)_{j,R}^{n+1} & \text{if } |(U_x)_{j,L}^n| > |(U_x)_{j,R}^n| \text{ and } (U_x)_{j,L}^n \cdot (U_x)_{j,R}^n > 0, \\ 0 & \text{if } (U_x)_{j,L}^n \cdot (U_x)_{j,R}^n \leq 0, \end{cases}$$

where

$$(U_x)_{j,L}^{n+1} = \frac{U_j^{n+1} - U_{j-1}^{n+1}}{\Delta x}, \qquad (U_x)_{j,R}^{n+1} = \frac{U_{j+1}^{n+1} - U_j^{n+1}}{\Delta x}.$$

Then (3) is the linear system of algebraic equations. If we use CFL $> 1$, the reconstructed function is piecewise constant and the scheme is of the first order of accuracy.

The linearization of the flux function is provided in the same manner as in (4). The approximation of the source terms is simply defined by (9) with the reconstruction values $C^{n+1,\pm}$, i.e.

$$\Psi_j^{n+1,(2)} = -g\frac{B(x_{j+1/2}) - B(x_{j-1/2})}{\Delta x} \cdot \frac{\left(C_{j+1/2}^{n+1,-} - B(x_{j+1/2})\right) + \left(C_{j-1/2}^{n+1,+} - B(x_{j-1/2})\right)}{2}.$$

and the scheme still preserves the steady state "rest at lake".

## 6 Numerical experiment

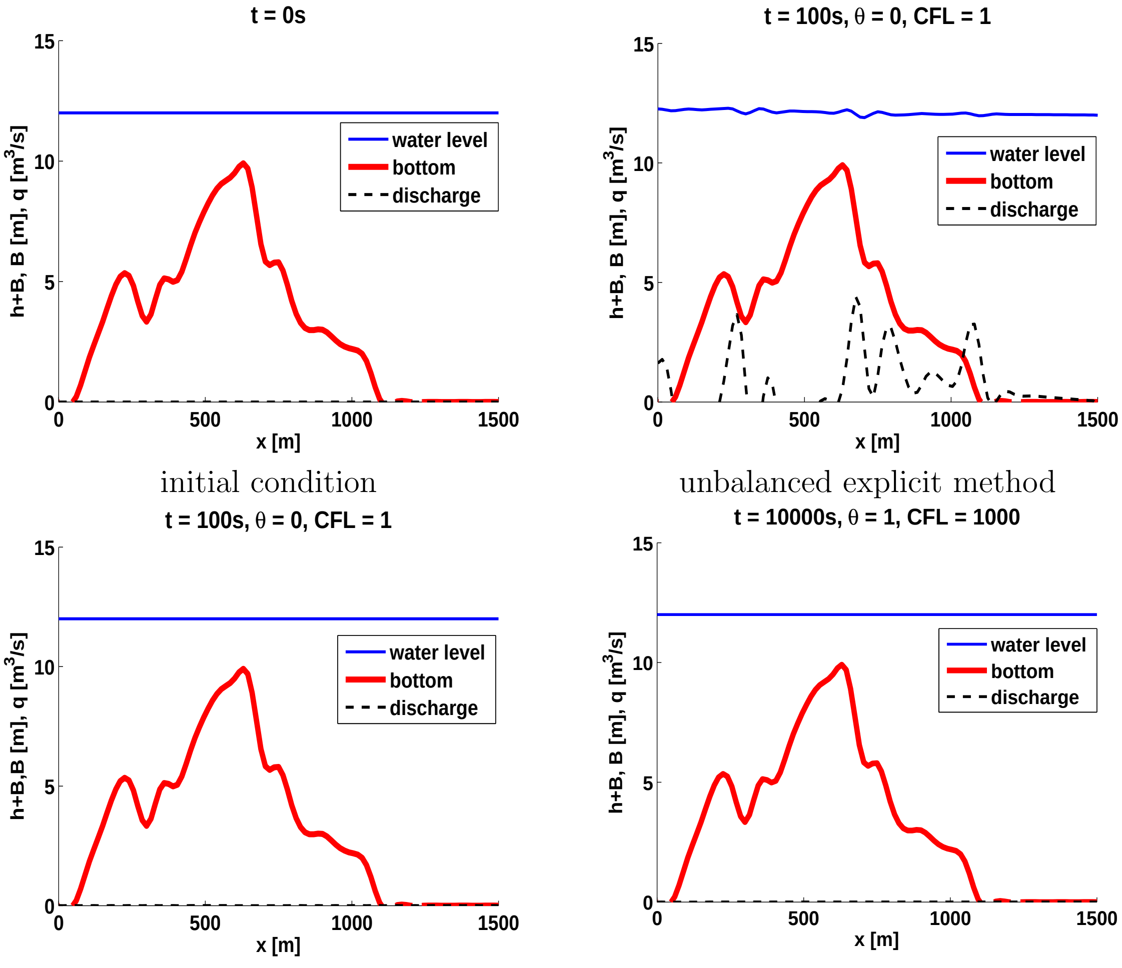
This experiment simulates the steady state "rest at lake". The described variants of the central-upwind method are used. The initial conditions (Figure 1, top left) are defined by

$$h(x,0) + B(x) = 12, \qquad v(x,0) = 0.$$

Boundary conditions are defined by zero discharge $q(0,t) = \text{const.} = 0$ and extrapolation of water level at the left boundary. The extrapolation of the discharge and water level is used on the right end of the interval. In Figure 1 we can see the comparison between the solutions computed by the balanced (bottom left) and unbalanced (top right) explicit method. In the case of balanced implicit method (bottom right) CFL $= 1000$ is used and the solution is depicted at the time $t = 10000s$.

## 7 Conclusions

We presented the high-resolution semi-implicit central upwind scheme for solving the Saint-Venant equations, which combines some of the advantages of implicit and explicit methods. As the basis for the implicit method we used the explicit method, which is positive, computationally efficient and preserves the special steady states. Since the method is nonlinear due to nonlinearity of the problem and the use of the limiter, we proposed the special linearized reconstruction of unknown functions at the time level $t_{n+1}$. The resulting semi-implicit method preserves the special steady states and it is also positive.



initial condition

unbalanced explicit method

balanced explicit method

balanced implicit method

**Fig. 1:** *Comparison of the approximate solutions for the steady state problem.*

## References

[1] Crnković, B., Črnjarić, N. and Kranjčević, L.: Improvements of semi-implicit schemes for hyperbolic balance laws applied on open channel flow equations. Comput. Math. Appl. **58** (2009), 292–309.

[2] Kurganov, A. and Levy, D.: Central-upwind schemes for the Saint-Venant system. M2AN Math. Model. Numer. Anal. **36** (2002), 397–425.

[3] LeVeque, R.J.: *Finite volume methods for hyperbolic problems.* Cambridge Texts in Applied Mathematics (No. 31), Cambridge University Press, Cambridge, 2004.

# INSTABILITY OF MIXED FINITE ELEMENTS FOR RICHARDS' EQUATION*

Jan Březina

**Abstract**

Richards' equation is a widely used model of partially saturated flow in a porous medium. In order to obtain conservative velocity field several authors proposed to use mixed or mixed-hybrid schemes to solve the equation. In this paper, we shall analyze the mixed scheme on 1D domain and we show that it violates the discrete maximum principle which leads to catastrophic oscillations in the solution.

## 1 Introduction

A standard model for the water flow in a partially saturated porous medium is Richards' equation which can by written as the system:

$$\partial_t \theta(h) + \operatorname{div}(\boldsymbol{u}) = f \qquad \text{in } (0,T) \times \Omega, \tag{1}$$

$$\boldsymbol{u} = -k(h)\nabla(h+z) \qquad \text{in } (0,T) \times \Omega. \tag{2}$$

The unknowns are the pressure head $h$ and the water velocity $\boldsymbol{u}$ while the other involved quantities are the density of volume water sources $f$, the $z$-coordinate, assumed to be in opposite direction to the gravity force, the water content $\theta$ and the hydraulic conductivity $k$, where $\theta$ and $k$ are given nonlinear function of $h$. Both equations are considered on the domain $\Omega \subset \mathbf{R}^N$ and during the time interval $(0,T)$. Through this work we consider the Dirichlet boundary condition $h_D$ on $\Gamma_D \subset \partial\Omega$, the homogeneous Neumann condition $\boldsymbol{u} = 0$ on the remaining part of the boundary, and the initial condition $h_0$ for the pressure head.

The characteristic functions $\theta(h)$ and $K(h)$ are empirical. We assume the most common Mualem – van Genuchten model [6], [5]:

$$\theta(h) = \theta_r + (\theta_s - \theta_r)\tilde{\theta}(h), \tag{3}$$

$$\tilde{\theta}(h) = (1 + (\alpha h)^n)^{-m}, \quad m = 1 - 1/n \tag{4}$$

$$k(h) = k_s \tilde{\theta}^{0.5} \left(1 - (1 - \tilde{\theta}^{1/m})^m\right)^2, \tag{5}$$

where $\theta_r$, $\theta_s$, $n$, $\alpha$, and $k_s$ are suitable soil parameters.

*This work was realized under the state subsidy of the Czech Republic within the research and development project "Advanced Remediation Technologies and Processes Center" 1M0554 – Programme of Research Centers PP2-D01 supported by Ministry of Education.

System $(1-2)$ represents a quasilinear degenerated parabolic-elliptic equation. The existence and uniqueness of the solution as well as some regularity properties were proved by Alt, and Luckhaus [1]. When solving Richards' equation numerically, we want to obtain a discrete velocity field which satisfies a discrete version of the continuity equation (1) up to the given tolerance of the nonlinear solver. This is important for a subsequent simulation of the water transport. That is why mixed or mixed-hybrid finite elements are used by many authors, e.g. [4], [3].

Motivated by these works, we want to develop a simulator that can solve coupled Richards' equations on domains of different dimension. Since the solution of Richards' equation evolves substantially only around a small wetting front region, adaptivity is crucial to achieve reasonable performance. To meet these two requirements, we have decided to try C++ finite element library DEAL II [2]. The library allows to produce a dimension independent code with $h$, $p$, and $hp$ versions of adaptivity and provides a rich palette of finite elements. The only but fundamental restriction of the library is that elements have to be topologically equivalent to hypercubes. However, during tests of our code we have observed serious oscillations of the solution. Aim of this paper is to present these observations and give an explanation of this behavior.

The paper is organized as follows. First, the mixed discretization is described. Then, in Section 3, we make its comparison with a primary discretization and we demonstrate the presence of instabilities. In the last section, we derive a condition under which the mixed scheme obeys a discrete maximum principle in 1D and we discuss some similar results.

## 2 Mixed finite elements

In order to derive mixed formulation of the system $(1-2)$, we multiply the first equation by a scalar test function $\varphi$, while in the second equation we divide by $k$, test by a vector valued function $\boldsymbol{\psi}$ and integrate by parts in the pressure term. Finally, we are looking for a solution $h \in L^2(\Omega)$, $\boldsymbol{u} \in H(div, \Omega)$ which satisfies

$$\int_\Omega k^{-1}(h)(\boldsymbol{u}\cdot\boldsymbol{\psi}) - \int_\Omega h \operatorname{div}\boldsymbol{\psi} = \int_\Omega z \operatorname{div}\boldsymbol{\psi} - \int_{\partial\Omega} (h_D + z)\boldsymbol{\psi}\cdot\boldsymbol{n}, \tag{6}$$

$$-\int_\Omega \partial_t\theta(h)\varphi - \int_\Omega \varphi \operatorname{div}\boldsymbol{u} = -\int_\Omega f\varphi \tag{7}$$

for all $\boldsymbol{\psi} \in H(div, \Omega)$ and $\varphi \in L^2(\Omega)$, where $H(div, \Omega)$ is a space of vector valued $L^2$-function with divergence in $L^2(\Omega)$.

Next, we consider a decomposition $\mathcal{T} = \{K_i\}$ of the domain $\Omega \subset \mathbf{R}^N$ into lines $(N = 1)$, quadrilaterals $(N = 2)$ or hexahedrons $(N = 3)$. On this computational grid we use Raviart-Thomas finite elements $RT_d$ with order $d$ for discretization of the velocity and discontinuous polynomial finite elements $P_d$ of order $d$ for discretization of the pressure head. More specifically, we consider discrete solution in a form

$$\boldsymbol{u}(t,\boldsymbol{x}) = \sum_i \tilde{u}_i(t)\boldsymbol{\psi}_i(\boldsymbol{x}), \qquad h(t,\boldsymbol{x}) = \sum_i \tilde{h}_i(t)\varphi_i(\boldsymbol{x}), \tag{8}$$

where $\tilde{u}$ and $\tilde{h}$ are unknown coefficient vectors. The backward Euler is used for temporal discretization. A fully implicit scheme is necessary to avoid oscillations on the saturated part of the domain where the equation becomes elliptic. Finally, we obtain a nonlinear system of equations which we solve by simple Picard iterations. Resulting linear system for the solution $\tilde{h}^k$, $\tilde{u}^k$ in iteration $k$ of time $t_n$ reads

$$A(h^{k-1})\tilde{u}^k + B\tilde{h}^k = F \tag{9}$$

$$B^T\tilde{u}^k + D(h^{k-1})\tilde{h}^k = G(h^{k-1}) \tag{10}$$

with

$$\begin{aligned}
A_{i,j}(h^{k-1}) &= \sum_{K\in\mathcal{T}} \int_K k^{-1}(h^{k-1})(\boldsymbol{\psi}_i \cdot \boldsymbol{\psi}_j),\\
B_{i,j} &= -\sum_{K\in\mathcal{T}} \int_K \varphi_i \operatorname{div} \boldsymbol{\psi}_j\,,\\
D_{i,j}(h^{k-1}) &= \sum_{K\in\mathcal{T}} \int_K -\frac{\theta'(h^{k-1})}{dt}\varphi_i\varphi_j\,,\\
F_i &= \sum_{K\in\mathcal{T}} \int_K z\operatorname{div} \boldsymbol{\psi}_i - \int_{K\cap\Gamma_D} (z+h_D)\boldsymbol{\psi}_i \cdot \boldsymbol{n},\\
G_i(h^{k-1}) &= \sum_{K\in\mathcal{T}} \int_K -\frac{\theta'(h^{k-1})h^{k-1}}{dt}\varphi_i + \frac{\theta(h^{k-1})-\theta^0}{dt}\varphi_i\,,
\end{aligned}$$

where $h^{k-1}$ is the actual discrete pressure head field according to (8) and $\theta^0$ is the water content field from the previous time $t_{n-1}$. Before solving system (9) – (10), we use the last pressure head $\tilde{h}^{k-1}$ to resolve equation (9) and compute a residuum $r^{k-1}$ of the equation (10). Iterations are stopped, when $l^2$-norm of the residuum drops under the prescribed tolerance. Then the residuum is subtracted from the actual water content which forms $\theta^0$ for the next time step. This way we achieve a perfect conservation of the total water content over the whole domain.

## 3 Comparison of mixed and primary discretization

The described mixed finite element approximation with the lowest element order $d = 0$ (MFE) have been compared with a mature one dimensional solver based on the primary linear finite element (FE) approximation of the pressure. The latter solver was thoroughly tested against experimental data in cooperation with Vogel et al. [7].

The setting of the one dimensional infiltration test problem was as follows: a vertical domain $(-5, 0)$ $[m]$, the constant initial pressure head $h_0 = -150$ $[m]$, the Dirichlet

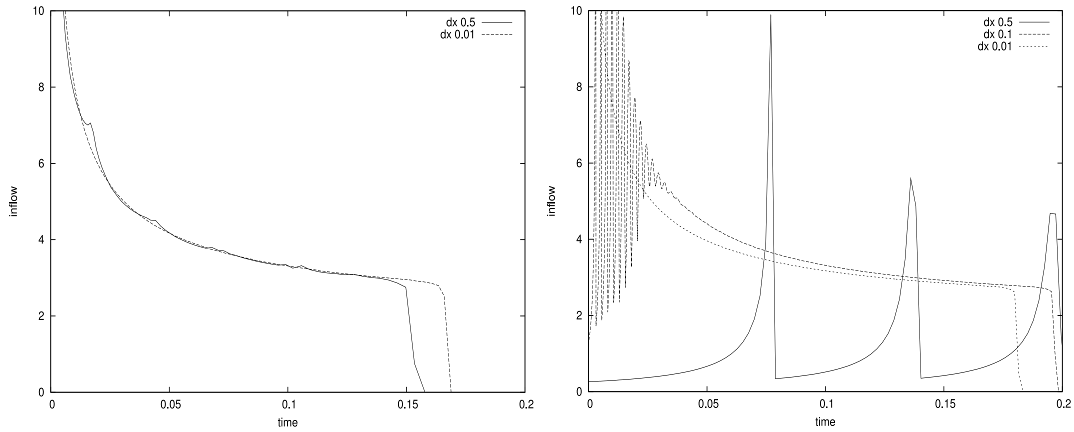


**Fig. 1:** *Infiltration velocity on the top of the vertical 1D domain. The stable FE scheme (left) and the unstable MFE scheme (right).*

boundary condition $h_D = 1\,[m]$ on the top and the homogeneous Neumann condition on the bottom. The parameters of the soil model were $n = 1.14$, $\alpha = 0.1\,[m^{-1}]$, $\theta_r = 0.01$, $\theta_s = 0.480$, $k_s = 2\,[mh^{-1}]$. This setting leads to a steep wetting front during the initial phase, thus we have to use short time steps. The wetting front goes from the top to the bottom so that the pressure head should be monotonous in time and space, increasing from $-150$ up to $1 + z$. The velocity should be always negative. The MFE code was run on meshes with steps 0.01, 0.1, and 0.5 the FE code was run only for steps 0.01 and 0.5. All simulations were started with the time step $10^{-6}$ and the time step is enlarged if the number of nonlinear iterations drops under 3.

Figure 1 shows the infiltration velocity on the top of the domain up to the full saturation of the domain. For the fine mesh step 0.01 the results are comparable. The infiltration computed by the MFE code takes just little bit longer compared to the FE code. On the other hand, for the coarser meshes, the MFE code produces terrible oscillations while the FE code still provides satisfactory results. The oscillations are not only in time but also in space and they get worse with shorter time steps or larger mesh steps. Values of the pressure head leave the valid interval $[-150, 1]$ and positive values of the velocity appear.

## 4 Discrete maximum principle

Maximum principle for elliptic PDEs states that a solution of the equation

$$\operatorname{div}(-\tilde{k}\nabla h) + \tilde{c}h = \tilde{f} \quad \text{on } \Omega, \quad h = \tilde{g} \quad \text{on } \partial\Omega, \tag{11}$$

with $\tilde{k} > 0$, $\tilde{c} \geq 0$, is non-negative provided $\tilde{f}$ and $\tilde{g}$ are non-negative. If a similar property holds for a discrete problem, we say that it obeys the discrete maximum principle (DMP).

In view of the previous section it seems that the MFE scheme violates DMP for the short time steps. To show this, we shall analyze one linear step, i.e. system (9)–(10), which can be viewed as the discretization of the linear elliptic problem (11) with $\tilde{k} = k(h)$, $\tilde{c} = \theta'(h)/dt$, and suitable positive $\tilde{f}$. We consider one dimensional domain with grid points $x_1 < x_2 < \cdots < x_n$ and the lowest order elements $d = 0$. Further, we use equivalent mixed-hybrid discretization of (11). On every element $K_i = (x_i, x_{i+1})$ the discrete solution is represented by the pressure head $h_i$ in the center of the element, by the traces $\mathring{h}_i^{1,2}$ on the element boundary, and by the velocity $\boldsymbol{u}_i = u_i^1\psi^1 + u_i^2\psi_2$. The velocity is linear combination of discontinuous $RT_0$ base functions

$$\psi_i^1(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}, \quad \psi_i^2(x) = \frac{x - x_i}{x_{i+1} - x_i}$$

where coefficients $u_i^{1,2}$ are the outer normal fluxes from the element $i$. Proceeding similarly as in the case of mixed formulation we obtain a discrete version of (11):

$$\sum_{j=1,2} \tilde{k}_i^{-1} u_i^j \int_{K_i} \psi_i^m \psi_i^j = h_i - \mathring{h}_i^m \quad \text{for } m = 1, 2 \tag{12}$$

$$\tilde{c}_i h_i |K_i| + u_i^1 + u_i^2 = \tilde{f}_i |K_i| \tag{13}$$

$$u_i^2 = -u_{i+1}^1, \quad \mathring{h}_i^2 = \mathring{h}_{i+1}^1. \tag{14}$$

We denote $\mathring{h}_i = \mathring{h}_i^2 = \mathring{h}_{i+1}^1$. The integral in (12) evaluates to $|K_i|/3$ and $-|K_i|/6$ for $m = j$ and $m \neq j$, respectively. On the Dirichlet boundary $x_n$ we set $\mathring{h}_n^1 = h_D$. Then, eliminating $h_i$ and $u_i^{1,2}$ from the system, we obtain an equation for $\mathring{h}_i$:

$$a_{i-1}\mathring{h}_{i-1} + (b_{i-1} + b_i)\mathring{h}_i + a_i \mathring{h}_{i+1} = c_{i-1} + c_i \tag{15}$$

where

$$a_i = \frac{2\tilde{k}_i}{|K_i|} - \frac{\alpha_i \alpha_i}{\beta_i}, \quad b_i = \frac{4\tilde{k}_i}{|K_i|} - \frac{\alpha_i \alpha_i}{\beta_i}, \quad c_i = \frac{\alpha_i |K_i| \tilde{f}_i}{\beta_i}, \tag{16}$$

$$\alpha_i = \frac{6\tilde{k}_i}{|K_i|}, \quad \beta_i = |K_i|\tilde{c}_i + 2\alpha_i. \tag{17}$$

Equation (15) is one row of a linear system $A\mathring{h} = c$, where vector $c$ is non-negative provided $\tilde{f}_i$ and $h_D$ are non-negative. In order to obtain a non-negative solution $\mathring{h}$, the matrix $A$ has to have positive inverse. This holds if $A$ is so called $M$-matrix, that is a matrix with positive diagonal entries, non-positive off diagonal entries, and positive row sums. In our case this is equivalent to $a_i \leq 0$, $b_i > 0$, and $a_i + b_i > 0$. The later two inequalities are always true, while the first one holds only if

$$\frac{|K_i|^2}{6} \leq \frac{\tilde{k}_i}{\tilde{c}_i} = dt \frac{k(h_i)}{\theta'(h_i)}. \tag{18}$$

For positive $\tilde{f}$ and $\tilde{g}$, this condition implies positive nodal pressures $\mathring{h}_i$. Then the elemental pressures $h_i$ are also positive since

$$h_i = \frac{|K_i|\tilde{f}_i + \alpha_i(\mathring{h}_i^1 + \mathring{h}_i^2)}{\beta_i}.$$

Further numerical experiments reveal that oscillations of the solution appear exactly on that elements where the condition (18) does not hold. Thus to get stable scheme one has to adapt the element size $|K_i|$ according to the condition. However, the right hand side tends to zero as $h_i \to -\infty$, at least for the soil model (3)–(5). It means we should use small mesh step on the dry region which is highly ineffective since the solution is mainly constant there. Situation is even worse for mixed elements on 2D quadrilaterals or 3D hexahedrons since they never lead to $M$-matrix even for $\tilde{c}_i = 0$.

In the paper due to Younes, Ackerer, and Lehmann [8] authors prove stability conditions similar to (18) for mixed-hybrid elements on triangular and tetrahedral meshes. We can conclude that the mixed scheme for the Richards' equation is stable only for large time steps and therefore is not suitable for a robust solver. However, one can try to modify the mixed scheme to make it more stable. In fact two such modifications were already proposed in [8].

## References

[1] Alt, H.W. and Luckhaus, S.: Quasilinear elliptic-parabolic differential equations. Mathematische Zeitschrift **183** (1983), 311–341.

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

[3] Bause, M. and Knabner, P.: Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods. Advances in Water Resources **27** (2004), 565–581.

[4] Bergamaschi, L. and Putti, M.: Mixed finite elements and Newton-Type linearizations for the solution of Richards' equation. International Journal for Numerical Methods in Engineering **45** (1999), 1025–1046.

[5] van Genuchten, M.T.: A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Science Society of America Journal **44** (1980), 892–898.

[6] Mualem, Y.: A new model for predicting the hydraulic conductivity of unsaturated porous media. Water Resources Research **12** (1976), 513–522.

[7] Vogel, T., Brezina, J., Dohnal, M., and Dusek, J.: Physical and numerical coupling in Dual-Continuum modeling of preferential flow. Vadose Zone Journal **9** (2010), 260–267.

[8] Younes, A., Ackerer, P., and Lehmann, F.: A new mass lumping scheme for the mixed hybrid finite element method. Int. J. Numer. Meth. Engng **67** (2006), 89–107.

# SOME REMARKS ON AVERAGING IN THE BDDC METHOD[*]

Marta Čertíková, Pavel Burda, Jaroslav Novotný, Jakub Šístek

## 1 Introduction

The Balancing Domain Decomposition based on Constraints (BDDC) method introduced in [1] is one of the latest domain decomposition methods. It can be understood as an improvement of the primal Neumann-Neumann domain decomposition method. As it has been recently shown in [3], a primal preconditioner of such type is determined by the choice of two operators: the injection $R$ and the averaging $E$. These two operators appear also in the estimate of the condition number of the preconditioned operator (see (4) bellow).

The choice of the operator $R$ can be formulated as the choice of continuity conditions across the interface (coarse unknowns). A lot of work has been invested into research of relations between the choice of coarse unknowns and the quality of preconditioning, and significant results were obtained (e.g. in [2, 3]).

On the other hand, the averaging operator $E$ seems to be aside from the main effort of the investigation so far. Standard choices of $E$ found already in [1] are arithmetic average and average weighted by diagonal entries of matrices of local problems.

In this paper, we introduce a general framework for derivation of the averaging operator, from which the standard choices are recovered by simplifications. Then, an alternative approach derived by another simplification is proposed and tested on a 2D example.

## 2 Reduction of the problem to the interface

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

$$\mathbf{Ku} = \mathbf{f}, \tag{1}$$

where $\mathbf{K}$ is large, sparse, symmetric positive definite (SPD) matrix and $\mathbf{f}$ is the vector of the right-hand side.

[*] This work was supported by Czech Academy of Sciences through AV0Z10190503, by grant No. 106/08/0403 of the Czech Science Foundation and by the project of the Research Centre 1M0554. We also acknowledge fruitful conversations with Jan Mandel, University of Colorado Denver.

Let us decompose the domain $\Omega$ into $N$ non-overlapping subdomains $\Omega_i$, $i = 1, \dots N$. Unknowns common to at least two subdomains form the *global interface* denoted as $\Gamma$. Remaining unknowns are classified as belonging to subdomain *interiors*. The global interface $\Gamma$ can be expressed as union of *local interfaces* $\Gamma_i$, $i = 1, \dots N$, containing interface unknowns involved just in subdomain $\Omega_i$.

The first step used in many domain decomposition methods including BDDC is the reduction of the problem to the interface. Without loss of generality, suppose that unknowns are ordered so that interior unknowns form the first part and the interface unknowns form the second part of the solution vector, i.e. $\mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathrm{o}} & \widehat{\mathbf{u}} \end{bmatrix}^T$, where $\mathbf{u}_{\mathrm{o}}$ stands for all interior unknowns and $\widehat{\mathbf{u}}$ for unknowns at interface. Now, system (1) can be formally rewritten to block form

$$\begin{bmatrix} \mathbf{K}_{\mathrm{oo}} & \mathbf{K}_{\mathrm{or}} \\ \mathbf{K}_{\mathrm{ro}} & \mathbf{K}_{\mathrm{rr}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathrm{o}} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{o}} \\ \widehat{\mathbf{f}} \end{bmatrix}. \tag{2}$$

The hat symbol ($\widehat{\ }$) is used to denote global interface quantities. If we suppose the interior unknowns ordered subdomain after subdomain, then the submatrix $\mathbf{K}_{\mathrm{oo}}$ is block diagonal with each diagonal block corresponding to one subdomain.

After eliminating all the interior unknowns from (2), we arrive at *Schur complement problem* for the interface unknowns

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

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

## 3 Primal DD methods and BDDC

The main idea of the primal DD substructuring methods of Neumann-Neumann type can be expressed as splitting the given residual of PCG method to subdomains, solving subdomain problems and projecting the result back to the global domain. The primal preconditioner can be written as $M = ES^{-1}E^T$, where operator $E^T$ represents splitting of the residual to subdomains, $S^{-1}$ stands for solution of subdomain problems, and $E$ represents projection of subdomain solutions back to the global problem by some averaging [3]. The condition number $\kappa$ of the preconditioned operator $M\widehat{S}$ is bounded by

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

where operator $R$ splits the global interface into subdomains and the energetic norm on the right-hand side is defined by the scalar product as $||u||_S^2 = \langle Su, u\rangle$. The relationship (4) was proved in [3] assuming that $ER = I$, which means that if the problem is split into subdomains and then projected back to the whole domain, the original problem is obtained.

If we use independent subdomain problems only (no continuity conditions across the interface), the operator $S$ is expressed by a block diagonal matrix $\mathbf{S}$ with diagonal blocks $\mathbf{S}_i$ representing local Schur complements on subdomains. Relationship between global and local problems can be expressed in matrix form as

$$\widehat{\mathbf{S}} \;=\; \mathbf{R}^{\mathrm{T}}\mathbf{S}\mathbf{R} \;=\; \sum_i \mathbf{R}^{i^{\mathrm{T}}}\mathbf{S}^i\mathbf{R}^i \,, \quad \mathbf{u} = \mathbf{R}\widehat{\mathbf{u}}\,, \quad \widehat{\mathbf{u}} = \mathbf{E}\mathbf{u}, \tag{5}$$

where $\mathbf{R}^i$ represents prolongation operator from local (subdomain) interface $\Gamma_i$ to the global interface $\Gamma$ and $\mathbf{E}$ performs some averaging.

The main idea of the BDDC ([1]) is to introduce a global *coarse problem* in order to achieve better preconditioning and to fix ‘floating subdomains’ by making their local Schur complements invertible. The matrix $\mathbf{S}$ is then positive definite, but it is not block diagonal any more, $R$ now represents splitting of the global interface into subdomains except the coarse unknowns, and $E^T$ distributes residual among neighbouring subdomains only in those interface unknowns which are not coarse. Thus in BDDC, only part of the global residual is split into subdomains; residual at the coarse unknowns is left undivided – it is processed by the global coarse problem.

## 4 Choice of the averaging operator E

We start by algebraic analysis of an elliptic problem on a domain divided into two subdomains, assuming coarse unknowns to be values at nodes only, and then generalize the results. For an illustration of this simple case see Figure 1.

### 4.1 Projection RE and its complement in matrix representation

Let us assume that on the interface there are $m$ coarse nodes and $n$ nodes which are not coarse. Suppose that nodes are ordered so that nodes that are not coarse are numbered subdomain by subdomain and the coarse nodes are the last. Then, in the simple case of two subdomains, the vectors $\widehat{\mathbf{u}}$ and $\mathbf{u}$ of values at the interface nodes and the matrices $\widehat{\mathbf{S}}$, $\mathbf{S}$ and $\mathbf{R}$ will have the following structure:

$$\widehat{\mathbf{S}} = \begin{bmatrix} \widehat{\mathbf{S}}_{\mathrm{rr}} & \widehat{\mathbf{S}}_{\mathrm{rc}} \\ \widehat{\mathbf{S}}_{\mathrm{cr}} & \widehat{\mathbf{S}}_{\mathrm{cc}} \end{bmatrix}, \; \mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathrm{r}}^1 \\ \mathbf{u}_{\mathrm{r}}^2 \\ \mathbf{u}_{\mathrm{c}} \end{bmatrix}, \; \mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mathrm{rr}}^1 & \mathbf{0} & \mathbf{S}_{\mathrm{rc}}^1 \\ \mathbf{0} & \mathbf{S}_{\mathrm{rr}}^2 & \mathbf{S}_{\mathrm{rc}}^2 \\ \mathbf{S}_{\mathrm{cr}}^1 & \mathbf{S}_{\mathrm{cr}}^2 & \widehat{\mathbf{S}}_{\mathrm{cc}} \end{bmatrix}, \; \mathbf{R} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}, \tag{6}$$

where $\mathbf{u}_{\mathrm{c}}$ represents coarse unknowns, $\mathbf{u}_{\mathrm{r}}^i$ local interface unknowns that are not coarse, $\mathbf{I}_k$ is the identity matrix of dimension $k$ and $\mathbf{S}_{\mathrm{rr}}^i$ is symmetric positive definite matrix of dimension $n$. Matrix $\mathbf{S}_{\mathrm{rr}}^i$ represents local Schur complement for $i$-th subdomain problem with zero values prescribed at coarse nodes. In the case of two subdomains, from (5) and (6) we have $\mathbf{S}_{\mathrm{rr}}^1 + \mathbf{S}_{\mathrm{rr}}^2 = \widehat{\mathbf{S}}_{\mathrm{rr}}$. From $\mathbf{E}\mathbf{R} = \mathbf{I}_{n+m}$ it follows

$$\mathbf{E} = \begin{bmatrix} \mathbf{A} & \mathbf{I}_n - \mathbf{A} & \mathbf{0} \\ \mathbf{C} & -\mathbf{C} & \mathbf{I}_m \end{bmatrix}, \tag{7}$$

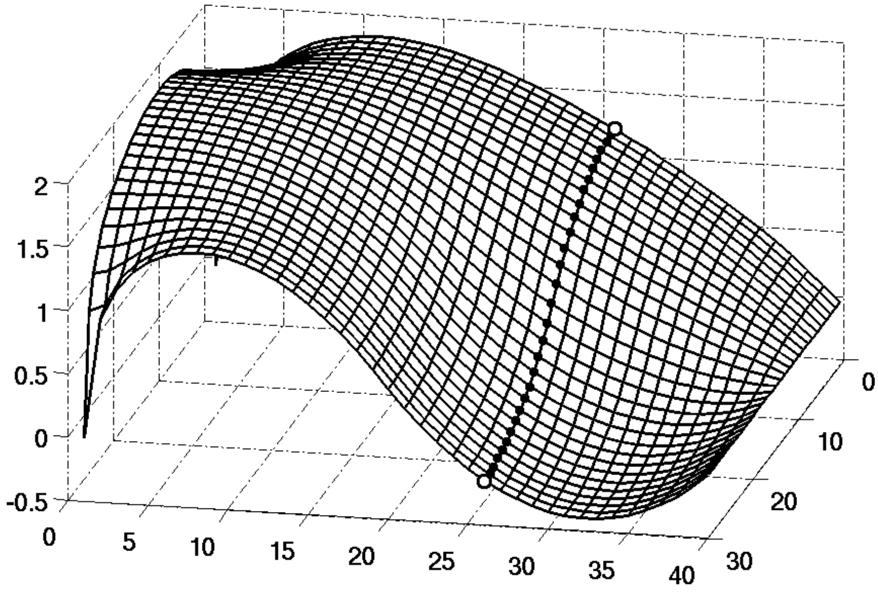


**Fig. 1:** *Test problem. 2D Poisson equation on a rectangular domain divided into two rectangular subdomains – left and right ones. Values of the solution at the interface nodes are marked by dots. The two coarse nodes are chosen on the opposite sides of the interface and are marked by circles.*

where $\mathbf{A}$ can be any weighting matrix for nodes that are not coarse and $\mathbf{C}$ is any matrix. Now we have the following decomposition of unity:

$$\mathbf{I} = \mathbf{RE} + (\mathbf{I} - \mathbf{RE}) = \begin{bmatrix} \mathbf{A} & \mathbf{I}_n - \mathbf{A} & \mathbf{0} \\ \mathbf{A} & \mathbf{I}_n - \mathbf{A} & \mathbf{0} \\ \mathbf{C} & -\mathbf{C} & \mathbf{I}_m \end{bmatrix} + \begin{bmatrix} \mathbf{I}_n - \mathbf{A} & \mathbf{A} - \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A} & \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{C} & \mathbf{0} \end{bmatrix} \tag{8}$$

(for brevity we write $\mathbf{I}$ instead of $\mathbf{I}_{2n+m}$). The projection $\mathbf{RE}$ can be viewed as some weighted average of values from adjacent subdomains at the interface nodes and the complementary projection $\mathbf{I} - \mathbf{RE}$ (which has the same energetic norm and is used in FETI-DP) as a weighted jump in these values. Its action on a given vector $\mathbf{u}$ of values at interface nodes can be expressed as

$$(\mathbf{I} - \mathbf{RE})\,\mathbf{u} = \begin{bmatrix} \mathbf{I}_n - \mathbf{A} & \mathbf{A} - \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A} & \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_\mathrm{r}^1 \\ \mathbf{u}_\mathrm{r}^2 \\ \mathbf{u}_\mathrm{c} \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{I}_n)\,\mathbf{d} \\ \mathbf{A}\,\mathbf{d} \\ \mathbf{C}\,\mathbf{d} \end{bmatrix}, \tag{9}$$

where $\mathbf{d} = \mathbf{u}_\mathrm{r}^2 - \mathbf{u}_\mathrm{r}^1$ is the jump in values at interface nodes that are not coarse.

For simplicity it is usually assumed that $\mathbf{C} = \mathbf{0}$ and $\mathbf{A}$ is diagonal. In what follows, we will try to achieve optimality only within this restricted class of choice of $\mathbf{E}$.

### 4.2 Approximate minimization of the energy norm of the projection

Our approach is to start with some fixed $\mathbf{u}$ with the interface jump $\mathbf{d}$ and try to find $\mathbf{E}$ so that it minimizes energetic difference between $\mathbf{u}$ and $\widehat{\mathbf{u}} = \mathbf{Eu}$. In other words, we are trying to minimize the energy norm of the projection $(\mathbf{I} - \mathbf{RE})\mathbf{u}$ of the given vector $\mathbf{u}$. The square of the energy norm can be expressed as

$$||(\mathbf{I} - \mathbf{RE})\mathbf{u}||_S^2 = \mathbf{u}^T(\mathbf{I} - \mathbf{RE})^T\mathbf{S}(\mathbf{I} - \mathbf{RE})\mathbf{u} = \mathbf{d}^T(\mathbf{A}^T\,\widehat{\mathbf{S}}_{\mathrm{rr}}\mathbf{A} - \mathbf{A}^T\mathbf{S}^1_{\mathrm{rr}} - \mathbf{S}^1_{\mathrm{rr}}\mathbf{A} + \mathbf{S}^1_{\mathrm{rr}}\,)\,\mathbf{d}\,.$$

Here we use the fact that $\widehat{\mathbf{S}}_{\mathrm{rr}} = \mathbf{S}^1_{\mathrm{rr}} + \mathbf{S}^2_{\mathrm{rr}}$ in the case of two subdomains.

A considerable effort is invested into minimization of this norm using the definition of $\mathbf{R}$ in the *adaptive BDDC* method [2]. Here, we follow a different path and concentrate on the matrix $\mathbf{E}$. Let $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Then the formula above can be seen as a quadratic function of variables $\alpha_i$, which can be minimised by computing all partial derivatives and equating them to zero:

$$\frac{\partial}{\partial\alpha_i}||(\mathbf{I} - \mathbf{RE})\mathbf{u}||_S^2 = 2d_i\left(\sum_j \widehat{s}_{ij}\alpha_j d_j - \sum_j s^1_{ij}d_j\right) = 0 \quad \forall\, i\,. \tag{10}$$

Here $d_i$ stands for the $i$-th component of the *jump vector* $\mathbf{d}$, elements of the matrix $\widehat{\mathbf{S}}_{\mathrm{rr}}$ are denoted as $\widehat{s}_{ij}$, and elements of the matrix $\mathbf{S}^1_{\mathrm{rr}}$ are denoted as $s^1_{ij}$. Problem (10) represents solution of a system of linear equations of dimension $n$ with a dense system matrix. Values of $\alpha_i$ obtained from (10) are tailored to the interface jump $\mathbf{d}$ of the given $\mathbf{u}$. Vector $\mathbf{u}$ changes in every iteration step, so values of $\alpha_i$ are also recomputed.

Solving (10) is in general nearly equally difficult as solving the original system (3). In order to solve this system with a reasonable effort, we can use some simplifying assumptions and solve it only approximately. One option is to omit all off-diagonal entries of matrices $\widehat{\mathbf{S}}_{\mathrm{rr}}$ and $\mathbf{S}^1_{\mathrm{rr}}$, which leads to the popular choice of

$$\alpha_i = s^1_{ii}/(s^1_{ii} + s^2_{ii})\,. \tag{11}$$

It is interesting to notice that in this case the solution does not depend on the chosen jump vector $\mathbf{d}$ and we can consider it as some approximation of minimising norm of the projection $\mathbf{I}-\mathbf{RE}$ as a whole. The main drawback of this choice is the necessity of computing the values of the diagonal entries of the matrices $\mathbf{S}$ and $\widehat{\mathbf{S}}$, which otherwise need not be explicitly computed. For this reason, corresponding values at diagonal of original matrices $\mathbf{K}^1$ and $\mathbf{K}^2$ are often used instead of diagonal entries of Schur complements in formula (11) (e.g. in [1]).

### 4.3 A new construction of the averaging operator

We propose another approach. Let us assume $\mathbf{d}$ to be some test vector chosen so that it simplifies the system of equations (10). One option is to choose all the cartesian basis vectors $\mathbf{e}_k$, one after another – then we again arrive at solution (11).

For less elementary test vectors $\mathbf{d}$ we make an additional simplification: Let us assume that all $\alpha_i$ are equal to the same value of $\alpha$ for some set of nodes (so we are going to find some average value). This is not as strange assumption as it may seem at the first glance: for large problems divided into a lot of relatively small subdomains by some automatic graph tool we probably can expect homogeneous behaviour along the interface for most pairs of adjacent subdomains. Then, after adding all equations (10) together, we get

$$\alpha = \mathbf{d}^T \mathbf{S}^1_{\text{rr}}\, \mathbf{d} / \mathbf{d}^T (\mathbf{S}^1_{\text{rr}} + \mathbf{S}^2_{\text{rr}})\, \mathbf{d}. \tag{12}$$

This formula requires only matrix-vector products that are already computed in the PCG method and it can be generalized to more than 2 subdomains. Our proposition is to choose several test vectors with nonzero values at some selected nodes only (typically face or edge) and compute corresponding value of $\alpha$.

## 5 Numerical results and conclusion

For a simple preliminary test depicted in Figure 1, a 2D Poisson equation on a rectangular domain was chosen. The domain was divided into two rectangular subdomains (different in size), both of which touch the boundary with prescribed Dirichlet boundary condition. The problem was discretized by FEM with bilinear elements. BDDC was used just as an iteration method, not as a preconditioner combined with PCG. Four different methods for choice of the averaging operator $E$ were tested:

I : arithmetic average, i.e. $\alpha = 0.5$,
II : weighted average (11), i.e. $\alpha_i = s^1_{ii}/(s^1_{ii} + s^2_{ii})$,
III : proposition (12) with $d = (1, \dots, 1)$, i.e. $\alpha = \sum_{i,j} s^1_{ij} / \sum_{i,j}(s^1_{ij} + s^2_{ij})$,
IV : proposition (12) with $d$ chosen as actual interface jump.

Table 1 contains norms of errors (differences from exact solution) at first 5 iterations. There are two different choices of coarse unknowns: either none (first part of the table), or 2 nodes at the opposite ends of the interface (second part). For Method II, computed values of $\alpha_i$ were between 0.499 and 0.500 in both cases (i.e. very close to the arithmetic average). For Method III, value of $\alpha$ was 0.191 for the first case and 0.341 for the second. For Method IV, values of $\alpha$ were recomputed in every step and are presented in the last column.

For this simple test problem, it seems that Methods III and IV outperform Methods I and II. An interesting observation is that for the first three methods, involving coarse unknowns lead to better performance as one would expect, but in the case of Method IV the opposite is true, and although Method IV was absolutely excellent in the first case, with coarse nodes it worsens so that Method III becomes slightly better. These are just preliminary results and more general numerical tests will be performed for other 2D as well as 3D problems.

| iter. | Method I | Method II | Method III | Method IV | $\alpha$ |
|---|---|---|---|---|---|
| without coarse nodes | | | | | |
| 1. | 1.7909 | 1.7851 | 0.9373 | 1.7909 | 0.500 |
| 2. | 1.1010 | 1.0938 | 0.3034 | 0.0022 | 0.193 |
| 3. | 0.6769 | 0.6702 | 0.0982 | 0.0004 | 0.273 |
| 4. | 0.4161 | 0.4107 | 0.0318 | 7e-07 | 0.475 |
| 5. | 0.2558 | 0.2517 | 0.0103 | 4e-11 | 0.191 |
| 2 coarse nodes | | | | | |
| 1. | 0.8663 | 0.8635 | 0.2690 | 0.8663 | 0.5 |
| 2. | 0.2576 | 0.2560 | 0.0302 | 0.0476 | 0.316 |
| 3. | 0.0766 | 0.0759 | 0.0035 | 0.0056 | 0.314 |
| 4. | 0.0227 | 0.0225 | 0.0004 | 0.0007 | 0.314 |
| 5. | 0.0068 | 0.0067 | 5e-05 | 8e-05 | 0.314 |

**Tab. 1:** *Comparison of discussed methods: errors at first 5 iterations for the test problem depicted in Figure 1, without (top) and with (bottom) coarse unknowns.*

## References

[1] Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25** (2003), 246–258. doi: 10.1137/S1064827502412887.

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

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

# INTERACTION OF COMPRESSIBLE FLOW WITH AN AIRFOIL[*]

Jan Česenek, Miloslav Feistauer

**Abstract**

The paper is concerned with the numerical solution of interaction of compressible flow and a vibrating airfoil with two degrees of freedom, which can rotate around an elastic axis and oscillate in the vertical direction. Compressible flow is described by the Navier-Stokes equations written in the ALE form. This system is discretized by the semi-implicit discontinuous Galerkin finite element method (DGFEM) and coupled with the solution of ordinary differential equations describing the airfoil motion. Computational results showing the flow induced airfoil vibrations are presented.

## 1 Formulation of the continuous problem

We consider 2D compressible viscous flow in a bounded domain $\Omega(t) \subset R^2$ depending on time $t \in [0,T]$. We assume that the boundary $\partial\Omega(t)$ of $\Omega(t)$ consists of three disjoint parts: $\partial\Omega(t) = \Gamma_I \cup \Gamma_O \cup \Gamma_W(t)$, where $\Gamma_I$ is inlet, $\Gamma_O$ is outlet and $\Gamma_W(t)$ is impermeable wall, whose part may move.

The time dependence of the domain is taken into account with the aid of a regular one-to-one ALE mapping (cf. [4]) $\mathcal{A}_t : \Omega_0 \longrightarrow \Omega_t$, i.e. $\mathcal{A}_t : X \longmapsto x = x(X,t) = \mathcal{A}_t(X)$. We define the ALE velocity $\tilde{\boldsymbol{z}}(X,t) = \partial \mathcal{A}_t(X)/\partial t$, $\boldsymbol{z}(x,t) = \tilde{\boldsymbol{z}}(\mathcal{A}^{-1}(x),t)$, $\quad t \in [0,T]$, $X \in \Omega_0$, $x \in \Omega_t$, and the ALE derivative of a function $f = f(x,t)$ defined for $x \in \Omega_t$ and $t \in (0,T)$: $D^A f(x,t)/Dt = \partial \tilde{f}(X,t)/\partial t$, where $\tilde{f}(X,t) = f(\mathcal{A}_t(X),t)$, $X \in \Omega_0$.

The system describing compressible flow consisting of the continuity equation, the Navier-Stokes equations and the energy equation (see, e.g. [2]) can be written in the ALE form

$$\frac{D^A \boldsymbol{w}}{Dt} + \sum_{s=1}^{2} \frac{\partial \boldsymbol{g}_s(\boldsymbol{w})}{\partial x_s} + \boldsymbol{w}\,\mathrm{div}\boldsymbol{z} = \sum_{s=1}^{2} \frac{\partial \boldsymbol{R}_s(\boldsymbol{w},\nabla\boldsymbol{w})}{\partial x_s}, \tag{1}$$

where for $i,j = 1,2$ we have

$$\begin{aligned}
&\boldsymbol{w} = (w_1,\dots,w_4)^T = (\rho,\rho v_1,\rho v_2,E)^T \in I\!R^4, \quad \boldsymbol{g}_i(\boldsymbol{w}) = \boldsymbol{f}_i(\boldsymbol{w}) - z_i\boldsymbol{w}, \qquad (2)\\
&\boldsymbol{f}_i(\boldsymbol{w}) = (f_{i1},\cdots,f_{i4})^T = (\rho v_i, \rho v_1 v_i + \delta_{1i}\,p, \rho v_2 v_i + \delta_{2i}\,p, (E+p)v_i)^T,\\
&\boldsymbol{R}_i(\boldsymbol{w},\nabla\boldsymbol{w}) = (R_{i1},\dots,R_{i4})^T = \left(0, \tau^V_{i1}, \tau^V_{i2}, \tau^V_{i1}\,v_1 + \tau^V_{i2}\,v_2 + k\partial\theta/\partial x_i\right)^T,\\
&\tau^V_{ij} = (-2\,\mathrm{div}\boldsymbol{v}/3\,\delta_{ij} + 2\,d_{ij}(\boldsymbol{v}))/Re, \;\; d_{ij}(\boldsymbol{v}) = (\partial v_i/\partial x_j + \partial v_j/\partial x_i)/2.
\end{aligned}$$

[*]The research of J. Česenek was supported by the Grant No. 12810 of the Grant Agency of the Charles University Prague. The research of M. Feistauer is a part of the research project MSM 0021620839 of the Ministry of Education of the Czech Republic. It was also partly supported by the grant No. 201/08/0012 of the Czech Science Foundation.

We use the following notation: $\rho$ - density, $p$ - pressure, $E$ - total energy, $\boldsymbol{v} = (v_1, v_2)$ - velocity, $\theta$ - absolute temperature, $\gamma > 1$ - Poisson adiabatic constant, $c_v > 0$ - specific heat at constant volume, $Re$ - the Reynolds number, $k$ - heat conduction. The vector-valued function $\boldsymbol{w}$ is called the state vector, the functions $\boldsymbol{f}_i$ are the so-called inviscid fluxes and $\boldsymbol{R}_i$ represent viscous terms. The above system is completed by the thermodynamical relations

$$p = (\gamma - 1)(E - \rho|\boldsymbol{v}|^2/2), \;\theta = \left(E/\rho - |\boldsymbol{v}|^2/2\right)/c_v$$

and equipped with the initial condition $\boldsymbol{w}(x, 0) = \boldsymbol{w}^0(x),\; x \in \Omega_0$, and the following boundary conditions:

$$\rho = \rho_D, \;\boldsymbol{v} = \boldsymbol{v}_D, \;\sum_{i,j=1}^{2} \tau_{ij}^V n_i v_j + k\frac{\partial \theta}{\partial n} = 0 \;\text{ on } \Gamma_I,$$

$$\boldsymbol{v}|_{\Gamma_{W_t}} = \boldsymbol{z}_D \;-\; \text{velocity of a moving wall, } \partial\theta/\partial n = 0 \text{ on } \Gamma_{W_t},$$

$$\sum_{i=1}^{2} \tau_{ij}^V n_i = 0, \;j = 1,\, 2, \;\partial\theta/\partial n = 0 \text{ on}\Gamma_O,$$

with given data $\boldsymbol{w}^0$, $\rho_D$, $\boldsymbol{v}_D$, $\boldsymbol{z}_D$.

The terms $\boldsymbol{R}_s$ and $\boldsymbol{f}_s$ satisfy the relations

$$\boldsymbol{R}_s(\boldsymbol{w}, \nabla\boldsymbol{w}) = \sum_{k=1}^{2} \mathbf{K}_{s,k}(\boldsymbol{w})\frac{\partial \boldsymbol{w}}{\partial x_k}, \quad \boldsymbol{f}_s(\boldsymbol{w}) = \mathbf{A}_s(\boldsymbol{w})\boldsymbol{w}, \tag{3}$$

where $\mathbf{K}_{s,k}(\boldsymbol{w}) \in R^{4\times 4}$ and $\mathbf{A}_s$ is the Jacobian matrix of $\boldsymbol{f}_s$.

## 2 Discretization

### 2.1 Discontinuous Galerkin space discretization

By $\Omega_h(t)$ we denote polygonal approximation of the domain $\Omega(t)$. Let $\mathcal{T}_h(t) = \{K_i\}_{i\in I(t)}$ be a triangulation of the domain $\Omega_h(t)$ formed by a finite number of closed triangles $K_i$ with mutually disjoint interiors. We set $h_K = diam(K)$ as the diameter of $K$, $h(t) = \max_{K\in\mathcal{T}_h(t)} h_K$, $|\,K|$ is the Lebesgue measure of $K$. All elements of $\mathcal{T}_h(t) = \{K_i\}_{i\in I(t)}$ will be numbered so that $I(t) \subset Z^+ = \{0, 1, 2, 3, ...\}$ is a suitable index set. If two elements have a common face, than we call them neighbours and put $\Gamma_{ij} = \Gamma_{ji} = \partial K_i \cap \partial K_j$. For each $i \in I(t)$ we define the index set $s(i)(t) = \{j \in I(t); K_j \text{ is a neighbour of } K_i\}$. The boundary $\partial\Omega_h(t)$ is formed by a finite number of sides of elements $K_i$ adjacent to $\partial\Omega_h(t)$. We denote all these boundary sides by $S_j$, where $j \in I_b(t) \subset Z^- = \{-1, -2, -3, ...\}$ and set $\gamma(i)(t) = \{j \in I_b(t); S_j \text{ is a side of } K_i\}, \Gamma_{ij} = S_j$ for $K_i \in \mathcal{T}_h(t)$ such that $S_j \subset \partial K_i, j \in I_b(t)$. For an element $K_i$, not containing any boundary side $S_j$, we set $\gamma(i)(t) = \emptyset$. Obviously $s(i)(t) \cap \gamma(i)(t) = \emptyset$ for all $i \in I(t)$. Moreover we define $S(i)(t) = s(i)(t) \cup \gamma(i)(t)$.

We shall look for an approximate solution of the problem in the space $\mathbf{S}_h(t) = \{v; v\,|_K \in P^r(K), \forall K \in \mathcal{T}_h(t)\}^4$, where $r \geq 0$ is an integer and $P^r(K)$ is the space of polynomials of degree at most $r$ on $K$. If $v \in \mathbf{S}$, then we use the notation $v|_{\Gamma_{ij}}$ and $v|_{\Gamma_{ji}}$ for the traces of $v$ on $\Gamma_{ij}$ from the side of the adjacent elements $K_i$ and $K_j$, respectively, $\langle v\rangle_{\Gamma_{ij}}$ for the average of traces of $v$ on the face $\Gamma_{ij}$ from the side of the adjacent elements and $[v]_{\Gamma_{ij}}$ the jump of $v$ on $\Gamma_{ij}$. By $\boldsymbol{n}_{ij}$ we denote the unit outer normal to the boundary of $K_i$ on $\Gamma_{ij}$.

For arbitrary $t \in [0,T]$ we can multiply the system by a test function $\varphi \in \mathbf{S}_h(t)$ integrate and sum over all $K_i \in \mathcal{T}_h(t)$, apply Green's theorem and introduce a numerical flux $\mathbf{H}$. Then we introduce the following forms (cf. [1]):

$$
\begin{aligned}
\tilde{b}_h(\boldsymbol{w},\boldsymbol{\varphi}_h) =& -\sum_{i\in I(t)}\int_{K_i}\sum_{s=1}^{2}\boldsymbol{g}_s(\boldsymbol{w})\frac{\partial\boldsymbol{\varphi}_h}{\partial x_s}\mathrm{d}x + \sum_{i\in I(t)}\sum_{i\in S(i)(t)}\int_{\Gamma_{ij}}\mathbf{H}(\boldsymbol{w}|_{\Gamma_{ij}},\boldsymbol{w}|_{\Gamma_{ji}},\boldsymbol{n}_{ij})\mathrm{d}S\\
\tilde{a}_h(\boldsymbol{w},\boldsymbol{\varphi}_h) =& -\sum_{i\in I(t)}\int_{K_i}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}_{s,k}(\boldsymbol{w})\frac{\partial\boldsymbol{w}}{\partial x_k}\cdot\frac{\partial\boldsymbol{\varphi}_h}{\partial x_s}\mathrm{d}x\\
&+\sum_{i\in I(t)}\sum_{\substack{j\in s(i)(t)\\ j<i}}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\left\langle\sum_{k=1}^{2}\mathbf{K}_{s,k}(\boldsymbol{w})\frac{\partial\boldsymbol{w}}{\partial x_k}\right\rangle(n_{ij})_s\cdot[\boldsymbol{\varphi}_h]\mathrm{d}S\\
&+\sum_{i\in I(t)}\sum_{j\in\gamma_D(i)(t)}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}_{s,k}(\boldsymbol{w})\frac{\partial\boldsymbol{w}}{\partial x_k}(n_{ij})_s\cdot\boldsymbol{\varphi}_h\mathrm{d}S\\
&+\ \Theta\sum_{i\in I(t)}\sum_{\substack{j\in s(i)(t)\\ j<i}}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\left\langle\sum_{k=1}^{2}\mathbf{K}^T_{k,s}(\boldsymbol{w})\frac{\partial\boldsymbol{\varphi}_h}{\partial x_k}\right\rangle(n_{ij})_s\cdot[\boldsymbol{w}]\mathrm{d}S\\
&+\ \Theta\sum_{i\in I(t)}\sum_{j\in\gamma_D(i)(t)}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}^T_{k,s}(\boldsymbol{w})\frac{\partial\boldsymbol{\varphi}_h}{\partial x_k}(n_{ij})_s\cdot\boldsymbol{w}\mathrm{d}S\\
J_h^\sigma(\boldsymbol{w},\boldsymbol{\varphi}_h) =& \sum_{i\in I(t)}\sum_{\substack{j\in s(i)(t)\\ j<i}}\int_{\Gamma_{ij}}\sigma[\boldsymbol{w}]\cdot[\boldsymbol{\varphi}_h]\mathrm{d}S + \sum_{i\in I(t)}\sum_{j\in\gamma_D(i)(t)}\int_{\Gamma_{ij}}\sigma\boldsymbol{w}\cdot\boldsymbol{\varphi}_h\mathrm{d}S\\
\tilde{l}_h(\boldsymbol{w},\boldsymbol{\varphi}_h) =&\Theta\sum_{i\in I(t)}\sum_{j\in\gamma_D(i)(t)}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}^T_{k,s}(\boldsymbol{w})\frac{\partial\boldsymbol{\varphi}_h}{\partial x_k}(n_{ij})_s\cdot\boldsymbol{w}_B\mathrm{d}S\\
&+\sum_{i\in I(t)}\sum_{j\in\gamma_D(i)(t)}\int_{\Gamma_{ij}}\sigma\boldsymbol{w}_B\cdot\boldsymbol{\varphi}_h\mathrm{d}S,
\end{aligned}
$$

where $\sigma\,|_{\Gamma_{ij}}= \frac{C_W}{h(\Gamma_{ij})Re}$, $C_W > 0$ is a suitable sufficiently large constants and $\boldsymbol{w}_B$ is a boundary state defined by the Dirichlet boundary condition and extrapolation. By $(\cdot,\cdot)$ we denote the $L^2(\Omega(t_{k+1}))$-scalar product. We set $\Theta = -1$ or 0 or 1 and get the so-called nonsymmetric or incomplete or symmetric version of the viscous form. In practical computations we use $\Theta = 1$.

Now we can define the discrete problem: Find $\boldsymbol{w}_h(t) \in \mathbf{S}_h(t)$ such that

$$\begin{aligned}
&\left(\frac{D^{\mathcal{A}}\boldsymbol{w}_h(t)}{Dt}, \boldsymbol{\varphi}_h\right) - (\mathrm{div}\boldsymbol{z}(t)\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) + \tilde{b}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) + \tilde{a}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) \\
+ \quad & J_h^\sigma(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = \tilde{l}_h(\boldsymbol{\varphi}_h) \quad \forall \boldsymbol{\varphi}_h \in \mathbf{S}_h(t), \quad \forall t \in (0, T), \\
& \boldsymbol{w}_h(0) = \boldsymbol{w}_h^0.
\end{aligned}$$

where $\boldsymbol{w}_h^0$ is the $\mathbf{S}_h(0)$-approximation of $\boldsymbol{w}^0$. It means that

$$\left(\boldsymbol{w}_h^0, \boldsymbol{\varphi}_h\right) = \left(\boldsymbol{w}^0, \boldsymbol{\varphi}_h\right) \quad \forall \boldsymbol{\varphi}_h \in \mathbf{S}_h(0).$$

## 2.2 Time discretization

Let us consider a partition $0 = t_0 < t_1 < ... < t_M$ of the interval $[0, T]$, $t_k = k\tau$, $\tau > 0$. We use the approximation $\boldsymbol{w}_h(t_l) \approx \boldsymbol{w}_h^l$, defined in $\Omega_h(t_l)$. Then we set $\hat{\boldsymbol{w}}_h^k(x) = \boldsymbol{w}_h^k(\mathcal{A}_{t_k}(\mathcal{A}_{t_{k+1}}^{-1}(x)))$, $\quad x \in \Omega_h(t_{k+1})$, and approximate the ALE-derivative using the first order backward difference:

$$\left(\frac{D^{\mathcal{A}}\boldsymbol{w}_h(t_{k+1})}{Dt}, \boldsymbol{\varphi}_h\right) \approx \left(\frac{\boldsymbol{w}_h^{k+1} - \hat{\boldsymbol{w}}_h^k}{\tau}, \boldsymbol{\varphi}_h\right).$$

Since the terms $\tilde{a}_h$ and $\tilde{b}_h$ are nonlinear, we shall linearized them. For $\tilde{b}_h$ we use the property (3) of $\boldsymbol{f}_s$ and the definition of $\boldsymbol{g}_s$. We get the approximation

$$\sum_{i \in I(t)} \int_{K_i} \sum_{s=1}^{2} \boldsymbol{g}_s(\boldsymbol{w}) \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s} \mathrm{d}x \approx \sigma_1 = \sum_{i \in I(t_{k+1})} \int_{K_i} \sum_{s=1}^{2} \left(\mathbf{A}_s(\hat{\boldsymbol{w}}_h^k) - z_s \mathbf{I}\right) \boldsymbol{w}_h^{k+1} \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s} \, \mathrm{d}x.$$

Now let us set $\mathbf{P}(\boldsymbol{w}, \boldsymbol{n}) := \sum_{s=1}^2 \left(\mathbf{A}_s(\boldsymbol{w}) - z_s\mathbf{I}\right) n_s$, $(\boldsymbol{n} = (n_1, n_2), n_1^2 + n_2^2 = 1)$. We have $\sum_{s=1}^2 \boldsymbol{g}_s(\boldsymbol{w}) n_s = \mathbf{P}(\boldsymbol{w}, \boldsymbol{n})\boldsymbol{w}$. It is possible to show that the matrix $\mathbf{P}$ is diagonalizable: $\mathbf{P} = \mathbf{T}\mathbf{D}\mathbf{T}^{\mathbf{-1}}$, where $\mathbf{T}$ is a nonsingular matrix, $\mathbf{D} = \mathrm{diag}(\lambda_1, ..., \lambda_4)$ is a diagonal matrix and $\lambda_i$ are the eigenvalues of $\mathbf{P}$. Then we can define the "positive" and "negative" parts of the matrix $\mathbf{P}$: $\mathbf{P}^\pm = \mathbf{T}\mathbf{D}^\pm\mathbf{T}^{\mathbf{-1}}$, where $\mathbf{D}^\pm = \mathrm{diag}(\lambda_1^\pm, ..., \lambda_4^\pm)$ and $\lambda^+ = \max(\lambda, 0)$, $\lambda^- = \min(\lambda, 0)$. Using this concept, we introduce the so-called Vijayasundaram numerical flux

$$\mathbf{H}_V(\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{n}) \;\; = \;\; \mathbf{P}^+\left(\frac{\boldsymbol{w}_1 + \boldsymbol{w}_2}{2}, \boldsymbol{n}\right)\boldsymbol{w}_1 + \mathbf{P}^-\left(\frac{\boldsymbol{w}_1 + \boldsymbol{w}_2}{2}, \boldsymbol{n}\right)\boldsymbol{w}_2.$$

Then we can approximate integrals over faces in the following way:

$$\begin{aligned}
&\sum_{i \in I(t)} \sum_{j \in S(i)(t)} \int_{\Gamma_{ij}} \mathbf{H}(\boldsymbol{w}|_{\Gamma_{ij}}, \boldsymbol{w}|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \, \mathrm{d}S \;\approx\; \sigma_2 := \\
&\sum_{i \in I(t_{k+1})} \sum_{j \in S(i)(t_{k+1})} \int_{\Gamma_{ij}} \mathbf{P}^+\left(\frac{\hat{\boldsymbol{w}}_h^k|_{\Gamma_{ij}} + \hat{\boldsymbol{w}}_h^k|_{\Gamma_{ji}}}{2}, \boldsymbol{n}_{ij}\right) \boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}} \cdot \boldsymbol{\varphi}_h \, \mathrm{d}S \\
+ \quad &\sum_{i \in I(t_{k+1})} \sum_{j \in S(i)(t_{k+1})} \int_{\Gamma_{ij}} \mathbf{P}^-\left(\frac{\hat{\boldsymbol{w}}_h^k|_{\Gamma_{ij}} + \hat{\boldsymbol{w}}_h^k|_{\Gamma_{ji}}}{2}, \boldsymbol{n}_{ij}\right) \boldsymbol{w}_h^{k+1}|_{\Gamma_{ji}} \cdot \boldsymbol{\varphi}_h \mathrm{d}S
\end{aligned}$$

and define the form $b_h(\hat{\boldsymbol{w}}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = -\sigma_1 + \sigma_2$.

Using (3), we linearize viscous terms:

$$a_h(\hat{\boldsymbol{w}}_h^k, \boldsymbol{w}_h^{k+1}\boldsymbol{\varphi}_h) = -\sum_{i\in I(t_{k+1})}\int_{K_i}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}_{s,k}(\hat{\boldsymbol{w}}_h^k)\frac{\partial \boldsymbol{w}_h^{k+1}}{\partial x_k}\cdot\frac{\partial \boldsymbol{\varphi}_h}{\partial x_s}\mathrm{d}x$$

$$+ \sum_{i\in I(t_{k+1})}\sum_{\substack{j\in s(i)(t_{k+1})\\ j<i}}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\left\langle\sum_{k=1}^{2}\mathbf{K}_{s,k}(\hat{\boldsymbol{w}}_h^k)\frac{\partial \boldsymbol{w}_h^{k+1}}{\partial x_k}\right\rangle(n_{ij})_s\cdot[\boldsymbol{\varphi}_h]\mathrm{d}S$$

$$+ \sum_{i\in I(t_{k+1})}\sum_{j\in\gamma_D(i)(t_{k+1})}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}_{s,k}(\hat{\boldsymbol{w}}_h^k)\frac{\partial \boldsymbol{w}_h^{k+1}}{\partial x_k}(n_{ij})_s\cdot\boldsymbol{\varphi}_h\mathrm{d}S$$

$$+ \Theta\sum_{i\in I(t_{k+1})}\sum_{\substack{j\in s(i)(t_{k+1})\\ j<i}}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\left\langle\sum_{k=1}^{2}\mathbf{K}_{k,s}^{T}(\hat{\boldsymbol{w}}_h^k)\frac{\partial \boldsymbol{\varphi}_h}{\partial x_k}\right\rangle(n_{ij})_s\cdot[\boldsymbol{w}_h^{k+1}]\mathrm{d}S$$

$$+ \Theta\sum_{i\in I(t_{k+1})}\sum_{j\in\gamma_D(i)(t_{k+1})}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}_{k,s}^{T}(\hat{\boldsymbol{w}}_h^k)\frac{\partial \boldsymbol{\varphi}_h}{\partial x_k}(n_{ij})_s\cdot\boldsymbol{w}_h^{k+1}\mathrm{d}S,$$

and the right-hand side form:

$$l_h(\hat{\boldsymbol{w}}_h^k, \boldsymbol{\varphi}_h) = \Theta\sum_{i\in I(t_{k+1})}\sum_{j\in\gamma_D(i)(t_{k+1})}\int_{\Gamma_{ij}}\sum_{s=1}^{2}\sum_{k=1}^{2}\mathbf{K}_{k,s}^{T}(\hat{\boldsymbol{w}}_h^k)\frac{\partial \boldsymbol{\varphi}_h}{\partial x_k}(n_{ij})_s\cdot\boldsymbol{w}_B^{k+1}\mathrm{d}S$$

$$+\sum_{i\in I(t_{k+1})}\sum_{j\in\gamma_D(i)(t_{k+1})}\int_{\Gamma_{ij}}\frac{C_W}{h(\Gamma_{ij})Re}\boldsymbol{w}_B^{k+1}\cdot\boldsymbol{\varphi}_h\mathrm{d}S$$

All these considerations lead us to the following semi-implicit scheme: For $k = 0, 1, ...$ find $\boldsymbol{w}_h^{k+1} \in \mathbf{S}_h(t_{k+1})$ such that

$$\left(\frac{\boldsymbol{w}_h^{k+1} - \hat{\boldsymbol{w}}_h^k}{\tau}, \boldsymbol{\varphi}_h\right) - \left(\mathrm{div}\boldsymbol{z}(t_{k+1})\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h\right) + b_h(\hat{\boldsymbol{w}}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) \tag{4}$$

$$+a_h(\hat{\boldsymbol{w}}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + J_h^\sigma(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = l_h(\hat{\boldsymbol{w}}_h^k, \boldsymbol{\varphi}_h) \quad \forall\boldsymbol{\varphi}_h \in \mathbf{S}_h(t_{k+1}).$$

## 3 Fluid-structure interaction

We shall simulate motion of a profile with two degrees of freedom: $H$ - displacement of the profile in the vertical direction and $\alpha$ - the rotation of the profile around the so-called elastic axis. The motion of the profile is described by the system of ordinary differential equations

$$\begin{aligned} m\ddot{H} + k_{HH}H + S_\alpha\ddot{\alpha} &= -L(t), \\ S_\alpha\ddot{H} + I_\alpha H + k_{\alpha\alpha}\alpha &= M(t), \end{aligned} \tag{5}$$

where we use the following notation: $m$ - mass of the airfoil, $L(t)$ - aerodynamic lift force, $M(t)$ - aerodynamic torsional moment, $S_\alpha$ - static moment of the airfoil

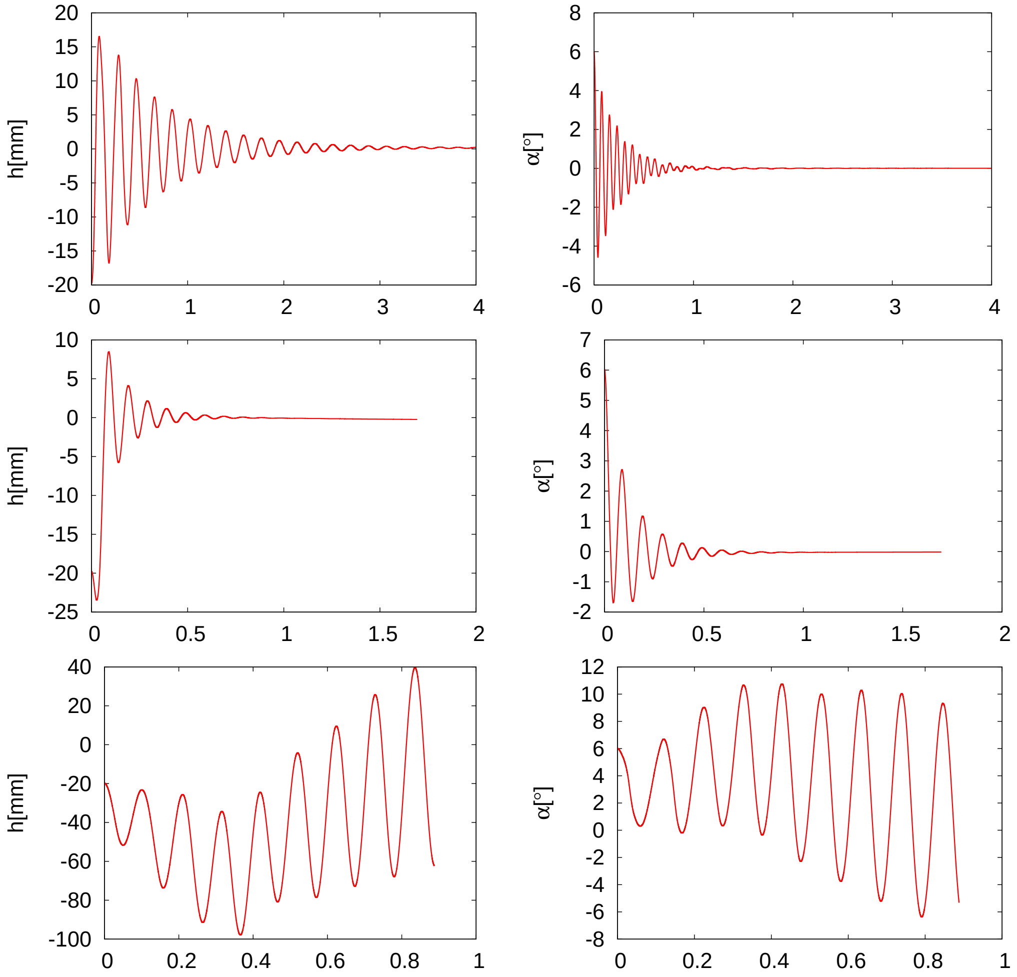


**Fig. 1:** *Displacement $H$ (left) and rotation angle $\alpha$ (right) of the airfoil in dependence on time for far-field velocity 10, 30 and 40 m/s.*

around the elastic axis, $I_\alpha$ - inertia moment of the airfoil around the elastic axis, $k_{HH}$ - bending stiffness, $k_{\alpha\alpha}$ - torsional stiffness. For the derivation of system (5) see, e.g. [5].

System (5) is transformed to a first-order system and solved by the fourth-order Runge-Kutta method together with the discrete flow problem (4). The ALE mapping is constructed on the new time level $t_{k+1}$ on the basis of the computed values $H(t_{k+1})$ and $\alpha(t_{k+1})$.

## 4 Numerical experiments

We perform numerical experiments with the following data and initial conditions: $m = 0.086622$ kg, $S_a = -0.000779673$ kg m, $I_a = 0.000487291$ kg m$^{-2}$, $k_{HH} = 105.109$ N/m, $k_{\alpha\alpha} = 3.696682$ Nm/rad, $l = 0.05$ m, $c = 0.3$ m, far-field density $\rho = 1.225$ kg m$^{-3}$, $H(0) = -20$mm, $\alpha(0) = 6°$, $\dot{H}(0) = \dot{\alpha}(0) = 0$.

Figure 1 shows the displacement $H$ and the rotation angle $\alpha$ in dependence on time for the far-field velocity 10, 30 and 40 m/s. We see that for the velocities 10 and 30 m/s the vibrations are damped, but for the velocity 40 m/s we get the flutter instability when the vibration amplitudes are increasing in time. The monotonous increase and decrease of the average values of $H$ and $\alpha$, respectively, shows that the flutter is combined with a divergence instability in the presented example.

These results are qualitatively comparable with vibrations of the airfoil NACA 0012 induced by viscous incompressible flow, contained in [3]. For low far-field velocity the differences of the presented results and results from [3] are small, because the compressibility of the fluid is not significant. For the far-field velocity 40 m/s the qualitative behaviour of the vibrations (flutter combined with divergence) is comparable with the results in [3] obtained by the finite element method. The quantitative difference is already larger probably due to compressibility taken into account in the present paper.

## References

[1] Dolejší, V.: Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. Commun. Comput. Phys. **4** (2008), 231–274.

[2] Feistauer, M., Felcman, J., and Straškraba, I.: *Mathematical and computational methods for compressible flow.* Clarendon Press, Oxford, 2003.

[3] Honzátko, R., Horáček, J., Kozel, K., and Sváček, P.: Simulation of free airfoil vibrations in incompressible viscous flow - comparison of FEM and FVM. Appl. Math. Comput. (submitted).

[4] Nomura, T. and Hughes, T.J.R.: An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body. Comput. Methods Appl. Mech. Engrg. **95** (1992), 115–138.

[5] Sváček, P., Feistauer, M., and Horáček, J.: Numerical simulation of flow induced airfoil vibrations with large amplitudes. J. of Fluids and Structures **23** (2007), 391–411.

# ELEMENTS OF UNCERTAINTY MODELING[*]

Jan Chleboun

**Abstract**

The goal of this contribution is to introduce some approaches to uncertainty modeling in a way accessible to non-specialists. Elements of the Monte Carlo method, polynomial chaos method, Dempster-Shafer approach, fuzzy set theory, and the worst (case) scenario method are presented.

## 1 Introductory comments on modeling and uncertain data

Where can uncertainty analysis be placed in computational modeling? Typically, uncertainty propagation analysis leads to solving “two level” problems. This means that we can distinguish both an inner problem and an outer problem that together constitute an uncertainty analysis problem. Such a structure is not uncommon; a PDE-constrained optimization also falls into this category, for instance. Indeed, if a problem of this kind is solved by successive optimization steps, then the inner PDE is repeatedly solved during the optimization process to deliver necessary data to a constrained optimization algorithm keeping the optimized variables in an admissible set.

In engineering-oriented problems, the inner problem, commonly known as the *state problem*, represents the mathematical model of a physical phenomenon or design (imagine a temperature field in a heated body or a mechanical stress distribution in a loaded body, for example). Often, the inner problem is rather standard and even easily solvable for given unique and crisp input values such as thermal conductivity coefficients, heat capacity, intensity of heat sources, boundary condition parameters, loading forces, Young modulus, etc.

The outer problem originates from the fact that the values of input parameters are usually not known exactly. Then a question arises how uncertainty can be measured in inputs, how it propagates through the state problem, and how it can be measured in state problem outputs.

We can view uncertainty in modeling from yet another perspective. To see this, let us recall the general layout of a modeling process.

(I) A situation (phenomenon) we wish to model to get an insight and to predict the behavior we are interested in.

(II) Available input information that we can use in our models. This step is closely related to (III) next.

[*]This research was supported by grant No. P105/10/1682 of the Czech Science Foundation. The author thanks Dr. Richard (Dick) Haas for hours of fruitful discussions.

(III) A model chosen from a hierarchy of mathematical models. This step is crucial because it determines the rest of the modeling sequence. Choosing an adequate model can be a difficult task in which the model complexity and solvability as well as the model adequacy to our needs have to be taken into account. Modeling a beam-like body can be a good example. Models of different complexity are at one's disposal, take Bernoulli beam, Timoshenko beam, perhaps a 2D shell if one dimension of the body is significantly smaller than the others, and a full 3D model. We can, however, also consider a hierarchy of material models from linear models to nonlinear ones. In choosing a model, a compromise has to be made to end up with a model that is not too complex and computationally demanding, but can still deliver information that we need.

(IV) Validation [16, 17]. It is a process of gaining trust in the mathematical model. Roughly speaking, the mathematical formulation should be adequate to both the phenomenon that we model and the questions we wish to answer through the model. Although a theoretical analysis is always valuable and can significantly contribute to the validation process, validation is unthinkable without computational modeling, see (VI).

(V) Approximation. Once the mathematical model is defined, we find ourselves in a situation similar to (III). We have to choose a numerical method to obtain an approximate solution. Again, different or even contradictory factors should be balanced.

(VI) Verification [16, 17]. It is a process of gaining trust in the numerical method, its implementation, and its accuracy. This trust originates from various sources. At least, it is necessary to solve benchmark problems and to numerically check theoretical convergence rates. Verification is a matter of mathematics and, unlike validation, it is independent of the modeled object. On the other hand, only verified numerical models allow us to put "hands on" the mathematical model through the numerical solution that is believed[1] to be close to the exact solution of the mathematical model.

(VII) Model output and the desired information. The former may not be equal to the latter, and a post-processing may be required. It is important to obtain outputs that contain, though possibly hidden, the desired information. As a consequence, (VII) is closely related to (III).

(VIII) Interpretation of results. This step can be more demanding than it appears to be at first glance. There is the danger of misinterpretation caused by our expectations that might be seemingly confirmed by the obtained results.

The goal of modeling is to step out of the area of well-proved solutions, and make a new prediction. Then the ultimate goal is a guaranteed prediction. That is, the solution accompanied by the evaluation of the inaccuracy caused by the model selection, by the approximation method, and by other effects. Among them, the effect of uncertainties in input data is of great importance.

[1] This belief must not be a blind belief, it should be a well-founded belief. Unfortunately, we can never be entirely sure that our (complex) software is correct.

To close this section, let us recall two kinds of uncertainty; see [15], for example.

*Epistemic uncertainty* is caused by the lack of knowledge. In principle, it can (often) be reduced through improving measuring instruments as well as data collecting and mining.

*Aleatory uncertainty* is caused by the inherent variation associated with the modeled system. Take, for example, the randomness of material parameters, or the variability of the weather.

Consequently, our mathematical models are burdened with uncertainty in input data.

### 1.1 Notation, basic setting

In the sequel, we will use $\mathcal{U}_{\mathsf{ad}}$ to denote the set of values of input parameters. For example, $\mathcal{U}_{\mathsf{ad}}$ is an interval if a scalar parameter is uncertain; $\mathcal{U}_{\mathsf{ad}}$ is a subset of $\mathbb{R}^n$ if an $n$-tuple of real values is uncertain; $\mathcal{U}_{\mathsf{ad}}$ is a set of functions if a function is uncertain.

Next, $D(a)u = f(a)$ will stand for a state problem dependent on $a \in \mathcal{U}_{\mathsf{ad}}$ where the right-hand side $f$ can also depend on $a$. Consequently, the state solution $u \equiv u(a)$ also depends on $a \in \mathcal{U}_{\mathsf{ad}}$. Examples include a boundary value problem for an ordinary or a partial differential equation dependent on $a \in \mathcal{U}_{\mathsf{ad}}$, an initial value problem dependent on $a \in \mathcal{U}_{\mathsf{ad}}$, or a variational inequality dependent on $a \in \mathcal{U}_{\mathsf{ad}}$ (then the equality symbol is inappropriate).

Finally, the quantity of interest, $\Phi(a, u(a)) \in \mathbb{R}$, will be the third ingredient of uncertainty modeling. The quantity of interest, also known as the criterion function (or criterion functional), evaluates the input data both directly and indirectly (through $u(a)$). Displacement, temperature, local mechanical stress or stress invariants, and concentration of chemicals can serve as examples.

We assume that $D(a)u = f(a)$ is uniquely solvable for each $a \in \mathcal{U}_{\mathsf{ad}}$ and that $\Psi(a) \equiv \Phi(a, u(a))$, where $a \in \mathcal{U}_{\mathsf{ad}}$, is continuous and bounded. We did not specified the set $\mathcal{U}_{\mathsf{ad}}$ but, generally speaking, we assume that $\mathcal{U}_{\mathsf{ad}}$ is a connected and compact subset of a Banach space.

## 2 Stochastic approaches to uncertainty

### 2.1 Monte Carlo method

The idea of the Monte Carlo method is quite simple. Random samples of $a$ taken from $\mathcal{U}_{\mathsf{ad}}$ are evaluated through $\Psi$, the values $\Psi(a)$ are collected, and the collection is then statistically analyzed to infer probabilistic characteristics of the model behavior; see [14, 18, 21]. An advantageous feature is that the method easily allows evaluating multiple samples in parallel to speed up collecting output data.

Figure 1 shows an example. Let us assume that we are to predict the tip displacement of a loaded cantilever beam with a constant but uncertain thickness $a$. It is further assumed that the thickness is random with a known probability distribution. By using this distribution, we generate (pseudo)random samples (Figure 1,

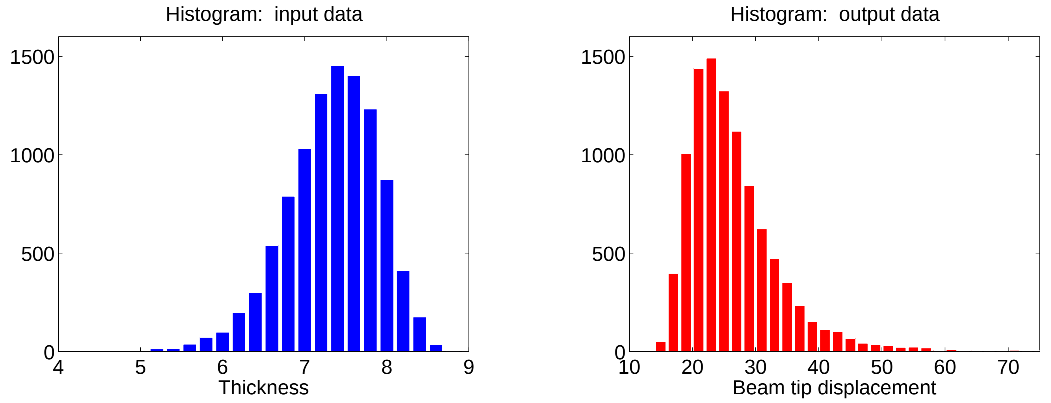


**Fig. 1:** *Monte Carlo method. Histogram of input data (left). Histogram of output data (right).*

left) of the thickness and calculate the beam tip displacement $\Psi(a)$ for each sample $a$. Then an approximation of the probabilistic behavior of $\Psi(a)$ can be inferred from the histogram of $\Psi(a)$ (Figure 1, right).

Although Monte Carlo simulation has proved fruitful and is commonly used in the modeling of uncertainty propagation through a model, one should be aware of some possible pitfalls.

The probability distribution of input data can be difficult to identify or its parameters can be uncertain.

Dependencies between input parameters are possible and, moreover, often uncertain, which complicates both the sampling procedure and the credibility of results.

If $N$ is the number of samples, then, in general, the convergence rate of estimated probabilistic parameters is equal to $\mathcal{O}(N^{-1/2})$. Although sophisticated sampling methods can be a partial remedy, the necessary number of state solutions can still be prohibitive if the state problem is computationally demanding.

## 2.2 Polynomial chaos

Polynomial chaos is also known as the Wiener polynomial chaos or Hermite chaos or, in a generalized form, the Askey chaos. In computational applications, the method is also called stochastic finite elements; see [11, 12, 22].

Analogously to the Monte Carlo method, it is assumed that the state problem input parameters can be represented by a random process; let us denote it by $X(\theta)$. The idea is to express or, in calculations, to approximate the random process through separate spatial (or temporal) deterministic variables and independent random variables.

To give an example [22], let us consider a two-dimensional expansion $\sum_{i=0}^{\infty} c_i\phi_i$ using Hermite polynomials and $\xi_1, \xi_2$, two independent Gaussian random variables with zero mean and unit variance, that is,

$$\begin{aligned}X(\theta) &= c_0 + c_1\xi_1 + c_2\xi_2 + c_3(\xi_1^2 - 1) + c_4(\xi_1\xi_2) + c_5(\xi_2^2 - 1) + \dots \\ &= c_0 + c_1\phi_1 + c_2\phi_2 + c_3\phi_3 + c_4\phi_4 + c_5\phi_5 \dots = \sum_{i=0}^{\infty} c_i\phi_i,\end{aligned}$$

where $c_i \in \mathbb{R}$ for $i = 0, 1, \dots$.

Next, a weight $w$ related to the used random variables is introduced to make the system $\{\phi_i\}$ $w$-orthogonal. For $\xi = (\xi_1, \xi_2)$ where $\xi_1$ and $\xi_2$ are Gaussian random variables, we take

$$w(\xi) = \frac{1}{2\pi}\mathrm{e}^{-(\xi_1^2+\xi_2^2)/2},$$

which simplifies the weighted inner product $\langle\cdot,\cdot\rangle_w$ of $\phi_i, \phi_j$, i.e.,

$$\langle\phi_i, \phi_j\rangle_w = \int_{S_w} \phi_i(\xi)\phi_j(\xi)w(\xi)\,\mathrm{d}\xi = \langle\phi_i, \phi_j\rangle_w\delta_{ij} \text{ (Kronecker } \delta),$$

where $S_w$ is the domain of $w$; it is $S_w = \mathbb{R}^2$ for our choice of $\xi$.

A finite part of the expansion of random inputs is employed in the state problem to obtain its approximate solution as a random process. Let us illustrate this through an example taken from [22].

Let us consider the following initial value problem

$$y'(t) = -ky(t), \quad y(0) = y_\star, \tag{1}$$

where $k \equiv k(\theta)$ is a random variable with probability density function $f$.

*Remark:* For (1) and a given $k \in \mathbb{R}$, the deterministic solution $y(t) = y_\star \mathrm{e}^{-kt}$ allows a direct probabilistic characterization of relevant quantities without the use of a polynomial chaos expansion. Take, for instance, the mean of the stochastic solution at $t$

$$\overline{y}(t) = y_\star \int_S \mathrm{e}^{-kt} f(k)\,\mathrm{d}k,$$

where $S$ is the support of $f$. For educational purposes, however, we will not use the approach based on the deterministic solution but we will apply the polynomial chaos expansion to (1). □

Let us assume that $k$ is a Gaussian random variable. Then it is recommended to expand the random variables into functions originating from Hermite polynomials applied to Gaussian random variables $\xi_1, \dots, \xi_n$ ($n = 2$ in our example). The solution $y$ of (1) is a random process $y(t, \theta)$ at any $t \in \mathbb{R} \setminus \{0\}$. We approximate both $y(t, \theta)$ and $k(\theta)$ by finite parts of their polynomial chaos expansions, i.e.,

$$y(t, \theta) \approx \widehat{y}(t, \xi) = \sum_{j=0}^{P} y_j(t)\phi_j(\xi), \quad k(\theta) \approx \widehat{k}(\xi) = \sum_{i=0}^{P} k_i\phi_i(\xi),$$

where $y_j$ are unknown functions of $t$, $k_i \in \mathbb{R}$ are unknown constants, and $\phi_i$ are known $w$-orthogonal functions. By inserting these finite sums into $y'(t) = -ky(t)$, we arrive at

$$\sum_{i=0}^{P} y_i'(t)\phi_i = -\sum_{i=0}^{P}\sum_{j=0}^{P} \phi_i\phi_j k_i y_j(t).$$

After multiplying by $\phi_\ell$ ($\ell = 0, 1, \dots, P$), integrating with the weight $w$, and exploiting the orthogonality, we obtain

$$\langle \phi_\ell, \phi_\ell \rangle_w y_\ell'(t) = -\sum_{i=0}^{P}\sum_{j=0}^{P} e_{ij\ell} k_i y_j(t), \quad \ell = 0, 1, \dots, P, \tag{2}$$

where $e_{ij\ell} = \int_{S_w} \phi_i(\xi)\phi_j(\xi)\phi_\ell(\xi)w(\xi)\,\mathrm{d}\xi$ and $S_w$ is the domain of $w$.

We observe that the original stochastic problem (1) is transformed into a system of numerically solvable (deterministic) ordinary differential equations (2).

By having $y(t,\theta) \approx \widehat{y}(t,\xi) = \sum_{j=0}^{P} y_j(t)\phi_j(\xi)$, the orthogonality, and $\phi_0 = 1$, we can approximate the mean solution at $t$ by

$$E[\widehat{y}(t,\xi)] = \overline{\widehat{y}(t,\xi)} = \int_{S_w} \sum_{j=0}^{P} y_j(t)\phi_j(\xi) 1 w(\xi)\,\mathrm{d}\xi = \langle \phi_0, \phi_0 \rangle_w y_0(t)$$

and the variance of the solution at $t$ by

$$\begin{aligned}
\operatorname{Var}(\widehat{y}(t,\xi)) &= E\left[\left(\widehat{y}(t,\xi) - \overline{\widehat{y}(t,\xi)}\right)^2\right] \\
&= \int_{S_w} \left(\sum_{j=0}^{P} y_j(t)\phi_j(\xi) - \langle \phi_0, \phi_0 \rangle_w y_0(t)\right)^2 w(\xi)\,\mathrm{d}\xi \\
&= \int_{S_w} \sum_{i=0}^{P}\sum_{j=0}^{P} y_i(t)y_j(t)\phi_i(\xi)\phi_j(\xi)w(\xi)\,\mathrm{d}\xi \\
&\qquad - 2\langle \phi_0, \phi_0 \rangle_w y_0(t) \int_{S_w} \sum_{j=0}^{P} y_j(t)\phi_j(\xi)w(\xi)\,\mathrm{d}\xi + \langle \phi_0, \phi_0 \rangle_w^2 y_0^2(t) \\
&= \sum_{i=0}^{P} \langle \phi_i, \phi_i \rangle_w y_i^2(t) - 2\langle \phi_0, \phi_0 \rangle_w^2 y_0^2(t) + \langle \phi_0, \phi_0 \rangle_w^2 y_0^2(t) \\
&= \sum_{i=1}^{P} \langle \phi_i, \phi_i \rangle_w y_i^2(t).
\end{aligned}$$

Although the complexity of the method increases with the number of random variables and, consequently, with the complexity of the expansion, the method can be

$10^2\times-10^4\times$ faster than the Monte Carlo method delivering the required probabilistic characteristics with the same accuracy; see [22].

Other probability density functions can be considered in input parameters. Relevant weights $w$ and $w$-orthogonal polynomials are then used in the analysis; see [22].

The approximate stochastic solution $\widehat{y}$ can be further evaluated by the criterion functional whose probability characteristics are to be inferred.

### 2.3 Transformation to a deterministic problem; Karhunen-Loève expansion (KLE)

The underlying idea is identical to the idea presented in the previous subsection. However, unlike the polynomial chaos expansion, which is rather heuristic, the Karhunen-Loève expansion is mathematically more rigorous. More details on the transformation can be found in [2, 3, 4, 5].

Let $\Omega \subset\ R^d$, where $d \in \{1, 2, 3\}$, be a domain. Let us consider $g$, a stochastic function on $\Omega$. Values $g(s)$ and $g(t)$, where $s, t \in \Omega$ and $s \neq t$, are random variables that can be coupled to some extent. In the probability theory, such non-deterministic couplings are characterized by the covariance function $\mathrm{cov}[g] : \Omega \times \Omega \to \mathbb{R}$ that is defined as follows

$$\mathrm{cov}[g](s,t) = E\left[\left(g(s) - E[g(s)]\right)\left(g(t) - E[g(t)]\right)\right], \quad s, t \in \Omega,$$

where $E[\omega]$ stands for the mean (expected value) of a random quantity $\omega$. We assume that $\mathrm{cov}[g]$ is continuous and bounded on $\Omega \times \Omega$.

We define an operator $T_g : L^2(\Omega) \to L^2(\Omega)$ by

$$T_g v(\cdot) = \int_\Omega \mathrm{cov}[g](x, .)v(x)\,\mathrm{d}x \quad \forall v \in L^2(\Omega).$$

It can be shown that the operator is compact, selfadjoint, and that its eigenvalues are non-negative. Let $\{\lambda_i\}_{i=1}^\infty$ be a non-increasing sequence of the eigenvalues of $T_g$ and let $\{b_i\}_{i=1}^\infty$ be a sequence of the corresponding $L^2(\Omega)$-orthonormal eigenfunctions, i.e., $T_g b_i = \lambda_i b_i$.

The stochastic function $g(x, \theta)$, where $x \in \Omega$, can be expressed by the Karhunen-Loève expansion from which, however, we take only the first $N$ terms to approximate $g$ by $g_N$, that is,

$$g_N(x,\theta) = E[g](x) + \sum_{i=1}^{N} \sqrt{\lambda_i} b_i(x) Y_i(\theta), \tag{3}$$

where the real random variables, $\{Y_i\}_{i=1}^N$, are uncorrelated, have zero mean and unit variance, i.e., $E[Y_i] = 0$ and $E[Y_i Y_j] = \delta_{ij}$.

The amplitude of $\lambda_i$ can provide guidance for choosing $N$ (if $\lambda_i$ is "small", we cut off the rest of the infinite expansion) and, analogously, for determining the size of the polynomial chaos expansion introduced in Subsection 2.2.

Let us illustrate the above ideas through a boundary value problem with stochastic functions; see [5].

Let $\Omega \subset \mathbb{R}^d$ be a domain. We consider the following elliptic boundary value problem

$$-\operatorname{div}(a(x,\theta)\nabla u(x,\theta)) = f(x,\theta) \quad \text{in } \Omega, \text{ i.e, } x \in \Omega, \tag{4}$$

$$u(x,\theta) = 0 \quad \text{on } \partial\Omega, \tag{5}$$

where $a, f$ (and $u$) are stochastic functions.

We assume that
a) $a(x,\theta) = a(x, Y_1(\theta), \dots, Y_N(\theta))$, where $Y_i$ are the functions introduced in (3);
b) $f(x,\theta) = f(x, Y_1(\theta), \dots, Y_N(\theta))$;
c) $\Gamma_i$, the range of $Y_i$, is a bounded interval in $\mathbb{R}$ for $i = 1, 2, \dots, N$;
d) the random variable $Y_i$ has a known density function $\rho_i : \Gamma_i \to \mathbb{R}^+$ (nonnegative real numbers) with $\rho_i \in L^\infty(\Gamma_i)$, where $i = 1, 2, \dots, N$.

As a consequence, $u(x,\theta) = u(x, Y_1(\theta), \dots, Y_N(\theta))$.

Let $\rho = \rho(y) : \Gamma \to \mathbb{R}^+$ be the joint probability density function of the random vector $Y = (Y_1, \dots, Y_N)$, where $\Gamma = \prod_{i=1}^N \Gamma_i \subset \mathbb{R}^N$.

We also assume that quantities $Y_i$ are not only uncorrelated but mutually independent. Consequently, $\rho(y) = \prod_{i=1}^N \rho_i(y_i)$, where $y = (y_1, \dots, y_N)$.

Like a deterministic boundary value problem, the stochastic problem (4)-(5) has its stochastic variational counterpart, see [5]. Although it is omitted here, we introduce its deterministic equivalent: Find $u \in H_0^1(\Omega) \otimes L_\rho^2(\Gamma)$ such that

$$\int_\Gamma \rho\, (a\nabla_x u, \nabla_x v)_{[L^2(\Omega)]^d}\, \mathrm{d}y = \int_\Gamma \rho\, (f, v)_{L^2(\Omega)}\, \mathrm{d}y \quad \forall v \in H_0^1(\Omega) \otimes L_\rho^2(\Gamma), \tag{6}$$

where $\nabla_x$ indicates that only the partial derivatives with respect to the spatial variables are included into the gradient, $H_0^1(\Omega)$ stands for the Sobolev space of once differentiable functions with traces vanishing on $\partial\Omega$,

$$L_\rho^2(\Gamma) = \left\{ v : \Gamma \to \mathbb{R} \mid \int_\Gamma \rho(y) v^2(y)\, \mathrm{d}y < +\infty \right\},$$

and the tensor space $H_0^1(\Omega) \otimes L_\rho^2(\Gamma)$ is a Hilbert space with the inner product defined as follows

$$(u, v)_{H_0^1(\Omega)\otimes L_\rho^2(\Gamma)} = \int_\Gamma \rho(y) (u(\cdot, y), v(\cdot, y))_{H^1(\Omega)}\, \mathrm{d}y.$$

Problem (6) is purely deterministic because the stochastic features of (4) have been transformed into the weight $\rho$. However, we pay for it by the increased number of dimensions. In (6), functions $a$, $f$, $u$, and $v$ are functions of $d + N$ variables.

Let us confine ourselves to a few comments on solving multidimensional boundary value problems though this subject would deserve a more detailed treatment.

In Monte Carlo Galerkin finite element method (FEM) [5], samples of the state solution $u$ are obtained via realizations of $a$ and $f$. For each realization, i.e., $a(\cdot, y_1, \ldots, y_N)$, $f(\cdot, y_1, \ldots, y_N)$, where $(y_1, \ldots, y_N)$ is fixed, the state solution is obtained through a standard Galerkin FEM. The samples are weighted by their probability[2] and the expected value (i.e., mean) of the stochastic state solution can be calculated, for example.

FEM methods for multidimensional BVPs have been designed. They use, for instance, $h$-FEM basis functions in the spatial variable ($d$-tuple) $x$ and $p$-FEM basis functions in the probabilistic variable ($n$-tuple) $y$; special polynomials to approximate the probabilistic part of $u$ and compute its mean efficiently; a reduced set of basis functions. It is said that problems exhibiting $\approx 10 - 20$ dimensions, or $\approx 100$ dimensions in special cases, are solvable by present means.

Although the KLE-based methods are anchored in a rigorous mathematical analysis and can deliver theoretical results as well as error estimates, one should be aware that the KLE and $\rho_i$ identification are crucial and demanding prerequisites.

*Remark:* We omit a huge class of differential equations perturbed by (white) noise (Brownian motion, Wiener process). They are known as stochastic differential equations and a special (Itô) calculus has been proposed for their analysis. □

We conclude the section on stochastic methods by the observation that these methods can deliver extremely important and valuable assessments of uncertainty and its propagation to model outputs. To perform well, however, they need input data that is sometimes (if not even often) difficult to obtain in the required quality and/or quantity.

## 3 Non-stochastic methods

Stochastic methods can also be viewed as methods where input values are weighted (by their probability) and the goal is to infer the weights of model output values. Since the weights have to fulfil rather strong assumptions (recall the requirements placed on probability measures), it can be advantageous to have methods whose assumptions about input data are relaxed.

We will present two such methods and one method where inputs are not weighted.

### 3.1 The worst (case) scenario method (WSM)

In this approach, input parameter values are considered equally possible and of the same weight. The name of the method reflects a common goal in practice — to find the particular value of a quantity of interest that is most unfavorable from an application point of view; see [13].

Since maxima are often important (maximum temperature, maximum mechanical stress, for example), to determine the “worst” scenario (also known as anti-optimization [6, 10]), we maximize $\Psi$ by searching for

[2] In Monte Carlo, it is common to divide the probability domain into subdomains of equal probability and to take a representative sample from each subdomain.

$$a^0 = \arg\max_{a \in \mathcal{U}_{\mathsf{ad}}} \Psi(a). \tag{7}$$

If also the “best” scenario

$$a_0 = \arg\min_{a \in \mathcal{U}_{\mathsf{ad}}} \Psi(a) = \arg\max_{a \in \mathcal{U}_{\mathsf{ad}}} (-\Psi(a)) \tag{8}$$

is found, then the range of $\Psi|_{\mathcal{U}_{\mathsf{ad}}}$ is given by

$$I_\Psi = [\Psi(a_0), \Psi(a^0)]. \tag{9}$$

To obtain (7)-(9), we assume that $\mathcal{U}_{\mathsf{ad}}$ is a compact and convex set and that $\Psi : \mathcal{U}_{\mathsf{ad}} \to \mathbb{R}$ is a continuous map. In practice, non-convex $\mathcal{U}_{\mathsf{ad}}$ makes maximization (minimization) of $\Psi$ difficult.

*Remark:* The formulation of the worst scenario problem (7) (or its modification (8)) is identical to the formulation of other already established problems; take optimization problems or parameter identification problems. Indeed, for the latter, where a desirable output $u_{\text{given}}$ is known on a domain $\Omega$, the goal could be to minimize

$$\Psi(a) = \int_\Omega (u(a) - u_{\text{given}})^2 \, \mathrm{d}x,$$

over $\mathcal{U}_{\mathsf{ad}}$, for instance. □

The method delivers the guaranteed range of $\Psi|_{\mathcal{U}_{\mathsf{ad}}}$, which fits to the concept of the guaranteed prediction, see Section 1. The WSM does not use weighted data, which eliminates the difficulties associated with the determination of weights. On the other hand, the method neglects the fact that the occurrence of extremal values of $\Psi$ is rare in many practical problems.

*Remark:* The worst scenario approach is also possible in stochastic problems. Take $\mathcal{U}_{\mathsf{ad}}$ representing a set of admissible cumulative distribution functions, for instance. The extremal probabilistic features of $\Psi$ are then in the focus of the analysis; see [7].

### 3.2 Dempster-Shafer evidence theory

In this approach, the entire sets of input data are weighted. The weights, though resembling probability measures, are defined less strictly, however.

Let us introduce the elements of the evidence theory; a more detailed treatment can be found in [1, 6, 8, 20].

Let $X$ be a universal set and $P_X$ be the power set of $X$. A map $m : P_X \to [0, 1]$, called the basic probability assignment, is defined. It holds $m(\emptyset) = 0$ and $\sum_{A \in P_X} m(A) = 1$. For simplicity, we assume that $m(A_i) > 0$ only for a finite number of sets $A_i \in P_X$, where $i = 1, 2, \ldots, k$; these sets $A_i$ are called *focal elements*. We can interpret $m(A_i)$ as the weight associated with $A_i$; see Figure 2 (left).

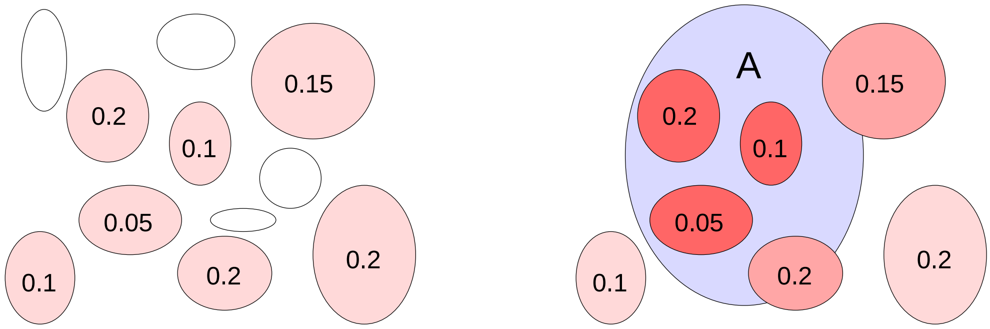


**Fig. 2:** *Focal elements and their weights (left). Focal elements and a set $A$ (right).*

Next, two functions are defined on $P_X$, namely *belief* and *plausibility*

$$\mathrm{Bel}(A) = \sum_{A_i \subseteq A} m(A_i), \quad \mathrm{Pl}(A) = \sum_{A_i \bigcap A \neq \emptyset} m(A_i), \quad A \in P_X. \tag{10}$$

For the set $A$ and the focal elements depicted in Figure 2, we obtain

$$\mathrm{Bel}(A) = 0.2 + 0.1 + 0.05 = 0.35, \quad \mathrm{Pl}(A) = 0.2 + 0.1 + 0.05 + 0.15 + 0.2 = 0.7.$$

Various interpretations of Bel and Pl can be found in [15]: Pl($A$) is the largest probability for $A$ that is consistent with all available evidence, Bel($A$) is the smallest probability for $A$ that is consistent with all available evidence. Or, Pl is an upper limit and Bel is a lower limit on the strength of evidence at hand.

The latter interpretation is close to our weight-oriented perspective. Indeed, we can interpret Pl($A$) (Bel($A$)) as an upper (lower) weight $A$ can have, given the weights of focal elements.

Our ultimate goal is to infer weights attributed to (at least) some sets of values produced by the criterion functional $\Psi$. In other words, we have to establish a set of focal elements in the space of the quantity of interest. To achieve this goal, we employ the worst scenario method.

Let $A_i$, where $i = 1, \dots, N$, be the focal elements of a probability assignment $m$ in the space of model inputs. Let us interpret each $A_i$ as an admissible set and calculate the intervals $\Psi|_{A_i}$, where $i = 1, \dots, N$; see (7)-(9).

These intervals can serve as focal elements, but we have to take into consideration that more than one admissible set can be mapped to one interval. For the quantity of interest, basic probability assignment $m_\Psi$ and its focal elements $I_\Psi^k$ are defined as follows (extension principle)

$$m_\Psi(I_\Psi^k) = \sum_{\{i:\ I_\Psi^k = \Psi|_{A_i}\}} m(A_i), \quad k = 1, \dots, K, \tag{11}$$

where $K$ is the total number of different intervals $\Psi|_{A_i}$.

Once $m_\Psi$, the basic probability assignment in the space of the quantity of interest, is established, the relationship between the focal elements $I_\Psi^k$ and various sets can be assessed through Bel and Pl.

*Example:* Let us imagine that some input parameters of a computational model of a structure are not given as crisp numbers but they are known only through inexact measurements performed by four groups of students. The measurements have resulted in four data sets denoted by $A_1, \ldots, A_4$. Since the groups do not share the same level of experience, the credibility of their results is also different, which is represented by weights $m_i$ that we attribute to $A_i$, where $i = 1, 2, 3, 4$.

The range of a quantity of interest $\Psi$ is calculated for each $A_i$; let

$$\Psi|_{A_1} = [5, 7.5], \quad \Psi|_{A_2} = [6, 10], \quad \Psi|_{A_3} = [6, 10], \quad \Psi|_{A_4} = [9.5, 12]. \tag{12}$$

By applying (11) to (12), we obtain three focal elements and their probability assignment, i.e.,

$$I_\Psi^1 = [5, 7.5], \quad I_\Psi^2 = [6, 10], \quad I_\Psi^3 = [9.5, 12]; \tag{13}$$

$$m_\Psi(I_\Psi^1) = m_1, \quad m_\Psi(I_\Psi^2) = m_2 + m_3, \quad m_\Psi(I_\Psi^3) = m_4. \tag{14}$$

To get some insight into the behavior of the quantity of interest, we will use $m_\Psi$ to calculate Bel and Pl for various intervals of length 3.

To this end, let us consider $x \in [1, 13]$ and define two functions

$$f_{\text{Bel}}(x) = \text{Bel}([x, x+3]), \quad f_{\text{Pl}}(x) = \text{Pl}([x, x+3]).$$

It is, for instance, $f_{\text{Pl}}(4.9) = \text{Pl}([4.9, 7.9]) = m_\Psi(I_\Psi^1) + m_\Psi(I_\Psi^2) = m_1 + m_2 + m_3$.

In Figure 3 ($m_1 = 0.2$, $m_2 = 0.1$, $m_3 = 0.25$, and $m_4 = 0.45$), two clusters of intervals attract our attention. We observe that intervals $[x, x+3]$ determined by $x \in [6.5, 7.5]$ have plausibility equal to 1 but zero belief and that intervals $[x, x+3]$

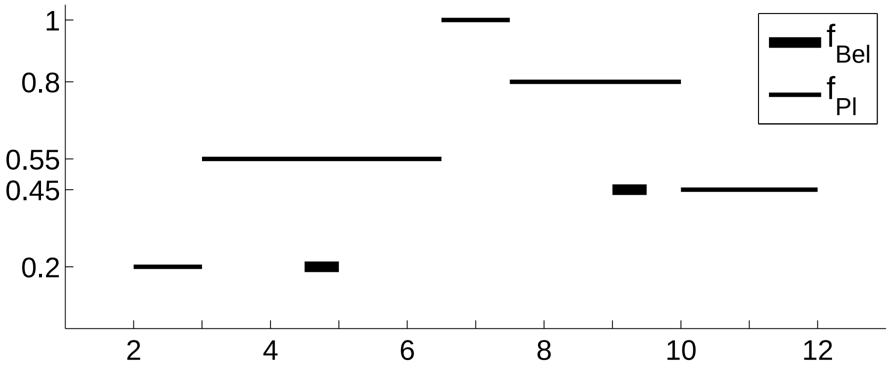


**Fig. 3:** *The graphs of $f_{Bel}$ and $f_{Pl}$.*

determined by $x \in [9, 9.5]$ have plausibility 0.8 and their belief value is equal to 0.45. The intervals from the former set can be in agreement with the quantity of interest "hidden" in the intervals $I^i_\Psi$ but no agreement is guaranteed. The intervals from the latter set cannot comply with all $I^i_\Psi$ but at least partial correspondence is guaranteed. □

It is worth noting that the Dempster-Shafer theory includes rules of combination that allow to combine two different probability assignments. In other words, different opinions of different experts can be combined into one assessment; see [1, 19], for instance. Although fruitful in uncertainty analysis, the combination fails if the included opinions are too different [19].

### 3.3 Fuzzy set theory

A fuzzy set $U \subset Z$, where $Z$ is a basic set, is identified through $\mu_U$, the *membership function*

$$\mu_U : Z \to [0, 1],$$

where the real value in $[0, 1]$ represents the degree to which $z \in Z$ belongs to the set $U$. The higher the value, the stronger the membership. For the theory and applications, see [1, 6, 9, 23].

For our purposes and in accordance with our weight-oriented standpoint, we assume that an admissible set $\mathcal{U}_{\mathsf{ad}}$ is given together with its membership function $\mu_{\mathcal{U}_{\mathsf{ad}}} : \mathcal{U}_{\mathsf{ad}} \to [0, 1]$ and that if $\mu_{\mathcal{U}_{\mathsf{ad}}}(x) = 0$, then $x \in \partial\mathcal{U}_{\mathsf{ad}}$ where $\partial\mathcal{U}_{\mathsf{ad}}$ is the boundary of $\mathcal{U}_{\mathsf{ad}}$. As in the worst scenario section, it is advantageous to assume that $\mathcal{U}_{\mathsf{ad}}$ is a compact and convex subset of a Banach space. We assume, moreover, that $\mu_{\mathcal{U}_{\mathsf{ad}}}$ is a continuous and concave function on $\mathcal{U}_{\mathsf{ad}}$.

For $\alpha \in [0, 1]$, a subset ${}^\alpha\mathcal{U}_{\mathsf{ad}}$ comprising all $x \in \mathcal{U}_{\mathsf{ad}}$ such that $\mu_{\mathcal{U}_{\mathsf{ad}}}(x) \geq \alpha$ is called $\alpha$-cut. Obviously $\mathcal{U}_{\mathsf{ad}} \equiv {}^0\mathcal{U}_{\mathsf{ad}}$.

By knowing the $\alpha$-cuts of a fuzzy set, we are able to restore its membership function, which is the goal we want to achieve in the set $I_\Psi = \{\Psi(a) |\ a \in \mathcal{U}_{\mathsf{ad}}\}$ that is fuzzy due to the fuzziness of $\mathcal{U}_{\mathsf{ad}}$.

To this end, we temporarily fix $\alpha \in [0, 1]$ and, by employing the best and the worst scenario (see (7)-(9)), we determine

$$ {}^\alpha I_\Psi = [\Psi(a_{0,\alpha}), \Psi(a^{0,\alpha})], \tag{15}$$

where ${}^\alpha I_\Psi$ is the $\alpha$-cut of $I_\Psi \equiv {}^0 I_\Psi$.

By computing (15) for all $\alpha \in [0, 1]$, we can construct the membership function $\mu_{I_\Psi}$ through

$$\mu_{I_\Psi}(y) = \max\{\alpha |\ y \in {}^\alpha I_\Psi\}, \quad y \in I_\Psi,$$

to asses the fuzziness of $\Psi$, the quantity of interest. In practice, of course, ${}^\alpha I_\Psi$ is found only for a finite number of input $\alpha$-cuts and only an approximation of $\mu_{I_\Psi}$ is obtained.

## 4 Concluding remarks

Uncertainty propagation analysis fits into the framework of decision analysis. The analyst has a variety of approaches at his or her disposal. Their applicability depends on what is known about input data, what kind of uncertainty is most relevant to the available data and to the problem under investigation.

In practice, the weights attributed to input data are often uncertain because they originate from a limited number of (inaccurate) measurements, expert opinions, hypotheses, and estimates. As a consequence, the validity of uncertainty analysis results is a delicate matter.

Although ignored in this paper, sensitivity analysis is an important part of uncertainty analysis. The goal of sensitivity analysis is to assess the influence of a (small) change in inputs on the quantity of interest. It helps to identify input parameters that have a weak influence on the value of the quantity of interest; these parameters can be excluded from the uncertainty analysis. Sensitivity analysis is also beneficial in algorithms searching for the minimum or maximum of a quantity of interest.

The worst scenario method can be used as a stand-alone method. It is also an integral part of other methods as we observed in the sections on the Dempster-Shafer theory and fuzzy set theory. Moreover, the WSM shares many features with optimal shape design, PDE constrained optimization, and inverse problems, for instance. As a consequence, various well-tried tools for theoretical as well as computational analysis are at our disposal. However, their tailoring for uncertain input data problems is desirable.

## References

[1] Ayyub, B.M. and Klir, G.J.: *Uncertainty modeling and analysis in engineering and the sciences.* Chapman & Hall/CRC, Taylor & Francis Group, Boca Raton, 2006.

[2] Babuška, I. and Chatzipantelidis, P.: On solving elliptic stochastic partial differential equations. Comput. Methods Appl. Mech. Engrg. **191** (2002), 4093–4122.

[3] Babuška, I. and Liu, K.M.: On solving stochastic initial-value differential equations. Math. Models Methods Appl. Sci. **13** (2003), 715–745.

[4] Babuška, I., Liu, K.M., and Tempone, R.: Solving stochastic partial differential equations based on the experimental data. Math. Models Methods Appl. Sci. **13** (2003), 415–444.

[5] Babuška, I., Tempone, R., and Zouraris, G.E.: Solving elliptic boundary value problems with uncertain coefficients by the finite element method: The stochastic formulation. Comput. Methods Appl. Mech. Engrg. **194** (2005), 1251–1294.

[6] Bernardini, A.: What are the random and fuzzy sets and how to use them for uncertainty modelling in engineering systems? In: I. Elishakoff (Ed.), *Whys and hows in uncertainty modelling, probability, fuzziness and anti-optimization.* CISM Courses and Lectures No. 388, Springer–Verlag, Wien, New York, 1999.

[7] Chleboun, J.: An approach to the Sandia workshop static frame challenge problem: A combination of elementary probabilistic, fuzzy set, and worst scenario tools. Comput. Methods Appl. Mech. Eng. **197** (2008), 2500–2516.

[8] Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Ann. Math. Stat. **38** (1967), 325–339.

[9] Dubois, D. and Prade, H. (Eds.): *Fundamentals of fuzzy sets. Foreword by Lotfi Zadeh*, *The Handbooks of Fuzzy Sets Series*, vol. 7. Kluwer Academic Publishers, Dordrecht, 2000.

[10] Elishakoff, I.: An idea of the uncertainty triangle. Shock Vib. Dig. **22** (1990), 1.

[11] Ghanem, R. and Red-Horse, J.: Propagation of probabilistic uncertainty in complex physical systems using a stochastic finite element approach. Physica D **133** (1999), 137–144.

[12] Ghanem, R.G. and Spanos, P.D.: *Stochastic finite elements: A spectral approach.* Springer–Verlag, New York, 1991.

[13] Hlaváček, I., Chleboun, J., and Babuška, I.: *Uncertain input data problems and the worst scenario method*, *North-Holland Series in Applied Mathematics and Mechanics*, vol. 46. Elsevier, Amsterdam, 2004.

[14] Madras, N.: *Lectures on Monte Carlo methods*, *Fields Institute Monographs*, vol. 16. American Mathematical Society, Providence, RI, 2002.

[15] Oberkampf, W.L., Helton, J.C., and Sentz, K.: Mathematical representation of uncertainty. Research Article AIAA 2001-1645, American Institute of Aeronautics and Astronautics, Reston, VA, 2001.

[16] Oberkampf, W.L. and Trucano, T.G.: Verification and validation in computational fluid dynamics. Progress in Aerospace Sciences **38** (2002), 209–272.

[17] Roache, P.J.: *Verification and validation in computational science and engineering.* Hermosa, Albuquerque, 1998.

[18] Rubinstein, R.Y. and Kroese, D.P.: *Simulation and the Monte Carlo method.* John Wiley & Sons, Hoboken, 2008, second edition.

[19] Sentz, K. and Ferson, S.: Combination of evidence in Dempster-Shafer theory. Sandia Report SAND2002-0835, Sandia National Laboratories, Albuquerque, New Mexico, 2002.

[20] Shafer, G.: *A mathematical theory of evidence.* Princeton University Press, Princeton, NJ, 1976.

[21] Sobol′, I.M.: *A primer for the Monte Carlo method.* CRC Press, Boca Raton, FL, 1994.

[22] Xiu, D. and Karniadakis, G.E.: The Weiner-Askey polynomial chaos for stochastic differential equations. Tech. Rep., Brown University, Providence, RI, 2003. Available on the Internet.

[23] Zimmermann, H.J.: *Fuzzy set theory — and its applications.* Kluwer Academic Publishers, Boston, 2001, fourth edition.

# A NONLINEAR SYSTEM OF DIFFERENTIAL EQUATIONS WITH DISTRIBUTED DELAYS

Pavol Chocholatý

**Abstract**

It is well-known that the environments of most natural populations change with time and that such changes induce variation in the growth characteristics of population which is often modelled by delay differential equations, usually with time-varying delay. The purpose of this article is to derive a numerical solution of the delay differential system with continuously distributed delays based on a composition of $p$-step methods ($p = 1, 2, 3, 4, 5$) and quadrature formulas. Some numerical results are presented compared to the known ones.

## 1 Introduction

Delay differential equations (DDE), also called functional differential equations, time-delay systems, are widely used for describing and mathematical modeling of various processes and systems in various applied problems. Theoretical aspects of DDE theory are elaborate with almost the same completeness as corresponding parts of ordinary differential equations (ODE) theory. However, unlike ODE, even for linear DDE there are no general methods of finding solutions in explicit forms. So elaboration of numerical methods for DDE is a very important problem. Presently, various specific numerical methods are constructed for solving specific DDEs. Most of investigations are devoted to numerical methods for systems with discrete delays and Volterra integro-differential equations, see e.g. [2]. In the framework of such approach one can construct for DDE analogies of all known numerical methods of ODE case. Moreover the coefficients of the corresponding numerical methods are the same in ODE and DDE cases. The approach described in this paper was applied to numerical solution of delay differential equations with distributed delays.

## 2 Delay differential equations

ODE – the Cauchy problem considered is

$$\begin{aligned} &x'(t) = f(t, x(t)), \quad t \geq t_0, \\ &x(t_0) = x_0, \end{aligned}$$

where $f$ is a nonlinear function, assumed to be Lipschitz continuous in $x$ and $x_0$ is a given initial value. We consider a single equation because simpler notation can be used in this case. Everything in this paper can be generalized to systems of differential equations in a straightforward manner. There is a variety of applications which are more naturally modelled as functional differential equations rather then ODEs. In such equations dependent variables are concurrently evaluated at more then one value of the independent variable.

A generic form for such equations considered here is

$$x'(t) = f(t, x(t), x(\omega_1(t)), x(\omega_2(t))), \quad t \geq t_0$$

where $\omega_1(t) < t < \omega_2(t)$. When there are $\omega_1(t)$ terms (corresponding to so-called delays), but no $\omega_2(t)$ terms (corresponding to so-called advances), then the functional differential equation is called a DDE.

DDE – the Cauchy problem considered is

$$\begin{aligned} x'(t) &= f(t, x(t+\tau_1), \ldots, x(t+\tau_k)), && t \geq t_0, \\ x(t) &= \Psi(t), && t \leq t_0, \end{aligned}$$

$f$ is a function where $t$ is the independent variable (usually considered as time), dependent variable $x(t)$ is a phase vector, and dependent variable $x(t+\tau_i)$, $\tau_i \in \langle -r, 0 \rangle$, $i = 1, 2, \ldots, k$ is the function which characterizes an influence of the pre-history of the phase vector on the dynamics of the system. A class of DDE with constant delay $\tau_i$, $i = 1, 2, \ldots, k$ is called DDEs with discrete delay. Supposed that delay $\tau_i = \tau_i(t)$ we speak about differential equations with discrete time-varying delay. One can see, it is insufficient to know only the initial value to define the phase vector $x(t)$, but it is also necessary to define an initial function (initial pre-history) $\Psi(t)$. So, DDEs are generalizations of ODEs when the velocity $x'(t)$ of a process depends also on the pre-history $x(t+\tau_i)$.

Delay can also be distributed

$$x'(t) = f(t, x(t), \int\limits_{\tau(t)}^{0} \alpha(t, s, x(t+s)) \mathrm{d}s).$$

The Volterra integro-differential equations

$$x'(t) = f(t, x(t), \int\limits_{0}^{t} \beta(t, s, x(s)) \mathrm{d}s)$$

represent a special class of DDE with distributed delays. However, in practical models distributed delays occur rather than concentrated one. Here, we study an equation which includes as special cases logistic equations with both "concentrated" and "continuous" delays.

Let us consider some of them, the delay logistic equation

$$x'(t) = r(t)x(t)\left(1 - \frac{x(g(t))}{K}\right), \quad g(t) \leq t$$

describes a delay population model and is known as Hutchinson's equation, if $r$ and $K$ are positive constants and function $g(t) = t + \tau$ for negative constant $\tau$. The oscillation of solutions of this equation was studied by Gopalsamy and Zhang [4]. It is well-known that the environments of most natural populations change with time and that such changes induce variation in the growth characteristics of populations. For instance, favourable weather conditions stimulate an increase in the body size an reproduction while unfavourable environments can lead to a decline in the birth rate and increase in mortality. Temporal variations of an environment of a population are usually incorporated in model systems by the introduction of time-dependent parameters in governing equations. The reader is referred to the monograph of Gopalsamy [1] for an extensive discussion of multispecies dynamics in temporally uniform environments governed by autonomous differential equations with continuously distributed delays.

Here, we conclude the Lotka-Volterra-like predator-prey model, which is a system of two delay differential equations with distributed delay. This system is frequently used to describe the dynamics of biological systems in which two species interact, one a predator $x_1(t)$ and one its prey $x_2(t)$

$$x_1'(t) = \left[ c - k_1 x_2(t) - \int_{-\tau}^{0} \alpha_1(x_2(t+s))\mathrm{d}s \right] x_1(t)$$

$$x_2'(t) = \left[ -c + k_2 x_1(t) - \int_{-\tau}^{0} \alpha_2(x_1(t+s))\mathrm{d}s \right] x_2(t)$$

where $x_1'(t)$ and $x_2'(t)$ represent the growth of the two populations against time, $c$, $k_i$, $\alpha_i$ are parameters representing the interaction of the two species.

Also, one of the models for human immunodeficiency virus (HIV) in a homogeneously mixed single-gender group with distributed waiting times can be described using equations with distributed delay. Such DDEs with distributed delay arise in a number of other scientific applications. In general, it is difficult to obtain solutions of such equations for arbitrary choices of parameters. We usually resort to a numerical method for obtaining an approximate solution of the problem. And we must obtain classes of numerical methods for a specific choice of the parameters. In [3], Kim and Pimenov have proposed an exact solution to a system of DDEs with distributed delay. Then, by considering the maximum absolute errors in the solution at grid points and tabulated in tables for different choices of step size, we can conclude how further presented approaches produce accurate results in comparison with those exact ones.

A solution $(x_1(t), x_2(t))^{\mathrm{T}}$ of a nonlinear system of two DDEs with distributed delays

$$x_1'(t) = -\frac{1}{2}\int\limits_{-\pi}^{0} x_1(t+s)\mathrm{d}s + \frac{2x_1(t) - \frac{\pi}{2}x_2(t)}{\sqrt{x_1^2(t)+x_2^2(t)}}$$

$$x_2'(t) = -\frac{1}{2}\int\limits_{-\pi}^{0} x_2(t+s)\mathrm{d}s + \frac{2x_2(t) + \frac{\pi}{2}x_1(t)}{\sqrt{x_1^2(t)+x_2^2(t)}}$$

corresponding to an initial function

$$\begin{matrix} \Psi_1(s) = (1+s)\cos(1+s) \\ \Psi_2(s) = (1+s)\sin(1+s) \end{matrix}, \qquad \text{for} \quad -\pi \le s \le 0,$$

and an initial time $t_0 = 1$, has the form

$$\begin{matrix} x_1(t) = t\cos(t) \\ x_2(t) = t\sin(t) \end{matrix}, \qquad \text{for} \ \ t \ge 1.$$

## 3 A numerical approach

The most popular numerical approaches for solving Cauchy problem of ODEs are called finite difference methods. Approximate values are obtained for the solution at a set of grid points $\{t_i : i = 0, 1, 2, \dots, N\}$ and the approximate value at each point $t_{i+1}$ is obtained by using some of values obtained in previous steps. Single-step methods for solving ODEs require only a knowledge of the numerical solution at the point $t_i$ in order to compute the next value at the point $t_{i+1}$. The best known one-step methods are Euler's methods (explicit, implicit), trapezoidal method and higher-order Runge-Kutta methods. This has obvious advantages over the $p$-step method that use several past values computed at the points $t_{i-p+1}\dots, t_i$. And $p$-step methods are used to produce predictor-corrector algorithms known as Milne's method, Milne-Simpson's method and higher-order Adams-Moulton methods.

Now, we analyze numerical methods for evaluating definite integrals $I(f) = \int_a^b f(t)\mathrm{d}t$. Most such integrals cannot be evaluated explicitly and with many others it is often faster to integrate them numerically rather than evaluating them exactly using a complicated antiderivative of $f(t)$. There are many numerical methods for evaluating $I(f)$, but most can be made to fit within the following framework. For integrand $f(t)$, find an approximating family $\{f_n : n = 1, 2, \dots\}$ and define $I_n(f) = \int_a^b f_n(t)\mathrm{d}t = I(f_n)$ and we usually require the approximations $f_n(t)$ to satisfy $\|f - f_n\|_\infty \to 0$ as $n \to \infty$. Most numerical integration formulas are based on defining $f_n(t)$ by using polynomial or piecewise polynomial interpolation. Formulas using such interpolation with evenly spaced grid points are the composite trapezoidal rule and the composite Simpson's rule, which are the first two cases $(k = 1, 2)$ of the Newton-Cotes integration formula. The complete formula for $k = 3$ is called the composite three-eights rule. Integration formulas in which one or both endpoints are missing are called open Newton-Cotes formulas, and the previous formulas are called closed formulas. Each Newton-Cotes formula ($k = 1, 2, 3$) can be used to construct a composite method with mentioned $p$-step methods ($p = 1, 2, 3, 4, 5$). The next question of interest is whether the obtained approximate values for the solution of our two DDEs with distributed delays converges to the exact ones.

So, the simplest way to solve our model equation

$$x'(t) = I + F(x(t)),$$

with initial function $\Psi(s)$, by applying Simpson's rule and explicit Euler's methods is outlined by the following algorithm:

```
h=pi/N;
for i=1:N+1
x{i}= Ψ(-pi+(i-1)*h);
end
for i=N+1:N+N*7
if(mod(i,2) ≈=0)
I=0;
for j=i-N:i-2
if(mod(j,2) ≈=0)
SUM=(h/3)*(x{j}+4*x{j+1}+x{j+2});
I=I+SUM;
end
end
end
if(mod(i,2) ==0)
I=0;
for j=i-N:i-2
if(mod(j,2) ==0)
SUM=(h/3)*(x{j}+4*x{j+1}+x{j+2});
I=I+SUM;
end
end
end
x{i+1}=x{i}+h*(I+F(x{i}));
end
```

## 4 Numerical experiments

In order to test the viability of the proposed composite methods and to demonstrate its convergence computationally we have considered several tests with some steps, to assess the convergence property and efficiency of methods developed in Section 3.

For instance, the idea is to calculate the numerical solution by Milne's predictor-corrector method with composite Simpson rule on an equidistant mesh $t_{i+1} - t_i = h$. We discretize the time-interval $t \in \langle 1, 1 + 7\pi \rangle$ on $N$ subintervals in order to obtain the approximate values for the solution at the grid points $t_i$. Here we are only interested in showing the errors for the solution at some grid points. To obtain an approximation of maximum errors at the endpoint, we compare the numerical solutions on two different meshes having $N$ and $10N$ subintervals, respectively, and having transition points $1 + c\pi/3$, $c = 1, 8, 14, 21$ at the same place (!irrational numbers) in both the meshes, with the exact solution of this problem. Numerical results are given in Table 1 for several values of $h$. The answers are given at only a few points, rather than at all points at which they were calculated.

This problem has been solved using our methods with different values of $N$, $(30, 300, 3000, 30000)$ and compared with exact solution. To illustrate the applicability and effectiveness of "the best" composite method obtained by Milne-Simpson's method of 5-th order and Simpson rule, we compare our results with exact ones in Table 2.

**Tab. 1**

| points | $1 + \pi/3$ | $1 + 8\pi/3$ | $1 + 14\pi/3$ | $1 + 21\pi/3$ |
|---|---|---|---|---|
| $x_1$, $h = \pi/300$ | $-0.938061642$ | $-9.386079514$ | $-15.69957230$ | $-12.67558987$ |
| $x_1$, $h = \pi/3000$ | $-0.938829650$ | $-9.369056409$ | $-15.64897895$ | $-12.44655301$ |
| $x_1$, exact | $-0.938812239$ | $-9.367137558$ | $-15.64332592$ | $-12.42217059$ |
| $x_2$, $h = \pi/300$ | 1.819802987 | 0.476277848 | 0.838558832 | $-19.34452329$ |
| $x_2$, $h = \pi/3000$ | 1.819233853 | 0.445332299 | 0.748045279 | $-19.34682070$ |
| $x_2$, exact | 1.819244182 | 0.442434527 | 0.738875399 | $-19.34638443$ |

**Tab. 2**

| grid $T(j)$ | exact $x_1$ | approx. $x_1$ | exact $x_2$ | approx. $x_2$ |
|---|---|---|---|---|
| 1 | $-0.93881224$ | $-0.93882868$ | 1.81924418 | 1.81923445 |
| 2 | 2.37949667 | 2.37914423 | $-4.61102368$ | $-4.61175244$ |
| 3 | $-3.82018110$ | $-3.81898822$ | 7.40280318 | 7.40510074 |
| 4 | 5.26086553 | 5.25835117 | $-10.19458268$ | $-10.19927812$ |
| 5 | $-6.70154995$ | $-6.69723186$ | 12.98662181 | 12.99428392 |
| 6 | 8.14223438 | 8.13562983 | $-15.77814168$ | $-15.79011818$ |
| 7 | $-9.58291881$ | $-9.57354472$ | 18.56992118 | 18.58678092 |

This table presents approximate values to the solution of our DDEs with distributed delays computed by step $h = \pi/3000$ at only a few points $T(j) = 1 + (3j - 2)\pi/3$, $j = 1, 2, 3, 4, 5, 6, 7$ rather than at all points at which they were calculated.

## References

[1] Gopalsamy, K.: *Stability and oscillation in delay differential equations of population dynamics.* Kluwer Academic Publishers, Dordrecht, Boston, London, 1992.

[2] Hairer, E., Norsett, S., and Wanner, G.: *Solving ordinary differential equations.* Nonstiff Problems. Springer, Berlin, 1987.

[3] Kim, A.V. and Pimenov, V.G.: Numerical methods for delay differential equations – application of $i$-smooth calculus. *Lecture Notes Series*, vol. 44. Seoul National University, Korea, 1999.

[4] Zhang, B. and Gopalsamy, K.: Oscillation and nonoscillation in a nonautonomous delay-logistic equation. Q. Appl. Math. **46** (1988), 267–273.

# COMPLEXITY OF THE METHOD OF AVERAGING*

Josef Dalík

**Abstract**

The general method of averaging for the superapproximation of an arbitrary partial derivative of a smooth function in a vertex $a$ of a simplicial triangulation $\mathcal{T}$ of a bounded polytopic domain in $\Re^d$ for any $d \geq 2$ is described and its complexity is analysed.

## 1 Introduction

We reserve the symbol $\mathcal{P}_d^{(m)}$ for the space of (real) polynomials in $d \geq 1$ (real) variables whose degree is less than or equal to $m$ for any $m \geq 1$, $\Omega$ for a bounded polytopic domain of dimension $d \geq 2$ and consider meshes of $\Omega$ consisting of $d$-dimensional simplices. For any simplex $T$, we put

$$h_T = \operatorname{diam}(T) \quad \text{and} \quad \varrho_T = \sup\{\operatorname{diam}(B) \,|\, B \subset T \text{ is a sphere}\}.$$

If $a$ is an inner vertex of a mesh $\mathcal{T}$ and $T_1, \ldots, T_n$ are the $\mathcal{T}$-simplices with vertex $a$ then we call $\Theta(a) = T_1 \cup \ldots \cup T_n$ a *neighbourhood* of $a$ and set $h(a) = \max\{h_{T_1}, \ldots, h_{T_n}\}$.

A *Lagrange finite element* $e = e_d^{(m)}$ of degree $m$ consists of

a) the simplex $T = \overline{a^1 \ldots a^{d+1}}$,

b) the *local space* $\mathcal{L}^{(m)}$ of restrictions of the polynomials from $\mathcal{P}_d^{(m)}$ to $T$,

c) the "set of parameters" relating the values $p(n^{i_1 \cdots i_d})$ to every $p \in \mathcal{L}^{(m)}$

$$\text{in the} \quad \begin{pmatrix} d+m \\ m \end{pmatrix} \quad \textit{nodes} \quad n^{i_1 \cdots i_d} = \sum_{j=1}^{d+1} \frac{i_j}{m} a^j$$

for the non-negative integers $i_1, \ldots, i_d$ and $i_{d+1}$ such that $i_1 + \ldots + i_{d+1} = m$. (The fractions $i_1/m, \ldots, i_{d+1}/m$ are the barycentric coordinates of the node $n^{i_1 \cdots i_d}$ in $T$.)

If $m$ is a positive integer, $T$ a $d$-dimensional simplex and $u \in C(T)$ then we denote by $\mathrm{P}_{T,m}[u]$ the $\mathcal{L}^{(m)}$–interpolant of $u$ in the nodes of $e_d^{(m)}$.

*This outcome has been achieved with the financial support of the Ministry of Education, Youth and Sports, project No. 1M0579, within activities of the CIDEAS research centre.

For any integer $m$, multiindex $\varrho$ with length $r = |\varrho|$ such that $m \geq r \geq 1$, function $u \in C^{m+2}(\overline{\Omega})$ and inner vertex $a$ of a mesh $\mathcal{T}$ it is well-known that the $\mathcal{T}$-simplices $T_1, \ldots, T_n$ with vertex $a$ satisfy

$$\frac{\partial^r(\mathrm{P}_{T_i,m}[u] - u)}{\partial x^\varrho}(a) = O\left((h_{T_i})^{m+1-r}\right).$$

The (general) method of averaging consists in the solution of the problem to construct a vector $f = [f_1, \ldots, f_n]^\top$ such that

$$\frac{\partial^r\left(f_1 \mathrm{P}_{T_1,m}[u] + \ldots + f_n \mathrm{P}_{T_n,m}[u] - u\right)}{\partial x^\varrho}(a) = O\left(h(a)^{m+2-r}\right). \tag{1}$$

The special method of averaging, related to the *special case* $d = 2, m = 1 = r$, is an old problem formulated already in [9], 1967, with the aim to get an accurate approximation of the strain tensor in the postprocessing of the elasticity problem. In many papers including [7], [10], [6], [3], various approaches to the solution of this special case are presented. They can be applied in the constructions of a posteriori error estimators of the finite element solutions of the second–order partial differential problems in the plane, see [3] and [1], in the sensitivity analysis of optimization problems and in other areas. Of course, the applicability of the solution of the general problem is essentially more extensive. A solution of an analogously general problem appeared in [8].

In Section 2, the vector $f$ satisfying (1) is shown to be the minimal 2-norm solution of a small underdetermined system of linear equations. In Section 3, we study the way in which the complexity of these linear equations depends on the given multiindex $\varrho$. In the last Section 4, the general method of averaging is applied to a concrete problem and an agreement of the order of error with (1) is illustrated numerically.

## 2 The general method of averaging

We describe the system of linear equations for the vector $f$ from (1) and conditions guaranteeing the order of error required in (1).

**Definition 1.** If $m$ is an integer, $\varrho$ a multiindex such that $m \geq r = |\varrho| \geq 1$, $a$ an inner vertex of a mesh $\mathcal{T}$ and $T_1, \ldots, T_n$ are the $\mathcal{T}$-simplices with vertex $a$ then $\mathcal{F}_{m,\varrho}(a)$ denotes the set of vectors $f = [f_1, \ldots, f_n]^\top$ satisfying

$$f_1 \frac{\partial^r \mathrm{P}_{T_1,m}[p]}{\partial x^\varrho}(a) + \ldots + f_n \frac{\partial^r \mathrm{P}_{T_n,m}[p]}{\partial x^\varrho}(a) = \frac{\partial^r p}{\partial x^\varrho}(a) \tag{2}$$

for all $p \in \mathcal{P}_d^{(m+1)}$.

**Remark 1.** If $p \in \mathcal{P}_d^{(m)}$ then $\mathrm{P}_{T_i,m}[p] = p$ for $i = 1, \ldots, n$. In this case the equation (2) is trivial when $\partial^r p / \partial x^\varrho(a) = 0$ and it is of the form

$$f_1 + \ldots + f_n = 1 \tag{3}$$

when $\partial^r p / \partial x^\varrho(a) \neq 0$. Obviously, the latter case appears for $p = x^\varrho$.

**Definition 2.** A system $\mathbf{T}$ of meshes of our domain $\Omega \subset \Re^d$ is said to be a *regular family* when the following conditions (a), (b) are satisfied.

(a) For every $\varepsilon > 0$ there is a mesh $\mathcal{T} \in \mathbf{T}$ such that $h_T < \varepsilon$ for all $T \in \mathcal{T}$.

(b) There exists a constant $\sigma$ such that $\sigma \geq h_T/\varrho_T$ for all simplices $T$ in any mesh from $\mathbf{T}$.

The following hypothesis, related to a regular family $\mathbf{T}$, parameter $m$ and to a multiindex $\varrho$ with $m \geq r = |\varrho| \geq 1$, has been proved in the special case for the regular family of triangulations consisting of triangles without obtuse inner angles in [3].

*Hypothesis* (H). There exists a constant $C_0$ such that a vector $f \in \mathcal{F}_{m,\varrho}(a)$ with the 2-norm $\|f\| \leq C_0$ can be found for every inner vertex $a$ of every mesh $\mathcal{T} \in \mathbf{T}$.

The following main statement has been proved in [4], Theorem 4.

**Theorem 1.** Let us assume that a regular family $\mathbf{T}$, an integer $m$ and a multiindex $\varrho$ such that $m \geq r = |\varrho| \geq 1$ satisfy the hypothesis (H). Then there exists a constant $C_1$ such that

$$\left| \frac{\partial^r (f_1 \mathrm{P}_{T_1,m}[u] + \ldots + f_n \mathrm{P}_{T_n,m}[u] - u)}{\partial x^\varrho}(a) \right| \leq C_1 |u|_{m+2,\infty} h(a)^{m+2-r}$$

for every function $u \in C^{m+2}(\overline{\Omega})$, all inner vertices $a$ of the meshes $\mathcal{T} \in \mathbf{T}$, the $\mathcal{T}$-simplices $T_1, \ldots, T_n$ with vertex $a$ and for the vectors $f \in \mathcal{F}_{m,\varrho}(a)$ with the property $\|f\| \leq C_0$.

Let us assume that a regular family $\mathbf{T}$, integer $m$ and a multiindex $\varrho$ such that $m \geq r = |\varrho| \geq 1$ satisfy the hypothesis (H). Then, for any inner vertex $a$ of a triangulation $\mathcal{T} \in \mathbf{T}$, the $\mathcal{T}$-simplices $T_1, \ldots, T_n$ with vertex $a$ and any function $u \in C^{m+2}(\overline{\Omega})$, the minimal 2-norm solution $f = [f_1, \ldots, f_n]^\top$ of the system of equations (2) satisfies $\|f\| \leq C_0$ and the related linear combination

$$\mathrm{G}_{m,\varrho}[u](a) \equiv f_1 \frac{\partial^r \mathrm{P}_{T_1,m}[u]}{\partial x^\varrho}(a) + \ldots + f_n \frac{\partial^r \mathrm{P}_{T_n,m}[u]}{\partial x^\varrho}(a) \tag{4}$$

approximates $\partial^r u / \partial x^\varrho(a)$ with an error $O(h(a)^{m+2-r})$ due to Theorem 1. As both sides of (2) are linear, the equations (2) for all $p \in \mathcal{P}_d^{(m+1)}$ are equivalent to the $\dim \mathcal{P}_d^{(m+1)}$ equations (2) for all $p$ from the basis

$$\begin{gathered} 1, x_1 - a_1, \ldots, x_d - a_d, (x_1 - a_1)^2, (x_1 - a_1)(x_2 - a_2), \ldots, (x_d - a_d)^2, \\ \ldots, (x_1 - a_1)^{m+1}, (x_1 - a_1)^m (x_2 - a_2), \ldots, (x_d - a_d)^{m+1}. \end{gathered} \tag{5}$$

Due to Remark 1, these equations are equivalent to the *reduced system* of

$$1 + \binom{m+d}{d-1}$$

| $m$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $d=2$ | 4 (6) | 5 (10) | 6 (15) | 7 (21) | 8 (28) |
| $d=3$ | 7 (10) | 11 (20) | 16 (35) | 22 (57) | 29 (84) |

**Tab. 1:** *The numbers of equations in the reduced systems and the dimensions of $\mathcal{P}_d^{(m+1)}$ (in brackets).*

equations consisting of the equation (3) and the equations (2) for the polynomials $p$ of degree $m+1$ from (5). In Table 1, the numbers of equations from the reduced systems are compared with the dimensions of the spaces $\mathcal{P}_d^{(m+1)}$ in brackets for $m = 1, \ldots, 5$ and $d = 2, 3$. The right-hand sides of the equations (2) for the polynomials of degree $m+1$ from (5) are equal to zero. In [3], the reduced systems of four equations in the special case are analysed completely and efficient procedures for their solution are suggested.

## 3 Complexity of the general method of averaging

Theorem 1 says that the order of error of approximation of any partial derivative of degree $r$ is proportional to the difference $m - r$ and the method of averaging increases this order from $m + 1 - r$ to $m + 2 - r$. In the special case there is $m = 1 = r$, i.e. the degree of the interpolants used on the triangles surrounding the given vertex $a$ is the least possible. The cases $m = r$ appear, among others, for the following reasons: The data necessary for the higher degree interpolants need not be available and, in the case $m = r$, the calculations of the method of averaging are most simple. In what follows, we restrict our analysis to the special case $m = r$ only. We investigate simplifications of the general method of averaging based on the following identities:

**Problem.** For a given simplex $T$ and non-zero multiindex $\varrho$ find non-zero multiindices $\sigma, \tau$ with lengths $s, t$ such that $\varrho = \sigma + \tau$ and

$$\frac{\partial^r \mathrm{P}_{T,r}[p]}{\partial x^\varrho} = \frac{\partial^s \mathrm{P}_{T,s}\left[\partial^t p/\partial x^\tau\right]}{\partial x^\sigma} \quad \forall\, p \in \mathcal{P}_d^{(r+1)}. \tag{6}$$

These identities give us the following information about the reduced systems of equations: If the multiindices $\sigma, \tau$ create a solution of the Problem then, as the partial derivatives $\partial^t p/\partial x^\tau$ of all polynomials $p$ of degree $r+1$ are just all polynomials of degree $s+1$, the system of equations (2) for all polynomials $p$ of degree $m = r+1$ is in fact the system of equations (2) for all polynomials $\partial^t p/\partial x^\tau$ of the smaller degree $m = s+1$. Hence the reduced system of $1 + \left(\begin{array}{c} r+d \\ d-1 \end{array}\right)$ equations is in fact a simpler reduced system of $1 + \left(\begin{array}{c} s+d \\ d-1 \end{array}\right)$ equations.

Identity (6) can be equivalently formulated by means of the space

$$\mathcal{Q}_T^{(r+1)} = \{q \in \mathcal{P}_d^{(r+1)} |\, \mathrm{P}_{T,r}[q] = o\}$$

in the following way.

**Theorem 2.** For all simplices $T$ and non-zero multiindices $\sigma, \tau$ with $\varrho = \sigma + \tau$, (6) is equivalent to the condition

$$\frac{\partial^s \mathrm{P}_{T,s}\left[\partial^t q / \partial x^\tau\right]}{\partial x^\sigma} = 0 \quad \forall\, q \in \mathcal{Q}_T^{(r+1)}. \tag{7}$$

*Proof.* Let us assume that the multiindices $\sigma, \tau$ satisfy condition (7) and consider a polynomial $p \in \mathcal{P}_d^{(r+1)}$. If we set $q = p - \mathrm{P}_{T,r}[p]$ then $q \in \mathcal{Q}_T^{(r+1)}$ so that $q$ satisfies (7) by assumption. But then

$$\frac{\partial^s \mathrm{P}_{T,s}\left[\partial^t p / \partial x^\tau\right]}{\partial x^\sigma} = \frac{\partial^s \mathrm{P}_{T,s}\left[\partial^t \left(\mathrm{P}_{T,r}[p] + q\right) / \partial x^\tau\right]}{\partial x^\sigma} = \frac{\partial^r \mathrm{P}_{T,r}[p]}{\partial x^\varrho}.$$

If (6) is true then we obtain (7) by inserting the polynomials $q \in \mathcal{Q}_T^{(r+1)}$ into (6).

The following solution of an analogy of our Problem in dimension $d = 1$ appears to be usefull in what follows.

**Theorem 3.** Let $r > 1$, $p \in \mathcal{P}_1^{(r+1)}$ and $a = x_0 < x_1 < \ldots < x_r = b$ be equidistant nodes. Then the Lagrange interpolant $\mathrm{P}_r[p] \in \mathcal{P}_1^{(r)}$ of $p$ in the nodes $a = x_0, x_1, \ldots, x_r = b$ and the Lagrange interpolant $\mathrm{P}_1\left[p^{(r-1)}\right] \in \mathcal{P}_1^{(1)}$ of $p^{(r-1)}$ in the nodes $a, b$ satisfy

$$\frac{d^r \mathrm{P}_r[p]}{dx^r} = \frac{1}{b-a}\int_a^b p^{(r)}(x)dx = \frac{d\mathrm{P}_1\left[p^{(r-1)}\right]}{dx}. \tag{8}$$

*Proof.* Of course,

$$\frac{d\mathrm{P}_1\left[p^{(r-1)}\right]}{dx} = \frac{p^{(r-1)}(b) - p^{(r-1)}(a)}{b-a} = \frac{1}{b-a}\int_a^b p^{(r)}(x)dx. \tag{9}$$

On the other hand, for every $x \in \langle a, b\rangle$ there is $\xi \in (a, b)$ such that

$$p(x) - \mathrm{P}_r[p](x) = \frac{p^{(r+1)}(\xi)}{(r+1)!}\,(x - x_0)(x - x_1)\ldots(x - x_r)$$

due to [2], Section 2.3. As $p \in \mathcal{P}_1^{(r+1)}$, there exists a constant $C$ such that $p^{(r+1)}(\xi) = C$ for all $\xi \in (a, b)$. This and the comparison of the $r$-th derivatives of both sides of the last identity lead to

$$\begin{aligned}
\frac{d^r \mathrm{P}_r[p]}{dx^r} &= p^{(r)}(x) - \frac{C}{(r+1)!}\left[(r+1)!\,x - r!\,(x_0 + \ldots + x_r)\right] \\
&= p^{(r)}(x) - Cx + \frac{C}{r+1}\,(x_0 + \ldots + x_r)
\end{aligned}$$

for all $x \in \langle a, b\rangle$. Integrating both sides of this identity over $\langle a, b\rangle$, dividing by $b-a$ and using the fact that $d^r \mathrm{P}_r[p]/dx^r$ is a constant, we obtain

$$\frac{d^r \mathrm{P}_r[p]}{dx^r} = \frac{1}{b-a}\int_a^b p^{(r)}(x)dx + C\left[\frac{x_0+\ldots+x_r}{r+1} - \frac{a+b}{2}\right].$$

As the nodes $a = x_0, x_1, \ldots, x_r = b$ are equidistant, this identity means

$$\frac{d^r \mathrm{P}_r[p]}{dx^r} = \frac{1}{b-a}\int_a^b p^{(r)}(x)dx.$$

**Lemma 1.** Under the assumptions of Theorem 3,

$$\frac{d^r \mathrm{P}_r[p]}{dx^r} = \frac{1}{h^r}\sum_{i=0}^{r}(-1)^{r-i}\begin{pmatrix} r \\ i \end{pmatrix} p\left(x_i\right) \quad \text{for} \quad h = \frac{b-a}{r}.$$

*Proof.* If we express the Lagrange interpolant $\mathrm{P}_r[p](x)$ in the Newton form for equidistant nodes then we obtain

$$\frac{d^r \mathrm{P}_r[p]}{dx^r} = \frac{\Delta^r p(0)}{h^r}.$$

The statement can be proved by induction using the recursive definition of the $r$-th forward difference $\Delta^r p(0)$.

In the following Theorem 4 we describe all solutions of our Problem in the special case of the partial derivatives in the variables $\xi_1, \ldots, \xi_d$ given by the directions of the catheti of the *unit simplices* $\hat{T} = \overline{a^1 \ldots a^{d+1}}$ with $a^1 = [0, 0, \ldots, 0]$, $a^2 = [1, 0, \ldots, 0]$, $\ldots$, $a^{d+1} = [0, 0, \ldots, 1]$ of the reference finite elements $\hat{e}_d^{(r)}$ with the *discretization step* $h = 1/r$. For the indices $i_1 = 0, \ldots, r$, $i_2 = 0, \ldots, r - i_1$, $\ldots$, $i_d = 0, \ldots, r - i_1 - \ldots - i_{d-1}$,

$$\hat{n}^{i_1 \ldots i_d} = [i_1 h, i_2 h, \ldots, i_d h] \tag{10}$$

are the nodes of $\hat{e}_d^{(r)}$. In Fig. 1, the black circles illustrate the nodes of the finite element $\hat{e}_2^{(r)}$.

**Theorem 4.** Let $\hat{T}$ be a unit simplex and $\varrho$ a non-zero multiindex with length $r$. The non-zero multiindices $\sigma, \tau$ of lengths $s, t$ create a solution of the Problem if and only if $\sigma = \tau \cdot s/t$.

*Proof.* Let us consider arbitrary indices

$$\begin{aligned} i_1 &= 0, \ldots, r+1, \\ i_k &= 0, \ldots, r+1-i_1-\ldots-i_{k-1} \quad \text{for} \quad k = 2, \ldots, d-1 \quad \text{and} \\ i_d &= r+1-i_1-\ldots-i_{d-1} \end{aligned} \tag{11}$$

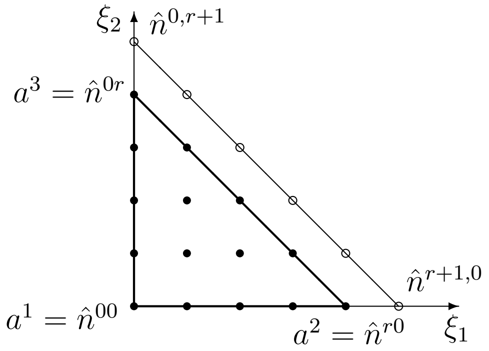


**Fig. 1:** *The nodes of the finite element* $\hat{e}_2^{(r)}$.

and set

$$f_{i_k}(\xi_k) = \textstyle\prod_{\iota=0}^{i_k-1}(\xi_k - \iota h) \quad \text{for} \quad k = 1, \dots, d,$$
$$q_{i_1 \dots i_d}(\xi_1, \dots, \xi_d) = f_{i_1}(\xi_1) \dots f_{i_d}(\xi_d). \tag{12}$$

As a matter of fact, $f_{i_k}$ is a polynomial of degree $i_k$ in the variable $\xi_k$ such that $f_{i_k}(\iota h) = 0$ for all indices $\iota$, $0 \le \iota < i_k$. Consequently, $\deg(q_{i_1 \dots i_d}) = r + 1$ and $q_{i_1 \dots i_d}$ is equal to zero in all nodes (10) of the finite element $\hat{e}_d^{(r)}$ as well as in the additional nodes $\hat{n}^{j_1 \dots j_d}$ with the indices $j_1 \dots j_d$ of the form (11) except the node $\hat{n}^{i_1 \dots i_d}$ itself. The additional nodes are indicated by the white circles in the case $d = 2$ in Fig. 1. These facts lead to the conclusion that the polynomials (12) create a basis in the space $\mathcal{Q}_{\hat{T}}^{(r+1)}$. This and the linearity of condition (7) mean that (7) is valid for all $q \in \mathcal{Q}_{\hat{T}}^{(r+1)}$ if and only if (7) is valid for the polynomials (12) related to all indices $i_1 \dots i_d$ of the form (11).

Let us now express the partial derivative from (7) for a function $q = q_{i_1 \dots i_d}$. Setting $\sigma = (\alpha_1, \dots, \alpha_d)$ and $\tau = (\beta_1, \dots, \beta_d)$, we obtain

$$\frac{\partial^t q_{i_1 \dots i_d}}{\partial \xi^\tau} = \frac{\partial^t q_{i_1 \dots i_d}}{\partial \xi_1^{\beta_1} \dots \partial \xi_d^{\beta_d}} = f_{i_1}^{(\beta_1)}(\xi_1) \dots f_{i_d}^{(\beta_d)}(\xi_d). \tag{13}$$

Observe that this derivative is different form zero if and only if

$$\beta_1 \le i_1, \dots, \beta_d \le i_d. \tag{14}$$

The next step towards the formulation of condition (7) for the functions $q_{i_1 \dots i_d}$ is to create the interpolant $\mathrm{P}_{\hat{T},s}\left[\partial^t q_{i_1 \dots i_d}/\partial \xi^\tau\right]$. We set $H = 1/s$ and, to every node $\hat{U} = \hat{N}^{u_1 \dots u_d}$ of the finite element $\hat{e}_d^{(s)}$, relate the function

$$L_{\hat{U}}^0(\xi_1, \dots, \xi_d) = F_{u_1}(\xi_1) \dots F_{u_d}(\xi_d) G_{\hat{U}}(\xi_1, \dots, \xi_d)$$

such that

$$F_{u_k}(\xi_k) \quad = \quad \prod_{\iota=0}^{u_k-1}(\xi_k - \iota H) \quad \text{for} \quad k = 1, \dots, d,$$

$$G_{\hat{U}}(\xi_1,\ldots,\xi_d) \;=\; \prod_{\iota=u_1+\ldots+u_d+1}^{s} (\iota H - \xi_1 - \ldots - \xi_d).$$

As $\deg(F_{u_1}) = u_1, \ldots, \deg(F_{u_d}) = u_d$ and $\deg(G_{\hat{U}}) = s - u_1 - \ldots - u_d$, we have $\deg(L^0_{\hat{U}}) = s$. Moreover, $L^0_{\hat{U}}(v_1,\ldots,v_d) = 0$ for every node $\hat{N}^{v_1\ldots v_d}$ of $\hat{e}^{(s)}_d$ different from $\hat{U}$. Indeed, if $v_1 + \ldots + v_d \le u_1 + \ldots + u_d$ then there exists an index $v_k < u_k$ so that $F_{u_k}(v_k) = 0$ and $G_{\hat{U}}(v_1,\ldots,v_n) = 0$ in the case $v_1 + \ldots + v_d > u_1 + \ldots + u_d$. As

$$L^0_{\hat{U}}(u_1,\ldots,u_d) = H^s u_1! \ldots u_d!(s - u_1 - \ldots - u_d)!,$$

we can see that

$$L_{\hat{U}}(\xi_1,\ldots,\xi_d) = \frac{1}{H^s u_1! \ldots u_d!(s - u_1 - \ldots - u_d)!}\, L^0_{\hat{U}}(\xi_1,\ldots,\xi_d) \tag{15}$$

is the Lagrange base function in the local space $\hat{\mathcal{L}}^{(s)} = \mathcal{P}^{(s)}_d$ of the reference finite element $\hat{e}^{(s)}_d$ related to the node $\hat{U}$. Then, due to (13),

$$\begin{aligned} &\mathrm{P}_{\hat{T},s}\left[\frac{\partial^t q_{i_1\ldots i_d}}{\partial \xi^\tau}\right] = \mathrm{P}_{\hat{T},s}\left[f^{(\beta_1)}_{i_1}(\xi_1)\ldots f^{(\beta_d)}_{i_d}(\xi_d)\right] \\ &= \sum_{u_1=0}^{s}\sum_{u_2=0}^{s-u_1}\cdots\sum_{u_d=0}^{s-u_1-\ldots-u_{d-1}} L_{\hat{U}}(\xi_1,\ldots,\xi_d)\, f^{(\beta_1)}_{i_1}(u_1 H)\ldots f^{(\beta_d)}_{i_d}(u_d H).\end{aligned}$$

In order to obtain the $\sigma$-th partial derivative of this interpolant, let us analyse the partial derivatives

$$\frac{\partial^s L_{\hat{U}}}{\partial \xi^\sigma} = \frac{\partial^s L_{\hat{U}}}{\partial \xi_1^{\alpha_1}\ldots\partial \xi_d^{\alpha_d}}. \tag{16}$$

As $\deg(L_{\hat{U}}) = s$, (16) is a constant depending on the coefficient $C$ of the maximal-order monomial $C\xi_1^{\alpha_1}\ldots\xi_d^{\alpha_d}$ of $L_{\hat{U}}$. Necessarily, this monomial is a product of the maximal-order monomials

$$\xi_1^{u_1},\ldots,\xi_d^{u_d} \tag{17}$$

from the factors $F_{u_1}$,…,$F_{u_d}$ of $L_{\hat{U}}$. But then

$$u_k \le \alpha_k \;\text{ for }\; k = 1,\ldots,d. \tag{18}$$

The nodes $[u_1 H,\ldots,u_d H]$ of the finite element $\hat{e}^{(s)}_2$ satisfying (18) are illustrated by the black circles in the case $d = 2$ in Fig. 2. A simple consideration tells us that the product of the monomials (17) with the maximal-order monomial

$$\frac{(-1)^{s-u_1-\ldots-u_d}(s - u_1 - \ldots - u_d)!}{(\alpha_1 - u_1)!\ldots(\alpha_d - u_d)!}\,\xi_1^{\alpha_1-u_1}\ldots\xi_d^{\alpha_d-u_d}$$

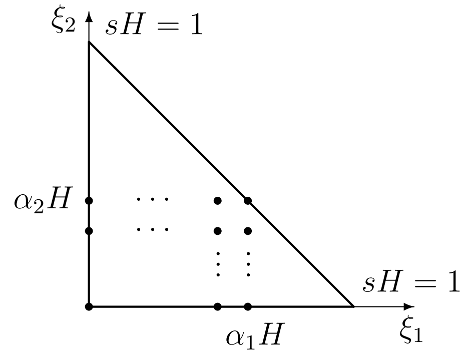


**Fig. 2:** *The nodes $[u_1H,\dots,u_dH]$ of the finite element $\hat{e}_2^{(s)}$.*

from the factor $G_{\hat{U}}$ appears in $L^0_{\hat{U}}$ and, due to (15),

$$\begin{aligned}\frac{\partial^s L_{\hat{U}}}{\partial\xi^\sigma} &= \frac{(-1)^{s-u_1-\ldots-u_d}}{H^s u_1!\ldots u_d!(\alpha_1-u_1)!\ldots(\alpha_d-u_d)!}\frac{\partial^s}{\partial\xi^\sigma}\xi_1^{\alpha_1}\ldots\xi_d^{\alpha_d}\\ &= \frac{(-1)^{s-u_1-\ldots-u_d}}{H^s}\begin{pmatrix}\alpha_1\\ u_1\end{pmatrix}\ldots\begin{pmatrix}\alpha_d\\ u_d\end{pmatrix}.\end{aligned}$$

Hence, by this result, (18) and Lemma 1, $\partial^s \mathrm{P}_{\hat{T},s}\left[\partial^t q_{i_1\ldots i_d}/\partial\xi^\tau\right]/\partial\xi^\sigma =$

$$\begin{aligned}&= \sum_{u_1=0}^{s}\sum_{u_2=0}^{s-u_1}\cdots\sum_{u_d=0}^{s-u_1-\ldots-u_{d-1}}\frac{\partial^s L_{\hat{U}}}{\partial\xi^\sigma} f_{i_1}^{(\beta_1)}(u_1H)\ldots f_{i_d}^{(\beta_d)}(u_dH)\\ &= \sum_{u_1=0}^{\alpha_1}\sum_{u_2=0}^{\alpha_2}\cdots\sum_{u_d=0}^{\alpha_d}\frac{(-1)^{s-u_1-\ldots-u_d}}{H^s}\begin{pmatrix}\alpha_1\\ u_1\end{pmatrix}\ldots\begin{pmatrix}\alpha_d\\ u_d\end{pmatrix} f_{i_1}^{(\beta_1)}(u_1H)\ldots f_{i_d}^{(\beta_d)}(u_dH)\\ &= \prod_{k=1}^{d}\frac{1}{H^{\alpha_k}}\sum_{u_k=0}^{\alpha_k}(-1)^{\alpha_k-u_k}\begin{pmatrix}\alpha_k\\ u_k\end{pmatrix} f_{i_k}^{(\beta_k)}(u_kH)\\ &= \prod_{k=1,\alpha_k>0}^{d}\frac{d^{\alpha_k}\mathrm{P}_{\alpha_k}\left[f_{i_k}^{(\beta_k)}\right]}{d\xi_k^{\alpha_k}}\prod_{k=1,\alpha_k=0}^{d} f_{i_k}^{(\beta_k)}(0). \qquad (19)\end{aligned}$$

Now, we characterize the non-zero multiindices $\sigma,\tau$ satisfying condition (7) for the polynomials $q_{i_1\ldots i_d}$ related to the indices $i_1\ldots i_d$ of the form (11). If $\deg(f_{i_k}^{(\beta_k)}) = i_k-\beta_k \le \alpha_k$ then $\mathrm{P}_{\alpha_k}\left[f_{i_k}^{(\beta_k)}\right] = f_{i_k}^{(\beta_k)}$ and

$$\frac{d^{\alpha_k}\mathrm{P}_{\alpha_k}\left[f_{i_k}^{(\beta_k)}\right]}{d\xi_k^{\alpha_k}} = f_{i_k}^{(\alpha_k+\beta_k)}.$$

Both this value for $\alpha_k>0$ and $f_{i_k}^{(\beta_k)}(0)$ for $\alpha_k=0$ is zero in the case $i_k-\beta_k<\alpha_k$ and non-zero when $i_k-\beta_k=\alpha_k$ for $k=1,\dots,d$. Hence, whenever there exists $k$ such that $i_k-\beta_k<\alpha_k$, the product (19) is zero. Let us analyse the remaining case

$$i_k-\beta_k\ge\alpha_k \quad\text{for}\quad k=1,\dots,d. \qquad (20)$$

By adding up these inequalities and using (11), we obtain $r+1-t \geq s$ or, equivalently, $s+1 \geq s$. Hence all inequalities from (20) except one are equalities and the exception is of the form $i_k - \beta_k = \alpha_k + 1$. As the factors in the product (19) related to the equalities are non-zero, (19) is equal to zero for all sequences of indices from (11) if and only if

$$\frac{d^{\alpha_k} \mathrm{P}_{\alpha_k}\left[f^{(\beta_k)}_{\alpha_k+\beta_k+1}\right]}{d\xi_k^{\alpha_k}} = 0 \quad \text{when} \quad \alpha_k > 0 \quad \text{and} \quad f^{(\beta_k)}_{\beta_k+1}(0) = 0 \quad \text{when} \quad \alpha_k = 0 \tag{21}$$

for $k = 1, \ldots, d$. In the case $\alpha_k > 0$, condition (21) is equivalent to

$$\frac{d\mathrm{P}_1\left[f^{(\alpha_k+\beta_k-1)}_{\alpha_k+\beta_k+1}\right]}{d\xi_k} = 0$$

due to Theorem 3. As $f_{\alpha_k+\beta_k+1}(\xi_k) = \prod_{\iota=0}^{\alpha_k+\beta_k}(\xi_k - \iota h) =$

$$\begin{aligned} &= \xi_k^{\alpha_k+\beta_k+1} - \frac{h}{2}(\alpha_k+\beta_k+1)(\alpha_k+\beta_k)\xi_k^{\alpha_k+\beta_k} \\ &+ \frac{h^2}{24}(\alpha_k+\beta_k+1)(\alpha_k+\beta_k)(\alpha_k+\beta_k-1)(3\alpha_k+3\beta_k+2)\xi_k^{\alpha_k+\beta_k-1} + p(\xi_k) \end{aligned}$$

for some polynomial $p$ with $\deg(p) \leq \alpha_k + \beta_k - 2$, we obtain $f^{(\alpha_k+\beta_k-1)}_{\alpha_k+\beta_k+1}(\xi_k) =$

$$= \frac{(\alpha_k+\beta_k+1)!}{2}\left[\xi_k^2 - h(\alpha_k+\beta_k)\xi_k + \frac{h^2}{12}(\alpha_k+\beta_k-1)(3\alpha_k+3\beta_k+2)\right].$$

Then

$$\begin{aligned} \frac{d\mathrm{P}_1\left[f^{(\alpha_k+\beta_k-1)}_{\alpha_k+\beta_k+1}(\xi_k)\right]}{d\xi_k} &= \frac{f^{(\alpha_k+\beta_k-1)}_{\alpha_k+\beta_k+1}(\alpha_k H) - f^{(\alpha_k+\beta_k-1)}_{\alpha_k+\beta_k+1}(0)}{\alpha_k H} \\ &= (\alpha_k+\beta_k+1)!\frac{\alpha_k H}{2}\left[\alpha_k H - (\alpha_k+\beta_k)h\right]. \end{aligned}$$

By putting $h = 1/(s+t)$ and $H = 1/s$, we can see that condition (21) is equivalent to the condition

$$\frac{(\alpha_k+\beta_k+1)!\alpha_k}{2s^2(s+t)}(\alpha_k t - \beta_k s) = 0$$

and this one is equivalent to $\alpha_k = \beta_k \cdot s/t$. In the case $\alpha_k = 0$, an evaluation of $f^{(\beta_k)}_{\beta_k+1}(0)$ tells us that the condition $f^{(\beta_k)}_{\beta_k+1}(0) = 0$ means $\beta_k = 0$.

The results obtained in both cases lead to $\sigma = \tau \cdot s/t$.

## 4 Conclusions

We formulate a corollary of Theorem 4 characterizing multiindices $\varrho$ such that our Problem has a solution on a unit simplex, illustrate the influence of the solutions of the Problem on the complexity of the method of averaging by an example and discuss some open problems.

**Definition 3.** Let $\varrho = (\gamma_1, \ldots, \gamma_d)$ be a multiindex of length $r$ and $l_\varrho$ the largest common divisor of $\gamma_1, \ldots, \gamma_d$. We call the multiindex $\varrho$ *reduced* when $l_\varrho = 1$. If $\varrho$ is non-reduced then we set $\overline{\gamma}_k = \gamma_k / l_\varrho$ for $k = 1, \ldots, d$ and say that the multiindex $\overline{\varrho} = (\overline{\gamma}_1, \ldots, \overline{\gamma}_d)$ is a *reduction* of $\varrho$ of length $\overline{r} = r/l_\varrho$.

**Corollary 1.** There exists a solution $\sigma, \tau$ of the Problem related to a $d$-dimensional unit simplex $\hat{T}$ and a non-zero multiindex $\varrho = (\gamma_1, \ldots, \gamma_d)$ if and only if $\varrho$ is non-reduced.

*Proof.* According to Theorem 4, non-zero multiindices $\sigma, \tau$ solve the Problem whenever $\sigma = \tau \cdot s/t$. As $s/t > 0$, we have $\varrho = \tau \cdot r/t$ and $r/t > 1$. Let us write $r/t = \overline{r}/\overline{t}$ so that the integers $\overline{r}, \overline{t}$ are relatively prime. Then, as $\overline{r} > \overline{t} \geq 1$ and

$$\gamma_k = \beta_k \cdot \frac{\overline{r}}{\overline{t}} \quad \text{for} \quad k = 1, \ldots, d,$$

the fractions $\beta_k / \overline{t}$ are integers for $k = 1, \ldots, d$. This and $\overline{r} > 1$ tell us that $\varrho$ is non-reduced. On the other hand, if $\varrho$ is non-reduced and $\varrho = l_\varrho \overline{\varrho}$ then the non-zero multiindices $\sigma = \overline{\varrho}, \tau = (l_\varrho - 1)\overline{\varrho}$ create a solution of the Problem.

It is an open question whether Corollary 1 can be generalized to arbitrary simplices. The statement of the following Lemma 2, see [4], Lemma 8, provides a partial positive answer to this question.

**Lemma 2.** If $r \in \{2, 3, \ldots\}$ and $k \in \{1, 2\}$ then

$$\frac{\partial^r \mathrm{P}_{T,r}(p)}{\partial x_k^r} = \frac{\partial \mathrm{P}_{T,1}(\partial^{r-1} p / \partial x_k^{r-1})}{\partial x_k}$$

for all 2-dimensional simplices $T$ and polynomials $p = p(x_1, x_2)$ of degree $r + 1$.

**Example 1.** For $u(x, y) = \ln(x^2 + 0.2y^4 + 0.5) \cdot \exp(xy - \sin(x + 2y) - 3)$ and an inner vertex $a = [0, 0]$ with the neighbours $ha^1, \ldots, ha^7$ of certain triangulations $\mathcal{T}_h$ – Fig. 3 illustrates the neighbourhood $\Theta(a) = T_1 \cup \ldots \cup T_7$ of $a$ in $\mathcal{T}_h$ – find the errors of the approximations of $\partial^3 u / \partial x^3(a)$ by means of the method of averaging with the parameters $m = 3 = r$ for such values of $h$ that $h(a) = 2^{-1}, \ldots, 2^{-8}$.

In this example, the multiindex $\varrho = (3, 0)$ is non-reduced. Setting $\sigma = \overline{\varrho} = (1, 0)$ and $\tau = (2, 0)$, we can see that the reduced systems of 6 equations in 7 unknowns indicated in Table 1 are in fact reduced systems of 4 equations in 7 unknowns due to Lemma 2. These systems are exactly the reduced systems for the superapproximation

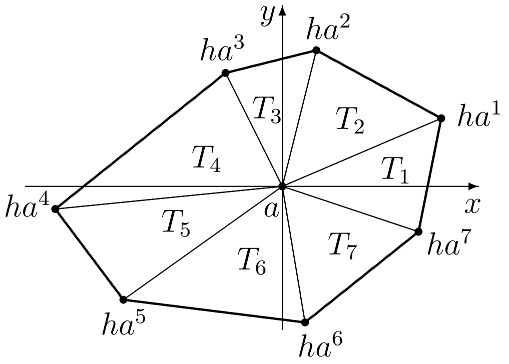


**Fig. 3:** *Neighbourhood* $\Theta(a) = T_1 \cup \ldots \cup T_7$ *of* $a$ *in* $\mathcal{T}_h$.

| $i$ | $h_i$ | $e_i$ | $\log \frac{e_i}{e_{i-1}} / \log \frac{h_i}{h_{i-1}}$ |
|---|---|---|---|
| 1 | 5E−1 | −1.57729E−1 | |
| 2 | 2.5E−1 | −4.38036E−2 | 1.84833 |
| 3 | 1.25E−1 | −1.13485E−2 | 1.94855 |
| 4 | 6.25E−2 | −2.86352E−3 | 1.98664 |
| 5 | 3.125E−2 | −7.17266E−4 | 1.99721 |
| 6 | 1.5625E−2 | −1.79366E−4 | 1.99960 |
| 7 | 7.8125E−3 | −4.48941E−5 | 1.99831 |
| 8 | 3.90625E−3 | −1.13726E−5 | 1.98096 |

**Tab. 2:** *The errors* $e_i = \partial^3 u/\partial x^3(a) - G_{3,(3,0)}[u](a)$ *and the estimates of the order of accuracy.*

of the first derivative $\partial u/\partial x(a)$ by the method of averaging with the parameters $m=1=r$. We solve these underdetermined systems of 4 equations by the Householder QR-algorithm described in [5] and use their solutions $f_1, \ldots, f_7$ in the computation of the approximation $\mathrm{G}_{3,(3,0)}[u](a)$ according to (4).

Table 2 presents the values of errors $e_i = \partial^3 u/\partial x^3(a) - \mathrm{G}_{3,(3,0)}[u](a)$ related to the parameters $h(a) = h_i = 2^{-i}$ for $i = 1, \ldots, 8$. The last column indicates that $e_i = O(h_i^2)$.

The special method of averaging ($d = 2, m = 1 = r$) has been analysed in [3] completely. On the contrary, concerning the general method, answers to many open questions would increase its applicability. Among them, besides the generalization of Corollary 1, validity of the hypothesis (H) and applicability of the method in the boundary vertices should be studied.

## References

[1] Ainsworth, M. and Oden, J.: *A posteriori error estimation in finite element analysis*. Wiley, New York, 2000.

[2] Berezin, I.S. and Shidkov, N.P.: *Numerical methods 1*. Nauka, Moscow, 1966.

[3] Dalík, J.: Averaging of directional derivatives in vertices of nonobtuse regular triangulations. Numer. Math. **116** (2010), 619–644.

[4] Dalík, J.: Approximations of the partial derivatives by averaging. Submitted for publication.

[5] Golub, G.H. and Van Loan, Ch.F.: *Matrix computations*. Third ed., The Johns Hopkins University Press, 1996.

[6] Hlaváček, I., Křížek, M. and Pištora, V.: How to recover the gradient of linear elements on nonuniform triangulations. Appl. Math. **41** (1996), 241–267.

[7] Křížek, M. and Neittaanmäki, P.: Superconvergence phenomenon in the finite element method arising from averaging gradients. Numer. Math. **45** (1984), 105–116.

[8] Zhang, Z. and Naga, A.: A new finite element gradient recovery method: superconvergence property. SIAM J. Sci. Comp. **26**, no. 4, (2005), 1192–1213.

[9] Zienkiewicz, O.C. and Cheung, Y.K.: *The finite element method in structural and continuum mechanics*. McGraw Hill, London, 1967.

[10] Zienkiewicz, O.C., Zhu, J.Z., The superconvergence patch recovery and *a posteriori* error estimates. Part 1: The recovery technique. Int. J. Numer. Methods Engrg. **33** (1992), 1331–1364.

# USAGE OF MODULAR SCISSORS IN THE IMPLEMENTATION OF FEM[*]

Dalibor Frydrych

### Abstract

Finite Element Method (*FEM*) is often perceived as a unique and compact programming subject. Despite the fact that many *FEM* implementations mention the Object Oriented Approach (*OOA*), this approach is used completely, only in minority of cases in most real-life situations. For example, one of building stones of OOA, the interface-based polymorphism, is used only rarely.

This article is focusing on the design reuse and at the same time it gives a complex view on *FEM*. The article defines basic principles of *OOA* and their use in *FEM* implementation. Using OOA *FEM* project is split in many smaller sub-projects which are interlinked together. Links between sub-projects are one way only and non circular. Such a setting gives opportunity to use the modular scissors. In addition, these individual sub-projects can be used directly, without additional adjustments, in similar projects.

## 1 Introduction

Development of computer programs have undergone significant changes in recent years. In the beginning, programming was seen as a kind of art. Programmers worked alone and quality of their work depended on their individual skills. The demand for computer programs increased significantly with rising use of computers. At the same time expectation from these programs were growing too, in parallel with the complexity of studied tasks. A new market was created and new companies emerged to develop computer programs. Due to complexity of tasks programs were usually written by several programmers. This brought along approach known, up to that date, mainly in mass production: analysis and planning standardization, quality assurance, process documentation, personal replaceability, labor efficiency and management. Development of computer programs became a full scale industry.

Process of program development is structured like other projects. Development is done by a development team. Team is managed by Project manager. A task is initially analyzed, if necessary also in depth at customer side, by an analyst. Designer is supposed to choose an appropriate software platform and language. Programmer (SW designer) is often called a "coder" and is responsible solely to write machine code using data input defined by analyst and designer.

[*] This work has been supported by Ministry of Education of the Czech Republic; under the project "Advanced remedial technologies and processes", code 1M0554.

Organizing the project development in different layers between different persons improves productivity. Team member on given level is an expert in his job. Communication between the layers is standardized by using defined documents (equivalent of technological procedures for example in industrial production). Definition of all these steps ensures personal replaceability and makes team management easier. In order to ensure that documentation is consistent in all levels it was necessary to adjust general approach to problem solving. Nowadays, the ideal approach seem to be Object Oriented Approach (*OOA*).

## 2 Object oriented approach

Common mistake committed by developers is, to use for solving of a problem the object oriented programming language, and to believe that it is *OOA*. *OOA* is mainly about giving a project right structure, than about using particular programming language. For example, one of important features, by which *OOA* can be recognized is the absence of circular associations.

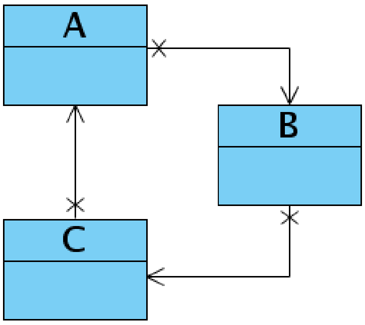


**Fig. 1:** *Class diagram of circular association.*

Circular associations are often responsible for tangling of code. Figure 1 demonstrates circular association between three classes **A**, **B** and **C**. Class diagram from Unified Modeling Language (*UML*) [2] is used for explanation. Modeling of processes, where classes are so heavily interlinked is very complicated as change in one class causes changes in other classes. During implementation, many such conflicts arise and what makes solving of problem even more difficult is the fact, that responsibility of individual classes is not clearly defined.

Solution to this problem is to cut the circular association, for example between class **C** and **A**, and make it linear. Then the responsibilities of classes are clearly defined. Implementation of class **C** is the starting point. Because class **C** is not dependent on any other class, its responsibility is clearly defined. Implementation is quick and easy. Next step is implementation of class **B**. This class is using class **C**. But class **C** is already implemented and tested, so it is ready to use. The same situation repeats in implementing of class **A**.

However, decision how to cut circular association is a crucial step and has to be done only after deep analysis of the whole system.

## 3 Analytical model of FEM

Now, it is time to describe *FEM*, very roughly and analytically, from system design point of view. Domain of task $\Omega$ is divided in particular sets of elements. Scalar products are calculated for each element. Values of individual scalar sets members depend on initial conditions, respectively on results of previous time step. Scalar products are placed in global matrix. Global matrix and the right hand side of linear equations set are modified according to boundary conditions. Set of linear equations is solved, and then interpreted as values of individual searched parameters. Then the calculation is repeated for next time step.

From the above description it is clear, that the *FEM* is element-centric method. In many implementations of *FEM*, the data structure defining an element is very big.

```
class Element {
   Node[] nodes;
   ...
   MaterialParameters[] materialParameters;
   DOF[] dofs;
   Results[] results;
   ...
}
```

It contains information about initial conditions, material properties, boundary conditions etc. Such a structure is too complicated and difficult to re-use. Defining too big structures is very common design mistake.

### 3.1 Reduced model of mesh

The key point of efficient design is making the data structure, which defines the element, smaller. The basic idea is the following: element defines only part of the task domain $\Omega$, an element needs only association to nodes.

Analytical model of mesh is then very simple, as can be seen in Figure 2. Mesh keeps only information about task domain $\Omega$, ensures the association to list of ele-

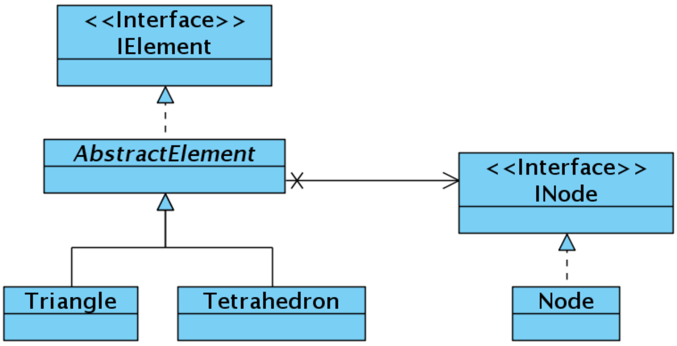


**Fig. 2:** *UML class diagram of mesh.*

ments and list of nodes. An element is a typical abstract structure. Mechanism of inheritance is used to specify concrete type of element (triangle, tetrahedron, ...). A model defined in this way has a clearly defined functionality, which can be used in future models with no need for additional modification - that is the idea of re-use.

### 3.2 Methodology $DF^2EM$

*Methodology* $DF^2EM$ (Developers Fab Finite Element Method) [5] describes individual particular models, their functionality and associations in between them, see Figure 3.

The lowest level represented by model *Tools* is dealing with system functions. The next level model *Math* is solving main mathematical parts. All these models work only with basic sets of data (int and double), eventually define their own (*Matrix*). On the next level there are three independent models:

- Model *Mesh* described above.
- Model *Material* implementing calculation with material parameters.
- Model *Scenario* solving definition of individual time steps for calculation of unsteady processes.

Model *Approx* is a database of approximation functions. Model *Local* interconnects, using association classes models *Mesh*, *Material* and *Approximation* and implement calculations of local matrices. Model *Formulation* defines basis of mathematics formulation of *FEM* (primal, mixed-hybrid, etc.). On the highest level there is model *Task*. It ensures initial phases, start up and management of the whole calculation. From Figure 3 it is clear that associations between individual models are NOT circular. To re-use any individual model, it is possible to use modular scissors. For example, to re-use model *Local*, it is necessary to take ONLY models on which this model is dependent (*Approx*, *Material*, *Mesh*, etc.) NOT all the models *Methodology* $DF^2EM$.

## 4 Implementation

Programming language *JAVA* was used for implementation. Implementation was named as *Project* $DF^2EM$ and was based on models created in *Methodology* $DF^2EM$. Interface-based polymorphism was exclusively used for implementation in order to ensure easy exchange of concrete implementation. *JAVA* language ensures the possibility to insert concrete implementation even in run-time (similar to principle of plug-in modules). User of *Project* $DF^2EM$ has therefore an option to exchange any part of implementation by a different one without need to modify the source texts *Project* $DF^2EM$.

### 4.1 Testing

An important part of each software development project is its testing. For implementation of *Project* $DF^2EM$ were used ideas of technology *Test driven development*. Division of *FEM* into small logically defined parts, enabled their thorough

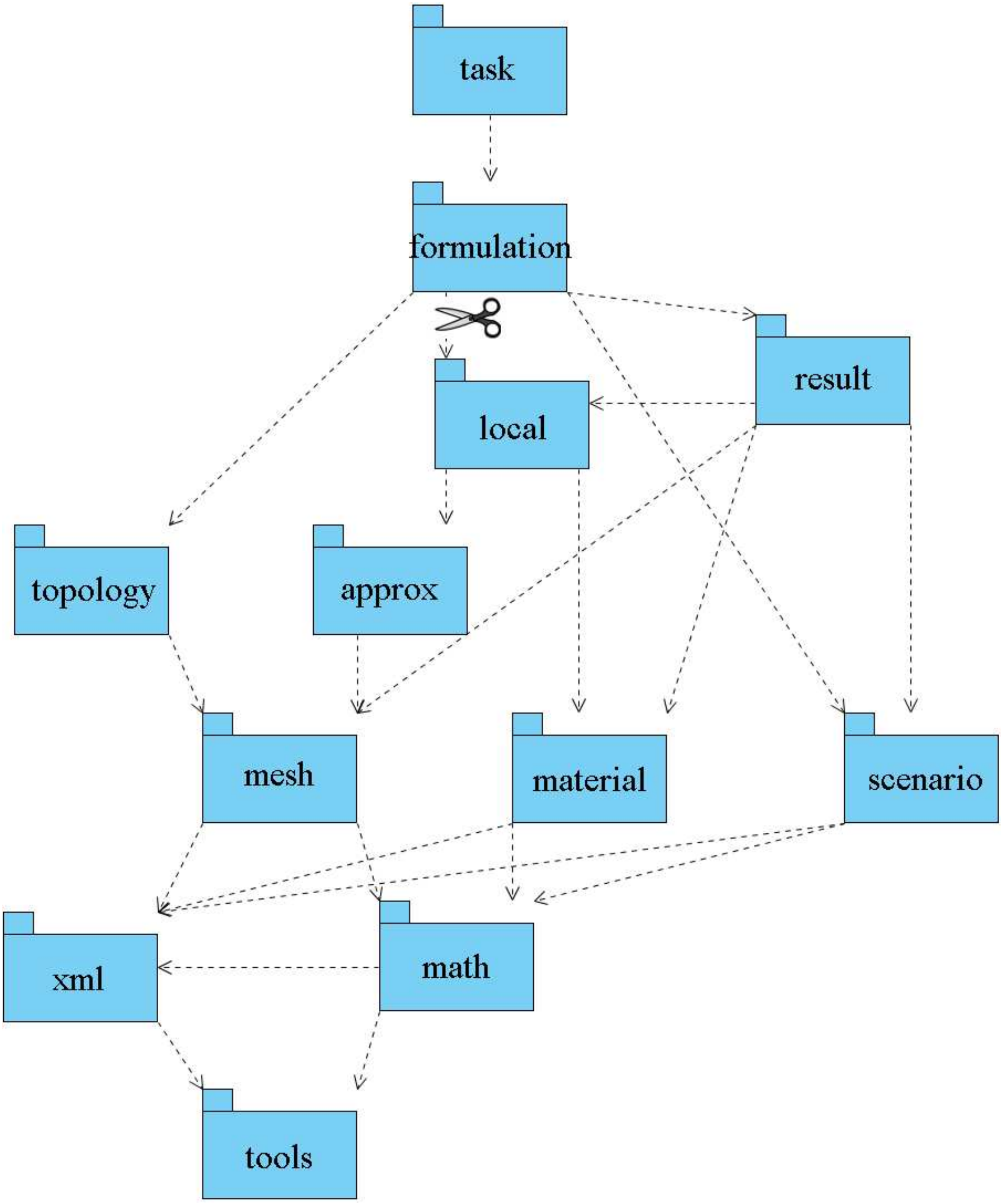


**Fig. 3:** *Schema of $DF^2EM$ models.*

testing. Framework *JUnit* [3] was used for testing. Thorough testing was possible by using support tool *Cobertura*. In perimeter of *JUnit* it is possible to create test file for each class. Then it is possible, within the class, to test all its methods. Tool *Cobertura* analyzes how all individual tests were running. Checks track of all tests passages through individual lines of source code and presents results in graphs and reports. In this way it enables very detailed testing of all the functions of individual classes.

## 5 Conclusion

This article defined a unique system of *OOA* to *FEM* called *Methodology* $DF^2EM$. Basis is *Methodology* $DF^2EM$ detailed *OOA* of *FEM*. This *OOA* is independent of used programming language. *FEM* is divided in small, logically defined modules - which are managing individual objects. Associations between modules were reduced so between them and also inside of them circular associations were not created. Avoidance of circular associations, allow as to separate individual modules for re-use, by modular scissors. Such models can be re-used in other project without additional corrections or adjustments.

*Methodology* $DF^2EM$ is not only a theoretical work with no practical use. Implementation was done in language *JAVA* and is called *Project* $DF^2EM$. *Project* $DF^2EM$ is used for implementation of several models based on *FEM*. Very good results were reached in model *ISERIT* [6].

## References

[1] Gamma, E., Helm, R., Johnson, R., and Vlissides, J.: *Design patterns: elements of reusable object-oriented software.* Addison Wesley Professional, 1994, ISBN 978-0201633610.

[2] UML©Resource Page, `http://www.uml.org`

[3] JUnit.org: Resources for Test Driven Development, `http://www.junit.org`

[4] Cobertura, `http://cobertura.sourceforge.net`

[5] Frydrych, D. and Lisal, J.: Introduction to methodology $DF^2EM$ - framework for efficient development of finite element based models. *Proceedings of ICCSA'08.* San Francisco, USA, 2008.

[6] Frydrych, D. and Hokr, M.: Verification of coupled heat and mass transfer model ISERIT by full-scale experiment. *Proceedings of ICMSC'08.* San Francisco, USA, 2008.

# ON THE WORST SCENARIO METHOD: APPLICATION TO UNCERTAIN NONLINEAR DIFFERENTIAL EQUATIONS WITH NUMERICAL EXAMPLES[*]

Petr Harasim

## 1 Introduction: The worst scenario method

A great many problems in science can be described and solved by means of suitable mathematical models. Nevertheless, since the input data of mathematical models is encumbered with various sorts of uncertainty, the output values are also uncertain. It is our goal to evaluate the uncertainty of output data if the uncertainty of input data is somehow specified.

The mentioned models are characterized by a state problem $\mathcal{P}(a, u)$, where $a$ represents input data and $u$ denotes a solution of the state problem, so called state solution. The state problem $\mathcal{P}(a, u)$ can be represented by a boundary value problem, for instance. We consider a state problem whose input data is uncertain. Thus, let $\mathcal{U}_{\mathrm{ad}}$ be a given set of admissible input data. Since the state solution $u$ depends on the input parameter $a \in \mathcal{U}_{\mathrm{ad}}$, we obtain a set of state solutions. As a rule, we are concerned with a real-valued quantity of interest related to the state solution and represented by a criterion functional $\Phi = \Phi(a, u(a))$, generally directly dependent on $a$. Due to the uncertainty of the state solution, we obtain a set of values of the criterion functional.

There exists a number of approaches to treatments of uncertainty in mathematical models. The choice of an acceptable approach depends largely on the amount of available information about the input data. If only the set of admissible input data is known, we wish to derive the corresponding set of outputs. In engineering applications, mainly large values of the quantity of interest (e.g. temperature at a selected point of a heated body, or local mechanical stress at a point of a loaded body) are important. Therefore, we search for an input parameter $a^0 \in \mathcal{U}_{\mathrm{ad}}$ such that the quantity of interest is maximal, i.e. we search for the worst scenario.

More precisely, let the state problem $\mathcal{P}(a, u)$ be given, $a \in \mathcal{U}_{\mathrm{ad}} \subset U$, $u \in V$, where $U$ and $V$are suitable Banach spaces, and let $\Phi$ be the criterion functional mentioned above. The goal is to solve the following worst scenario problem: Find $a^0 \in \mathcal{U}_{\mathrm{ad}}$ such that

$$a^0 = \arg\max_{a \in \mathcal{U}_{\mathrm{ad}}} \Phi(a, u(a)). \tag{1}$$

[*]The work was supported by the Academy of Science of the Czech Republic, Institutional Research Plan No. AV0Z 30860518. The author would like to thank Dr. J. Chleboun and Dr. P. Byczanski for their help during the work.

The existence of the solution to problem (1) can be proved via the convergence of the solutions to approximate worst scenario problems, see [2]. The approximate worst scenario problem is defined as follows: Find $a_h^{M0} \in \mathcal{U}_{\mathrm{ad}}^M$ such that

$$a_h^{M0} = \arg \max_{a^M \in \mathcal{U}_{\mathrm{ad}}^M} \Phi(a^M, u_h(a^M)), \tag{2}$$

where $\mathcal{U}_{\mathrm{ad}}^M \subset \mathcal{U}_{\mathrm{ad}}$ is a $M$-dimensional approximation of the admissible set $\mathcal{U}_{\mathrm{ad}}$, $u_h(a^M) \in V_h \subset V$ is the solution of the state problem in a finite-dimensional subspace $V_h$ of space $V$ (usually, we use a finite element space). This approach also provides a way to calculate, at least approximately, the worst scenario $a^0$ and the corresponding value $\Phi(a^0, u(a^0))$.

For a more detailed mathematical treatment of the worst scenario method, see, e.g., [1, 2, 3, 4, 5].

## 2 Application to a one dimensional nonlinear boundary value problem

### 2.1 Definition of the problem

We consider the state problem examined in [2] and motivated by a boundary value problem with an ordinary differential equation: Find $u \in H_0^1(0,1)$ such that

$$\int_0^1 a(u'^2)u'v'\mathrm{d}x = \int_0^1 fv\,\mathrm{d}x \qquad \forall v \in H_0^1(0,1), \tag{3}$$

where $H_0^1(0,1)$ is the usual Sobolev space, the function $a \in \mathcal{U}_{\mathrm{ad}}$ is an admissible coefficient, $f \in L^2(0,1)$. Let the admissible set $\mathcal{U}_{\mathrm{ad}}$ be a set of Lipschitz continuous functions $a$ defined on $\mathbb{R}_0^+$ (nonnegative real numbers) and such that

$$\begin{aligned} 0 \le \frac{\mathrm{d}a}{\mathrm{d}x} \le& C_{\mathrm{L}} && \text{a.e. in } [0, x_C], \\ a(x) =& a(x_{\mathrm{C}}) && \text{for} \quad x \ge x_{\mathrm{C}}, \\ 0 < a_{\min} \le a(x) \le& a_{\max} && \forall x \in \mathbb{R}_0^+, \end{aligned}$$

where $C_{\mathrm{L}},\ x_{\mathrm{C}},\ a_{\min},\ a_{\max}$ are positive constants such that the admissible set is not empty.

Further, let $T_j$, $j \in \{1, \ldots, M\}$, be equally spaced points in $[0, x_C]$, $T_1 = 0$ and $T_M = x_C$. We define the set $\mathcal{U}_{\mathrm{ad}}^M \subset \mathcal{U}_{\mathrm{ad}}$ of functions $a \in \mathcal{U}_{\mathrm{ad}}$ such that $a|_{[T_j, T_{j+1}]} \in P_1([T_j, T_{j+1}])$, $j \in \{1, \ldots, M-1\}$, where $P_1([T_j, T_{j+1}])$ denotes the linear polynomials on the interval $[T_j, T_{j+1}]$. Moreover, we introduce equally spaced points $x_0 = 0 < x_1 < \ldots < x_{N+1} = 1$ into interval $[0,1]$ and define $V_h \subset H_0^1(0,1)$, the space of functions continuous on $[0,1]$, linear on the interval $[x_i, x_{i+1}]$, $i = 0, \ldots, N$, and with vanishing value at 0 and 1.

### 2.2 Algorithm and numerical results

In the following section, we show a procedure to find, at least approximately, a solution of problem (2), and present some numerical results. The computations were performed in MATLAB.

At first, we set $\Psi(a) = \Phi(a, u(a))$, so that we will examine $a$-dependent functional $\Psi$ defined on $\mathcal{U}_{\text{ad}}^M$. Furthermore, the finite-dimensional admissible set $\mathcal{U}_{\text{ad}}^M$ can be identified with a compact subset $\widehat{\mathcal{U}}_{\text{ad}}^M \subset \mathbb{R}^M$, if we define

$$\begin{aligned}\widehat{\mathcal{U}}_{\text{ad}}^M = \{\alpha \in \mathbb{R}^M : \exists a \in \mathcal{U}_{\text{ad}}^M \quad \alpha &= (\alpha_1, \dots, \alpha_M) \\ &= (a(x_1), \dots, a(x_M))\},\end{aligned}$$

see also [1]. In this sense, the functional $\Psi$ is, as a matter of fact, a real function $\widehat{\Psi} = \widehat{\Psi}(\alpha)$, where $\alpha = (\alpha_1, \dots, \alpha_M) \in \widehat{\mathcal{U}}_{\text{ad}}^M$. To obtain the value of function $\widehat{\Psi}$ at any point $\alpha \in \widehat{\mathcal{U}}_{\text{ad}}^M$, it is necessary to solve the following nonlinear problem (a finite-dimensional analogy to (3)): Find $u_h \in V_h$ such that

$$\int_0^1 a(u_h'^2) u_h' v' \mathrm{d}x = \int_0^1 f v \, \mathrm{d}x \qquad \forall v \in V_h, \tag{4}$$

where $a \in \mathcal{U}_{\text{ad}}^M$, $a(x_i) = \alpha_i$, $i = 1, \dots, M$. An approximation of the solution to problem (4) is obtained by using the Kachanov method, that is, by means of a (finite) sequence of the solutions to linearized problems, more detailed treatment can be found, e.g., in [3]. Subsequently, the criterion functional $\Phi$ is evaluated. The ultimate goal is to solve the following global optimization problem arising from (2): Find $\alpha^0 \in \widehat{\mathcal{U}}_{\text{ad}}^M$ such that

$$\alpha^0 = \arg \max_{\alpha \in \widehat{\mathcal{U}}_{\text{ad}}^M} \widehat{\Psi}(\alpha).$$

To find the element $\alpha^0$ at least approximately, we use the Nelder-Mead simplex method. This method is implemented by the standard MATLAB function *fminsearch* (this algorithm requires to enter an initial point). However, to be able to solve our global optimization problem by the unconstrained optimization routine *fminsearch*, we establish a transformation $T : \mathbb{R}^M \to \widehat{\mathcal{U}}_{\text{ad}}^M$ and search for the maximum of the composite function $\widehat{\Psi} \circ T : \mathbb{R}^M \to \mathbb{R}$. In the concrete, for $x = (x_1, \dots, x_M) \in \mathbb{R}^M$ we obtain the corresponding value $T(x) = \alpha = (\alpha_1, \dots, \alpha_M) \in \widehat{\mathcal{U}}_{\text{ad}}^M$ as follows: For the first component of $\alpha$ we define

$$\alpha_1 = a_{\min} + \frac{(a_{\max} - a_{\min})(\frac{\pi}{2} + \arctan x_1)}{\pi},$$

for $\alpha_i$, $i = 2, \dots, M$, we define

$$\alpha_i = \alpha_{i-1} + \frac{K(\frac{\pi}{2} + \arctan x_i)}{\pi},$$

where $K = \min\{\frac{C_L x_C}{M-1}, a_{\max} - \alpha_{i-1}\}$.

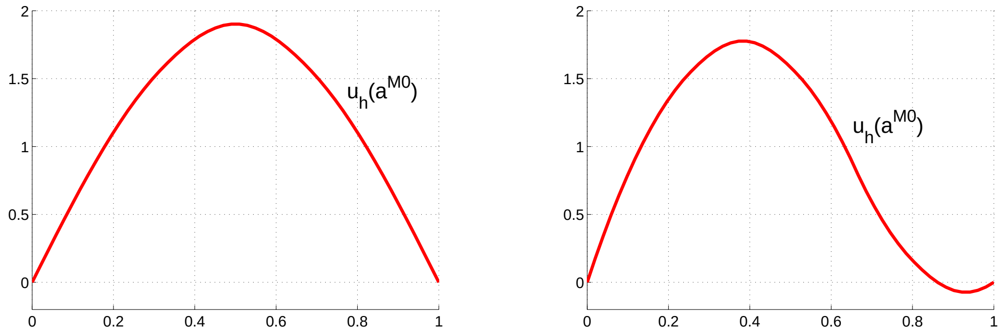


**Fig. 1:** *The state solution of the problem (4) with the parameter $a^{M0}$ and the right-hand side $f_1$ (on the left) and $f_2$ (on the right).*

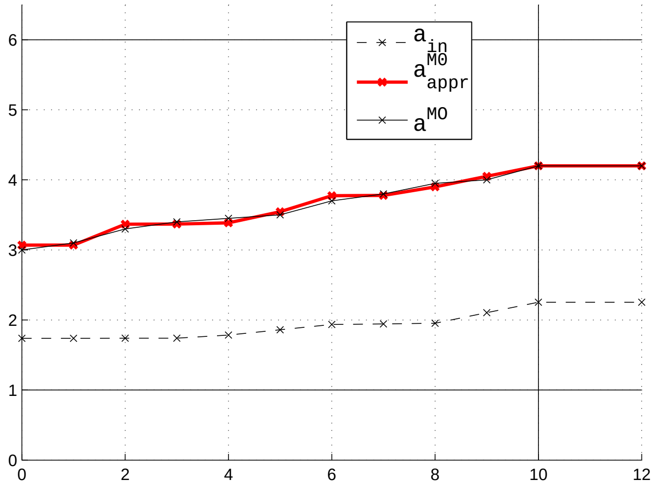


**Fig. 2:** *The approximation $a^{M0}_{\text{appr}}$ of the searched parameter $a^{M0}$ for the right-hand side $f_1$ and a given initial point $\alpha_{\text{in}} \in \widehat{\mathcal{U}}^M_{\text{ad}}$ corresponding to a parameter $a_{\text{in}} \in \mathcal{U}^M_{\text{ad}}$ ($\widehat{\Psi}(\alpha_{\text{in}}) = -1.2828 \times 10^6, \widehat{\Psi}(\alpha^{M0}_{\text{appr}}) = -0.86 \times 10^{-2}$).*

Now, we present concrete numerical examples. Let the parameters of admissible set $\mathcal{U}_{\text{ad}}$ be: $a_{\min} = 1$, $a_{\max} = 6$, $C_L = 0.3$, and $x_C = 10$. Let the dimension of $\mathcal{U}^M_{\text{ad}}$ be $M = 11$ and the dimension of the finite element space $V_h$ be $N = 50$. We solve the state problem (4) with two different right-hand sides $f_1$ and $f_2$. Concretely, $f_1(x) = 300x(1-x)$, and

$$f_2(x) = \begin{cases} 100 & \text{for} \quad 0 \le x \le \frac{2}{3} \\ -100 & \text{for} \quad \frac{2}{3} < x \le 1. \end{cases}$$

The worst scenario problem (2) is solved with the following criterion functional:

$$\Phi(a, u(a)) = -10^6 \int_0^1 [u(a) - u_h(a^{M0})]^2 \,\mathrm{d}x,$$

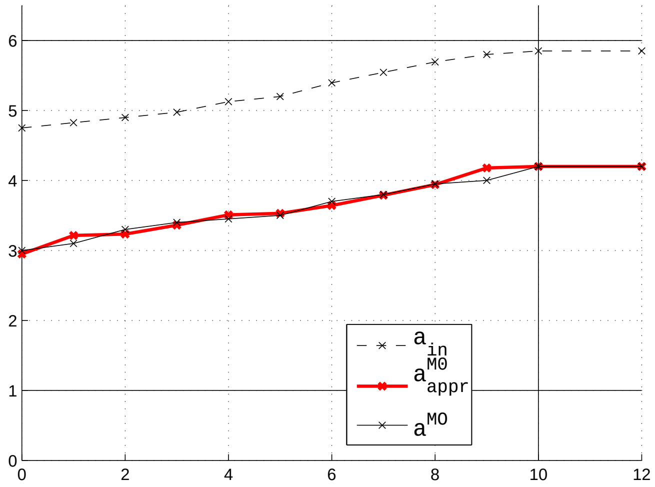


**Fig. 3:** *The approximation* $a^{M0}_{\mathrm{appr}}$ *of the searched parameter* $a^{M0}$ *for the right-hand side* $f_2$ *and a given initial point* $\alpha_{\mathrm{in}} \in \widehat{\mathcal{U}}^M_{\mathrm{ad}}$ *corresponding to a parameter* $a_{\mathrm{in}} \in \mathcal{U}^M_{\mathrm{ad}}$ $(\widehat{\Psi}(\alpha_{\mathrm{in}}) = -9.7035 \times 10^4, \widehat{\Psi}(\alpha^{M0}_{\mathrm{appr}}) = -0.126 \times 10^{-1})$.

where $u_h(a^{M0}) \in V_h$ is the solution of problem (4) computed for a chosen (and afterwards searched) parameter $a^{M0}$, determined by the vector of nodal values $\alpha^0 = (3.00, 3.10, 3.30, 3.40, 3.45, 3.50, 3.70, 3.80, 3.95, 4.00, 4.20) \in \widehat{\mathcal{U}}^M_{\mathrm{ad}}$. In this setting, the worst scenario problem turns into a parameter identification problem and, naturally, it holds $\widehat{\Psi}(\alpha^0) = 0$. The following figures present some numerical results.

## References

[1] Chleboun, J.: Reliable solution for a 1D quasilinear elliptic equation with uncertain coefficients. J. Math. Anal. Appl. **234** (1999), 514–528.

[2] Harasim, P.: On the worst scenario method: A modified convergence theorem and its application to an uncertain differential equation. Appl. Math. **53** (2008), 583–598.

[3] Harasim, P.: On the worst scenario method: Application to a quasilinear elliptic 2D-problem with uncertain coefficiens. Appl. Math. Accepted

[4] Hlaváček, I.: Reliable solution of elliptic boundary value problems with respect to uncertain data. Nonlinear Anal. **30** (1997), 3879–3890.

[5] Hlaváček, I., Chleboun, J. and Babuška I.: *Uncertain input data problems and the worst scenario method.* Elsevier, Amsterdam, 2004.

# NONLOCAL TANGENT OPERATOR<br>FOR DAMAGE PLASTICITY MODEL[*]

Martin Horák, Mathieu Charlebois, Milan Jirásek, Philippe K. Zysset

## 1 Introduction

Realistic description of the mechanical behaviour of quasi-brittle materials requires a constitutive law with softening. Softening is one of the destabilising factors that may lead to localisation of inelastic processes into narrow bands. Standard "local" models fail to describe this phenomenon in an objective way. The boundary value problem becomes ill-posed due to the loss of ellipticity of the governing differential equation and results obtained numerically are not objective with respect to the discretisation. To avoid pathological sensitivity of the numerical results to the finite element mesh, the model is regularised by a nonlocal formulation based on a spatial averaging procedure, which acts as a localisation limiter. The return mapping algorithm based on the closest-point projection is developed and the corresponding consistent algorithmic stiffness is derived using an extension of the approach proposed in [2] for nonlocal damage models.

## 2 Constitutive model

In this section a model combining anisotropic elasticity and anisotropic plasticity coupled with isotropic damage is described. This model was first presented in [4]. The main feature of plasticity models is irreversibility of plastic strain while irreversible processes related to damage lead to degradation of stiffness. The basic equations include an additive decomposition of total strain into elastic (reversible) part and plastic (irreversible) part,

$$\varepsilon_{ij} = \varepsilon^{e}_{ij} + \varepsilon^{p}_{ij},$$

the stress strain law,

$$\sigma_{ij} = \left(1 - \omega\left(\kappa\right)\right)\bar{\sigma}_{ij} = \left(1 - \omega\left(\kappa\right)\right) D^{e}_{ijkl}\varepsilon^{e}_{kl},$$

loading-unloading conditions in Kuhn-Tucker form,

$$f(\bar{\sigma}_{ij},\kappa) \leq 0 \qquad \dot{\lambda} \geq 0 \qquad \dot{\lambda} f(\bar{\sigma}_{ij},\kappa) = 0,$$

[*]This work was supported by the Czech Science Foundation (grant GAČR 106/08/1508).

evolution laws for plastic strain,

$$\dot{\varepsilon}^p_{ij} = \dot{\lambda}\frac{\partial f}{\partial \bar{\sigma}_{ij}},$$

and for cumulated plastic strain,

$$\dot{\kappa} = \sqrt{\dot{\varepsilon}^p_{ij}\dot{\varepsilon}^p_{ij}},$$

the law governing the evolution of the damage variable,

$$\omega(\kappa) = \omega_c(1 - \mathrm{e}^{-a\kappa}),$$

and the hardening law,

$$\sigma_Y(\kappa) = 1 + \sigma_H(1 - \mathrm{e}^{-s\kappa}).$$

In the equations above, $\bar{\sigma}_{ij}$ is the effective stress tensor, $D^e_{ijkl}$ is the elastic stiffness tensor, $f$ is the yield function, $\lambda$ is the plastic multiplier, $\omega$ is the damage variable, $\kappa$ is the cumulated plastic strain, $\sigma_Y$ is the yield stress and $s$, $a$, $\sigma_H$ and $\omega_c$ are positive material parameters, to be identified from experiments. Superior dot marks the derivative with respect to time. To complete the formulation, the specific form of yield function needs to be introduced:

$$f(\bar{\sigma}_{ij}, \kappa) = \sqrt{\bar{\sigma}_{ij}F_{ijkl}\bar{\sigma}_{kl}} - \sigma_Y(\kappa).$$

Material anisotropy is characterised by the second-order fabric tensor. The eigenvectors of the fabric tensor determine the directions of material orthotropy and the components of the elastic stiffness tensor $D^e_{ijkl}$ are linked to eigenvalues of the fabric tensor. Similar relations are postulated for the components of the fourth-order tensor $F_{ijkl}$ used in the yield condition.

## 3 Nonlocal formulation

The standard elasto-plasto-damage model based on continuum approach was described in the previous section. However, such a model fails after the loss of ellipticity, which leads to an ill-posed boundary value problem. From the numerical point of view, ill-posedness is manifested by a pathological sensitivity of the numerical results to the size of finite elements. One possible regularisation technique is a nonlocal formulation based on spatial averaging. The model is regularised by the over-nonlocal formulation with damage driven by a combination of local and nonlocal cumulated plastic strain,

$$\hat{\kappa} = (1 - m)\kappa + m\bar{\kappa},$$

where

$$\bar{\kappa}(x) = \int_V \alpha(x, s)\kappa(s)\,\mathrm{d}s \qquad (1)$$

is the nonlocal cumulated plastic strain and $m$ is a model parameter that should exceed unity to suppress the sensitivity of the numerical solution to the mesh shape. The nonlocal weight function is usually defined as

$$\alpha(x,s) = \frac{\alpha_0(\|x-s\|)}{\int_V \alpha_0(\|x-t\|)\,\mathrm{d}t}$$

where

$$\alpha_0(r) = \begin{cases} (1-r^2/R^2)^2 & \text{if } r < R \\ 0 & \text{if } r \geq R \end{cases}$$

is a nonnegative function, for $r < R$ monotonically decreasing with increasing distance $r = \|x-s\|$, and $V$ denotes the domain occupied by the investigated material body. The key idea is that the damage evolution at a certain point depends not only on the cumulated plastic strain at that point, but also on points at distances smaller than the interaction radius $R$, considered as a new material parameter. Note that the over-nonlocal cumulated plastic strain affects only damage evolution while the yield condition remains local.

## 4 Numerical algorithm

To implement the constitutive model into a displacement-driven finite element code, the governing equations need to be expressed in an incremental form, and an algorithm for the evaluation of the stress increment from a given strain increment must be developed. In plasticity, this procedure is often called the stress-return algorithm. Within a computational increment number $n+1$, the mapping of strain $\boldsymbol{\varepsilon}^{n+1}$ at the end of the step onto the effective stress $\bar{\boldsymbol{\sigma}}^{n+1}$ at the end of the step, provided by the stress-return algorithm, is denoted as function $\boldsymbol{\theta}$, and the mapping of strain $\boldsymbol{\varepsilon}^{n+1}$ onto the cumulated plastic strain $\kappa^{n+1}$ at the end of the step is denoted as function $\eta$. The Jacobi matrix of $\boldsymbol{\theta}$, denoted as $\partial\boldsymbol{\theta}/\partial\boldsymbol{\varepsilon}$, is the consistent elasto-plastic material stiffness. Using the standard finite element assembly procedure, the consistent structural tangent stiffness can be constructed. However, for an elasto-plastic model with damage, it is necessary to take into account additional terms that result from damage growth, and if damage is driven by the over-nonlocal cumulated plastic strain $\hat{\kappa}$, such terms have a more complicated structure than usual, but are still manageable. The resulting nonlocal tangent stiffness matrix of the structural (finite element) model is used in equilibrium iterations of the Newton-Raphson type and leads to quadratic convergence, provided that the linearisation is done in a fully consistent manner.

### 4.1 Predictor-corrector scheme

The stress return algorithm is based on elastic-plastic operator split, which consist of a trial elastic predictor followed by the return mapping algorithm. The over-nonlocal formulation described in the previous section has computational advantages

**Algorithm 1** Return mapping algorithm

```
given $\varepsilon^{n+1}$, $\varepsilon^{p,n}$, $\kappa^n$, $\omega^n$
compute elastic predictor
    $\sigma_{ij}^{tr} = D_{ijkl}^{e}(\varepsilon_{kl}^{e,n+1} - \varepsilon_{kl}^{p,n})$ and $f^{tr} = f(\sigma_{ij}^{tr}, \kappa^n)$
check for plastic process
if $f \leq 0$ then
  elastic step: set $\bar{\boldsymbol{\sigma}}^{n+1} = \bar{\boldsymbol{\sigma}}^{tr}$, $\boldsymbol{\varepsilon}^{p,n+1} = \boldsymbol{\varepsilon}^{p,n}$, $\kappa^{n+1} = \kappa^n$, $\omega^{n+1} = \omega^n$
else
  return mapping algorithm
  1. solve system of nonlinear equations
  $\bar{\sigma}_{ij}^{n+1} = \bar{\sigma}_{ij}^{tr} - \Delta\kappa \dfrac{D_{ijkl}^{e} F_{klmn} \bar{\sigma}_{mn}^{n+1}}{\|F_{ijkl}\bar{\sigma}_{kl}^{n+1}\|}$        $\sqrt{\bar{\sigma}_{ij}^{n+1} F_{ijkl} \bar{\sigma}_{kl}^{n+1}} - \sigma_Y(\kappa + \Delta\kappa) = 0$
  $\Longrightarrow \bar{\boldsymbol{\sigma}}^{n+1} = \theta(\boldsymbol{\varepsilon}^{n+1})$; $\Delta\kappa = \eta(\boldsymbol{\varepsilon}^{n+1})$
  2. update state variables
  $\varepsilon_{ij}^{p,n+1} = \varepsilon_{ij}^{p,n} + \Delta\kappa \frac{F_{ijkl}\bar{\sigma}_{kl}^{n+1}}{\|F_{ijkl}\bar{\sigma}_{kl}^{n+1}\|}$,   $\kappa^{n+1} = \kappa^n + \Delta\kappa$,   $\hat{\kappa}^{n+1} = (1-m)\kappa^{n+1} + m\bar{\kappa}^{n+1}$
  $\omega^{n+1} = \omega(\hat{\kappa}^{n+1})$,   $\sigma_{ij}^{n+1} = (1 - \omega^{n+1})\bar{\sigma}_{ij}^{n+1}$
end if
```

because the plastic part of the model remains local and the standard return mapping algorithm can be applied at each Gauss point separately. After that, the nonlocal cumulated plastic strain and damage are evaluated in a fully explicit manner. This procedure is summarised in Algorithm 1.

### 4.2 Consistent tangent operator

The concept of a consistent tangent operator was first presented in [1] for the case of a local elastoplastic problem. As shown in [2], the quadratic convergence is preserved also for a nonlocal damage problem, but only with a consistent nonlocal tangent operator. The consistent stiffness operator is obtained by differentiating the internal force vector with respect to the nodal displacements:

$$\boldsymbol{K} = \frac{\partial \boldsymbol{f}_{int}}{\partial \boldsymbol{d}}.$$

The internal force vector is defined as

$$\int_V \boldsymbol{B}^T \boldsymbol{\sigma} \, \mathrm{d}x \approx \sum_r w_r \boldsymbol{B}_r^T \boldsymbol{\sigma}_r = \boldsymbol{f}_{int} \tag{2}$$

In the above, subscript $r$ refers to the integration points of the finite element model, $w_r$ are the corresponding integration weights and $\boldsymbol{B}$ is the usual strain-displacement matrix.

Using the expression for stress at Gauss point $r$,

$$\boldsymbol{\sigma}_r = (1 - \omega_r)\bar{\boldsymbol{\sigma}}_r,$$

we can expand (2) as

$$\boldsymbol{f}_{int} = \sum_r w_r \boldsymbol{B}_r^T (1-\omega_r)\bar{\boldsymbol{\sigma}}_r.$$

The effective stress at Gauss point $r$ is given by the return mapping evaluated for strain at Gauss point $r$:

$$\bar{\boldsymbol{\sigma}}_r = \theta_r(\boldsymbol{\varepsilon}_r).$$

One can then express damage as

$$\omega_r = \omega(\hat{\kappa}_r) = \omega\left(m\bar{\kappa}_r + (1-m)\,\kappa_r\right)$$

and after numerical approximation of integral (1) by

$$\bar{\kappa} \approx \sum_s \alpha_{rs}\kappa_s$$

one gets

$$\omega_r = \omega\left(m\sum_s \alpha_{rs}\kappa_s + (1-m)\,\kappa_r\right),$$

where

$$\kappa_s = \eta(\boldsymbol{\varepsilon}_s)$$

is also supplied by the return mapping algorithm. Combining all this with the standard relation $\boldsymbol{\varepsilon}_s = \boldsymbol{B}_s\boldsymbol{d}$, one can evaluate the consistent nonlocal tangent stiffness operator as

$$\begin{aligned} K = \sum_r w_r(1-\omega_r)\boldsymbol{B}_r^T \frac{\partial\theta(\boldsymbol{\varepsilon}_r)}{\partial\boldsymbol{\varepsilon}}\boldsymbol{B}_r - (1-m)\sum_r w_r\omega_r'\boldsymbol{B}_r^T\bar{\boldsymbol{\sigma}}_r\left(\frac{\partial\eta(\boldsymbol{\varepsilon}_r)}{\partial\boldsymbol{\varepsilon}}\right)^T\boldsymbol{B}_r \\ -m\sum_r\sum_s w_r\omega_r'\alpha_{rs}\boldsymbol{B}_r^T\bar{\boldsymbol{\sigma}}_r\left(\frac{\partial\eta(\boldsymbol{\varepsilon}_s)}{\partial\boldsymbol{\varepsilon}}\right)^T\boldsymbol{B}_s \end{aligned} \tag{3}$$

## 5 Numerical example

The algorithm described in section 4 has been implemented into the open-source finite element code OOFEM [5, 6]. Properties of the model have been explored for several examples in [4], but with the secant stiffness matrix, which provides only linear convergence rate. The compression of a cylinder is simulated in 100 incremental steps, using a three-dimensional model containing 915 nodes and 609 linear brick elements. As follows from (3), the nonlocal tangent operator is nonsymmetric. One important consequence of nonlocality is a growing profile of the stiffness matrix, caused by the stepwise activation of interaction between pairs of Gauss points belonging to different elements. The evolution of error versus the number of iteration for three steps, corresponding to a pre-peak, peak and post-peak state as indicated in the load-displacement curve in Figure 1(a), is depicted in Figure 1(b) in a semilogarithmic scale. The convergence curves are approximately parabolic, i.e., the convergence rate is quadratic and the equilibrium is reached in a few iterations.

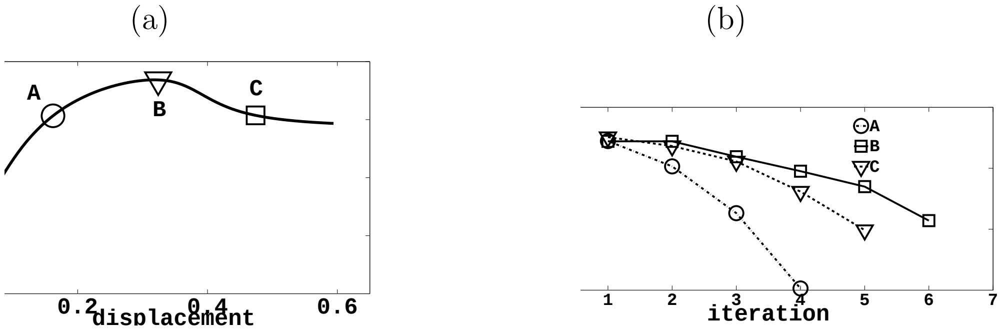


**Fig. 1:** *(a) Load-displacement curve. (b) Evolution of error during the equilibrium iteration process.*

## 6 Conclusions

The constitutive law combining anisotropic elasticity, anisotropic plasticity and isotropic damage with the over-nonlocal regularisation is presented. The stress-return algorithm is described and the nonlocal consistent tangent operator is derived. It is shown by a numerical example that the nonlocal consistent tangent operator leads to a quadratic rate of convergence, even if the tangent operator is nonsymmetric and the profile of nonzero elements is growing during the simulation.

## References

[1] Simo, J.C. and Taylor, R.L.: Consistent tangent operators for rate-independent elastoplasticity. Comput. Methods. Appl. Mech. Eng. **48** (1985), 101–118.

[2] Jirásek, M. and Patzák, B.: Consistent tangent stiffness for nonlocal damage models. Comput. Struct. **80** (2002), 1279–1293.

[3] Jirásek, M. and Rolshoven, S.: Comparison of integral-type nonlocal plasticity models for strain-softening materials. Int. J. Eng. Sci. **41** (2003), 1553–1602.

[4] Charlebois, M., Jirásek, M., and Zysset, P.K.: A nonlocal constitutive model for trabecular bone softening in compression. Biomech. Model. Mechanobiol. **9** (2010), 597–611.

[5] Patzák, B. and Bittnar, Z.: Design of object oriented finite element code. Advan. Eng. Soft. **32** (2001), 759–767.

[6] Patzák, B.: OOFEM project home page `http://www.oofem.org`, 2000–2010.

# A SHOCK-CAPTURING DISCONTINUOUS GALERKIN METHOD FOR THE NUMERICAL SOLUTION OF INVISCID COMPRESSIBLE FLOW[*]

Jiří Hozman

## 1 Introduction

A specific wide class of problems of fluid mechanics is formed of inviscid compressible flow, which is described by the system of the compressible Euler equations. The solutions of such problems usually contain subdomains, where steep gradients or discontinuities are presented (e.g., shock waves or contact discontinuities). To solve these problems in a sufficiently robust, efficient and accurate way, the *discontinuous Galerkin method* (DGM) is popularly used. DGM is based on a piecewise polynomial but discontinuous approximation, for a survey, see, e.g., [2], [3]. However when DGM is applied to the compressible inviscid fluid flow, the resulting solutions suffer from Gibbs-type oscillations, which arise in the vicinage of discontinuities, spread into the computational domain and corrupt the solution. In order to suppress these non-physical oscillations and improve a prediction of crucial flow phenomena the standard DGM is treated with a *shock-capturing* technique, see, e.g., [5], [7].

This article extends a shock capturing approach from [7], which is based on adding the artificial diffusion term to the original system, in a view of technique presented in [6], where the amount of added artificial viscosity is abided by the residual of the entropy equation. The resulting scheme denoted by SC-DGM is applied to a classical benchmark problem of inviscid steady-state flow.

## 2 Compressible Euler equations

We consider the compressible Euler equations in an open domain $Q_T = \Omega \times (0,T)$, where $T > 0$ is the final time and $\Omega \subset I\!R^2$ is the flow domain. We denote the boundary of $\Omega$ by $\partial\Omega$, it consists of several disjoint parts — inlet, outlet and impermeable walls. The system of the Euler equations describing a motion of inviscid compressible fluids can be written in conservative variables $\boldsymbol{w} = (\rho,\, \rho v_1,\, \rho v_2,\, e)^T$ in the dimensionless form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \nabla \cdot \vec{\boldsymbol{f}}(\boldsymbol{w}) = 0 \quad \text{in } Q_T, \tag{1}$$

where $\vec{\boldsymbol{f}} = (\vec{f}_1, \vec{f}_2)^T$ are the *inviscid (Euler) fluxes*, defined by

$$\vec{f}_s(\boldsymbol{w}) = (\rho v_s,\, \rho v_s v_1 + \delta_{s1} p,\, \rho v_s v_2 + \delta_{s2} p,\, (e+p)\, v_s)^T,\ \ s = 1, 2. \tag{2}$$

[*]This work was supported by the ESF Project No. CZ.1.07/2.3.00/09.0155.

We use a notation: $\rho$ - density, $\boldsymbol{v} = (v_1, v_2)^T$ - velocity field, $e$ - total energy, $p$ - pressure and $\delta_{sk}$ - Kronecker delta. The system (1) is closed with the equation of state of a perfect gas and equipped with the initial condition and the set of boundary conditions on appropriate parts of boundary, see [3].

## 3 DG discretization

Let $\mathcal{T}_h$ $(h > 0)$ represents a partition of the closure $\overline{\Omega}$ of the domain $\Omega$ into a finite number of closed elements $K$ with mutually disjoint interiors. We call $\mathcal{T}_h = \{K\}_{K \in \mathcal{T}_h}$ a *triangulation* of $\Omega$ and do not require the conforming properties from the finite element method. By $\mathcal{F}_h$ we denote the set of all open edges of all elements $K \in \mathcal{T}_h$. Further, the symbol $\mathcal{F}_h^I$ stands for the set of all $\Gamma \in \mathcal{F}_h$ that are contained in $\Omega$ (inner edges). Finally, for each $\Gamma \in \mathcal{F}_h$, we define a unit normal vector $\vec{n}_\Gamma = (n_1, n_2)^T$. We assume that $\vec{n}_\Gamma$, $\Gamma \subset \partial\Omega$, has the same orientation as the outer normal of $\partial\Omega$. For $\vec{n}_\Gamma$, $\Gamma \in \mathcal{F}_I$, the orientation is arbitrary but fixed for each edge.

DGM allows to treat with different polynomial degrees over elements. Therefore, we assign a positive integer $p_K$ as a *local polynomial degree* to each $K \in \mathcal{T}_h$. Then we set the vector $\mathsf{p} = \{p_K, K \in \mathcal{T}_h\}$. Over the triangulation $\mathcal{T}_h$ we define the finite dimensional space of discontinuous piecewise polynomial functions

$$S_{h\mathsf{p}} = \{v; v|_K \in P_{p_K}(K)\ \forall\, K \in \mathcal{T}_h\}, \tag{3}$$

where $P_{p_K}(K)$ denotes the space of all polynomials of degree $\leq p_K$ on $K$, $K \in \mathcal{T}_h$. Then we seek the approximate solution of the system (1) in the space of vector-valued functions $\mathbf{S}_{h\mathsf{p}} = [S_{h\mathsf{p}}]^4$.

For each $\Gamma \in \mathcal{F}_h^I$ there exist two elements $K_L$, $K_R \in \mathcal{T}_h$ such that $\Gamma \subset K_L \cap K_R$. We use a convention that $K_R$ lies in the direction of $\vec{n}_\Gamma$ and $K_L$ in the opposite direction of $\vec{n}_\Gamma$. For $v \in S_{h\mathsf{p}}$, by $v|_\Gamma^{(L)} =$ trace of $v|_{K_L}$ on $\Gamma$, $v|_\Gamma^{(R)} =$ trace of $v|_{K_R}$ on $\Gamma$ we denote the *traces* of $v$ on edge $\Gamma$, which are different in general. Moreover, $[v]_\Gamma = v|_\Gamma^{(L)} - v|_\Gamma^{(R)}$ and $\langle v \rangle_\Gamma = \frac{1}{2}\left(v|_\Gamma^{(L)} + v|_\Gamma^{(R)}\right)$ denote the *jump* and *mean value* of function $v$ over the edge $\Gamma$, respectively. For $\Gamma \in \partial\Omega$ there exists an element $K_L \in \mathcal{T}_h$ such that $\Gamma \subset K_L \cap \partial\Omega$. Then for $v \in S_{h\mathsf{p}}$, we put: $v|_\Gamma^{(L)} =$ trace of $v|_{K_L}$ on $\Gamma$, $\langle v \rangle_\Gamma = [v]_\Gamma = v|_\Gamma^{(L)}$ .

Now, we recall the space semi-discrete DG scheme presented in [3]. The crucial item of the DG formulation of the Euler equations is the treatment of the inviscid terms. We employ the concept of numerical flux $I\!\!H(\cdot,\cdot,\cdot)$, namely the Vijayasundaram numerical flux, see [5].

Therefore, a function $\boldsymbol{w}_h \in C^1([0,T]; \mathbf{S}_{h\mathsf{p}})$ is called the *semi-discrete solution* of (1) if

$$\left(\frac{\partial \boldsymbol{w}_h(t)}{\partial t}, \boldsymbol{\varphi}_h\right) + \boldsymbol{b}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = 0 \qquad \forall\, \boldsymbol{\varphi}_h \in \mathbf{S}_{h\mathsf{p}},\ \forall\, t \in (0,T), \tag{4}$$

where $(\cdot,\cdot)$ denotes the $L^2$-scalar product and

$$\boldsymbol{b}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) = \sum_{\Gamma \in \mathcal{F}_h} \int_\Gamma I\!H(\boldsymbol{w}_h|_\Gamma^{(L)}, \boldsymbol{w}_h|_\Gamma^{(R)}, \vec{n}_\Gamma)\, [\boldsymbol{\varphi}_h]_\Gamma \, \mathrm{d}S - \sum_{K \in \mathcal{T}_h} \int_K \vec{\boldsymbol{f}}(\boldsymbol{w}_h) \cdot \nabla \boldsymbol{\varphi}_h \, \mathrm{d}x. \quad (5)$$

The problem (4) represents a system of ordinary differential equations (ODEs) for $\boldsymbol{w}_h(t)$ which has to be discretized in time by a suitable method. Since these ODEs belong to the class of stiff problems whose solutions by an explicit scheme are rather inefficient, it is advantageous to use a *semi-implicit* approach.

According to [3], we define the semi-implicit time discretization of (4) by

$$(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + \tau_k \boldsymbol{b}_h^L(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = (\boldsymbol{w}_h^k, \boldsymbol{\varphi}_h) \qquad \forall \boldsymbol{\varphi}_h \in \mathbf{S}_{h\mathsf{p}},\ k = 0, 1 \ldots, r, \quad (6)$$

where $\boldsymbol{w}_h^k \in \mathbf{S}_{h\mathsf{p}}$, $k = 0, \ldots, r$, denote approximate solutions at time levels $t_k$, $k = 0, \ldots, r$, $\tau_k = t_{k+1} - t_k$ is the size of the time step and $\boldsymbol{b}_h^L(\cdot, \cdot, \cdot)$ formally represents a linearization of the DG discretization of the inviscid fluxes (5), see [3].

## 4 Shock-capturing scheme

We have proposed a *viscosity limiter* approach, which is based on adding artificial diffusion term to the system (1) in the form which corresponds to the viscous part of the compressible Navier-Stokes equations but with the variable Reynolds number $Re$ in the whole computational domain as in [7]. This variable choice of $Re$ plays a role as an artificial viscosity $\mu_{art}$, which depends on the solution of the system (1) in a special way, i.e., $Re^{-1} \approx \mu_{art}(\boldsymbol{w}_h)$.

We modify (1) and get new system

$$\frac{\partial \boldsymbol{w}}{\partial t} + \nabla \cdot \vec{\boldsymbol{f}}(\boldsymbol{w}) = \mu_{art}(\boldsymbol{w}) \nabla \cdot \vec{\boldsymbol{R}}(\boldsymbol{w}, \nabla \boldsymbol{w}) \quad \text{in } Q_T, \quad (7)$$

where

$$\vec{\boldsymbol{R}}(\boldsymbol{w}, \nabla \boldsymbol{w}) = (\vec{R}_1, \vec{R}_2) \text{ and } \vec{R}_s = \begin{pmatrix} 0 \\ \left(\frac{\partial v_s}{\partial x_1} + \frac{\partial v_1}{\partial x_s}\right) - \frac{2}{3}\mathrm{div}(\boldsymbol{v})\, \delta_{s1} \\ \left(\frac{\partial v_s}{\partial x_2} + \frac{\partial v_2}{\partial x_s}\right) - \frac{2}{3}\mathrm{div}(\boldsymbol{v})\, \delta_{s2} \\ \sum_{k=1}^2 \vec{R}_s^{(k)} v_k + \frac{\gamma}{Pr} \frac{\partial \theta}{\partial x_s} \end{pmatrix}, \ s = 1, 2, \quad (8)$$

with a notation: $\gamma$ - Poisson adiabatic constant, $Pr$ - Prandtl number and $\theta$ - temperature. Finally, the system (7) has to be closed by the equation of the total energy, see, e.g., [4].

Let us note that this proposed artificial viscosity approach corresponds to the solution of the compressible Navier-Stokes equations with "do-nothing" boundary condition.

According to (6) and (7) we obtain a shock-capturing scheme (SC):

$$\begin{aligned} (\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) \ &+\ \tau_k \left( \boldsymbol{b}_h^L(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + \boldsymbol{a}_h^L(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + \boldsymbol{J}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) \right) \\ &=\ (\boldsymbol{w}_h^k, \boldsymbol{\varphi}_h) \qquad \forall \boldsymbol{\varphi}_h \in \mathbf{S}_{h\mathsf{p}},\ k = 0, 1 \ldots, r, \qquad (9) \end{aligned}$$

where

$$\begin{aligned}\boldsymbol{a}_h^L(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) \quad &= \sum_{K\in\mathcal{T}_h} \int_K \sum_{s=1}^{2} \mu_{art}(\boldsymbol{w}_h^k) \left( \sum_{k=1}^{d} \boldsymbol{K}_{s,k}(\boldsymbol{w}_h^k) \frac{\partial \boldsymbol{w}_h^{k+1}}{\partial x_k} \right) \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s} \,\mathrm{d}x \qquad (10)\\ &- \sum_{\Gamma\in\mathcal{F}_h^I} \int_\Gamma \sum_{s=1}^{2} \left\langle \mu_{art}(\boldsymbol{w}_h^k) \left( \sum_{k=1}^{d} \boldsymbol{K}_{s,k}(\boldsymbol{w}_h^k) \frac{\partial \boldsymbol{w}_h^{k+1}}{\partial x_k} \right) \right\rangle_\Gamma n_s \cdot [\boldsymbol{\varphi}_h]_\Gamma \,\mathrm{d}S\\ &- \Theta \sum_{\Gamma\in\mathcal{F}_h^I} \int_\Gamma \sum_{s=1}^{2} \left\langle \mu_{art}(\boldsymbol{w}_h^k) \sum_{k=1}^{d} \boldsymbol{K}_{s,k}^T(\boldsymbol{w}_h^k) \frac{\partial \boldsymbol{\varphi}_h}{\partial x_k} \right\rangle_\Gamma n_s \cdot [\boldsymbol{w}_h^{k+1}]_\Gamma \,\mathrm{d}S\end{aligned}$$

represents the linearized viscous fluxes (8). The detailed description of the matrices $\mathbb{K}_{s,k} \in \mathbb{R}^{4\times 4},\ k = 1, 2, s = 1, 2$, can be found in [5] and $\Theta$ is a stabilization parameter which can take the values $\{-1; 0; 1\}$ according to the chosen variant of stabilization. In order to replace inter-element discontinuities of the scheme (SC) we introduce the *penalty*

$$\boldsymbol{J}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = \sum_{\Gamma\in\mathcal{F}_h^I} \int_\Gamma \mu_{art}(\boldsymbol{w}_h^k) C_W |\Gamma|^{-1} [\boldsymbol{w}_h^{k+1}]_\Gamma \cdot [\boldsymbol{\varphi}_h]_\Gamma \,\mathrm{d}S, \qquad (11)$$

where $C_W > 0$ is a suitable constant depending on the used variant of stabilization and on the degree of polynomial approximation. The scheme (9) requires a solution of linear algebraic problem at each time level and gives practically unconditionally stable scheme, see [4].

The key ingredient of the scheme (SC) is the nonlinear viscosity $\mu_{art}$ which is chosen proportionally to the residual of the entropy equation in the spirit of [6]. It is known from thermodynamics that $S = \frac{1}{\gamma-1} \ln(p/\rho^\gamma)$ is an entropy functional for perfect gas which satisfies the following energy equation (see [5]) written in the entropy form

$$\frac{\partial \rho S}{\partial t} + \mathrm{div}(\rho S \boldsymbol{v}) = \frac{D(\boldsymbol{v})}{\theta} + \mu_{art} \frac{\gamma}{Pr} \frac{\mathrm{div}(\nabla\theta)}{\theta}, \qquad (12)$$

where $D(\boldsymbol{v}) = -\frac{2}{3}\mu_{art}(\mathrm{div}(\boldsymbol{v}))^2 + 2\mu_{art}\boldsymbol{D}(\boldsymbol{v})\cdot\boldsymbol{D}(\boldsymbol{v})$ is a dissipation and $\boldsymbol{D}(\boldsymbol{v})$ denotes symmetric part of the velocity gradient.

To construct $\mu_{art}$, we first evaluate the discrete entropy residual $r_S = r_S(\boldsymbol{w}_h)$, which is considered in the following weak formulation as $r_S \in S_{h\mathsf{p}}$ such that

$$\int_\Omega r_S \cdot \varphi_h \,\mathrm{d}x = \int_\Omega \left( \frac{\partial \rho S}{\partial t} + \mathrm{div}(\rho S \boldsymbol{v}) - \frac{D(\boldsymbol{v})}{\theta} - \mu_{art} \frac{\gamma}{Pr} \frac{\mathrm{div}(\nabla\theta)}{\theta} \right) \varphi_h \,\mathrm{d}x \quad \forall\, \varphi_h \in S_{h\mathsf{p}}. \qquad (13)$$

In view of (13), the function $r_S$ is $L^2$-projection onto $S_{h\mathsf{p}}$, i.e. $r_S|_K \in P_{p_K}(K)$, $K \in \mathcal{T}_h$. Further, we construct a piecewise constant limiting viscosity as follows

$$\mu_{max}^K = \frac{\mathrm{diam}(K)}{p_K} \max_K \left( \rho|\boldsymbol{v}| + \rho\sqrt{\gamma\theta} \right) \Big|_K, \quad K \in \mathcal{T}_h \qquad (14)$$

and finally set

$$\mu_{art}(\boldsymbol{w}_h)|_K = \min\left(\mu^K_{max}, \beta L \, \mathrm{diam}(K) \Big| r_S|_K(\boldsymbol{w}_h) \Big| \right), \quad K \in \mathcal{T}_h, \tag{15}$$

where $L$ denotes the characteristic length (e.g. length of channel or airfoil) and $\beta$ is a user-dependent parameter, typically $\beta$ can reasonably be chosen in the range $[0.05, 5]$ without that choice dramatically affecting the results.

## 5 Numerical example

We consider inviscid steady transonic flow past a single NACA0012 airfoil of unit length at free stream Mach number $M_\infty = 0.8$ with angle of attack $\alpha = 1.25°$. The computation domain is a circle with radius of 50. We use a fixed relatively coarse triangular mesh having 4544 elements which was adaptively refined and uses curved elements along the airfoil. The characteristic feature of this flow is a relatively strong shock at the suction side and a very weak shock at the pressure side.

We carried out computations with the shock capturing scheme (SC) by $P_1$, $P_2$ and $P_3$ approximations and set $\Theta = 1$ (non-symmetric variant) with $C_W = 1$. These values guarantee the stability of the scheme (SC), for more details see [4]. The initial condition was set as a constant vector taken from the prescribed boundary conditions at infinity: $\rho = 1$, $v_1 = 0.999762027$, $v_2 = 0.021814885$ and the Mach number $M_\infty = 0.8$. This test case represents a stationary problem. Therefore, the computational process was stopped, after the residue of the solution had reached the prescribed tolerance.

Table 1 illustrates the asymptotic convergence of drag ($c_D$) and lift ($c_L$) coefficients and comparison with reference values from [1]. Figure 1 shows the pressure coefficient $c_p$ along the airfoil with resolved shocks. We obtained satisfactory results and quite good agreement was already achieved for piecewise cubic approximation with reference results from [1] using $P_5$ approximation.

| method | $c_D$ | $c_L$ | $\#DOF$ |
|---|---|---|---|
| SC-DGM – $P_1$ | 0.02426 | 0.33684 | 54 528 |
| SC-DGM – $P_2$ | 0.02300 | 0.34065 | 109 056 |
| SC-DGM – $P_3$ | 0.02277 | 0.35587 | 181 760 |
| ref. value [1] – $P_5$ | 0.02276 | 0.35366 | 381 696 |

**Tab. 1:** *Computed values of force coefficients in comparison with [1].*

## 6 Conclusion

We dealt with the numerical solution of the compressible Euler equations via discontinuous Galerkin method. We presented the shock-capturing technique avoiding a failure of computational processes and most of Gibbs phenomena. Preliminary numerical example gives promising results.

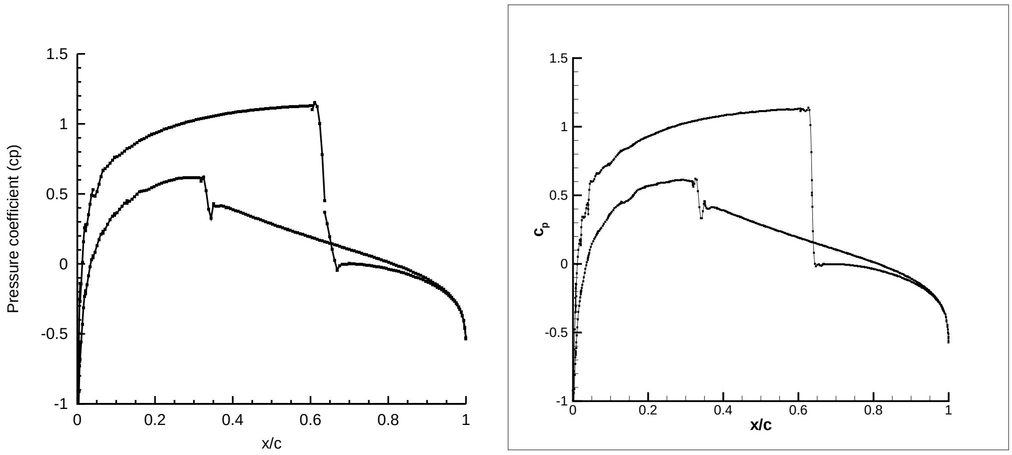


**Fig. 1:** *Pressure coefficient comparison, SC-DGM scheme with $P_3$ (left), DG scheme with $P_5$ described in [1] (right).*

## References

[1] ADIGMA. Adaptive higher-order variational methods for aerodynamic applications in industry, Specific Targeted Research Project no. 30719 supported by European Commison. URL: `http: //www.dlr.de/as/en/Desktopdefault.aspx/tabid-2035/2979_read-4582/`.

[2] Cockburn, B.: Discontinuous Galerkin methods for convection dominated problems. In: T.J. Barth and H. Deconinck, (Eds.), *High–order methods for computational physics*, *Lecture Notes in Computational Science and Engineering*, vol. 9, pp. 69–224. Springer, Berlin, 1999.

[3] Dolejší, V. and Feistauer, M.: Semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. J. Comp. Phys., **198(2)** (2004), 727–746.

[4] Dolejší, V.: Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. Commun. Comput. Phys. **4 (2)** (2008), 231–274.

[5] Feistauer, M., Felcman, J., and Straškraba, I.: *Mathematical and computational methods for compressible flow.* Oxford University Press, Oxford, 2003.

[6] Guermond, J.-L. and Pasquetti, R.: Entropy-based nonlinear viscosity for Fourier approximations of conservation laws. C. R. Acad. Sci. Paris, Ser. I **346** (2008), 801–806.

[7] Persson, P.-O. and Peraire J.: Sub-cell shock capturing for discontinuous Galerkin methods. In: *Proc. of the 44th AIAA Aerospace Sciences Meeting and Exhibit.* AIAA- 2006-1253, Reno, Nevada, 2006.

# NUMERICAL MODELLING OF NEWTONIAN AND NON-NEWTONIAN FLUIDS FLOW IN THE BRANCHING CHANNEL BY FINITE VOLUME METHOD[*]

Radka Keslerová, Karel Kozel

## 1 Mathematical model

The governing system of the equations is the system of Navier-Stokes equations for incompressible fluids. This system for generalized Newtonian fluids can be written in the conservative form [1]:

$$\tilde{R}W_t + F_x^c + G_y^c = F_x^v + G_y^v, \quad \tilde{R} = \mathrm{diag}(0,1,1), \tag{1}$$

$$W = \begin{pmatrix} p \\ u \\ v \end{pmatrix}, \quad F^c = \begin{pmatrix} u \\ u^2 + p \\ uv \end{pmatrix}, \quad G^c = \begin{pmatrix} v \\ uv \\ v^2 + p \end{pmatrix}, \tag{2}$$

$$F^v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \end{pmatrix}, \quad G^v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \end{pmatrix}. \tag{3}$$

where $p = \frac{P}{\rho}$, $P$ is the pressure, $u, v$ are the components of the velocity vector, $\rho$ is the constant density. The vector $W$ is the vector of unknowns. The vectors $F^c$, $G^c$ are inviscid physical fluxes and $F^v$, $G^v$ are viscous physical fluxes. The viscous stress $\tau$ is defined as follows

$$\tau = 2\eta(\dot{\gamma})D, \quad \dot{\gamma} = \sqrt{\mathrm{tr}D^2}, \quad D_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{4}$$

where tensor $D$ is the symmetric part of the velocity gradient and where $i$ and $j$ can take on the values $x, y$ or 1, 2. The quantities $x_1$ and $x_2$ in the derivatives denote Cartesian coordinates $x, y$. Similarly $v_1$ and $v_2$ denote the velocity vector components $u, v$.

Newtonian and non-Newtonian fluids differ through the choice of the viscosity function. One of the simplest viscosity function is the power-law model [2]

$$\eta(\dot{\gamma}) = \nu\left(\sqrt{\mathrm{tr}D^2}\right)^r, \tag{5}$$

[*]This work was partly supported by grant GACR No. 101/09/1539, Research Plan MSM 684 077 0003 and Research Plan MSM 684 077 0010.

where $\nu$ is a constant, e.g. the kinematic viscosity for Newtonian fluids. The power $r$ is the power-law index. The power-law model includes Newtonian fluids as a special case ($r = 0$). For $r > 0$ the power-law fluid is shear thickening (increasing viscosity with shear rate), while for $r < 0$ it is shear thinning (decreasing viscosity with shear rate).

## 2 Numerical solution

### 2.1 Steady computation

In this first part the steady state solution is considered. In such a case an artificial compressibility method can be applied, i.e. the continuity equation is completed by a term $\frac{1}{\beta^2}p_t$. In the non-dimensional form this yields

$$\tilde{R}_\beta W_t + F^c_x + G^c_y = \frac{1}{\mathrm{Re}}\,\epsilon\,(F^v_x + G^v_y), \quad \tilde{R}_\beta = \mathrm{diag}\left(\frac{1}{\beta^2}, 1, 1\right), \quad \beta \in \mathcal{R}^+ \tag{6}$$

where in non-dimensional form $F^c, G^c$ are inviscid physical fluxes and $F^v, G^v$ are viscous physical fluxes. The symbol $\epsilon$ represents $\left(\sqrt{\mathrm{tr}D^2}\right)^r$. The symbol Re denotes Reynolds number and it's defined by the expression

$$\mathrm{Re} = \frac{U^* L^*}{\nu}, \tag{7}$$

where $U^*, L^*$ are the reference velocity and length, $\nu$ is the kinematic Newtonian viscosity. The parameter $\beta$ has dimension of a speed. In the case of non-dimensional equations, $\beta$ is then divided by a reference velocity $U^*$. This is usually an upstream velocity, which does not significantly differ from the maximum velocity in the flow field. Hence, in the case of non-dimensional equations, $\beta = 1$ is used in presented steady numerical simulations.

Eq. (6) is space discretized by the finite volume method[3], [5] and the arising system of ODEs is time discretized by the explicit multistage Runge-Kutta scheme of the second order of accuracy in the time

$$\begin{aligned} W_i^n &= W_i^{(0)} \\ W_i^{(r)} &= W_i^{(0)} - \alpha_{r-1}\Delta t \mathrm{Res}(W)_i^{(r-1)} \\ W_i^{n+1} &= W_i^{(m)} \qquad\qquad r = 1, \ldots, m, \end{aligned} \tag{8}$$

where $m = 3$, $\alpha_0 = \alpha_1 = 0.5, \alpha_2 = 1.0$, the steady residual $\mathrm{Res}(W)_i$ is defined by finite volume method as

$$\mathrm{Res}(W)_i = \frac{1}{\mu_i}\sum_{k=1}^{4}\left[\left(\overline{F}^c_k - \frac{1}{\mathrm{Re}}\epsilon\,\overline{F}^v_k\right)\Delta y_k - \left(\overline{G}^c_k - \frac{1}{\mathrm{Re}}\epsilon\,\overline{G}^v_k\right)\Delta x_k\right], \tag{9}$$

where $\mu_i$ is the volume of the finite volume cell, $\mu_i = \int\int_{C_i} dx\, dy$. The symbols $\overline{F}^c_k, \overline{G}^c_k$ and $\overline{F}^v_k, \overline{G}^v_k$ denote the numerical approximation of the inviscid and viscous physical fluxes. The symbol Re is Reynolds number defined by (7). The symbol $\epsilon$ represents $\left(\sqrt{\mathrm{tr}D^2}\right)^r$, where for power $r$ three values are choosed: $r = 0$ for Newtonian fluids, $r = 0.5$ for shear thickening fluids and $r = -0.5$ for shear thinning fluids.

### 2.2 Unsteady computation

The dual-time stepping method is used for the unsteady flows for Newtonian fluids. The principle of dual-time stepping method is following. The artificial time $\tau$ is introduced and the artificial compressibility method in the artificial time is applied. The system of Navier-Stokes equations is extended to unsteady flows by adding artificial time derivatives $\partial W/\partial\tau$ to all equations [4]

$$\tilde{R}_\beta W_\tau + \tilde{R} W_t + F^c_x + G^c_y = F^v_x + G^v_y \tag{10}$$

with matrices $\tilde{R}$, $\tilde{R}_\beta$ given by Eq. (1), (6). The vector of the variables $W$, the inviscid fluxes $F^c, G^c$ and the viscous fluxes $F^v, G^v$ are given by Eq. (2).

The derivatives with respect to the real time $t$ are discretized using a three-point backward formula, it defines the form of unsteady residual

$$\tilde{R}_\beta \frac{W^{l+1} - W^l}{\Delta\tau} = -\tilde{R}\frac{3W^{l+1} - 4W^n + W^{n-1}}{2\Delta t} - \mathrm{Res}(W)^l = -\overline{\mathrm{Res}}(W)^{l+1}, \tag{11}$$

where $\Delta t = t^{n+1} - t^n$ and $\mathrm{Res}(W)$ is the steady residual defined as for steady computation, see Eq. (9). The symbol $\overline{\mathrm{Res}}(W)$ denotes unsteady residual. The superscript $n$ denotes the real time index and the index $l$ is associated with the pseudo-time. The integration in pseudo-time can be carried out by explicit multistage Runge-Kutta scheme.

The solution procedure is based on the assumption that the numerical solution at real time $t^n$ is known. Setting $W^l_i = W^n_i, \forall i$, the iteration in $l$ using explicit Runge-Kutta method are performed until the condition

$$\|\overline{\mathrm{Res}}(W)^l\|_{\mathrm{L}^2} = \sqrt{\sum_i \left(\frac{W^{l+1}_i - W^l_i}{\Delta\tau}\right)^2} \leq \epsilon \tag{12}$$

is satisfied for a chosen small positive number $\epsilon$. The symbol $\overline{\mathrm{Res}}(W)^l$ stands for the vector formed by the collection of $\overline{\mathrm{Res}}(W)^l_i, \forall i$. Once the condition (12) is satisfied for a particular $l$, one sets $W^{n+1}_i = W^{l+1}_i, \forall i$. Then the index representing real-time level can be shifted one up. History of the convergence of unsteady residual in dual time from $t^n$ to $t^{n+1}$ is plotted in decadic logarithm.

The unsteady boundary conditions are defined as follows. In the inlet, in the solid wall and in one of the outlet part the steady boundary conditions are prescribed. In the second outlet part the unsteady boundary conditions are defined. The velocity is computed by the extrapolation from the domain. The pressure value is prescribed by the function

$$p_{21} = \frac{1}{4}\left(1 + \frac{1}{2}\sin(\omega t)\right), \tag{13}$$

where $\omega$ is the angular velocity defined as $\omega = 2\pi f$, where $f$ is the frequency.

## 3 Numerical results

### 3.1 Two dimensional steady solution

In this section the steady numerical results of two dimensional incompressible laminar viscous flows for generalized Newtonian fluids are presented.

The following choices of the power-law index were used. For Newtonian fluid $r = 0$ is used. For shear thickening and shear thinning non-Newtonian fluid values $r = 0.5$ (shear thickening) and $r = -\ 0.5$ (shear thinning) are used. The flow is computed through the branching channel. In the inlet the velocity is prescribed by the parabolic function. Reynolds number is equal to 400 for tested cases of the fluids.

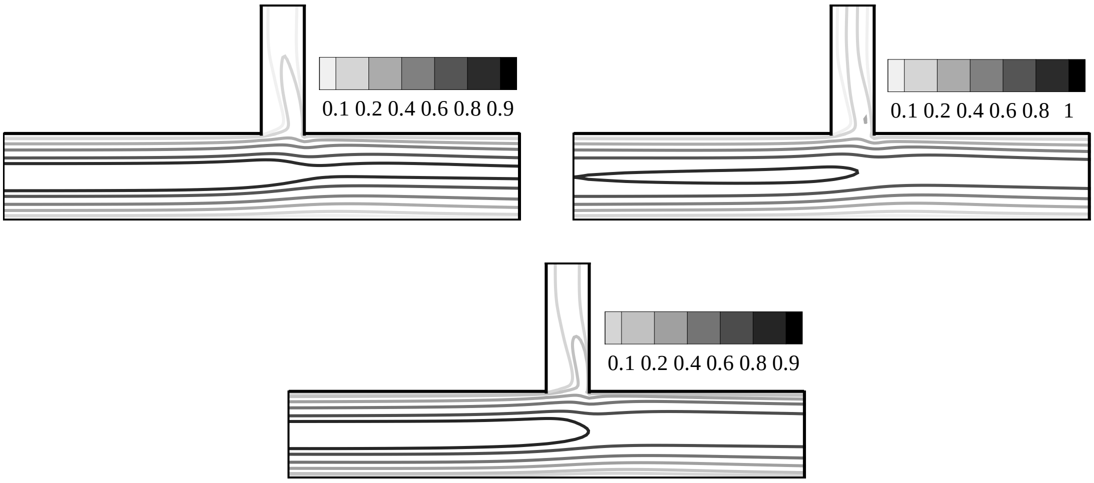


**Fig. 1:** *Velocity isolines of steady flows for generalized Newtonian fluids - a) Newtonian - b) shear thickening non-Newtonian - c) shear thinning non-Newtonian.*

In the Figures 1 the velocity isolines for 2D tested fluids are presented. One of the main differencies between Newtonian and non-Newtonian fluids is given by the size of the separation region.

In the Figure 2 the nondimensional axial velocity profile for steady fully developed flow of Newtonian, shear thickening and shear thinning fluids in 2D branching channel is shown.

### 3.2 Two dimensional unsteady numerical solution

In this section two dimensional unsteady numerical results for Newtonian fluid through the branching channel are presented. The dual-time stepping method are used. The unsteady boundary conditions were considered. As initial data the numerical solution of steady fully developed flow of Newtonian fluids in the branching channel were used. Reynolds number is 400. The frequency in the pressure function (13) is 2.

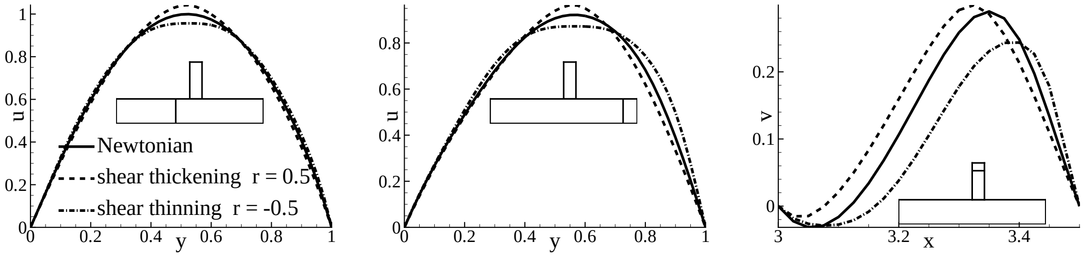


**Fig. 2:** *Nondimensional velocity profile for steady fully developed flow of generalized Newtonian fluids in the branching channel (the line legend in all three panels is the same).*

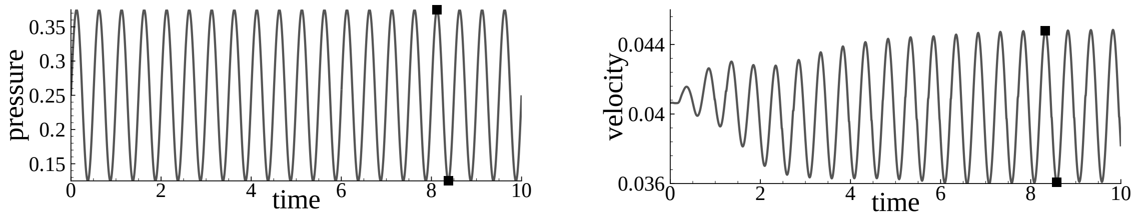


**Fig. 3:** *The graphs of the pressure and the velocity computed by the dual-time stepping method.*

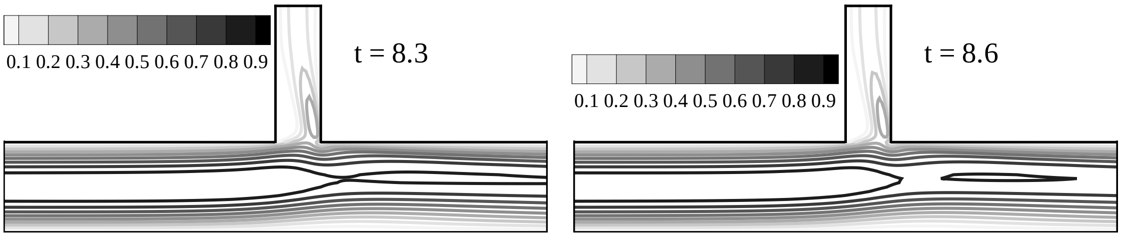


**Fig. 4:** *Velocity isolines of unsteady flows of Newtonian fluids - dual-time stepping method.*

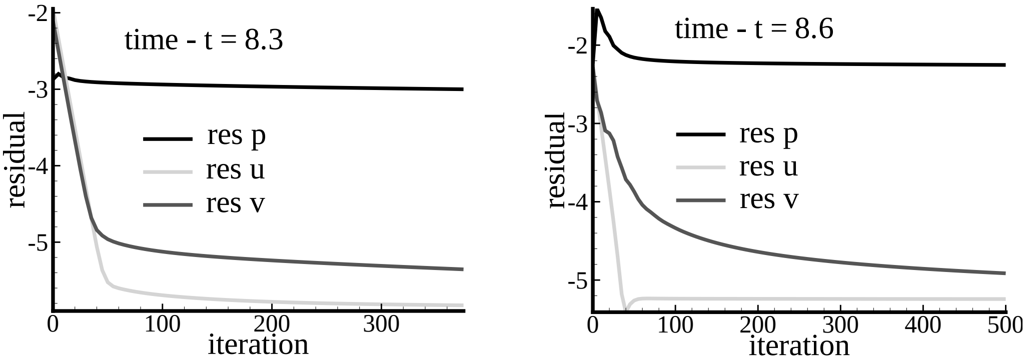


**Fig. 5:** *Decadic logarithm of the $L^2$ norm of the unsteady residual - dual-time stepping method.*

In Figure 3 the graphs of the pressure $p_{21}(t)$ and the velocity as the function of the time for Newtonian fluid are shown. By the square symbols the positions of the unsteady numerical results for dual-time stepping method shown in the Figure 4 are sketched. In the Figure 5 the decadic logarithm of the $\mathrm{L}^2$ norm of unsteady residual by the dual-time stepping method is shown.

## 4 Conclusions

In this paper a finite volume solver for two and three dimensional incompressible laminar viscous flows in the branching channel was described. The numerical results obtained by this method for Newtonian and non-Newtonian (shear thickening and shear thinning) fluid flows were presented. For the generalized Newtonian fluids the power-law model was used. The explicit Runge-Kutta method was considered for numerical modelling. The convergence history confirms the robustness of the applied method.

## References

[1] Dvořák, R. and Kozel, K.: *Mathematical modelling in aerodynamics (in Czech).* CTU, Prague, Czech Republic, 1996.

[2] Robertson, A.M., Sequeira, A., and Kameneva, M.V.: *Hemorheology.* Birkhäuser Verlag, Basel, Switzerland, 2008.

[3] LeVeque, R.: *Finite-volume methods for hyperbolic problems.* Cambridge University Press, 2004.

[4] Gaitonde, A.L.: A dual-time method for two dimensional unsteady incompressible flow calculations. International Journal for Numerical Methods in Engineering **41** (1998), 1153–1166.

[5] Keslerová, R. and Kozel, K.: Numerical modelling of incompressible flows for Newtonian and non-Newtonian fluids. Mathematics and Computers in Simulation **80** (2010), 1783–1794.

# INSENSITIVITY ANALYSIS OF MARKOV CHAINS[*]

Martin Kocurek

**Abstract**

Sensitivity analysis of irreducible Markov chains considers an original Markov chain with transition probability matix $P$ and modified Markov chain with transition probability matrix $\tilde{P}$. For their respective stationary probability vectors $\pi, \tilde{\pi}$, some of the following charactristics are usually studied: $\|\pi - \tilde{\pi}\|_p$ for asymptotical stability [3], $|\pi_i - \tilde{\pi}_i|$, $\frac{|\pi_i - \tilde{\pi}_i|}{\pi_i}$ for componentwise stability or sensitivity[1]. For functional transition probabilities, $P = P(t)$ and stationary probability vector $\pi(t)$, derivatives are also used for studying sensitivity of some components of stationary distribution with respect to modifications of $P$ [2].

In special cases, modifications of matrix $P$ leave certain stationary probabilities unchanged. This paper studies some special cases which lead to this behavior of stationary probabilities.

## 1 Introduction

A Markov chain is a sequence of random variables $X_1, X_2, X_3, \dots,$ with the Markov property, namely that, given the present state, the future and past states are independent. Formally,

$$P(X_{n+1} = x | X_1 = x_1, X_2 = x_2 \dots, X_n = x_n) = P(X_{n+1} = x | X_n = x_n),$$

where the possible values of $X_i$ form a countable state space $S$ of the chain. Markov chains are often described by a directed graph, where the edges are labeled by the probabilities $p_{ij}$ of moving from state $i$ to the other state $j$. These probabilities are called *transition probabilities* and together they form a *transition probability matrix* denoted by $P$, with row sums equal to 1. We will study finite Markov chains (a finite chain has a finite state space $S = \{x_1, ..., x_n\}$). A state $i$ has period $p$ if any return to state $i$ must occur in multiples of $p$ time steps. Formally, the period of a state $i$ is defined as $p = gcd\{k : P(X_k = i | X_0 = i) > 0\}$. If $p = 1$, then the state is said to be aperiodic i.e. returns to state $i$ can occur at irregular times. Otherwise ($p > 1$), the state is said to be periodic with period $p$. If all states are periodic with period $p$, the chain is called $p$-cyclic.

Let us denote

$$e = (1, \dots, 1)^T,\ e_i = (0, \dots, 0, 1, 0, \dots, 0)^T = (\delta_{i,j})_{j=1}^n,\ i = 1, \dots, n,\ P = (P_{ij})_{i,j=1}^n.$$

A Markov chain is called irreducible, if there exists a connection between every two states. That means, matrix $P$ is irreducible. In this case, matrix $P$ has a unique

[*] This work was supported by grant OHK1-062/10 of the Czech Technical University.

eigenvalue 1 (which equals to spectral radius $\rho(P)$ of $P$) and unique left and right eigenvectors associated with this eigenvalue, $\pi = (\pi_1, \ldots, \pi_n)$ and $e$, so that

$$\pi P = \pi, \;\; Pe = e.$$

Vector $\pi$ is called *stationary probability vector*, we usually normalise this vector to $\pi e = \|\pi\|_1 = 1$; $i$-th component $\pi_i$ of $\pi$ shows, how often the chain "visits state $i$",

$$\pi_i = \lim_{m\to\infty} \frac{|\{j; X_j = x_i, j = 1, \ldots, m\}|}{m}.$$

We will also use a different normalisation, $\pi_k = 1$ and in this case, the eigenvector will be denoted by $\pi_{(k)}$, so that $\pi_{(k)k} = 1$.

In the following, we will partition matrix $P$ and vector $\pi$ into subblocks,

$$\pi = (\pi^{(1)}, \ldots, \pi^{(N)}), \quad P = \begin{pmatrix} P_{11} & \ldots & P_{1N} \\ \vdots & \ddots & \vdots \\ P_{N1} & \ldots & P_{NN} \end{pmatrix}, \tag{1}$$

where $N$ is the number of subblocks in matrix $P$, $n_1, ..., n_N$ will be respective dimensions of subblocks. Conformally with partitioning of $P$ we shall partition vector $e = (e^{(1)T}, \ldots, e^{(N)T})^T$, where $e^{(i)}$ is a vector $(1, \ldots, 1)^T$ with $n_i$ components.

As an example we will use a Markov chain with the following matrix:

$$P_c = \frac{1}{64}\begin{pmatrix}
62 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
62 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 60 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
62 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 63 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
62 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 60 & 2 & 0 & 0 & 0 & 0 & 0 \\
62 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 63 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 62 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 0 & 60 & 0 & 0 \\
0 & 0 & 0 & 0 & 62 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 60
\end{pmatrix}. \tag{2}$$

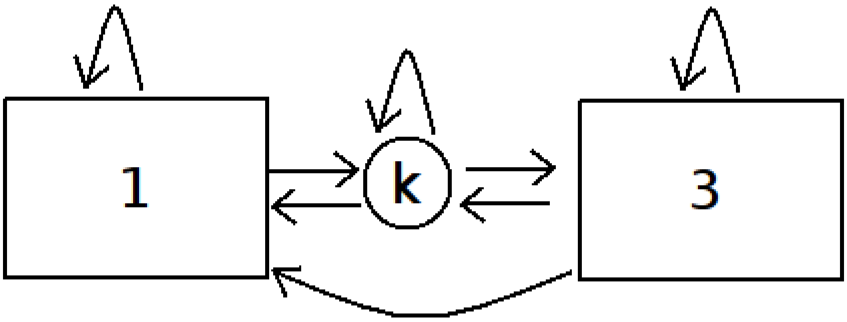
## 2 Normalisation $\pi_k = 1$

Normalisation $\pi_k = 1$ is useful for computing the eigenvector as a solution of a system of equations $\pi_{(k)}P = \pi_{(k)}$, or $P^T\pi^T_{(k)} = \pi^T_{(k)}$, $(I - P^T)\pi^T_{(k)} = 0$. By replacing an arbitrary equation with equation $\pi_{(k)}e_k = 1$, or equivalently, $e_k^T\pi^T_{(k)} = 1$, we obtain a system with better spectral properties, than when using condition $e^T\pi^T = 1$ [4].

When we use this normalization, we can state the following simple theorem.

**Theorem 1.** *Let the state space of a Markov chain can be decomposed into three groups $S_1, \{x_k\} = S_2, S_3$, so that in oriented graph of the Markov chain each path from $S_1$ to $S_3$ contains a vertex $x_k$. Then no modifications of transition probabilities between states of $S_1$ affect components in $\pi_{(k)}$ associated with states from $S_3$*

**Proof:** With given restrictions, the graph of the chain can be simplified into



At this picture, $S_1$ is denoted by 1, $x_k$ by k, $S_3$ by 3. It then follows that nonzero structure of $P$ is

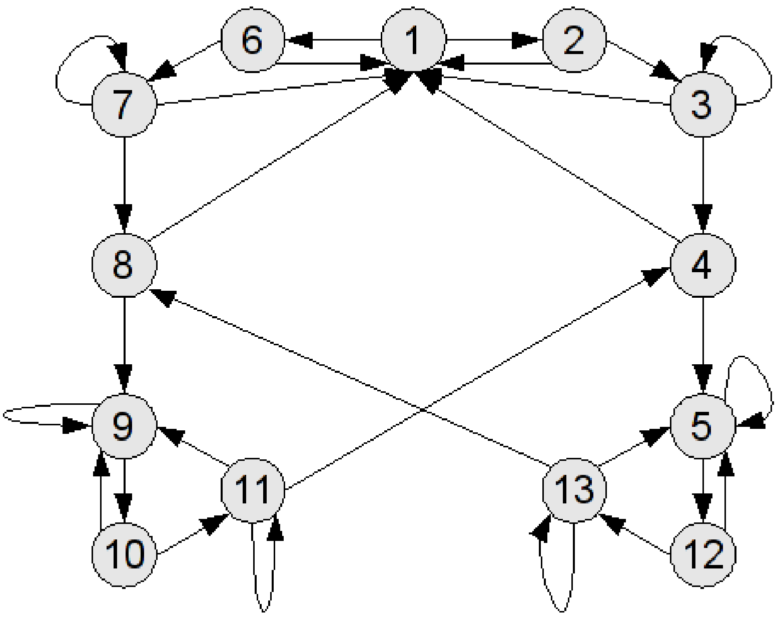
$$P = \left(\begin{array}{ccc|c|ccc} X & \dots & X & X & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X & \dots & X & X & 0 & \dots & 0 \\ \hline X & \dots & X & X & X & \dots & X \\ \hline X & \dots & X & X & X & \dots & X \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X & \dots & X & X & X & \dots & X \end{array}\right).$$

After forming left-hand side matrix $(I - P^T)$, we remove $k$-th equation and replace it with $e_k^T \pi_{(k)}^T = 1$. This way we obtain a system of equations with matrix $A^{(k)}$ and right-hand side $e_k$. $A^{(k)}$ has the following nonzero structure

$$A^{(k)} = \left(\begin{array}{ccc|c|ccc} X & \dots & X & X & X & \dots & X \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X & \dots & X & X & X & \dots & X \\ \hline 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \hline 0 & \dots & 0 & X & X & \dots & X \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & X & X & \dots & X \end{array}\right),$$

it is clearly reducible. Thus, no modifications of transition probabilities between states in $S_1$ (block $1, 1$ in $A^{(k)}$) will affect $k$-th,$\dots, n$-th components of $\pi_{(k)}$

**Example:** In example with $P_c$, we may draw an oriented graph:



We see that vertices $2, 3, 6, 7$ are accesible only through vertex 1. Thus if we fix the first element of $\pi$, no modifications of transition probabilities between vertices 4, 5, 8, 9, 10, 11, 12, 13 will affect components no. $2, 3, 6, 7$ in $\pi_{(1)}$.

### 3 Normalisation $\pi e = 1$

For a more usual normalisation $\pi e = 1$, let us first introduce a concept of lumpability

**Definition:** *Let us partition a transition probability matrix $P$ into blocks $(P_{ij})_{i,j=1}^{N}$ so that for every block $P_{ij}$ and vector $e^{(j)}$ of appropriate dimensions*

$$P_{ij}e^{(j)} = \alpha_{ij}e^{(j)}$$

*for some $\alpha_{ij} \in \mathbf{R}$.Then matrix $P$ is said to be* lumpable.

**Theorem 2.** *Let $P(t)$ be a perturbed transition probability matrix of an irreducible finite aperiodic Markov chain, whose state space divided into subsets $S_1, \ldots, S_{N+1}$, where states of $S_{N+1}$ are accessible only through $S_N$. Let perturbations depend on a variable $t$ and be restricted to lumpable submatrix of blocks $(P_{ij}(t))_{i,j=1}^{N-1}$ . If for every $i = 1, \ldots, N-1$ exists a column vector $x^{(i)}$ such that*

$$P_{i,N} = e^{(i)} \cdot x^{(i)T}, \tag{3}$$

*then subblocks $\pi^{(N)}, \pi^{(N+1)}$ are independent of t*

**Proof:** From the assumption it follows that

$$P_{ij}e^{(j)} = \alpha_{ij}e^{(j)},\ P_{i,N+1} = 0,\ i,j = 1,\ldots,N-1. \tag{4}$$

We will prove the theorem by using a power method for computing $\pi$. Assumptions guarrantee the existence of a unique steady point – eigenvector $\pi$ [4].

Let us choose a $\pi^{(0)} = \left(\pi_1^{(0)},\ldots,\pi_{N+1}^{(0)}\right)$, for $l = 1,2,\ldots$

$$\pi^{(l+1)} = \pi^{(l)}P.$$

a) At first we will show by induction, that for every $l$ the $\|\cdot\|_1$-norms of subvectors $\pi_1^{(l)},\ldots,\pi_{N+1}^{(l)}$ of $\pi^{(l)}$ do not depend on $t$. $\pi^{(0)}$ does not depend on $t$. The $l_1$-norm of the $j$-th subvector, $j = 1,\ldots,N-1$, in the $(l+1)$-th iteration is

$$\|\pi_j^{(l+1)}\|_1 = \pi^{(l)}P_{*,j}e^{(j)} = \sum_{i=1}^{N-1}\pi_i^{(l)}P_{i,j}(t)e^{(j)} + \sum_{i=N}^{N+1}\pi_i^{(l)}P_{i,j}e^{(j)} =$$

$$\sum_{i=1}^{N-1}\pi_i^{(l)}\alpha_{i,j}e^{(j)} + \sum_{i=N}^{N+1}\pi_i^{(l)}P_{i,j}e^{(j)},$$

which does not depend on $t$. For $j = N, N+1$ subblocks $P_{*,j}$ do not depend on $t$, thus $\|\pi_j^{(l+1)}\|_1 = \pi^{(l)}P_{*,j}e^{(j)}$ is also independent of $t$.

b) Now let us suppose that in iteration $\pi^{(l)}$ subvectors $N, N+1$ are independent of $t$. First, by (4) we have

$$\pi_{N+1}^{(l+1)} = \sum_{i=1}^{N+1}\pi_i^{(l)}P_{i,N+1} = \sum_{i=N}^{N+1}\pi_i^{(l)}P_{i,N+1},$$

which by induction hypothesis does not depend on $t$.

Finally, because of (3),

$$\pi_N^{(l+1)} = \sum_{i=1}^{N+1}\pi_i^{(l)}P_{i,N} = \sum_{i=1}^{N-1}\pi_i^{(l)}e^{(i)}x^{(i)T} + \pi_N^{(l)}P_{N,N} + \pi_{N+1}^{(l)}P_{N+1,N} =$$

$$= \sum_{i=1}^{N-1}\|\pi_i^{(l)}\|_1 x^{(i)T} + \pi_N^{(l)}P_{N,N} + \pi_{N+1}^{(l)}P_{N+1,N},$$

with all terms independent of $t$.

**Remark:** The above theorem holds also for periodic chains. If $P$ is a transition probability matrix of $p$-cyclic chain, it has exactly $p$ eigenvalues on a unit circle (one of them being 1). If we transform matrix $P$ onto

$$\tilde{P} = \alpha P + (1-\alpha)I,$$

we obtain a matrix with submatrix of subblocks $(\tilde{P}_{ij}(t))_{i,j=1}^{N-1}$ remaining lumpable and for $i = 1, \ldots, N-1$ we will have $P_{i,N} = e^{(i)} \cdot \alpha x^{(i)T}$. Furthermore $\pi\tilde{P} = \pi$ and all eigenvalues other than 1 will be inside the unit circle, ensuring convergence of power method.

**Example:** If we change the order of states in Markov chain represented by $P_c$ to 13, 11, 12, 10, 5, 9, 4, 8, 1, 2, 6, 3, 7, then the resulting chain has a transition probability matrix (zeros omited)

$$\bar{P}_c = \frac{1}{64}\left(\begin{array}{cc|cc|cc|cc|c|cc|cc}
60 & & & & 2 & & & 2 & & & & & \\
 & 60 & & & & 2 & 2 & & & & & & \\
\hline
2 & & & & 62 & & & & & & & & \\
 & 2 & & & & 62 & & & & & & & \\
\hline
 & & 1 & & 63 & & & & & & & & \\
 & & & 1 & & 63 & & & & & & & \\
\hline
 & & & & 2 & & & & 62 & & & & \\
 & & & & & 2 & & & 62 & & & & \\
\hline
 & & & & & & & & 62 & 1 & 1 & & \\
\hline
 & & & & & & & & 62 & & & 2 & \\
 & & & & & & & & 62 & & & & 2 \\
\hline
 & & & & & & 2 & & 2 & & & 60 & \\
 & & & & & & & 2 & 2 & & & & 60
\end{array}\right),$$

which is lumpable and if we have perturbations for example

$$\bar{p}_{11}(t) = \frac{60}{64} - t,\ \bar{p}_{12}(t) = t,\ \bar{p}_{66(t)} = \frac{63}{64} - 2t,\ \bar{p}_{65}(t) = 2t,$$

the resulting matrix satisfies conditions of the theorem.

## 4 Summary

This paper intends to present some conditions for insensitivity of a Markov chain towards perturbations in transition probability matrix. These conditions involve existence of cutpoints and regularity described by the concept of lumpability.

## References

[1] Cho, G.E. and Meyer, C.D.: Markov chain sensibility measured by mean first passage times. Lin. Alg. Appl. **316** (2000), 21–28.

[2] Deutch, E. and Neumann, M.: On the derivatives of the Perron vector. Portugaliae Mathematica **43** (1985-1986), 35–42.

[3] Kirkland, S.J., Neumann, M. and Sze, N.-S.: On optimal condition numbers for Markov chains. Numerische Mathematik **110** (2008), 521–537.

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

# PARALLEL SVD COMPUTATION[*]

Petr Kotas, Vít Vondrák, Pavel Praks

## 1 Introduction

The aim of this paper is to present experiments with parallel implementation of large scale singular value decomposition (SVD). The SVD has remarkable properties and it is widely used as a tool in matrix computations. However, there are problems with enormous computational demands of SVD. Recently there are many new SVD applications in the computational science. Just for an illustration we mention eigenfaces [5], which is probably one of the earliest computationally demanding application of the eigenvalue analysis applied to large data sets. Another widely used application is the Latent Semantic Indexing (LSI) [6]. LSI is used in data-mining and information retrieval communities for reducing dimension of a problem and for uncovering so called latent semantic, which is hidden in analyzed data.

In this paper we present a parallel implementation of bidiagonalization routine. Our main goal is to solve SVD for large matrices which cannot fit into the memory of standard PC and speed-up current algorithms porting them on massively parallel computers. We have implemented our version of parallel SVD algorithm in C++ programming language using Message Passing Interface (MPI). This allows us to utilize distributed resources and to load even huge data directly to the computer memory. Although other parallel implementations exist, many of them utilize multicore architectures to gain more speed-up with the same amount of local memory [8].

This paper is organized as follows: In Section 2 we present our parallel implementation of the bidiagonalization algorithm. Furthermore, in Section 3 we present efficiency of our algorithm on numerical experiments. Final comments and conclusions are presented in Section 4.

## 2 Computing SVD

The SVD computation consists of three consecutive steps: (i) bidiagonalization, (ii) computation of singular values and vectors, (iii) post-multiplication of results from previous two steps. In preprocessing stage the Householder bidiagonalization is used. This method utilizes the Householder reflection

$$H = I - 2vv^*, \tag{1}$$

[*] This work was supported by grant #GD103/09/H078 of Grant Agency of the Czech Republic and grant #SP/2010173 of Students Grant Competition VSB-Technical University of Ostrava.

**Algorithm 1** Parallel bidiagonalization

```
Input:$A$ distributed to all nodes
Output:$B$, bidiagonal matrix
$[m, n] \leftarrow$ size($A$)
for $k = 1$ to $min(m, n)$ do
    activeColumn $\leftarrow$ allGather($A_{loc}(:, k)$)
    $v \leftarrow$ householder(activeColumn)
    for $j = 1$ to $n$ do
        $\gamma_{loc}(j) = v^T A_{loc}(:, j)$
    end for
    $\gamma \leftarrow$ allReduce($\gamma_{loc}$)
    for $j = k$ to $n$ do
        $A_{loc}(:, j) \leftarrow A_{loc}(:, j) - 2\frac{\gamma(j)}{v^T v} v$
    end for
    if $k < (n - 2)$ then
        if node has $A_{loc}(k, :)$ then
            broadcast($A_{loc}(k, :)$)
            activeRow $\leftarrow A_{loc}(k, :)$
        else
            activeRow $\leftarrow$receive($A_{loc}(k, :)$)
        end if
        $v \leftarrow$householder(activeRow)
        for $i = k$ to $m$ do
            $\gamma \leftarrow A_{loc}(i, :)v$
            $A_{loc}(i, :) \leftarrow A_{loc}(i, :) - 2\frac{\gamma}{v^T v} v$
        end for
    end if
end for
$B \leftarrow A$
```

where $v = x \pm \|x\|_2 e, v \in \mathbb{R}^n$ is the Householder vector. For further details on bidiagonalization see [1]. In Algorithm 1 we propose our parallel version of basic bidiagonalization routine defined in [1]. This is optimized version without implicit accumulation of orthogonal transformation matrices. Function `householder` in Algorithm 1 denotes the standard Householder reflector as in [1]. The singular values of bidiagonal matrix computed by Algorithm 1 are the same as the singular values of the original matrix $A$. This is obvious fact since the Householder reflection preserves the orthogonality among singular vectors, as has been proved in [1].

In the second step a diagonalization of bidiagonal matrix $B$ computed in first step is performed. The resulting diagonal matrix consists only from singular values of the bidiagonal matrix $B$. Our diagonalizatin routine uses sequential implicit QR algorithm as it is described in [2], and can be theoretically implemented for massively parallel computers. At this time, we use the `LAPACK`[1] sequential function `_BDSDC`, which computes singular values of real bidiagonal matrix $B$. Therefore, the diagonalization part represents the bottleneck of our algorithm.

In the third step of SVD, the Householder matrices ($U_H$ and $V_H$), the matrix of singular values ($\Sigma$) as well as singular vectors of the bidiagonal matrix ($U_B$ and $V_B$) are assembled. To complete the whole decomposition, one only needs to multiply

$$\begin{aligned} U &= U_H \cdot U_B, \\ V &= V_B \cdot V_H. \end{aligned}$$

After this final step the full SVD decomposition of an arbitrary real matrix $A$ is obtained.

## 3 Numerical experiments

The overall execution time of $T_p$ Algorithm 1 is given by the following equation

$$T_p = t_c \frac{n^3}{3p} + 2t_s n + t_w \frac{n^2}{\sqrt{p}}, \qquad (2)$$

where $t_c$ is time needed for computing one FLOP[2], $t_s$ and $t_w$ denote both send and wait latencies of MPI, $n = \max(rows, cols)$ is dimension of the matrix $A$ and $p$ is the number of processors.

All experiments presented in this section are computed on cluster Teri with hardware configuration: 32x Intel Xeon QuadCore 2.5 GHz, 18 GB RAM, 4XDDR IB Mezzanine HCA 20Gb/s FullDuplex (per node). All experiments were performed on dense random matrices with sizes: $8 \times 8$, $16 \times 16$, $32 \times 32$, ..., $4096 \times 4096$.

Figure 1 compares theoretical time computed by (2) with real execution times of our implementation of Algorithm 1. Times $t_s$, $t_w$ and $t_c$ are estimated from measurements performed on cluster Teri. However, latency times $t_s$ and $t_w$ are heavily dependent on current load of computational cluster, which means execution time $T_p$ could vary. Significant increase of execution times for 64 and 128 processors is caused by raising rate of MPI communication. This problem could be solved with more suitable decomposition scheme.

In order to measure the performance of our implementation of bidiagonalization with fixed number of processors, we carried out several series of tests with varying size of the problem. Figure 2 shows total bidiagonalization time (including MPI communication) and MPI communication itself.

Similar tests were run to show behavior of our algorithm with increasing number of processors.

[1] `LAPACK` - Linear Algebra PACKage `http://www.netlib.org/lapack/`

[2] FLOP is abbreviation for floating point operation.

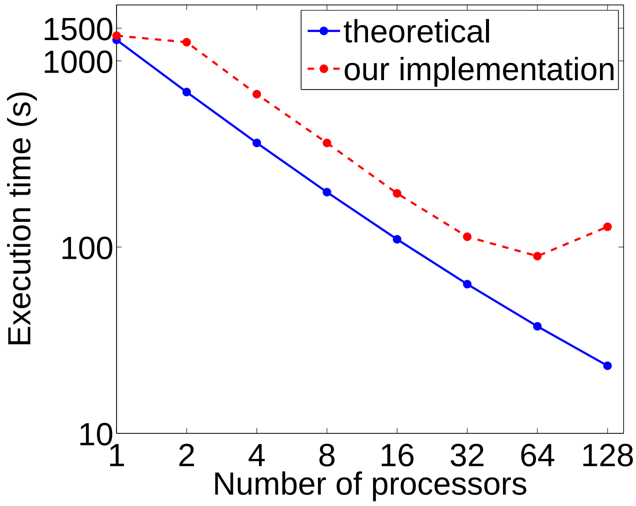
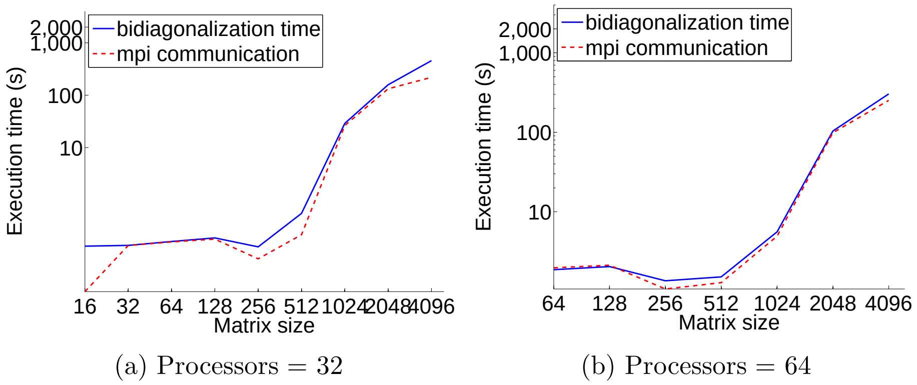


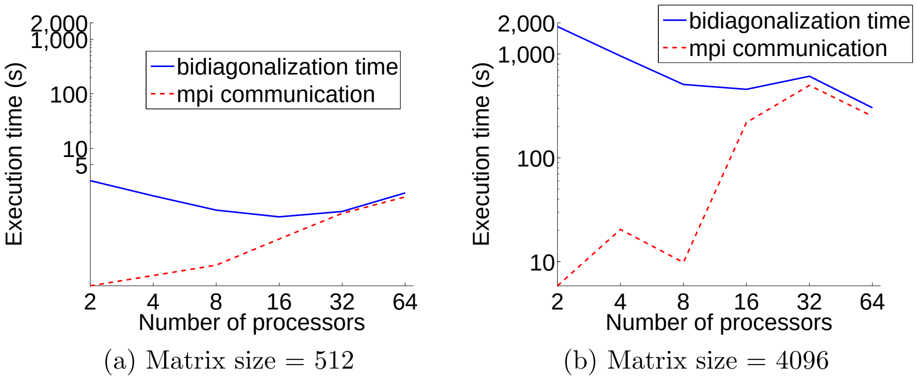
**Fig. 1**: *Bidiagonalization of the 4096 × 4096 matrix.*



(a) Processors = 32

(b) Processors = 64

**Fig. 2**: *Fixed number of processors.*



(a) Matrix size = 512

(b) Matrix size = 4096

**Fig. 3**: *Fixed matrix size.*

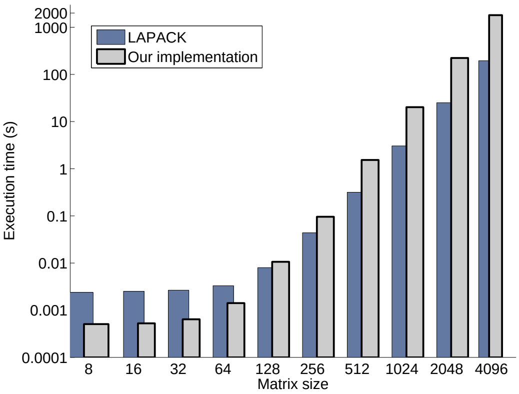


**Fig. 4**: *Sequential version of bidiagonalization compared to* `LAPACK`.

Figure 4 plots the execution times taken by bidiagonalization routine DGBERD from `LAPACK/ATLAS` library and our sequential version of Algorithm 1, respectively. We can see that both implementations have similar time evolution. But finally, the `LAPACK` sequential implementation seems to be faster because it uses optimized `BLAS` libraries.

The main advantage of our algorithm is its ability to process even large-scale data. We tried to decompose some very large problems. The largest matrix decomposed by our algorithm had dimension 32768 × 32768, which required $8.1GB$ of memory. We used 32 processors and our algorithm had been running for 32.32 hours. The MPI communication required 1.79 hours.

## 4 Comments and conclusions

The advantage of our implementation is effective handling of large dense problems. On the other hand, it seems that our algorithm is less effective in term of parallel scalability for more than 32 processors. This problem could be solved by more sophisticated decomposition scheme, which is left for further research. Further, improvements could be done utilizing the parallel implementation of the diagonalization routine and by using both MPI and OpenMP libraries. These improvements could lead to a significant speed-up, especially for large tasks running on large numbers of processors.

## References

[1] Golub, G.H. and Van Loan, Ch.F.: *Matrix computations*. The Johns Hopkins University Press; 3rd edition, 1998.

[2] Kotas, P.: *Efficient implementation of SVD and its application to biometric data processing*. VŠB - Diploma thesis, 2009.

[3] Gu, M. and Eisenstat, S.C.: A divide-and-conquer algorithm for the bidiagonal SVD. SIAM J. Mat. Anal. Appl. **16** (1995), 79–92.

[4] Jessup, E. and Sorensen, D.: A parallel algorithm for computing the singular value decomposition of a matrix. Mathematics and Computer Science Division Report ANL/MCS-TM-102, Argonne National Laboratory, Argonne, IL, December 1987.

[5] Turk, M. and Pentland, A.: Face recognition using eigenfaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.

[6] Letsche, T.A. and Berry, M.W.: Large-scale information retrieval with latent semantic indexing. Information Sciences **100** 1-4 (August 1997), 105–137.

[7] Larsen, R.M.: Lanczos bidiagonalization with partial reorthogonalization. Part of documentation to software package PROPACK, 1998.

[8] Ltaief, H., Kurzak, J., and Dongarra, J.: Parallel two-sided matrix reduction to band bidiagonal form on multicore architectures. IEEE Transactions on Parallel and Distributed Systems **99** (2009), 417–423.

# INTRODUCTION TO ALGORITHMS FOR MOLECULAR SIMULATIONS

Martin Kramář

**Abstract**

In the first part of the paper we survey some algorithms which describe time evolution of interacting particles in a bounded domain. Applications to macroscale as well as microscale are presented on two examples: motion of planets and collision of two bodies. In the second part of the paper we present solution to stationary Schrödinger equation for simple molecular models.

## 1 Algorithm for dynamic simulation

We consider a system of $n$ particles, which are determined by their weights $\{m_1, \ldots, m_n\}$, positions $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and velocities $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. We assume the following computational domain: $\Omega = [0, L_1] \times [0, L_2]$ or $\Omega = [0, L_1] \times [0, L_2] \times [0, L_3]$ in two or three dimensions, respectively. Further we assume that particles that left $\Omega$ do not interact any longer with those in $\Omega$. The time evolution is described by Newton's equation of motion $m_i\dot{\mathbf{v}}_i = \mathbf{F}_i$ or $m_i\ddot{\mathbf{x}}_i = \mathbf{F}_i, i = 1, \ldots, n$ [1].

### 1.1 Velocity-Störmer-Verlet method

We consider the time interval $[0, t_{end}]$ to be discretized into subintervals with the step $\delta t$ so that Newton's equation of motion are replaced with an algebraic system at $n \cdot \delta t$ where $n = 1, \cdots, n_{t_{end}}$, while using the second central difference $\left[\frac{d^2x}{dt^2}\right]_n = \dfrac{1}{\delta t^2}(x(t_{n+1}) - 2x(t_n) + x(t_{n-1}))$. This leads to the so-called velocity Störmer-Verlet method

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \delta t \mathbf{v}_i^n + \mathbf{F}_i^n \cdot \delta t^2/(2m_i), \tag{1}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + (\mathbf{F}_i^n + \mathbf{F}_i^{n+1})\delta t/(2m_i). \tag{2}$$

where we denote the positions by $\mathbf{x}_i^n = \mathbf{x}_i(t_n)$ and analogously $\mathbf{F}_i^n$ and $\mathbf{v}_i^n$ stand for forces and velocities, respectively.

We consider the gravitational force $\mathbf{F}_i = \sum_{j=1, j\neq i}^{n} \mathbf{F}_{ij}$ where $\mathbf{F}_{ij} = \dfrac{m_i m_j}{r_{ij}^3}\mathbf{r}_{ij}$. The method is demonstrated on a simplified 2-dimensional model, which consists of the Sun, the Earth, the Jupiter, and Halley's Comet. Figure 1 shows the resulting orbits and initial data.

| | | |
|---|---|---|
| $m_{sun} = 1$ | $\mathbf{x}^0_{sun} = (0, 0)$ | $\mathbf{v}^0_{sun} = (0, 0)$ |
| $m_{Earth} = 3 \cdot 10^{-6}$ | $\mathbf{x}^0_{Earth} = (0, 1)$ | $\mathbf{v}^0_{Earth} = (-1, 0)$ |
| $m_{Jupiter} = 9.55 \cdot 10^{-4}$ | $\mathbf{x}^0_{Jupiter} = (0, 5.36)$ | $\mathbf{v}^0_{Jupiter} = (-0.425, 0)$ |
| $m_{Halley} = 1 \cdot 10^{-14}$ | $\mathbf{x}^0_{Halley} = (34.75, 0)$ | $\mathbf{v}^0_{Halley} = (0, 0.0296)$ |
| $\delta t = 0.015$ | $t_{end} = 468.5$ | |

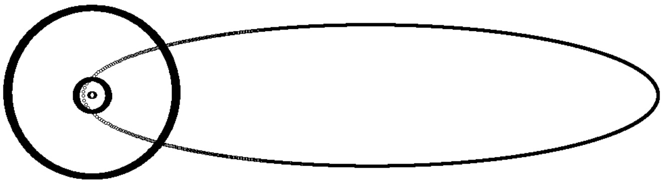


**Fig. 1:** *Trajectories of Halley's Comet, the Sun, the Earth and Jupiter. In this model are all masses divided by the mass of the Sun, all distances are divided by the $d_{ES}$ (distance from the Earth to the Sun) and all velocities are divided by the $v_E$ (velocity of the Earth). It means that the time is divided by $d_{ES}/v_E$.*

### 1.2 Cutoff radius

Calculation of forces is very time consuming for system with thousands and more mutually interacting particles. We shall accelerate the computation by considering only interactions of particles in a given neighbourhood by which we reduce the complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.
As a model, we choose the Lennard-Jones potential [1]

$$U(r_{ij}) = 4 \cdot \varepsilon \left(\frac{\sigma}{r_{ij}}\right)^6 \cdot \left(\left(\frac{\sigma}{r_{ij}}\right)^6 - 1\right), \tag{3}$$

where $\sigma > 0$ is the distance, at which the force switches between repulsive and attractive and $\varepsilon$ is depth of the potential.

The approximation of the potential function for $n$ particles is truncated double sum

$$V(\mathbf{x}_1, \ldots, \mathbf{v}_n) = \sum_{i=1}^{n} \sum_{j:0<r_{ij}\leq r_{cut}}^{n} U(r_{ij}), \tag{4}$$

and the approximation of the corresponding force $\mathbf{F}_i$ on the particle $i$ is given by

$$\mathbf{F}_i = -\nabla_{x_i} V(\mathbf{x}_1, \ldots, \mathbf{x}_n) = 24 \cdot \varepsilon \cdot \sum_{0<r_{ij}\leq r_{cut}}^{n} \frac{1}{r_{ij}^2} \left(\frac{\sigma}{r_{ij}}\right)^6 \cdot \left(1 - 2 \cdot \left(\frac{\sigma}{r_{ij}}\right)^6\right) \mathbf{r}_{ij}. \tag{5}$$

where $r_{cut}$ is chosen $2.5 \cdot \sigma$ typically [1].

The algorithm was applied to a problem of collision of two bodies which are created from 10 × 10 and 30 × 10 particles of equal mass, respectively, arranged on a lattice of mesh size $2^{1/6} \cdot \sigma$. For the initial data and numerical simulation we refer to Figure 2.

$$\begin{array}{lllll} L_1 = 50 & \varepsilon = 5 & \mathbf{v} = (0,-10) & N_1 = 100 & r_{cut} = 2.5\sigma \\ L_2 = 50 & \sigma = 1 & m = 1 & N_2 = 300 & \delta t = 0.00005 \end{array}$$

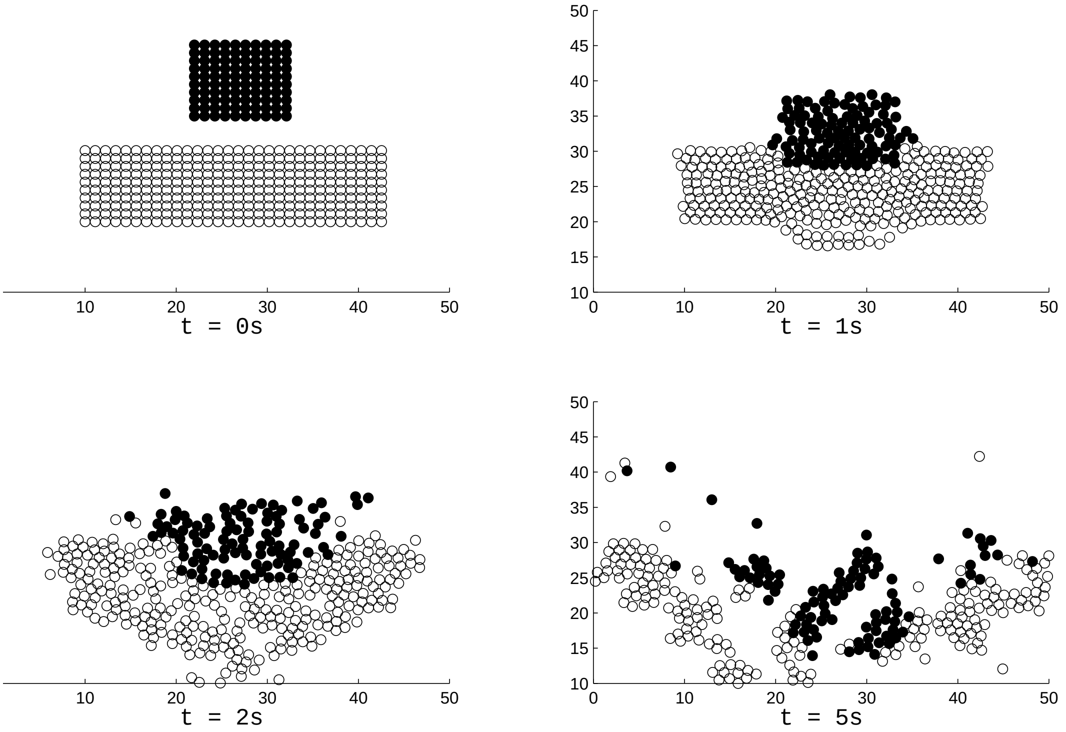


**Fig. 2:** *Collision of two bodies. Time evolution of the distribution of the particles.*

## 2 Time indenpendent Schrödinger equation

Consider a single particle. Schrödinger wave function $\Psi(x,y,z)$ is quantity which describes state of the particle. It is related to the probability $\rho(x,y,z)$ that a particle is at a given position by $\rho = \Psi^* \cdot \Psi$ where $\Psi^*$ is complex conjugate to function $\Psi$. The Schrödinger equation reads as follows: $\mathcal{H}\Psi = E\Psi$ with the Hamilton operator

$$\mathcal{H} = -\frac{h^2}{8\pi^2 m}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + \mathcal{V}(x), \tag{6}$$

where $\mathcal{V}(x)$ is operator of the potential energy, $m$ is mass of the particle and $h$ is the Planck constant. After adjustment we can write the Schrödinger wave equation in the form

$$\frac{\mathrm{d}^2\Psi}{\mathrm{d}x^2} + \frac{8\pi^2 m}{h^2}(E - \mathcal{V})\Psi = 0. \tag{7}$$

### 2.1 Particle in the potential well

We consider a particle moving inside one-dimensional potential well in the direction of the $x$-axis. We assume that the particle has the same potential energy at any point of the well. It is useful to put the potential energy equal to zero in the well and equal to infinity elsewhere. Schrödinger wave equation for the particle in the potential well is written in the form (because $\mathcal{V} = 0$)

$$\frac{\mathrm{d}^2\Psi(x)}{\mathrm{d}x^2} + \frac{8\pi^2 m}{h^2} E\Psi(x) = 0, \tag{8}$$

$$\Psi(0) = \Psi(a) = 0. \tag{9}$$

The solution to (8) and (9) reads as follows:

$$\Psi_n(x) = \sqrt{2/a}\sin(n\pi x/a), \quad E_n = (nh)^2/(8ma^2), \quad n = 1, 2, 3, \ldots, \tag{10}$$

where normalization factor $\sqrt{2/a}$ results from $\int_0^a \Psi^2(x)\mathrm{d}x = 1$ and $a$ is the width of the well. The analytical solution is shown in Figure 3.

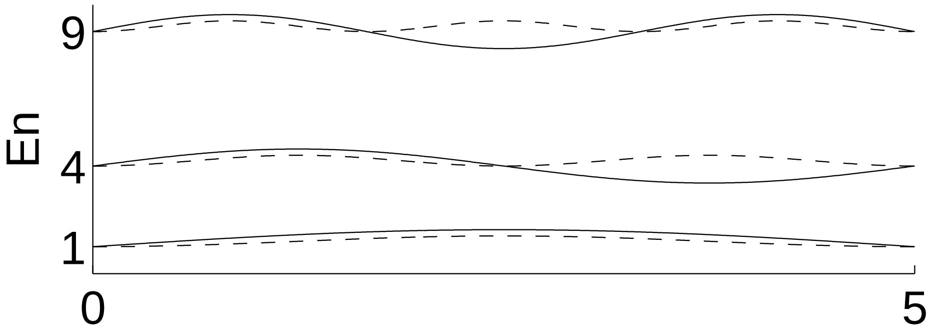


**Fig. 3:** *Potential well. Wave functions $\Psi_i$ (solid line) and the probability functions $\Psi_i^*\Psi_i$ (dashed line) shifted vertically by the related energies $E_i$ for $i = 1, 2, 3$.*

### 2.2 Harmonic oscillator

Another simple system of quantum mechanics is a harmonic motion of a particle. This system is interesting for us because it plays important role in the reasoning in molecular spectroscopy.

We consider a particle of the mass $m$ which is moving along the $x$-axis alternately in positive and negative direction so that point $x = 0$ is the equilibrium. The acting force is given by $F = -kx$ with $k > 0$, which results in the potential energy $V = -\int_0^x(-kx)\mathrm{d}x = \frac{1}{2}kx^2$ and the following Schrödinger equation

$$\frac{\mathrm{d}^2\Psi(x)}{\mathrm{d}x^2} + \frac{8\pi^2 m}{h^2}(E - \frac{1}{2}kx^2)\Psi(x) = 0. \tag{11}$$

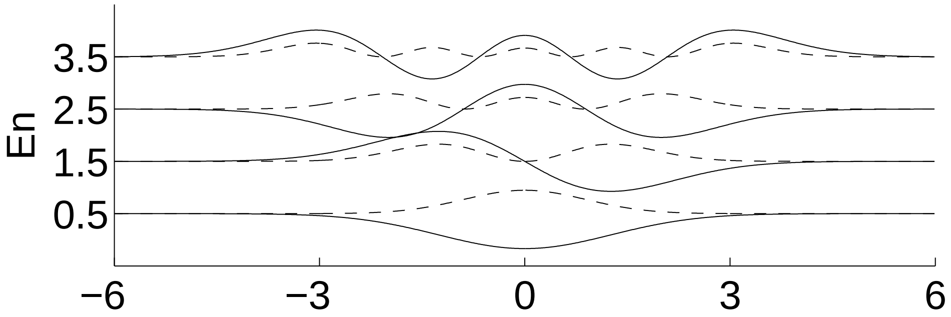


**Fig. 4:** *Harmonic oscillator. Wave functions $\Psi_i$ (solid line) and the probability functions $\Psi_i^*\Psi_i$ (dashed line) shifted vertically by the related energies $E_i$ for $i = 1, 2, 3, 4$.*

The solution can be found by cutting off infinite power series and total energy can be computed as [2]

$$E_n = h/(2\pi) \cdot \sqrt{k/m} \cdot (n + 1/2), \quad n = 0, 1, 2, \dots . \tag{12}$$

The solution can be found also by using numerical solvers. We solve problem (11), after adjustment we obtain

$$\frac{\mathrm{d}^2\Psi(x)}{\mathrm{d}x^2} + (\lambda - b^2x^2)\Psi(x) = 0, \quad \lambda = \frac{8\pi^2 mE}{h^2} \quad b = \frac{2\pi\sqrt{mk}}{h} \tag{13}$$

$$\Psi(x) = 0, \quad \text{for } |x| \to \infty. \tag{14}$$

We assume that the wave function is close to zero at distance $l$. We can write variational formulation of problem (13) and (14) and look for $\lambda > 0$ and $\Psi(x) \in H_0^1(-l, l)$ such that

$$\int_{-l}^{l} \Psi'(x)v'(x)\mathrm{d}x + b^2 \int_{-l}^{l} x^2\Psi(x)v(x)\mathrm{d}x = \lambda \int_{-l}^{l} \Psi(x)v(x)\mathrm{d}x, \quad \forall v \in H_0^1(-l, l).$$

We employ a finite element discretization to the latter formulation which leads to an approximate solution $\Psi_h(x) = \sum_{i=1}^{n} \psi_i\varphi_i(x)$ with continuous piecewise linear basis function $\varphi_i(x)$. The coefficients vector $\bar{\Psi}$ and energies $\lambda$ solve the following eigenvalue problem

$$\begin{aligned} &A\bar{\Psi} = \lambda B\bar{\Psi}, \\ &[A]_{ij} = \int_{-l}^{l} \varphi_i'(x)\varphi_j'(x)\mathrm{d}x + b^2 \int_{-l}^{l} x^2\varphi_i(x)\varphi_j(x)\mathrm{d}x, \\ &[B]_{ij} = \int_{-l}^{l} \varphi_i(x)\varphi_j(x)\mathrm{d}x. \end{aligned}$$

The Matlab software was use to solve the eigenvalue problem. Figure 4 shows the solution.

## Conclusion

In this paper we have presented some methods for dynamic simulation of particle effects. In the second part we showed analytical solution of the simple problems in quantum mechanics and numerical approach using the finite element method.

## References

[1] Griebel, M., Knapek, S., and Zumbusch, G.: *Numerical simulation in molecular dynamics.* Springer-Verlag, Berlin Heidelberg, 2007.

[2] Polák, R. and Zahradník, R.: *Kvantová chemie: Základy teorie a aplikace.* SNTL, Praha, 1985.

# A NEW RECONSTRUCTION-ENHANCED DISCONTINUOUS GALERKIN METHOD FOR TIME-DEPENDENT PROBLEMS*

Václav Kučera

**Abstract**

This work is concerned with the introduction of a new numerical scheme based on the discontinuous Galerkin (DG) method. We propose to follow the methodology of higher order finite volume schemes and introduce a reconstruction operator into the DG scheme. This operator constructs higher order piecewise polynomial reconstructions from the lower order DG scheme. Such a procedure was proposed already in [2] based on heuristic arguments, however we provide a rigorous derivation, which justifies the increased order of accuracy. Numerical experiments are carried out.

## 1 Problem formulation and notation

In this paper we shall be concerned with a nonlinear nonstationary scalar hyperbolic equation in a bounded domain $\Omega \subset I\!R^d$ with a Lipschitz-continuous boundary $\partial\Omega$. Let $Q_T := \Omega \times (0,T)$. We treat the following problem:

$$\frac{\partial u}{\partial t} + \mathrm{div}\boldsymbol{f}(\mathrm{u}) = 0 \quad \mathrm{in\ Q_T} \tag{1}$$

along with an appropriate initial and boundary condition. Here $\boldsymbol{f} = (f_1, \cdots, f_d)$ and $f_s, s = 1, \dots, d$ are Lipschitz continuous fluxes in the direction $x_s, s = 1, \dots, d$.

Let $\mathcal{T}_h$ be a partition (triangulation) of the closure $\overline{\Omega}$ into a finite number of closed simplices $K \in \mathcal{T}_h$. In general we do not require the standard conforming properties of $\mathcal{T}_h$ used in the finite element method (i.e. we admit the so-called hanging nodes). We shall use the following notation. By $\partial K$ we denote the boundary of an element $K \in \mathcal{T}_h$ and set $h_K = \mathrm{diam}(K)$, $h = \max_{K\in\mathcal{T}_h} h_K$.

Let $K, K' \in \mathcal{T}_h$. We say that $K$ and $K'$ are *neighbours*, if they share a common *face* $\Gamma \subset \partial K$. By $\mathcal{F}_h$ we denote the system of all faces of all elements $K \in \mathcal{T}_h$. Further, we define the set of all interior and boundary faces, respectively, by

$$\mathcal{F}_h^I = \{\Gamma \in \mathcal{F}_h;\ \Gamma \subset \Omega\}, \qquad \mathcal{F}_h^B = \{\Gamma \in \mathcal{F}_h;\ \Gamma \subset \partial\Omega\}$$

For each $\Gamma \in \mathcal{F}_h$ we define a unit normal vector $\mathbf{n}_\Gamma$, such that for $\Gamma \in \mathcal{F}_h^B$ the normal $\mathbf{n}_\Gamma$ has the same orientation as the outer normal to $\partial\Omega$.

Over a triangulation $\mathcal{T}_h$ we define the *broken Sobolev spaces*

$$H^k(\Omega, \mathcal{T}_h) = \{v;\ v|_K \in H^k(K),\ \forall K \in \mathcal{T}_h\}.$$

[*] This work is a part of the research project No. 201/08/0012 of the Czech Science Foundation.

For each face $\Gamma \in \mathcal{F}_h^I$ there exist two neighbours $K_\Gamma^{(L)}$, $K_\Gamma^{(R)} \in \mathcal{T}_h$ such that $\Gamma \subset K_\Gamma^{(L)} \cap K_\Gamma^{(R)}$. We use the convention that $\mathbf{n}_\Gamma$ is the outer normal to $K_\Gamma^{(L)}$. For $v \in H^1(\Omega, \mathcal{T}_h)$ and $\Gamma \in \mathcal{F}_h^I$ we introduce the following notation:

$$\begin{aligned} v|_\Gamma^{(L)} &= \text{ the trace of } v|_{K_\Gamma^{(L)}} \text{ on } \Gamma, & v|_\Gamma^{(R)} &= \text{ the trace of } v|_{K_\Gamma^{(R)}} \text{ on } \Gamma, \\ \langle v \rangle_\Gamma &= \tfrac{1}{2}\big(v|_\Gamma^{(L)} + v|_\Gamma^{(R)}\big), & [v]_\Gamma &= v|_\Gamma^{(L)} - v|_\Gamma^{(R)}. \end{aligned}$$

On boundary edges $\Gamma \in \mathcal{F}_h^B$, we define $v|_\Gamma^{(R)} = 0$, $[v]_\Gamma = v|_\Gamma^{(L)}$.

Let $n \geq 1$ be an integer. The approximate solution will be sought in the space of discontinuous piecewise polynomial functions

$$S_h^n = \{v;\, v|_K \in P^n(K), \forall K \in \mathcal{T}_h\},$$

where $P^n(K)$ denotes the space of all polynomials on $K$ of degree $\leq n$.

## 2 Discontinuous Galerkin (DG) formulation

We multiply (1) by an arbitrary $\varphi_h^n \in S_h^n$, integrate over an element $K \in \mathcal{T}_h$ and apply Green's theorem. By summing over all $K \in \mathcal{T}_h$ and rearranging, we get

$$\frac{d}{dt}\int_\Omega u(t)\,\varphi_h^n\,dx + \sum_{\Gamma\in\mathcal{F}_h}\int_\Gamma \boldsymbol{f}(u)\cdot\mathbf{n}\,[\varphi_h^n]\,dS - \sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{f}(u)\cdot\nabla\varphi_h^n\,dx = 0. \tag{2}$$

The boundary convective terms will be treated similarly as in the finite volume method, i.e. with the aid of a numerical flux $H(u, v, \mathbf{n})$:

$$\int_\Gamma \boldsymbol{f}(u)\cdot\mathbf{n}\,[\varphi_h^n]\,dS \approx \int_\Gamma H(u^{(L)}, u^{(R)}, \mathbf{n})[\varphi_h^n]\,dS. \tag{3}$$

We assume that $H$ is *Lipschitz continuous, consistent* and *conservative*, cf. [4].

Thus, we obtain the following standard DG formulation

$$\frac{d}{dt}\big(u_h(t), \varphi_h^n\big) + b_h\big(u_h(t), \varphi_h^n\big) = 0, \quad \forall \varphi_h^n \in S_h^n,\ \forall t \in (0, T), \tag{4}$$

where $b_h(\cdot,\cdot)$ is the *convective form* defined for $v, \varphi \in H^1(\Omega, \mathcal{T}_h)$:

$$b_h(v, \varphi) = \int_{\mathcal{F}_h} H(v^{(L)}, v^{(R)}, \mathbf{n})[\varphi]\,dS - \sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{f}(v)\cdot\nabla\varphi\,dx.$$

## 3 Reconstructed discontinuous Galerkin (RDG) formulation

For $v \in L^2(\Omega)$, we denote by $\Pi_h^n v$ the $L^2(\Omega)$-projection of $v$ on $S_h^n$:

$$\Pi_h^n v \in S_h^n, \quad (\Pi_h^n v - v,\, \varphi_h^n) = 0, \qquad \forall\, \varphi_h^n \in S_h^n. \tag{5}$$

The basis of the proposed method lies in the observation that (2) can be viewed as an equation for the evolution of $\Pi_h^n u(t)$, where $u$ is the exact solution of (1). In other words, due to (5), $\Pi_h^n u(t) \in S_h^n$ satisfies the following equation for all $\varphi_h^n \in S_h^n$:

$$\frac{d}{dt}\int_\Omega \Pi_h^n u(t)\, \varphi_h^n\, dx + \int_{\mathcal{F}_h} \boldsymbol{f}(u)\cdot \mathbf{n}\, [\varphi_h^n]\, dS - \sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{f}(u)\cdot\nabla\varphi_h^n\, dx = 0. \tag{6}$$

Now, let $N > n$ be an integer. We assume, that there exists a piecewise polynomial function $U_h^N(t) \in S_h^N$, which is an approximation of $u(t)$ of order $N+1$, i.e.

$$U_h^N(x,t) = u(x,t) + O(h^{N+1}), \quad \forall x \in \Omega,\ \forall t \in [0,T]. \tag{7}$$

This is possible, if $u$ is sufficiently regular in space, e.g. $u(t) \in W^{N+1,\infty}(\Omega)$, cf.[1]. Now we incorporate the approximation $U_h^N(t)$ into (6): the exact solution $u$ satisfies

$$\frac{d}{dt}\big(\Pi_h^n u(t), \varphi_h^n\big) + b_h\big(U_h^N(t), \varphi_h^n\big) = E(\varphi_h^n), \quad \forall \varphi_h^n \in S_h^n,\ \forall t \in (0,T), \tag{8}$$

where $E(\varphi_h^n)$ is an error term defined as

$$E(\varphi_h^n) = b_h\big(U_h^N(t), \varphi_h^n\big) - b_h\big(u(t), \varphi_h^n\big). \tag{9}$$

**Lemma 31** *The following estimate holds:*

$$E(\varphi_h^n) = O(h^N)\|\varphi_h^n\|_{L^2(\Omega)}. \tag{10}$$

*Proof:* Due to assumptions (H1) and (H2) it is easy to see that on an edge $\Gamma \in \mathcal{F}_h$

$$\boldsymbol{f}(u)\cdot\mathbf{n} - H(U_h^{N,(L)}, U_h^{N,(R)}, \mathbf{n}) = H(u,u,\mathbf{n}) - H(U_h^{N,(L)}, U_h^{N,(R)}, \mathbf{n}) = O(h^{N+1}).$$

Furthermore, due to the Lipschitz-continuity of $f_s$, we have on element $K \in \mathcal{T}_h$

$$\boldsymbol{f}(u) - \boldsymbol{f}(U_h^N) = O(h^{N+1}).$$

Estimate (10) follows from these results and the application of the *inverse* and *multiplicative trace inequalities*, cf [4]. □

It remains to construct a sufficiently accurate approximation $U_h^N(t) \in S_h^N$ to $u(t)$, such that (7) is satisfied. This leads to the following problem.

**Definition 32 (Reconstruction problem.)** *Let $v : \Omega \to I\!R$ be sufficiently regular. Given $\Pi_h^n v \in S_h^n$, find $v_h^N \in S_h^N$ such that $v - v_h^N = O(h^{N+1})$ in $\Omega$. We define the corresponding reconstruction operator $R : S_h^n \to S_h^N$ by $R\,\Pi_h^n v := v_h^N$.*

By setting $U_h^N(t) := R\,\Pi_h^n u(t)$ in (8)-(10), we obtain the following equation for the $L^2(\Omega)$-projections of the exact solution $u$ onto the space $S_h^n$:

$$\frac{d}{dt}\big(\Pi_h^n u(t), \varphi_h^n\big) + b_h\big(R\,\Pi_h^n u(t), \varphi_h^n\big) = O(h^N)\|\varphi_h^n\|_{L^2(\Omega)}, \quad \forall \varphi_h^n \in S_h^n. \tag{11}$$

By neglecting the right-hand side term and introducing the approximation $u_h^n(t) \approx \Pi_h^n u(t)$, we arrive at the following definition of the *reconstructed discontinuous Galerkin* (RDG) scheme. We seek $u_h^n$ such that

$$\frac{d}{dt}\big(u_h^n(t), \varphi_h^n\big) + b_h\big(Ru_h^n(t), \varphi_h^n\big) = 0, \quad \forall \varphi_h^n \in S_h^n,\ \forall t \in (0,T). \tag{12}$$

There are several points worth mentioning.

- The derivation of the RDG scheme follows the methodology of higher order finite volume schemes and spectral volume schemes, cf. [7]. The basis of these schemes is an equation for the evolution of averages of the exact solution on individual elements (i.e. an equation for $\Pi_h^0 u(t)$). Equation (11) is a direct generalization for the case of higher order $L^2(\Omega)$-projections $\Pi_h^n u(t)$, $n \geq 0$.
- Both $u_h^n(t)$ and $\varphi_h^n$ lie in $S_h^n$. Only $Ru_h^n(t)$, lies in the higher dimensional space $S_h^N$. Despite this fact, equation (11) indicates, that we may expect $u - Ru_h^n = O(h^{N+1})$, although $u - u_h^n = O(h^{n+1})$.
- Numerical quadrature must be employed to evaluate surface and volume integrals in (12). Since test functions are in $S_h^n$, as compared to $S_h^N$ in the corresponding $N$th order standard DG scheme, we may use lower order (i.e. more efficient) quadrature formulae as compared to standard DG.
- In practice, an explicit time discretization must be applied to (12) The upper limit on stable time steps, given by a CFL-like condition, is more restrictive with growing $N$. However, in the RDG scheme, stability properties are inherited from the lower order scheme, therefore a larger time step is possible as compared to the standard DG scheme.

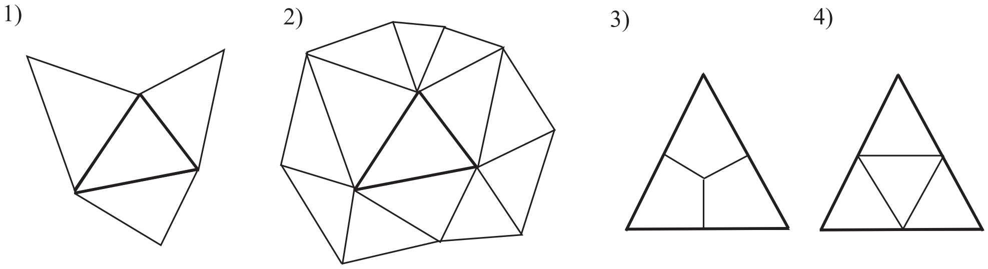


**Fig. 1:** *1) FV stencil for linear reconstruction, 2) FV stencil for quadratic reconstruction, 3) Control volumes in a spectral volume for linear reconstruction, 4) Analogy to the SV approach for DG - partition of triangle into control volumes, e.g. cubic reconstruction from linear data.*

### 3.1 Construction of the reconstruction operator

In analogy to the construction of reconstruction operators in higher order FV schemes, we propose two approaches.

### 3.1.1 'Standard' approach

In the *standard approach*, a stencil (a group of neighboring elements and the element under consideration) is used to build an $N$th-degree polynomial approximation to $u$ on the element under consideration ([5] [6]). In the FV method, the von Neumann neighborhood of an element is used as a stencil to obtain a piecewise linear reconstruction, Figure 1, 1). However, for higher order reconstructions, the size of the stencil increases dramatically, Figure 1, 2), rendering higher degrees than quadratic very time consuming. In the case of the RDG scheme, we need not increase the stencil size to obtain higher order accuracy, it suffices to increase the order of the underlying DG scheme.

The reconstruction operator $R$ is constructed analogously as in the FV method, so that $R\Pi_h^n$ is in some sense *polynomial preserving.* Specifically, for each element $K$ and its corresponding stencil $S$, we require that for all $p \in P^N(S)$

$$\Big((R\Pi_h^n)\big|_S\, p\Big)\Big|_K = p\big|_K. \tag{13}$$

This requirement allows us to study approximation properties of $R$ using the Bramble-Hilbert technique as in the standard finite element method, [1]. The disadvantage of this approach is that for unstructured meshes, the coefficients of the reconstruction operator must be stored for each individual stencil.

### 3.1.2 Spectral volume approach

In the *spectral volume approach*, we start with a partition of $\Omega$ into so-called *spectral volumes* $S$, for example triangles in 2D. The triangulation $\mathcal{T}_h$ is formed by subdividing each spectral volume $S$ into sub-cells $K$, called *control volumes*, [7]. In the FV method, the order of accuracy of the reconstruction determines the number of control volumes to be generated in each spectral volume. For example, for a linear reconstruction on a triangle, the triangle is divided into three control volumes, Figure 1, 3). Again, in the RDG scheme, we may use only the smallest available partition into control volumes, and increase the accuracy by increasing the order of the underlying scheme, cf. Figure 1, 4). The reconstruction operator should again be polynomial preserving, i.e. constructed similarly to (13).

The advantage of this approach is that, since all spectral volumes are affine equivalent, it is sufficient to construct the reconstruction operator $R$ only on a reference spectral volume.

## 4 Numerical experiments

We present preliminary numerical experiments for the periodic advection of a 1D sine wave on uniform meshes. Experimental orders of accuracy $\alpha$ in various norms on meshes with $N$ elements are given in Tables 1 and 2. The increase in accuracy due to reconstruction is clearly visible.

| $N$ | $\|\|e_h\|\|_{L^\infty(\Omega)}$ | $\alpha$ | $\|\|e_h\|\|_{L^2(\Omega)}$ | $\alpha$ | $\|e_h\|_{H^1(\Omega,\mathcal{T}_h)}$ | $\alpha$ |
|---|---|---|---|---|---|---|
| 4 | 5.82E-03 | – | 3.49E-03 | – | 3.65E-02 | – |
| 8 | 7.53E-05 | 6.27 | 4.43E-05 | 6,30 | 1.06E-03 | 5,11 |
| 16 | 9.07E-07 | 6.38 | 5.95E-07 | 6,22 | 3.58E-05 | 4,89 |
| 32 | 1.82E-08 | 5.64 | 8.70E-09 | 6,10 | 1.16E-06 | 4,95 |
| 64 | 3.41E-10 | 5.74 | 1.33E-10 | 6,03 | 3.67E-08 | 4,98 |

**Tab. 1:** *1D advection of sine wave, $P^1$ RDG scheme with $P^5$ reconstruction.*

| $N$ | $\|\|e_h\|\|_{L^\infty(\Omega)}$ | $\alpha$ | $\|\|e_h\|\|_{L^2(\Omega)}$ | $\alpha$ | $\|e_h\|_{H^1(\Omega,\mathcal{T}_h)}$ | $\alpha$ |
|---|---|---|---|---|---|---|
| 4 | 2.90E-03 | – | 1.85E-03 | – | 1.63E-02 | – |
| 8 | 7.75E-06 | 8.55 | 3.56E-06 | 9.02 | 1.03E-04 | 7.30 |
| 16 | 2.10E-08 | 8.53 | 6.64E-09 | 9.07 | 4.34E-07 | 7.89 |
| 32 | 7.21E-11 | 8.18 | 4.02E-11 | 7.37 | 1.76E-09 | 7.94 |

**Tab. 2:** *1D advection of sine wave, $P^2$ RDG scheme with $P^8$ reconstruction.*

## References

[1] Ciarlet, P.G.: *The finite elements method for elliptic problems.* North-Holland, Amsterdam, New York, Oxford, 1979.

[2] Dumbser, M., Balsara, D., Toro, E.F., and Munz, C.D.: A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes. J. Comput. Phys. **227** (2008), 8209–8253.

[3] Feistauer, M., Felcman, J., and Straškraba, I.: *Mathematical and computational methods for compressible flow.* Oxford University Press, Oxford, 2003.

[4] Feistauer, M. and Kučera, V.: Analysis of the DGFEM for nonlinear convection-diffusion problems. Electronic Transactions on Numerical Analysis **32** (2008), 33–48.

[5] Kröner, D.: *Numerical schemes for conservation laws.* Wiley und Teubner, 1996.

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

[7] Wang, Z.J.: Spectral (finite) volume method for conservation laws on unstructured grids. J. Comput. Phys. **178** (2002), 210–251.

# INTEGRATION IN HIGHER-ORDER FINITE ELEMENT METHOD IN 3D[*]

Pavel Kůs

## 1 Introduction

Integration of higher-order basis functions is an important issue, that is not as straightforward as it may seem. In traditional low-order FEM codes, the bulk of computational time is a solution of resulting system of linear equations. In the case of higher-order elements the situation is different. Especially in three dimensions the time of integration may represent significant part of the computation.

In first part of the text we describe Gauss quadrature and product quadrature rules on the reference brick. In Section 4.1 we describe algorithm calculating the local stiffness matrix en bloc, which allows to save a lot of calculations that would be repeated for many of the integrals of the stiffness matrix. Calculation is than much faster thanks to creation of auxiliary fields and multiple use of the values, which is shown in Section 5.

## 2 Gauss quadrature rules

The choice of quadrature type is very important. Even though two quadrature rules integrate exactly polynomials up to certain order, their performance can differ significantly when integrating non-polynomial functions (which is in reality always the case, since the inverse Jacobi matrix is non-polynomial for general mesh elements). The usual choice for higher-order integration are Gauss quadrature rules. A 1D integral over the segment $(-1, 1)$ is then approximated by the formula

$$\int_{-1}^{1} f(\xi)\mathrm{d}\xi \approx \sum_{i=1}^{n} w_{n,i} f(\xi_{n,i}), \tag{1}$$

where $\xi_{n,i}$ and $w_{n,i}$ are integration points and weights.

## 3 Product quadrature rules

Since the integration is performed on reference element, which is a cube in our case, the most natural choice of the integration rules is to use tensor products of 1D Gauss rules described in the previous section. A construction of such integration rules is described in [4].

[*] This work was supported by grant GAAVCR IAA100760702.

### 3.1 Computational cost of the integration

Let us estimate the computational cost of calculation of the local stiffness matrix. Consider hexahedral element. In *hp*-FEM it is usually equipped by basis functions constructed as products of 1D polynomials of degrees up to $p$. In total there are $(p+1)^3$ basis functions. In order to evaluate the local stiffness matrix, we have to calculate integral from the weak form for each pair of basis functions. Therefore we have $(p+1)^3 \times (p+1)^3$ integral evaluations. Integrand is always a product of two basis functions, its polynomial order therefore is up to $2p$ in each direction. Quadrature rule that will calculate integrals exactly has approximately $p^3$ points (each 1D rule has approximately $p$ points). Thus, calculation of one integral costs $O(p^3)$ function evaluations. Since we have to do $(p+1)^3 \times (p+1)^3$ such calculations, total asymptotic complexity of the evaluation of the local stiffness matrix is $O(p^9)$.

It is obvious, that this is extremely unfavorable and makes assembling procedure very time-consuming. For the numerical solution of partial differential equations in more than 3 dimensions, this estimate is even more severe and makes it virtually impossible to use such integration. For truly high-dimensional calculations, which are becoming more and more desirable for example for financial problems, completely different ways towards estimation of the values of the integrals, such as Smoljak's schemes are used. For the main idea see Section 4.2.

### 3.2 Hierarchical elements

In the following we describe several ideas how to make the calculation more economical. If we use hierarchical rather then nodal basis, the basis of an element of order $p$ is obtained by adding several polynomial functions of order $p$ to the basis of an element of order $p-1$. Therefore, the basis consist of polynomials of various orders from 1 up to $p$ and obviously it would be waste to integrate product of two low-degree polynomials with quadrature rule which is exact for product of polynomials of degree $p$. We consider basis functions in the form (2).

Assume we have to calculate product of two functions of degrees $(p_x, p_y, p_z)$ and $(q_x, q_y, q_z)$. Obviously, the rule capable of exact calculation is of order $(p_x+q_x,$ $p_y+q_y,$ $p_z+q_z)$. However, using such rules has slight drawback. When we calculate the value of the integral, we have to store precalculated values of all shape functions in all integration points of the particular rule. If we had precalculated values of all shape functions for all rules of order $(p_x, p_y, p_z)$, $p_x, p_y, p_z \in \{1 \dots P\}$, where $P$ is the maximal degree of polynomials used in the basis, the size of the tables would occupy a big portion of the computer memory. Possible solution of this problem is to use only quadrature rules with the same order in all directions, i.e. instead of a rule of order $(p_x, p_y, p_z)$ we use a rule of order $(p_m, p_m, p_m)$, where $p_m = \max(p_x, p_y, p_z)$ even though it has more points than necessary.

## 4 Alternative approaches to quadrature

In the previous section we described how a simple numerical quadrature works. We have seen, however, that this approach may lead to quadrature rules with very high number of integration points. In this section we want to describe two different approaches. The first is based upon the works [2], [1]. Ideas used there for 2D are adapted to 3D case and to different technique of construction of basis functions, which allows it's substantial simplification.

The second alternative is presented mainly for reference. Smoljak's schemes are used for integration in partial differential equations in more dimensions, where all conventional approaches fail due to the "curse of dimensionality".

### 4.1 Reordering of quadrature

We will use the fact, that both basis functions and integration rules are constructed as cartesian products of 1D functions and integration rules. Thanks to this structure, we can reorder the whole calculation, save some results into auxiliary fields and use them for more integrals of the stiffness matrix.

#### 4.1.1 General algorithm

In the articles [2], [1], the authors distinguish between vertex, edge and bubble basis functions and use slightly different algorithm for each group. Our algorithm does not do that.

We consider basis functions on the reference domain $K = [-1, 1]^3$ in the form

$$F_{k_1,k_2,k_3}(\xi^1, \xi^2, \xi^3) = f^1_{k_1}(\xi^1) f^2_{k_2}(\xi^2) f^3_{k_3}(\xi^3), \tag{2}$$

where $(k_1, k_2, k_3) \in M = \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \{1, \dots, n_3\}$. Our goal is to calculate all integrals

$$\int_K F_k(\xi) F_{k'}(\xi) Z(\xi)\, \mathrm{d}\xi, \tag{3}$$

where $k, k' \in M$. $Z(\xi)$ stands for the rest of the integrand independent on the basis functions. It can be Jacobian of reference mapping, material parameter or anything else. Of course this part of the integrals does not have product structure like the basis functions, but, on the other hand, is the same for all integrals calculated. The integrals will be approximated by one quadrature rule obtained as a product of three 1D rules with sufficiently high order in each direction. Individual 1D rules may have different order:

$$\begin{aligned} R_1 &= \{(w^1_i, \xi^1_i), i = 1, \dots, m_1\}, \\ R_2 &= \{(w^2_i, \xi^2_i), i = 1, \dots, m_2\}, \\ R_3 &= \{(w^3_i, \xi^3_i), i = 1, \dots, m_3\}, \end{aligned}$$

where $w^j_i$ stands for weight and $\xi^j_i$ for integration point. The compound rule has then the form:

$$R = \{(w^1_{i_1} w^2_{i_2} w^3_{i_3},\ (\xi^1_{i_1},\ \xi^2_{i_2},\ \xi^3_{i_3})),\ \ i_1 = 1, \dots, m_1,\ i_2 = 1, \dots, m_2,\ i_3 = 1, \dots, m_3\},$$

the number of integration points being $m = m_1 m_2 m_3$. The integral from (3) can be approximated as

$$\sum_{i=1}^{m} w_i F_k(\xi_i) F_{k'}(\xi_i) Z(\xi_i). \tag{4}$$

Using the product structure of basis functions and integration rules, the latest can be expanded to

$$\sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} w^1_{i_1} w^2_{i_2} w^3_{i_3} \; f^1_{k_1}(\xi^1_{i_1}) f^2_{k_2}(\xi^2_{i_2}) f^3_{k_3}(\xi^3_{i_3}) \; f^1_{k'_1}(\xi^1_{i_1}) f^2_{k'_2}(\xi^2_{i_2}) f^3_{k'_3}(\xi^3_{i_3}) Z(\xi^1_{i_1}, \xi^2_{i_2}, \xi^3_{i_3}). \tag{5}$$

Now the summation can be reordered:

$$\sum_{i_1=1}^{m_1} w^1_{i_1} f^1_{k_1}(\xi^1_{i_1}) f^1_{k'_1}(\xi^1_{i_1}) \sum_{i_2=1}^{m_2} w^2_{i_2} f^2_{k_2}(\xi^2_{i_2}) f^2_{k'_2}(\xi^2_{i_2}) \sum_{i_3=1}^{m_3} w^3_{i_3} f^3_{k_3}(\xi^3_{i_3}) f^3_{k'_3}(\xi^3_{i_3}) Z(\xi^1_{i_1}, \xi^2_{i_2}, \xi^3_{i_3}). \tag{6}$$

Let us introduce auxiliary field $G(k_3, k'_3, i_1, i_2)$, where

$$G(k_3, k'_3, i_1, i_2) = \sum_{i_3=1}^{m_3} w^3_{i_3} f^3_{k_3}(\xi^3_{i_3}) f^3_{k'_3}(\xi^3_{i_3}) Z(\xi^1_{i_1}, \xi^2_{i_2}, \xi^3_{i_3}). \tag{7}$$

It is important to realize, that the just defined term really depends only on $k_3$, $k'_3$, $i_1$ and $i_2$. Indeed, all terms depending on $k_1$, $k'_1$, $k_2$ and $k'_2$ were put in front of the last sum and $i_3$ is being summed over.

Similarly, let us introduce another auxiliary field $H(k_2, k'_2, k_3, k'_3, i_1)$:

$$H(k_2, k'_2, k_3, k'_3, i_1) = \sum_{i_2=1}^{m_2} w^2_{i_2} f^2_{k_2}(\xi^2_{i_2}) f^2_{k'_2}(\xi^2_{i_2}) G(k_3, k'_3, i_1, i_2). \tag{8}$$

This field depends also on $k_2$ and $k'_2$, but, thanks to the summation, does not depend on $i_2$. Now the integral (3) can be approximated as

$$\int_K F_k(\xi) F_{k'}(\xi) Z(\xi) \,\mathrm{d}\xi \approx \sum_{i_1=1}^{m_1} w^1_{i_1} f^1_{k_1}(\xi^1_{i_1}) f^1_{k'_1}(\xi^1_{i_1}) H(k_2, k'_2, k_3, k'_3, i_1) \tag{9}$$

When generating the matrix of the integrals, we first precalculate the field G, than the field H and finally use it to calculate all the integrals (9), where $k, k' \in M$.

### 4.1.2 Asymptotic analysis

Now let us estimate the amount of work needed to generate the stiffness matrix. The numerical comparisons are presented in Section 5, here we want to do just a rough estimate. As in Section 3.1, we assume, that the polynomial degree of our basis functions is up to $p$ in each direction. Therefore we have $p^3$ functions and the 1D integration rules, that comprise the final integration rule, have approximately $p$ integration points.

In Section 3.1, we approximated the work needed to generate the stiffness matrix to $O(p^9)$. In the algorithm described above, we first precalculate field $G$, which requires $O(p^4)$ work. Then the field $H$ is precalculated, which requires $O(p^5)$. That should be negligible in comparison with the main part, which are calculations using the formula (9). There are $p^3$ functions, therefore we have to calculate $p^6$ integrals. But in the formula (9) there is only one summation, with respect to $i_1$. Other summations are hidden in the auxiliary fields. Therefore the complexity of this part is $O(p^7)$. Comparisons of real number of operations needed to calculate the matrix will be presented in Section 5.

### 4.2 Sparse schemes

The idea of sparse schemes was first introduced by Smolyak in [3]. The goal of this approach is to construct an integration grid, similar to the simple product grid, but with fewer points. The reason, why this is possible, is that slight under-integration does not always spoil the convergence.

From the experiments and comparisons we made it seems that this approach is not the most successful for problems in three dimensions. It's role starts to be vital for problems in much more dimensions, which arise in various fields including financial math. Sparse grids seems to be the only method capable to cope with the "curse of dimensionality", when number of integration points rise exponentially with number of dimensions.

## 5 Comparisons

In this section we want to compare different approaches to quadrature with respect to number of operations needed.

### 5.1 CPU time of assembling

The solving process in our code has two main parts, assembling and solving the stiffness matrix. A CPU time needed to perform each part very strongly depends on the problem setting. It depends not only on the number of elements of the mesh and the polynomial order used in the finite element space, but also on the structure of the mesh and use of hanging nodes. Often the assembling time exceeds the time needed to solve the resulting linear system, so faster quadrature can be very welcomed in some cases.

### 5.2 Performance of different quadrature techniques

In Figure 1 we can see a comparison of number of operations needed to assembly a mass matrix of the element of various orders. On the graph we can see ratios of number of operations of individual methods with respect to the simple product method (it is therefore 1 for all polynomial orders.)

The usefulness of the faster quadrature depends on the order used, but even for order 5 we get ten times faster algorithm, comparing to the integration of each integral with optimal, but isotropic order.

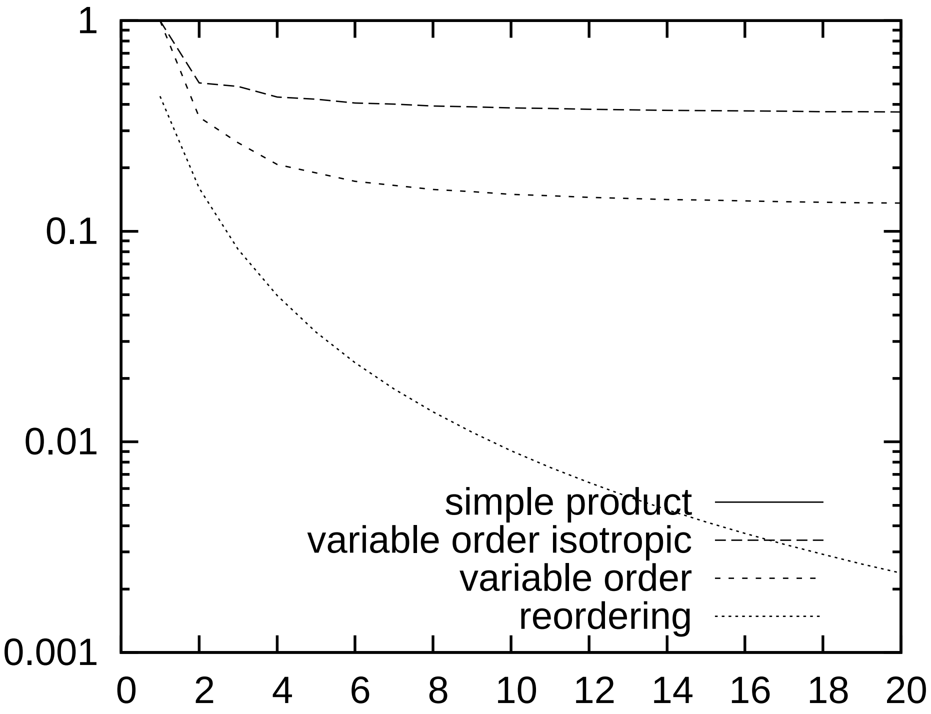


**Fig. 1:** *Comparison of performance of described methods with respect to the simple product method. On the $x$ axis is order of an element, on the $y$ axis quotient of number of operations of each method.*

### 5.3 Conclusions

We have shown, that the concept of reordering of summation works well and decreases number of operations needed to construct the local stiffness matrix. It's effect grows with growing order of an element. Even though incorporating into the code might bring certain complications, it is definitively worth considering.

### References

[1] Eibner, T. and Melenk, J.M.: Fast algorithms for setting up the stiffness matrix in *hp*-FEM: a comparison. Numerical Analysis Report 3/05 .

[2] Melenk, J.M., Gerdes, K., and Schwab, C.: Fully discrete *hp*-finite elements: Fast quadrature. Research Report No. 99-15 (1999).

[3] Smolyak, S.A.: Quadrature and interpolation formulas for tensor product of certain classes of functions. Dokl. Akad. Nauk (1963), 240–243.

[4] Solin, P., Segeth, K., and Dolezel, I.: *Higher-Order Finite Element Methods.* Chapman & Hall/CRC Press, Boca Raton, 2004.

# ROBUST PRECONDITIONERS FOR THE MATRIX FREE TRUNCATED NEWTON METHOD[*]

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

**Abstract**

New positive definite preconditioners for the matrix free truncated Newton method are given. Corresponding algorithms are described in detail. Results of numerical experiments that confirm the efficiency and robustness of the preconditioned truncated Newton method are reported.

## 1 Introduction

We consider the unconstrained minimization problem

$$x^* = \arg\min\nolimits_{x \in R^n} F(x), \quad F \in \mathcal{C}^2 : R^n \to R, \quad n - \text{large}$$

and use the notation

$$g(x) = \nabla F(x), \quad G(x) = \nabla^2 F(x),$$

$$\|G(x)\| \le \overline{G}, \quad \forall x \in R^n.$$

Numerical methods for unconstrained minimization are iterative and their iteration step has the form

$$x_{k+1} = x_k + \alpha_k s_k, \quad k \in N,$$

where $s_k$ is a direction vector and $\alpha_k$ is a step-length. In this contribution, we will deal with the Newton method, which uses the quadratic model

$$F(x_k + s) \approx Q(x_k + s) = F(x_k) + g^T(x_k)s + \frac{1}{2}s^T G(x_k)s$$

for direction determination in such a way that

$$s_k = \arg\min_{s \in \mathcal{M}_k} Q(x_k + s).$$

There are two basic possibilities for direction determination: the line-search method, where

$$\mathcal{M}_k = R^n,$$

and the trust-region method, where

[*]This work was supported by the Czech Science Foundation, project No. 201/09/1957, and the institutional research plan No. AV0Z10300504.

$$\mathcal{M}_k = \{s \in R^n : \|s\| \le \Delta_k\}$$

(here $\Delta_k > 0$ is the trust region radius). We suppose that matrix $G = G(x)$ and its structure are not explicitly known. The direction vector (a minimum of a quadratic function) is in this case computed iteratively by the preconditioned conjugate gradient (PCG) method with preconditioner $C$. The outer index $k$ is for the sake of simplicity mostly omitted.

**Algorithm 1** *Direction determination by the PCG method (the line-search method). Data: Relative precision* $0 \le \omega < 1$*.*

```
$s_1 = 0, \quad g_1 = g, \quad h_1 = C^{-1} g_1, \quad \rho_1 = g_1^T h_1, \quad p_1 = -h_1.$
Do $i = 1$ to $m$
$q_i = G p_i, \quad \sigma_i = p_i^T q_i.$
If $\sigma_i \le 0$ then $s = s_i$, stop.
$\alpha_i = \rho_i / \sigma_i, \quad s_{i+1} = s_i + \alpha_i p_i, \quad g_{i+1} = g_i + \alpha_i q_i,$
$h_{i+1} = C^{-1} g_{i+1}, \quad \rho_{i+1} = g_{i+1}^T h_{i+1}.$
If $\|g_{i+1}\| \le \omega \|g_1\|$ or $i = m$ then $s = s_i$, stop.
$\beta_i = \rho_{i+1} / \rho_i, \quad p_{i+1} = -h_{i+1} + \beta_i p_i.$
End do
```

**Algorithm 2** *Direction determination by the PCG method (the trust-region method) Data: Relative precision* $0 \le \omega < 1$*, trust region radius* $\Delta > 0$*.*

```
$s_1 = 0, \quad g_1 = g, \quad h_1 = C^{-1} g_1, \quad \rho_1 = g_1^T h_1, \quad p_1 = -h_1.$
Do $i = 1$ to $m$
$q_i = G p_i, \quad \sigma_i = p_i^T q_i.$
If $\sigma_i \le 0$ then $s = s_i + \lambda_i p_i,\ \lambda_i > 0,\ \|s_i + \lambda_i p_i\| = \Delta$, stop.
$\alpha_i = \rho_i / \sigma_i.$
If $\|s_i + \alpha_i p_i\| \ge \Delta$ then $s = s_i + \lambda_i p_i,\ \lambda_i > 0,\ \|s_i + \lambda_i p_i\| = \Delta$, stop.
$s_{i+1} = s_i + \alpha_i p_i, \quad g_{i+1} = g_i + \alpha_i q_i,$
$h_{i+1} = C^{-1} g_{i+1}, \quad \rho_{i+1} = g_{i+1}^T h_{i+1}.$
If $\|g_{i+1}\| \le \omega \|g_1\|$ or $i = m$ then $s = s_i$, stop.
$\beta_i = \rho_{i+1} / \rho_i, \quad p_{i+1} = -h_{i+1} + \beta_i p_i.$
End do
```

Since matrix $G$ is not given explicitly, we use numerical differentiation instead of matrix multiplication. Thus the product $q = Gp$ is replaced by the difference

$$G(x)p \approx \frac{g(x + \delta p) - g(x)}{\delta}$$

where $\delta = \varepsilon/\|p\|$ (usually $\varepsilon = \sqrt{\varepsilon_M}$ and $\varepsilon_M$ is a machine precision). The following theorems are proved in [4], Section 8.4.

**Theorem 1** *Let function $F \in \mathcal{C}^2 : R^n \to R$ have Lipschitz continuous second order derivatives (with a constant $\overline{L}$). Let $q = G(x)p$ and*

$$\tilde{q} = \frac{g(x+\delta p) - g(x)}{\delta}, \quad \delta = \frac{\varepsilon}{\|p\|}.$$

*Then it holds*

$$\|\tilde{q} - q\| \le \frac{1}{2}\varepsilon \overline{L}\|p\|.$$

**Theorem 2** *Consider the conjugate gradient method applied to the system of linear equations $G(x)s + g = 0$, where the vectors $q_i = G(x)p_i$ are replaced by the vectors $\tilde{q}_i = (g(x+\delta_i p_i) - g(x))/\delta_i$, $\delta_i = \varepsilon/\|p_i\|$. Suppose that the assumptions of Theorem 1 are satisfied and denote*

$$s_{m+1} = s_1 + \sum_{i=1}^{m} \alpha_i p_i, \quad g_{m+1} = g_1 + \sum_{i=1}^{m} \alpha_i q_i, \quad \tilde{g}_{m+1} = g_1 + \sum_{i=1}^{m} \alpha_i \tilde{q}_i$$

*(thus $g_{m+1} = g + G(x)s_{m+1}$ if the computation is exact). Then it holds*

$$\|\tilde{g}_{m+1} - g_{m+1}\| \le \overline{\vartheta}\|s_{m+1}\|, \quad \overline{\vartheta} = \frac{m}{2}\varepsilon\overline{L}.$$

**Remark 1** *Assume that $\|\tilde{g}_{m+1}\| \le \overline{\omega}\|g\|$, $0 < \overline{\omega} < 1$, in the m-th step of the conjugate gradient method. If we set $s = s_{m+1}$ and $\tilde{g} = \tilde{g}_{m+1}$, then we can write*

$$\frac{\|\tilde{G}s + g\|}{\|g\|} \le \overline{\omega}, \quad \frac{\|(\tilde{G} - G)s\|}{\|s\|} \le \overline{\vartheta},$$

*see Theorem 2, where $\tilde{G}$ is a symmetric matrix for which it holds $\tilde{G}s + g = \tilde{g}$ and $\overline{\vartheta} = m\varepsilon\overline{L}/2$. These expressions allow us to estimate the asymptotic rate of convergence.*

A disadvantage of the difference version of the truncated Newton method consists in the fact that it requires a large number of inner iterations (i.e. a large number of gradient evaluations) if matrix $G = G(x)$ is ill-conditioned. Therefore, the conjugate gradient method must be suitably preconditioned. Standard approaches cannot be used because matrix $G$ is unknown. The following possibilities will be studied:

- Preconditioning based on the limited memory BFGS (Broyden, Fletcher, Goldfarb, Shanno) method.
- Band preconditioners obtained by the standard BFGS method equivalent to the preconditioned conjugate gradient method.
- Band preconditioners obtained by numerical differentiation.
- Tridiagonal preconditioners determined by the Lanczos method equivalent to the unpreconditioned conjugate gradient method.

## 2 Preconditioning based on the limited memory BFGS method

The idea of limited memory preconditioners is very simple (see [7]). Matrix $C_k^{-1} = H_k = H_k^k$, used as a preconditioner in the $k$-th step of the Newton method, is determined recurrently in such a way that $H_{k-l}^k = \gamma_{k-l} I$ where $l$ is the number of updates (usually $l = 3$) and

$$\begin{aligned} H_{j+1}^k &= H_j^k + \left( \frac{y_j^T H_j^k y_j}{y_j^T d_j} + 1 \right) \frac{d_j d_j^T}{y_j^T d_j} - \frac{H_j^k y_j d_j^T + d_j (H_j^k y_j)^T}{y_j^T d_j} \\ &= V_j^T H_j^k V_j + \frac{d_j d_j^T}{y_j^T d_j} \end{aligned}$$

for $k - l \le j \le k - 1$ with

$$V_j = I - \frac{y_j d_j^T}{y_j^T d_j}, \quad d_j = x_{j+1} - x_j, \quad y_j = g_{j+1} - g_j.$$

Matrix $H_k$ is not computed explicitly. In the $i$-th inner step of the conjugate gradient method used in the $k$-th outer step of the Newton method, a vector $h_i = C_k^{-1} g_i = H_k\, g_i$ is determined by the Strang recurrences [6]. First, we set $u_k = g_i$ and compute numbers and vectors

$$\sigma_j = \frac{d_j^T u_{j+1}}{y_j^T d_j} \quad \text{and} \quad u_j = u_{j+1} - \sigma_j y_j, \quad k - l \le j \le k - 1,$$

respectively, using backward recurrences. Then we set $v_{k-l} = \gamma_{k-l} u_{k-l}$ and compute vectors

$$v_{j+1} = v_j + \left( \sigma_j - \frac{y_j^T v_j}{y_j^T d_j} \right) d_j, \quad k - l \le j \le k - 1,$$

using forward recurrence. Finally, we set $h_i = v_k$.

## 3 Band preconditioners obtained by the standard BFGS method

The BFGS method with perfect line search applied to a strictly convex quadratic function (with matrix $G$ in the quadratic term) is equivalent to the conjugate gradient method with the same step-length choice. The BFGS method generates a sequence of matrices $B_i$, $1 \le i \le m$, in such a way that $B_1 = C$ and

$$B_{i+1} = B_i + \frac{y_i y_i^T}{d_i^T y_i} - \frac{B_i d_i (B_i d_i)^T}{d_i^T B_i d_i} = B_i + \frac{G p_i (G p_i)^T}{p_i^T G p_i} + \frac{g_i g_i^T}{p_i^T g_i}$$

for $1 \le i \le m$, where $d_i = s_{i+1} - s_i = \alpha_i p_i$ and $y_i = g_{i+1} - g_i = G d_i$. Vectors $p_i$ and $g_i$ are byproducts of the conjugate gradient method. If we use vectors $\tilde{q}_i$ (given by numerical differentiation) and $\tilde{g}_i$ instead of vectors $q_i = G p_i$ and $g_i$, respectively, we can write $B_1 = C$ and

$$B_{i+1} = B_i + \frac{\tilde{q}_i \tilde{q}_i^T}{p_i^T \tilde{q}_i} + \frac{\tilde{g}_i \tilde{g}_i^T}{p_i^T \tilde{g}_i}, \quad 1 \le i \le m.$$

From the above formulation, it is evident that only vectors generated by the preconditioned conjugate gradient method (with matrix multiplication replaced by numerical differentiation) are used for determination of matrices $B_i$, $1 \leq i \leq m$. These matrices do not occur in correction terms, so we can save only their selected parts (see [8]). If the vectors $\tilde{q}_i$ and $\tilde{g}_i$ are good approximations of the vectors $q_i$ and $g_i$, then the matrices $B_i$, $1 \leq i \leq m$, are positive definite. Further, if the number of steps of the conjugate gradient method is sufficiently large, the matrix $B = B_{m+1}$ is a good approximation of matrix $G$ so we can use it (or its part) as a preconditioner in the next step of the Newton method. We will investigate three special cases.

### 3.1 Diagonal preconditioning

If $C = D$, where $D$ is a diagonal matrix containing diagonal elements of $B$, no problem arises because positive definite matrix $B$ has positive numbers on the main diagonal. Diagonal preconditioning for problems with sparse Hessian matrices justifies the following theorem proved in [3].

**Theorem 3** *Let $\mathcal{D}_n$ be the set of all diagonal matrices of order $n$ and let $D$ be a diagonal matrix containing diagonal elements of matrix $G$. Then it holds*

$$\kappa(GD^{-1}) \leq l \min_{M \in \mathcal{D}_n} \kappa(GM^{-1})$$

*where $\kappa$ is a spectral condition number and $l$ is a maximal number of nonzero elements in rows of matrix $G$ ($l = 5$ for pentadiagonal matrix $G$).*

### 3.2 Tridiagonal preconditioning

Let now $C = T$ where $T$ is a tridiagonal matrix containing elements of three main diagonals of matrix $B$. In this case the matrix $C$ need not be positive definite (even if $B$ was positive definite). Consider, as an example, matrices

$$B = \begin{bmatrix} 2 & -2 & 2 \\ -2 & 3 & -3 \\ 2 & -3 & 4 \end{bmatrix}, \qquad T = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 3 & -3 \\ 0 & -3 & 4 \end{bmatrix}.$$

Both these matrices have positive elements on the main diagonal and positive main subdeterminants of the second order. But it holds that $\det B = 2$ and $\det T = -10$ so $T$ is not positive definite, although $B$ is positive definite. In order to remove this drawback, we have to modify matrix $T$ to be positive definite.

**Lemma 1** *Consider a tridiagonal matrix*

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & \dots & 0 & 0 \\ \beta_1 & \alpha_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & \dots & \beta_{n-1} & \alpha_n \end{bmatrix}$$

*(elements $\alpha_i$ have differrent meaning than step-sizes $\alpha_i$ used in previous sections) and denote $\Delta_i$ a main subdeterminant of the i-th order of matrix $T$ containing rows and columns with indexes $1, 2, \ldots, i$). Then it holds $\Delta_1 = \alpha_1$ and*

$$\Delta_i = \alpha_i \Delta_{i-1} - \beta_{i-1}^2 \Delta_{i-2}, \qquad 2 \le i \le n,$$

*where $\Delta_0 = 1$.*

This well-known lemma can be used in the proof of the next theorem (see [1], [4]).

**Theorem 4** *A tridiagonal matrix $T$ is positive definite if and only if $\gamma_i > 0$ for $1 \le i \le n$, where $\gamma_1 = \alpha_1$ and*

$$\gamma_i = \alpha_i - \frac{\beta_{i-1}^2}{\gamma_{i-1}}, \qquad 2 \le i \le n.$$

Theorem 4 can be utilized in such a way that we compute numbers $\gamma_i$, $1 < i \le n$, and as soon as $\gamma_i \le 0$ for some index $i$, we decrease the off-diagonal element $\beta_{i-1}$ so that $\beta_{i-1}^2 < \gamma_{i-1}\alpha_i$ (e.g. we set $\beta_{i-1}^2 = \lambda_{i-1}\gamma_{i-1}\alpha_i$, where $0 < \lambda_{i-1} < 1$). The trouble is that if we choose $\lambda_{i-1}$ unsuitably, the resulting tridiagonal matrix can be ill-conditioned. For practical purposes it is more convenient to use the following theorem and its corollary (see [4]), Section 8.4.

**Theorem 5** *Consider a tridiagonal matrix $T$ with positive numbers on the main diagonal. If matrices*

$$\begin{bmatrix} 2\alpha_1 & 2\beta_1 \\ 2\beta_1 & \alpha_2 \end{bmatrix}, \qquad \begin{bmatrix} \alpha_i & 2\beta_i \\ 2\beta_i & \alpha_{i+1} \end{bmatrix}, \qquad \begin{bmatrix} \alpha_{n-1} & 2\beta_{n-1} \\ 2\beta_{n-1} & 2\alpha_n \end{bmatrix},$$

*where $2 \le i < n - 2$, are positive semidefinite and at least one of them is positive definite, then matrix $T$ is positive definite.*

**Corollary 1** *Let a tridiagonal matrix $T$ contain the main diagonal and halves of subdiagonals of the positive definite matrix $B$ (thus $\alpha_i = b_{i,i}$, $1 \le i \le n$, and $\beta_i = b_{i,i+1}/2$, $1 \le i \le n-1$). Then $T$ is positive definite.*

Corollary 1 can be utilized so that the subdiagonal elements of matrix $B$ are divided by two. Thereafter, the resulting tridiagonal matrix is positive definite. Theorem 5 can be utilized so that we compute determinants $\alpha_i\alpha_{i+1} - 4\beta_i^2$, $1 \le i \le n-1$, and as soon as $\alpha_i\alpha_{i+1} - 4\beta_i^2 < 0$ holds for some index $i$, we decrease the subdiagonal element $\beta_i$ so that $\beta_i^2 = \alpha_i\alpha_{i+1}/4$.

### 3.3 Pentadiagonal preconditioning

Assertions of Theorem 5 and Corollary 1 can also be generalized for an arbitrary band matrix. We will show the corresponding procedure in case of the following pentadiagonal matrix

$$P = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \dots & 0 & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \dots & 0 & 0 & 0 \\ \gamma_1 & \beta_2 & \alpha_3 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \alpha_{n-2} & \beta_{n-2} & \gamma_{n-2} \\ 0 & 0 & 0 & \dots & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & 0 & \dots & \gamma_{n-2} & \beta_{n-1} & \alpha_n \end{bmatrix}$$

on which we will often refer. The following theorem and its corollary are proved in [4], Section 8.4.

**Theorem 6** *Consider a pentadiagonal matrix $P$ with positive elements on the main diagonal. If matrices*

$$\begin{bmatrix} \alpha_i & (3/2)\beta_i & 3\gamma_i \\ (3/2)\beta_i & \alpha_{i+1} & (3/2)\beta_{i+1} \\ 3\gamma_i & (3/2)\beta_{i+1} & \alpha_{i+2} \end{bmatrix}, \qquad 1 \leq i < n-2,$$

*are positive semidefinite, then matrix $P$ is positive definite.*

**Corollary 2** *Let a pentadiagonal matrix $P$ contain the main diagonal, two thirds of subdiagonals, and one third of subsubdiagonals of a positive definite matrix $B$ (thus $\alpha_i = b_{i,i}$, $1 \leq i \leq n$, $\beta_i = 2b_{i,i+1}/3$, $1 \leq i \leq n-1$, and $\gamma_i = b_{i,i+2}/3$, $1 \leq i \leq n-2$). Then $P$ is positive definite.*

Corollary 2 can be utilized so that we take two thirds of subdiagonal elements and one third of subsubdiagonal elements of matrix $B$. Thereafter, the resulting pentadiagonal matrix is positive definite. Theorem 6 can be utilized so that we first compute subdeterminants $\alpha_i\alpha_{i+1} - (9/4)\beta_i^2$, $1 \leq i \leq n-1$, and as soon as one of them is negative, we decrease the subdiagonal element $\beta_i$ so that $\beta_i^2 = (4/9)\alpha_i\alpha_{i+1}$. Finally, we compute the determinants of the matrices mentioned in Theorem 6 as long as they are nonnegative. If one of them is negative, the corresponding element $\gamma_i$ is modified using the following theorem is proved in [4], Section 8.4.

**Theorem 7** *Determinants $\Delta_i$ of the matrices mentioned in Theorem 6 can be computed according to the formula*

$$\Delta_i = \alpha_{i+1}\left(\alpha_i\alpha_{i+2} - 9\gamma_i^2\right) - \frac{9}{4}\left(\alpha_i\beta_{i+1}^2 + \alpha_{i+2}\beta_i^2 - 6\beta_i\beta_{i+1}\gamma_i\right).$$

*The determinant $\Delta_i$ is nonnegative if and only if $\underline{\gamma}_i \leq \gamma_i \leq \overline{\gamma}_i$ where*

$$\begin{aligned}\underline{\gamma}_i &= \frac{1}{3\alpha_{i+1}}\left(\frac{9}{4}\beta_i\beta_{i+1} - \sqrt{D_i}\right),\\ \overline{\gamma}_i &= \frac{1}{3\alpha_{i+1}}\left(\frac{9}{4}\beta_i\beta_{i+1} + \sqrt{D_i}\right)\end{aligned}$$

*are the roots of the quadratic equation* $\Delta_i = 0$ *and*

$$D_i = \left(\alpha_i\alpha_{i+1} - \frac{9}{4}\beta_i^2\right)\left(\alpha_{i+1}\alpha_{i+2} - \frac{9}{4}\beta_{i+1}^2\right)$$

*is the discriminant, divided by 36, of this equation, which is nonnegative provided that both multipliers are nonnegative.*

**Remark 2** *Theorem 7 offers two possibilities how to choose a new element* $\gamma_i$ *in case that* $\Delta_i < 0$*. If* $\gamma_i < \underline{\gamma}_i$*, we set* $\gamma_i := \underline{\gamma}_i$*. If* $\gamma_i > \overline{\gamma}_i$*, we set* $\gamma_i := \overline{\gamma}_i$*. However, more advantageous is to set*

$$\gamma_i = \frac{1}{2}(\underline{\gamma}_i + \overline{\gamma}_i) = \frac{3}{4}\frac{\beta_i\beta_{i+1}}{\alpha_{i+1}},$$

*because this choice is computationally simpler and gives better practical results.*

## 4 Band preconditioners obtained by numerical differentiation

Suppose that the Hessian matrix has a band structure (even if it was not true in fact). The elements of this fictitious matrix that will be used as a preconditioner can be determined by numerical differentiation. It is performed only once at the beginning of the outer step of the Newton method.

In order to determine all elements of a band matrix which has $k-1$ couples of subdiagonals (thus $k = (l+1)/2$ where $l$ is a band width), it suffices to use $k$ gradient differences, which means to compute $k$ extra gradients during each outer step of the Newton method. We will investigate three special cases again.

### 4.1 Diagonal preconditioning

**Remark 3** *Assume that the Hessian matrix is diagonal. Then all its elements can be approximated using one gradient difference*

$$G(x)v \approx g(x+v) - g(x), \qquad v = [\delta_1, \dots, \delta_n]^T,$$

*where* $\delta_1, \dots, \delta_n$ *are suitable differences. Diagonal matrix* $C = D = \operatorname{diag}(\alpha_1, \dots, \alpha_n)$ *where* $Dv = g(x+v) - g(x)$ *is then used as a preconditioner. After substitution we obtain* $\alpha_i\delta_i = g_i(x+v) - g_i(x)$ *or*

$$\alpha_i = \frac{g_i(x+v) - g_i(x)}{\delta_i}, \qquad 1 \le i \le n.$$

**Remark 4** *The differences can be chosen in two different ways:*

*(1) We set $\delta_i = \delta$, $1 \le i \le n$, so $v = \delta e$, where $e$ is a vector with all elements equal to one. We can choose (similarly as in Theorem 1) $\delta = \sqrt{\varepsilon_M}/\|e\| = \sqrt{\varepsilon_M/n}$.*

*(2) We set $\delta_i = \sqrt{\varepsilon_M}\max(|x_i|, 1)$, $1 \le i \le n$. This choice is less sensitive to rounding errors.*

*In both cases we can write $\delta_i = \varepsilon\overline{\delta}_i$, $1 \le i \le n$, where $\varepsilon = \sqrt{\varepsilon_M}$ and either $\overline{\delta}_i = 1/\sqrt{n}$ or $\overline{\delta}_i = \max(|x_i|, 1)$ for $1 \le i \le n$.*

A disadvantage of preconditioners based on numerical differentiation is the fact that they need not be positive definite. Consider a strictly convex quadratic function $F : R^2 \to R$:

$$F(x) = \frac{1}{2}\, x^T \begin{bmatrix} 1 & -2 \\ -2 & 6 \end{bmatrix} x, \qquad g(x) = \begin{bmatrix} 1 & -2 \\ -2 & 6 \end{bmatrix} x.$$

Then it holds

$$\frac{g(x+\delta e) - g(x)}{\delta} = \begin{bmatrix} 1 & -2 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 4 \end{bmatrix},$$

thus

$$De = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 4 \end{bmatrix},$$

which gives $\alpha_1 = -1$, $\alpha_2 = 4$, and so matrix $D$ is not positive definite. This drawback can be removed by setting

$$\alpha_i = \frac{|g_i(x+v) - g_i(x)|}{\delta_i}, \qquad 1 \le i \le n.$$

This modification is justified by the following theorem proved in [10].

**Theorem 8** *Let $\mathcal{D}_n$ be the set of all diagonal matrices of order $n$ and let $D = \mathrm{diag}(\alpha_1, \dots, \alpha_n)$ be a diagonal matrix such that*

$$\alpha_i = \sum_{j=1}^{n} |G_{ij}|, \qquad 1 \le j \le n,$$

*where $G_{ij}$, $1 \le j \le n$, are the elements of the $i$-th row of matrix $G$. Then it holds*

$$\kappa_1(GD^{-1}) = \min_{M \in \mathcal{D}_n} \kappa_1(GM^{-1}),$$

*where $\kappa_1$ is an $l_1$ condition number (the product of $l_1$ norms of a matrix and its inverse).*

If matrix $G$ has only positive numbers and if we set $v = \delta e$, we can write $De = (g(x+\delta e) - g(x))/\delta \approx Ge$, so

$$\alpha_i \approx \sum_{j=1}^{n} G_{ij} = \sum_{j=1}^{n} |G_{ij}|$$

and matrix $D$ is according to Theorem 8 an ideal preconditioner (in $l_1$ norm) for the system of equations $Gs + g = 0$. If matrix $G$ does not contain only positive numbers, it holds

$$|\alpha_i| \approx \left| \sum_{j=1}^{n} G_{ij} \right| \leq \sum_{j=1}^{n} |G_{ij}|,$$

so the elements of modified matrix $D$ form the lower bound for the elements of an ideal preconditioner.

### 4.2 Tridiagonal preconditioning

**Theorem 9** *Let the Hesian matrix of function $F$ be tridiagonal (as matrix $T$). Set $v_1 = [\delta_1, 0, \delta_3, 0, \delta_5, 0, \ldots]$, $v_2 = [0, \delta_2, 0, \delta_4, 0, \delta_6, \ldots]$, where $\delta_i = \varepsilon \overline{\delta}_i$, $1 \leq i \leq n$. Then for $1 < i < n$ it holds*

$$\begin{aligned}
\alpha_1 &= \lim_{\varepsilon \to 0} \frac{g_1(x+v_1) - g_1(x)}{\delta_1}, \quad \beta_1 = \lim_{\varepsilon \to 0} \frac{g_1(x+v_2) - g_1(x)}{\delta_2}, \\
\alpha_i &= \lim_{\varepsilon \to 0} \frac{g_i(x+v_1) - g_i(x)}{\delta_i}, \quad \beta_i = \lim_{\varepsilon \to 0} \frac{g_i(x+v_2) - g_i(x) - \delta_{i-1}\beta_{i-1}}{\delta_{i+1}}, \quad \operatorname{mod}(i,2) = 1, \\
\alpha_i &= \lim_{\varepsilon \to 0} \frac{g_i(x+v_2) - g_i(x)}{\delta_i}, \quad \beta_i = \lim_{\varepsilon \to 0} \frac{g_i(x+v_1) - g_i(x) - \delta_{i-1}\beta_{i-1}}{\delta_{i+1}}, \quad \operatorname{mod}(i,2) = 0, \\
\alpha_n &= \lim_{\varepsilon \to 0} \frac{g_n(x+v_1) - g_n(x)}{\delta_n}, \quad \operatorname{mod}(n,2) = 1, \\
\alpha_n &= \lim_{\varepsilon \to 0} \frac{g_n(x+v_2) - g_n(x)}{\delta_n}, \quad \operatorname{mod}(n,2) = 0.
\end{aligned}$$

**Remark 5** *Theorem 9, proved in [4], Section 8.4, gives us the way how to construct a tridiagonal preconditioner. A fixed number $\varepsilon$ is chosen (e.g. $\varepsilon = \sqrt{\varepsilon_M}$) and the elements of matrix $C = T$ are computed according to formulas mentioned in Theorem 9 (in which the limit is omitted).*

Matrix $C = T$ obtained by Remark 5 need not be positive definite even if the Hessian matrix was positive definite. Tridiagonal matrix $T$ obtained by application of Theorem 9 (with $\overline{\delta}_i = \overline{\delta}$, $1 \leq i \leq n$) to a strictly convex quadratic function of three variables with the positive definite Hessian matrix

$$G = \begin{bmatrix} 1 & -1 & -2 \\ -1 & 4 & -1 \\ -2 & -1 & 8 \end{bmatrix}$$

can serve as an example. We will state two theorems supporting a choice of tridiagonal preconditioning in cases when the actual Hessian matrix is pentadiagonal (see [4]).

**Theorem 10** *Let the Hessian matrix $G(x)$ be pentadiagonal, positive definite, and diagonally dominant. Then, if $\delta_i = \varepsilon\overline{\delta}$, $1 \le i \le n$, and if the number $\varepsilon$ is sufficiently small, matrix $C = T$ obtained by Remark 5 is positive definite and diagonally dominant.*

**Remark 6** *Theorem 10, proved in [4], Section 8.4, requires all differences to be equal, which is fulfilled for instance when $\delta_i = \sqrt{2\varepsilon_M/n}$, $1 \le i \le n$. But the numerical experiments show that the choice $\delta_i = \sqrt{\varepsilon}\max(|x_i|, 1)$, $1 \le i \le n$, is usually more advantageous.*

Matrix $T$ is positive definite for a lot of practical problems. Consider a boundary value problem for the second order ordinary differential equation

$$y''(t) = \varphi(y(t)), \quad 0 \le t \le 1, \quad y(0) = y_0, \quad y(1) = y_1,$$

where function $\varphi : R \to R$ is twice continuously differentiable. If we divide the interval $[0, 1]$ onto $n+1$ parts using nodes $t_i = ih$, $0 \le i \le n+1$, where $h = 1/(n+1)$ is the step-size and if we replace the second order derivatives in nodes with differences

$$y''(t_i) = \frac{y(t_{i-1}) - 2y(t_i) + y(t_{i+1})}{h^2}, \quad 1 \le i \le n,$$

we will obtain a system of $n$ nonlinear equations

$$h^2\varphi(x_i) + 2x_i - x_{i-1} - x_{i+1} = 0,$$

where $x_i = y(t_i)$, $0 \le 1 \le n+1$, so $x_0 = y_0$ and $x_{n+1} = y_1$. If we solve this system by the least squares method, the minimized function has the form

$$F(x) = \frac{1}{2}\sum_{i=1}^{n} f_i^2(x) = \frac{1}{2}\sum_{i=1}^{n} \left(h^2\varphi(x_i) + 2x_i - x_{i-1} - x_{i+1}\right)^2,$$

where $x = [x_1, \ldots, x_n]^T$. The following theorem is proved in [4], Section 8.4.

**Theorem 11** *Let the difference version of the Newton method be applied to the sum of squares given above with a linear function $\varphi : R \to R$. Then, if $\delta_i = \varepsilon\overline{\delta}$, $1 \le i \le n$, and if the number $\varepsilon$ is sufficiently small, matrix $C = T$ obtained by Remark 5 is positive definite.*

### 4.3 Pentadiagonal preconditioning

**Theorem 12** *Let the Hessian matrix of function $F$ be pentadiagonal (as matrix $P$). Set $v_1 = [\delta_1, 0, 0, \delta_4, 0, 0, \ldots]$, $v_2 = [0, \delta_2, 0, 0, \delta_5, 0, \ldots]$, $v_3 = [0, 0, \delta_3, 0, 0, \delta_6, \ldots]$, where $\delta_i = \varepsilon\overline{\delta}_i$, $1 \le i \le n$. Then it holds*

$$\alpha_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_1)-g_i(x)}{\delta_i}, \qquad \beta_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_2)-g_i(x)-\delta_{i-2}\gamma_{i-2}}{\delta_{i+1}},$$
$$\gamma_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_3)-g_i(x)-\delta_{i-1}\beta_{i-1}}{\delta_{i+2}}, \qquad \mathrm{mod}(i,3)=1,$$
$$\alpha_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_2)-g_i(x)}{\delta_i}, \qquad \beta_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_3)-g_i(x)-\delta_{i-2}\gamma_{i-2}}{\delta_{i+1}},$$
$$\gamma_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_1)-g_i(x)-\delta_{i-1}\beta_{i-1}}{\delta_{i+2}}, \qquad \mathrm{mod}(i,3)=2,$$
$$\alpha_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_3)-g_i(x)}{\delta_i}, \qquad \beta_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_1)-g_i(x)-\delta_{i-2}\gamma_{i-2}}{\delta_{i+1}},$$
$$\gamma_i = \lim_{\varepsilon\to 0} \frac{g_i(x+v_2)-g_i(x)-\delta_{i-1}\beta_{i-1}}{\delta_{i+2}}, \qquad \mathrm{mod}(i,3)=0,$$

This theorem is proved in [4], Section 8.4.

## 5 Tridiagonal preconditioners determined by the Lanczos method

The elements of a tridiagonal matrix $T$ obtained by the Lanczos method can be determined from the coefficients of the conjugate gradient method (which will be denoted with a tilde) by transformations $\alpha_1 = 1/\tilde{\alpha}_1$ and

$$\beta_i^2 = \frac{\tilde{\beta}_i}{\tilde{\alpha}_i^2}, \quad \alpha_{i+1} = \frac{\tilde{\beta}_i}{\tilde{\alpha}_i} + \frac{1}{\tilde{\alpha}_{i+1}}, \quad 1 \le i \le m,$$

where $m$ is the number such that $\tilde{\alpha}_i > 0$ for $1 \le i \le m$. The following theorems are proved in [4], Section 8.4.

**Theorem 13** *Consider the conjugate gradient method (applied to the quadratic function with the Hessian matrix $G$) such that $\tilde{\alpha}_i > 0$ for $1 \le i \le m$. Then the tridiagonal matrix $T_m$ of order $m$ with the elements given by the above transformations is positive definite.*

**Remark 7** *The tridiagonal matrix $T_m$ has the dimension $m \le n$. In order to obtain a preconditioner with the dimension $n$, we set*

$$C = [Q_m, Q_{n-m}] \begin{bmatrix} T_m & 0 \\ 0 & I_{n-m} \end{bmatrix} [Q_m, Q_{n-m}]^T = (I - Q_m Q_m^T) + Q_m T_m Q_m^T$$

*where $Q_m$ is a matrix with $m$ orthonormal columns obtained with the symmetric Lanczos process and $Q_{n-m}$ is a matrix with $n-m$ orthonormal columns such that matrix $[Q_m, Q_{n-m}]$ is square and orthogonal.*

**Theorem 14** *Let the assumptions of Theorem 13 be fulfilled. Then the preconditioner mentioned in Remark 7 is positive definite and it holds*

$$C^{-1} = (I - Q_m Q_m^T) + Q_m T_m^{-1} Q_m^T.$$

## 6 Rejecting of preconditioners

It is important to be able to decide whether the preconditioner will be used or rejected. Indefinite preconditioner is inappropriate also in case the Hessian matrix is not positive definite.

The Gill-Murray decomposition, proposed in [2], is a suitable means for testing positive definiteness and ill-conditioning of a matrix. If a pivot is during the elimination step less than $\delta \max(1, \max_{1 \le i \le n}(|\alpha_i|))$, where $\delta$ is a prescribed bound, then the decomposition of a preconditioner is terminated and the preconditioner is rejected. It is not worth performing the whole Gill-Murray decomposition and using the obtained positive definite matrix as a preconditioner (numerical experiments prove this claim). The number $\delta$ is usually chosen such that $\delta = 10^{-12}$. Sometimes, however, we have to choose a larger value (e.g. $\delta = 10^{-2}$).

## 7 Concluding remarks

- Preconditioning based on the limited memory BFGS method does not require any corrections. It is rather robust, but not very efficient.
- Band preconditioners obtained by the standard BFGS method have to be modified in advance, otherwise they are mostly rejected during the decomposition. Modifications based on Theorem 5, when the subdiagonal elements are decreased in order negative subdeterminants were zero, have proved to be very successful. It is shown that it is necessary to reject the preconditioners obtained in this way more often (e.g. to choose $\delta = 10^{-2}$).
- Band preconditioners obtained by numerical differentiation can be modified in a simple way that the diagonal elements are replaced with their absolute values. It suffices to choose $\delta = 10^{-12}$ for rejecting (except for diagonal preconditioners which are more prone to rejecting).
- It is not necessary to modify tridiagonal preconditioners determined by the Lanczos method (they are positive definite by Theorem 14). However, they can be determined only in unpreconditioned steps of the Newton method. This causes a lot of technical difficulties (the iteration process of the conjugate gradient method have to be modified).

## 8 Numerical comparison

The difference versions of the Newton method which use various preconditioners were tested using a set of 71 test problems with 1000 variables. The results are reported in the table containing the following data: `NIT` – the total number of iterations, `NFV` – the total number of function evaluations, `NFG` – the total number of gradient evaluations, `NCG` – the total number of inner iterations, `NCN` – the total number of preconditioned outer iterations, `NCP` – the total number of problems with enlarged bound for rejecting, `Time` – the total computational time.

The methods tested: TN – the unpreconditioned Newton method, TNLM – preconditioning using the limited memory BFGS method, TNVM – band preconditioning using the standard BFGS method (1 – diagonal, 2 – tridiagonal, 3 – pentadiagonal), TNND – band preconditioning using numerical differentiation (1 – diagonal, 2 – tridiagonal, 3 – pentadiagonal), TNLT – tridiagonal preconditioning using the Lanczos method, LMVM – the limited memory BFGS method, CG – the nonlinear conjugate gradient method. Methods LMVM and CG are mentioned only for comparison (they have nothing in common with the Newton method studied in this contribution).

| Method | NIT | NFV | NFG | NCG | NCN | NCP | Time |
|---|---|---|---|---|---|---|---|
| TN | 7425 | 11827 | 372789 | 359505 | - | - | 66.08 |
| TNLM | 7270 | 12521 | 233269 | 219347 | 7270 | - | 42.55 |
| TNVM-1 | 7095 | 10303 | 274344 | 262855 | 4335 | 37 | 50.43 |
| TNVM-2 | 6751 | 9252 | 139989 | 129933 | 4260 | 37 | 27.47 |
| TNVM-3 | 6803 | 8857 | 229501 | 219820 | 4027 | 36 | 51.67 |
| TNND-1 | 6522 | 8491 | 347384 | 331709 | 3857 | 40 | 59.51 |
| TNND-2 | 7573 | 11245 | 147391 | 119434 | 4409 | 3 | 25.45 |
| TNND-3 | 7107 | 10726 | 125262 | 91665 | 4943 | 4 | 24.57 |
| TNLT | 7398 | 11672 | 352199 | 339081 | 6808 | 1 | 55.61 |
| LMVM | 121314 | 127189 | 127189 | - | - | - | 39.59 |
| CG | 109166 | 325994 | 325994 | - | - | - | 75.72 |

From the results reported in this table, we can deduce several conclusions:

- The difference versions of the Newton method converge very fast, but they require more gradient computations.
- The unpreconditioned Newton method is not competitive with the limited memory BFGS method.
- Diagonal preconditioners and preconditioners obtained by the Lanczos method are not too efficient.
- Band preconditioners obtained by the standard BFGS method have to be often modified. Moreover, the bound for rejecting has to be often increased.
- Band preconditioners given by numerical differentiation rarely require corrections. The Newton method modified in this way is more efficient than the limited memory BFGS method.

## References

[1] El-Mikkawy, M.E.A.: Notes on linear systems with positive definite tridiagonal matrices. Indian Journal on Pure and Applied Mathematics (2002), 1285–1293.

[2] Gill, P.E. and Murray, W.: Newton type methods for unconstrained and linearly constrained optimization. Math. Programming **7** (1974), 311–350.

[3] Higham, N.J.: *Accuracy and stability of numerical algorithms.* SIAM, Philadelphia, 2002.

[4] Lukšan, L.: Numerické optimalizační metody. Tech. Rep. V-1058, Institute of Computer Science AS CR, Prague, 2009 (`www.cs.cas.cz/luksan/lekce4.pdf`).

[5] Lukšan, L., Matonoha, C., and Vlček, J.: Sparse test problems for unconstrained optimization. Tech. Rep. V-1064, Institute of Computer Science AS CR, Prague, 2010 (`ftp.cs.cas.cz/pub/reports/v1064-10.ps`).

[6] Matthies, H. and Strang, G.: The solution of nonlinear finite element equations. Int. J. for Numerical Methods in Engineering **14** (1979), 1613–1623.

[7] Morales, J.L. and Nocedal, J.: Automatic preconditioning by limited memory quasi-Newton updating. SIAM J. Optimization **10** (2000), 1079–1096.

[8] Nash, S.G.: Preconditioning of truncated-Newton methods. SIAM Journal on Scientific and Statistical Computation **6** (1985), 599–616.

[9] Nocedal, J.: Updating quasi-Newton matrices with limited storage. Mathematics of Computation **35** (1980), 773–782.

[10] Roma, M.: Dynamic scaling based preconditioning for truncated Newton methods in large scale unconstrained optimization. Optimization Methods and Software **20** (2005), 693–713.

# REALIZATION OF DIRICHLET CONDITIONS IN RKPM

Vratislava Mošová

## 1 Introduction

Meshless methods are a group of numerical algorithms that serve for solving boundary value problems. These methods are alternative to the popular and efficient FEM. The greatest advantage of meshless methods is that they need no connectivity condition, like the FEM, in the beginning of computation.

We can specify meshless methods as Galerkin methods where basis functions are replaced by shape functions built in a special way. The construction of the shape functions differs for different meshless methods. Some shape functions are approximations of the kernel in the integral transform

$$u(x) = \int_\Omega K(x,y)u(y)\,\mathrm{d}y \tag{1}$$

(see [10], [4]). Some are constructed by means of the moving least squares method (see [3], [8]). Shape functions based on the idea of partition of unity, that are a composition of an extrinsic and an intrinsic basis form, form the next specific group (see [2], [12]).

The meshless methods have received their place among numerical techniques. They were used for instance in solving problems from mechanic of solid body (see [4]), biomechanics (see [1]) or structural dynamic (see [9]). They are successfully used in the modelling of large deformations, crack propagation or moving boundary. A serious limitation is the fact that the meshless methods do not reproduce the Dirichlet, more generally essential boundary conditions.

Several attempts to solve the problem involving Dirichlet conditions are discussed in this contribution. Our attention will be focused only on one of meshless methods – the reproducing kernel particle method (briefly the RKP method or the RKPM). The construction of the RKP-shape functions and an application of the RKPM to an elliptic boundary value problem are presented in Section 2. Methods that enable to realize the Dirichlet condition in the RKPM are introduced in Section 3.

## 2 RKPM approximation

Consider the problem

$$-\Delta u(x) = f(x) \quad \text{in} \quad \Omega \subset \mathbb{R}^n, \tag{2}$$

$$\frac{\partial u}{\partial n}(x) = g(x) \quad \text{on} \quad \partial\Omega_1, \tag{3}$$

$$u(x) = u^0(x) \quad \text{on} \quad \partial\Omega_0. \tag{4}$$

Denote

$$V = \{v \in W^{1,2}(\Omega) |\ v(x) = 0 \text{ on } \partial\Omega_0 \ \text{ in the sense of traces}\}$$

and find a weak solution $u \in W^{1,2}(\Omega)$ of the problem (2)–(4) such that

$$u - u^0 \in V,$$

$$\int_\Omega \sum_{i=1}^n \frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_i}\, \mathrm{d}x = \int_\Omega f v \,\mathrm{d}x + \int_{\partial\Omega_1} g v \,\mathrm{d}s, \ \forall v \in V. \tag{5}$$

The numerical solution of the problem (2)–(4) will be constructed at points $x^1, \dots, x^N \in \Omega$. At first it is necessary to choose the monomial basis $p$ of degree $s$ and some one-dimensional weight function $\Phi_1$.

**Definition 1** The points $x^1, \dots, x^N \in \Omega$, which are used for construction of the RKP approximation, are called particles.

**Remark 1** The particles $x^1, \dots, x^N$ differ mutually, they can be distributed uniformly or nonuniformly.

**Remark 2** For example, $p(x) = (1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1^2, x_2^2, x_3^2)^{\mathrm{T}}$ is the second degree monomial basis in $\mathbb{R}^3$.

The space generated by monomials of degree less than or equal to $s$ will be denoted by $\mathbb{P}^s$.

**Remark 3** The most often weight functions chosen are the *Gaussian function*

$$\Phi_1(x) = \begin{cases} e^r(x^2-1)/(1-e^r) & \text{if } \ |x| \le 1, \\ 0 & \text{if } \ |x| > 1, \end{cases}$$

with $r > 0$, the *cubic spline*

$$\Phi_1(x) = \begin{cases} \frac{2}{3} - 4x^2 + 4|x|^3 & \text{if } \quad |x| \le \frac{1}{2}, \\ \frac{4}{3} - 4|x| + 4x^2 - \frac{4}{3}|x|^3 & \text{if } \ \frac{1}{2} < |x| \le 1, \\ 0 & \text{if } \quad |x| > 1 \end{cases}$$

and the *conic function*

$$\Phi_1(x) = \begin{cases} (1-x^2)^k & \text{if } \ |x| \le 1, \\ 0 & \text{if } \ |x| > 1, \end{cases}$$

such that $k > 1$. See [2], [5], [4].

The $n$-dimensional weight function can be constructed from a one-dimensional weight function $\Phi_1$ by putting

$$\Phi(x) = \prod_{i=1}^n \Phi_1(x_i), \ \text{ where } x = (x_1, x_2, \dots, x_n).$$

**Definition 2** Let the particles $x^1, \ldots, x^N \in \Omega$, the degree $s$ of the monomial basis $p$ and the weight function $\Phi_1$ be given. Interpolants constructed by means of the RKPM are the linear combinations

$$\tilde{u}(x) = \sum_{I=1}^{N} \Psi_I(x) u_I \tag{6}$$

of the RKP shape-functions $\Psi_I$ with coefficients $u_I$. The shape functions are of the form

$$\Psi_I(x) = p^T \left( \frac{x^I - x}{\rho} \right) b(x) \, \Phi \left( \frac{x^I - x}{\rho} \right) \Delta V_I. \tag{7}$$

Here $\rho > 0$ is a dilatation parameter[1], $\Delta V_I$ is the quadrature weight and the function $b(x)$ is the solution of the linear equations

$$M(x) b(x) = p(0) \tag{8}$$

with the *moment matrix*

$$M(x) = \sum_{I=1}^{N} p \left( \frac{x^I - x}{\rho} \right) p^T \left( \frac{x^I - x}{\rho} \right) \Phi \left( \frac{x^I - x}{\rho} \right) \Delta V_I. \tag{9}$$

**Remark 4** Because the RKPM is based on the approximation of the kernel in the integral transformation (1) and this integral is discretized by means of numerical quadrature, the quadrature weight $\Delta V_I$ occurs in (7).

**Remark 5** There are some principles how to choose the particles $x^1, \ldots, x^N$ to receive suitable results. Especially, the necessary condition for the unique solvability of (8) is that

$$\operatorname{card} \left\{ x_I \middle| \, x \in \operatorname{supp} \Phi \left( \frac{x^I - x}{\rho} \right) \right\} \geq \dim \mathbb{P}^s$$

$\forall x \in \mathbb{R}^n$, see [2].

If we put $v = \Psi_K$ and insert the form (6) of the approximate solution into the weak formulation (5), we receive

$$\int_\Omega \sum_{i=1}^{n} \left( \sum_{I=1}^{N} u_I \frac{\partial \Psi_I}{\partial x_i} \right) \frac{\partial \Psi_K}{\partial x_i} \, \mathrm{d}x = \int_\Omega f \Psi_K \, \mathrm{d}x + \int_{\partial \Omega_1} g \Psi_K \, \mathrm{d}s, \;\; K = 1, 2, \ldots, N.$$

The matrix form of these equations for an unknown vector $u = (u_1, \ldots, u_N)^{\mathrm{T}}$ is

$$Au = b,$$

where

$$A = (A_{IK})_{I,K=1}^{N}, \quad A_{IK} = \int\limits_\Omega \sum_{i=1}^{n} \frac{\partial \Psi_I}{\partial x_i} \frac{\partial \Psi_K}{\partial x_i} \, \mathrm{d}x, \tag{10}$$

[1] The role of $\rho$ is to specify the size of supp $\Phi$.

$$b = (b_1, \dots, b_N)^T, \quad b_K = \int\limits_{\Omega} f\Psi_K \,\mathrm{d}x + \int\limits_{\partial\Omega_1} g\Psi_K \,\mathrm{d}s.$$

## 3 Methods for problems involving the Dirichlet boundary condition

As soon as we compute the components of the vector $u$, the approximation $\tilde{u}(x) = \sum_{I=1}^{N} \Psi_I(x)u_I$ is known. But there is a problem – there can be particles $x^J \in \partial\Omega_0$ such that $\Psi_I(x^J) \neq \delta_{IJ}$. Consequently,

$$\tilde{u}(x^J) = \sum_I \Psi_I(x^J)u_I \neq u_J.$$

It is the reason why the imposition of Dirichlet boundary conditions is not trivial. We will deal with the question how to remove this trouble.

### 3.1 Method of weight functions

The first idea how to satisfy the Dirichlet condition (for instance, see the article [11]) is to multiply the weight function $\Phi$ by a smooth function $w$ that is equal to one on $\Omega_0$ and declines to zero near $\partial\Omega_0$ successively. This correction is then reflected in the relations (9) and (7). The new moment matrix is

$$M(x) = \sum_{I=1}^{N} p\left(\frac{x^I - x}{\rho}\right) p^T \left(\frac{x^I - x}{\rho}\right) w(x)\Phi\left(\frac{x^I - x}{\rho}\right) \Delta V_I$$

and the new shape functions

$$\Psi_I(x) = p^T\left(\frac{x^I - x}{\rho}\right) b(x)\, w(x)\Phi\left(\frac{x^I - x}{\rho}\right) \Delta V_I.$$

Both the idea of the method and its implementation are simple. But, because the smoothness of the RKP-approximation depends on the smoothness of $w\Phi$, the smoothness of the approximate solution may become worse.

### 3.2 Transform method

The approximation (6) has to satisfy the Dirichlet condition. But the Dirichlet boundary conditions are prescribed for the real nodal values $\tilde{u}(x^I)$ and not for the unknowns $u_I$. This discrepancy can be removed in the following way: If we denote $\tilde{u}(x^I) = \tilde{u}_I$ and $\Psi_I(x^J) = T_{IJ}$, then the approximation (6) can be written in the form

$$\tilde{u}_J = \sum_{I=1}^{N} T_{IJ}u_I$$

now, or shortly $\tilde{u} = Tu$. If the matrix $T$ is non-singular, there exists an inverse matrix $T^{-1}$ such that $T^{-1}\tilde{u} = u$, i.e.

$$\sum_{J=1}^{N} T_{JI}^{-1}\tilde{u}_J = u_I.$$

Returning to the formula (6), we obtain

$$\tilde{u}(x) = \sum_{I=1}^{N} \Psi_I(x) u_I = \sum_{I=1}^{N} \Psi_I(x) \sum_{J=1}^{N} T_{JI}^{-1} \tilde{u}_J = \sum_{J=1}^{N} \left( \sum_{I=1}^{N} \Psi_I(x) T_{JI}^{-1} \right) \tilde{u}_J$$
$$= \sum_{J=1}^{N} \tilde{\Psi}_J(x) \tilde{d}_J.$$

The transformed functions $\tilde{\Psi}_J(x)$ have the Kronecker delta property now.

The transform method is based on manipulation with the matrix of values of the shape functions at given particles. In general, this matrix is full and, moreover, it is required to be non-singular. These facts belong to the disadvantages of the method. On the other hand, the Dirichlet condition is satisfied exactly at the particles from $\partial\Omega_0$. An application of the transform method can be found in the article [4].

### 3.3 Method of Lagrange multipliers

This method is based on a modification of the weak formulation of the problem given. The main idea is to minimize the functional

$$I(u, \lambda) = \frac{1}{2} \int_\Omega \sum_{i=1}^{n} \left( \frac{\partial u}{\partial x_i} \right)^2 \mathrm{d}x - \int_\Omega f u \,\mathrm{d}x - \int_{\partial\Omega_1} g u \,\mathrm{d}s + \int_{\partial\Omega_0} \lambda(u - u^0) \,\mathrm{d}s$$

with respect to $u$ and $\lambda$. We put

$$\tilde{u} = \sum_{I=1}^{N} \Psi_I u_I, \qquad \tilde{\lambda} = \sum_{I=1}^{N_0} \theta_I \lambda_I$$

in this case. The shape functions $\Psi_I$ are the same as in Section 2, $\theta_I$ are the linear Lagrange basis functions and $N_0$ is the number of points discretizing $\partial\Omega_0$. The method leads to the system of linear equations $Au = b$ such that

$$A = \begin{pmatrix} H & G \\ G^T & 0 \end{pmatrix}, \ u = (u_1, \ldots, u_N, \lambda_1, \ldots, \lambda_{N_0})^T, \ b = (b_1, \ldots, b_N, c_1, \ldots, c_{N_0})^T,$$

$$H_{IK} = \int\limits_\Omega \sum_{i=1}^{n} \frac{\partial \Psi_I}{\partial x_i} \frac{\partial \Psi_K}{\partial x_i} \,\mathrm{d}x, \ I = 1, \ldots, N, \ K = 1, \ldots, N,$$

$$G_{IK} = \int\limits_{\partial\Omega_0} \Psi_I \theta_K \,\mathrm{d}s, \ I = 1, \ldots, N, \ K = 1, \ldots, N_0,$$

$$b_K = \int\limits_\Omega f \Psi_K \,\mathrm{d}x + \int\limits_{\partial\Omega_1} g \Psi_K \,\mathrm{d}s, \ K = 1, \ldots, N, \ c_K = \int\limits_{\partial\Omega_0} u^0 \theta_K \,\mathrm{d}s, \ K = 1, \ldots, N_0.$$

In this method the matrix $A$ is more complicated than the matrix (10). It is also necessary to compute more unknown parameters. The advantage of this method is that it is general and accurate. The method was used for instance in the article [4].

## 4 Conclusions

This contribution deals with the question how to discretize the Dirichlet boundary conditions in the RKPM that occurs when the Dirichlet condition has to be realized in the RKPM. Three approaches – the method of weight functions, the transform method and the method of Lagrange multipliers – are described and their essential properties are discussed.

## References

[1] Alturi, S.N. and Zhu, T.: New concepts in meshless methods. Internat. J. Numer. Methods Engrg. **47** (2000), 537–556.

[2] Babuška, I., Banerjee, U., and Osborn, J.E.: Survey of meshless and generalized finite element mehods: A unified approach. Acta Numer. (2003), 1–125.

[3] Belytschko, T., Lu, Y., and Gu, I.: Element-free Galerkin methods. Internat. J. Numer. Methods Engrg. **37** (1994), 229–256.

[4] Chen, J.S., Pan, C., and Wu, C.T.: Large deformation analysis of rubber based on a reproducing kernel particle methods. Comput. Mech. **19** (1997), 211–227.

[5] Fries, T.P. and Matthies, H.G.: Classification and overview of meshfree methods. Inst. of Scientific Computing, Tech. University Brunswick, Germany, 2004.

[6] Han, W. and Meng, X.: Error analysis of the reproducing kernel particle method. Comp. Methods Appl. Mech. Engrg. **190** (2001), 6157–6181.

[7] Krysl, P. and Belytschko T.: Element-free Galerkin method: Convergence of the continuous and discontinuous shape functions. Computer Methods in Applied Mechanics and Engineering **148** (1997), 257–277.

[8] Li, S. and Liu, W.K.: Reproducing kernel hierarchical partition of unity. Internat. J. Numer. Methods Engrg. **45** (1999), 251–317.

[9] Liu, W.K., Jun, S., Li, S., Adee, J., and Belytschko, T.: Reproducing kernel particle methods for structural dynamics. Internat. J. Numer. Methods Engrg. **38** (1999), 1655–1679.

[10] Monaghan, J.J.: Why particle methods work. Sci. Stat. Comput. **3** (1982), 422–433.

[11] Mošová, V.: How to choose basis functions in meshless methods? In: *Lecture Notes in Computer Science*, vol. 5434, pp. 423–430. Springer-Verlag, Berlin, 2009.

[12] Strouboulis, T., Babuška, I., and Copps, K.: The design and analysis of the generalized finite element method. Comput. Methods Appl. Mech. Engrg. **181** (2000), 43–69.

# A POSTERIORI ERROR ESTIMATES OF THE DISCONTINUOUS GALERKIN METHOD FOR PARABOLIC PROBLEM[*]

Ivana Šebestová, Vít Dolejší

**Abstract**

We deal with a posteriori error estimates of the discontinuous Galerkin method applied to the nonstationary heat conduction equation. The problem is discretized in time by the backward Euler scheme and a posteriori error analysis is based on the Helmholtz decomposition.

## 1 Introduction

Our aim is to develop a sufficiently accurate and efficient numerical method for simulations of unsteady flows. A promising technique is a combination of the discontinuous Galerkin finite element method (DGFEM) for the space discretization and the backward difference formula for the time discretization, see [1], [2]. In order to both apply an adaptive algorithm and assess the discretization error, a posteriori error estimates have to be developed.

Within this paper, we focus on simplified model problem, represented by the heat equation, which is discretized by the high order DGFEM and the backward Euler method. We develop a posteriori error estimates based on the Helmholtz decomposition of the gradient of the error, see [4]. Therefore, this paper represents an extension of results from [6] where low order DGFEM was considered.

## 2 Problem definition

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be a bounded simply connected polyhedral Lipschitz domain with a boundary $\partial\Omega$, $T > 0$ and $Q_T = \Omega \times (0, T)$. Let us consider the problem:

$$\begin{array}{rcll} \partial u/\partial t - \Delta u & = & f & \text{in} \quad Q_T, \\ u & = & 0 & \text{on} \quad \partial\Omega \times (0,T), \\ u(x,0) & = & u^0(x) & \text{in} \quad \Omega. \end{array} \tag{1}$$

We use a standard notation for the Lebesgue, Sobolev and Bochner spaces. We introduce a weak formulation of (1).

[*]This work is a part of the research projects MSM 0021620839 (V. Dolejší). The research of I. Šebestová was supported by grant No. 10209/B-MAT/MFF of the Grant Agency of Charles University Prague.

**Definition 1.** *The function $u : Q_T \to \mathbb{R}$ such that $u \in L^2(0,T; H_0^1(\Omega))$ and $\partial u/\partial t \in L^2(0,T; H_0^1(\Omega))$ is the weak solution of the problem* (1) *if*

$$\begin{array}{rcl} \langle \partial u(t)/\partial t, v\rangle + \int_\Omega \nabla u(t)\cdot\nabla v\,dx & = & \langle f(t), v\rangle \quad \forall v \in H_0^1(\Omega),\ \textit{for a.a. } t\in(0,T), \\ u(x,0) & = & u^0(x) \quad \textit{in } \Omega, \end{array} \tag{2}$$

*where $\langle\cdot,\cdot\rangle$ denotes the duality pairing between $H_0^1(\Omega)$ and $H^{-1}(\Omega)$ and where we assume $f \in C(0,T; H^{-1}(\Omega))$ and $u^0 \in L^2(\Omega)$.*

## 3 Discretization

### 3.1 Time semidiscretization

Let $0 = t_0 < t_1 < ... < t_{\bar{N}} = T$ be a partition of the time interval $[0,T]$ and set $\tau_n = t_n - t_{n-1}$, $\tau = \max\{\tau_n : 1 \le n \le \bar{N}\}$. We use the backward Euler scheme in (2) and get the semi-discrete problem: Find a sequence $\{u^n\}_{1\le n\le\bar{N}}$, $u^n \in H_0^1(\Omega)$ such that

$$\int_\Omega \frac{u^n - u^{n-1}}{\tau_n} v\,dx + \int_\Omega \nabla u^n\cdot\nabla v\,dx = \int_\Omega f^n v\,dx \quad \forall v \in H_0^1(\Omega),$$

where $f^n = f(\cdot, t_n)$.

### 3.2 Space discretization

We will carry out the space discretization with the aid of the DGFEM. On each time level $t_n$, $n = 1,\dots,\bar{N}$, we consider a family $\{\mathcal{T}_{h,n}\}_{h>0}$ of partitions of $\Omega$ into a finite number of closed triangles in 2D and tetrahedra in 3D with mutually disjoint interiors. We assume that the following conditions are satisfied.

$$\text{shape regularity:}\quad \exists C_s > 0 : \frac{h_K}{\rho_K} \le C_s\ \forall\, K \in \mathcal{T}_{h,n}, \tag{3}$$

$$\text{local quasi-uniformity:}\quad \exists C_H > 0 : \ h_K \le C_H h_{K'}\ \forall K, K^{'} \in \mathcal{T}_{h,n} \text{ sharing a face}, \tag{4}$$

where $h_K = \operatorname{diam}(K)$ for $K \in \mathcal{T}_{h,n}$, $\rho_K$ denotes the radius of the largest $d$-dimensional ball inscribed into $K$, and $\partial K$ denotes the boundary of element $K$. Moreover, we assume that there exists a triangulation $\widetilde{\mathcal{T}}_{h,n}$ satisfying (3) and (4) which is a refinement of both $\mathcal{T}_{h,n-1}$ and $\mathcal{T}_{h,n}$, $1 \le n \le \bar{N}$ and such that

$$\exists C_{HT} > 0 : \ \forall 1 \le n \le \bar{N}\ \forall K \in \widetilde{\mathcal{T}}_{h,n}\ \forall K^{'} \in \mathcal{T}_{h,n}, K \subset K^{'} : \ \frac{h_{K'}}{h_K} < C_{HT}.$$

By $\mathcal{F}_{h,n}^I$ and $\mathcal{F}_{h,n}^D$ we denote the set of all interior faces (edges for $d = 2$) and faces (edges for $d = 2$) on $\partial\Omega$, respectively. For a simplicity, we put $\mathcal{F}_{h,n} = \mathcal{F}_{h,n}^I \cup \mathcal{F}_{h,n}^D$. Further, we set $h_\Gamma = \operatorname{diam}(\Gamma)$ for $\Gamma \in \mathcal{F}_{h,n}$. For each $\Gamma \in \mathcal{F}_{h,n}^I$ there exist two elements $K_\Gamma^L$ and $K_\Gamma^R$ such that $\Gamma \subset \overline{K_\Gamma^L} \cap \overline{K_\Gamma^R}$. We define a unit normal vector $\boldsymbol{n}_\Gamma$

to each $\Gamma \in \mathcal{F}_h^I$ so that it points out of $K_\Gamma^L$. Finally, we assume that $\boldsymbol{n}_\Gamma$, $\Gamma \in \mathcal{F}_{h,n}^D$, has the same orientation as the outward normal to $\partial\Omega$.

Over the triangulation $\widetilde{\mathcal{T}}_{h,n}$ we define the so-called broken Sobolev space

$$H^s(\Omega, \widetilde{\mathcal{T}}_{h,n}) = \{v; v|_K \in H^s(K)\, \forall K \in \widetilde{\mathcal{T}}_{h,n}\}$$

equipped with the norm $\|v\|^2_{H^s(\Omega,\widetilde{\mathcal{T}}_{h,n})} = \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \|v\|^2_{H^s(K)}$. For $v \in H^1(\Omega, \widetilde{\mathcal{T}}_{h,n})$ we define the broken gradient $\nabla_h v$ of $v$ by $(\nabla_h v)|_K = \nabla(v|_K)$ for $\forall K \in \widetilde{\mathcal{T}}_{h,n}$ and use the following notation: $v_\Gamma^L$ stands for the trace of $v|_{K_\Gamma^L}$ on $\Gamma$, $v_\Gamma^R$ is the trace of $v|_{K_\Gamma^R}$ on $\Gamma$, $\langle v\rangle_\Gamma = \frac{1}{2}(v_\Gamma^L + v_\Gamma^R)$, $[v]_\Gamma = v_\Gamma^L - v_\Gamma^R$, $\Gamma \in \mathcal{F}_{h,n}^I$. Further, for $\Gamma \in \mathcal{F}_{h,n}^D$, we define $v_\Gamma^L$ as the trace of $v|_{K_\Gamma^L}$ on $\Gamma$, and $\langle v\rangle_\Gamma = [v]_\Gamma = v_\Gamma^L$. If $[\cdot]_\Gamma$ and $\langle\cdot\rangle_\Gamma$ appear in an integral of the form $\int_\Gamma \dots dS$, we will omit the subscript $\Gamma$ and write $[\cdot]$ and $\langle\cdot\rangle$ instead. Finally, we define the space of discontinuous piecewise polynomial functions

$$S_{hp}^n = \{v; v \in L^2(\Omega), v|_K \in P^p(K)\, \forall\, K \in \widetilde{\mathcal{T}}_{h,n}\},$$

where $P^p(K)$ is the space of all polynomials on $K$ of degree $p$.

Now, we can state the discrete problem: For a given approximation $u_h^0 \in S_{hp}^0$ of an initial condition $u^0$ find a sequence $\{u_h^n\}_{1\le n\le \bar{N}}$, $u_h^n \in S_{hp}^n$ such that

$$\int_\Omega \frac{u_h^n - u_h^{n-1}}{\tau_n} v_h\, dx + \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_K \nabla u_h^n \cdot \nabla v_h\, dx - \sum_{\Gamma\in\mathcal{F}_{h,n}} \int_\Gamma \langle \nabla u_h^n \cdot \boldsymbol{n}\rangle [v_h]\, dS$$
$$+\theta \sum_{\Gamma\in\mathcal{F}_{h,n}} \int_\Gamma \langle \nabla v_h \cdot \boldsymbol{n}\rangle [u_h^n]\, dS + \sum_{\Gamma\in\mathcal{F}_{h,n}} \int_\Gamma \sigma [u_h^n][v_h]\, dS = \int_\Omega f^n v_h\, dx$$

for all $v_h \in S_{hp}^n$, where $\theta = -1$, $\theta = 1$, and $\theta = 0$ corresponds to the symmetric, nonsymmetric, and incomplete variants of the DGFEM, respectively.

In this section, we derive a residual-based a posteriori error estimate of the discretization error based on the Helmholtz decomposition of the gradient of the error. This approach was developed in [4], where the heat equation was solved with the aid of the combination of the Crouzeix-Raviart nonconforming finite elements in space and the backward Euler scheme in time.

Since the time error estimation is almost the same as in [4], we focus on the spatial error estimation.

**Definition 2.** *Let $\{u^n\}_{1\le n\le\bar{N}}$ be the semi-discrete solution and $\{u_h^n\}_{1\le n\le\bar{N}}$ be the discrete solution of* (1)*. Then we set*

$$\{e^n\}_{1\le n\le\bar{N}} = \{u^n - u_h^n\}_{1\le n\le\bar{N}}.$$

We will need an interpolation operator that maps $H^1(\Omega, \widetilde{\mathcal{T}}_{h,n})$ into $S_{hp}^n \cap H_0^1(\Omega)$.

### 3.3 Oswald interpolation operator

Let $\mathcal{N}^0_{h,n}$ be the set of all Lagrangian vertices of the elements of $\widetilde{\mathcal{T}}_{h,n}$. According to, e.g., [3], we define the Oswald interpolation operator $\mathcal{I}^0_{Os} : S^n_{hp} \to S^n_{hp} \cap H^1_0(\Omega)$ by

$$\begin{aligned}\mathcal{I}^0_{Os}(v_h)(\nu) &= \frac{1}{\mathrm{card}(\omega_\nu)} \sum_{K\in\omega_\nu} v_h|_K(\nu), \quad \nu \in \mathcal{N}^0_{h,n} \backslash \mathcal{N}^B_{h,n} \\ &= 0, \quad \nu \in \mathcal{N}^B_{h,n}\end{aligned}$$

where $\omega_\nu = \{K \in \widetilde{\mathcal{T}}_{h,n};\ \nu \in K\}$, $\mathcal{N}^B_{h,n} = \{\nu \in \mathcal{N}^0_{h,n}; \nu \in \partial\Omega\}$. Moreover, we define the interpolation operator $I^0_{h,n} : H^1(\Omega, \widetilde{\mathcal{T}}_{h,n}) \to S^n_{hp} \cap H^1_0(\Omega)$ by

$$I^0_{h,n}(v) = \mathcal{I}^0_{Os}(\Pi_{hp}(v)) \quad \forall\, v \in H^1(\Omega, \widetilde{\mathcal{T}}_{h,n}),$$

where $\Pi_{hp}$ denotes the $L^2$-projection of $v$ on the space $S^n_{hp}$.

In order to overcome difficulties with the nonconformity of $S^n_{hp}$, the Helmholtz decomposition of the gradient of the error is carried out as follows (see, e.g., [5]):

$$\nabla_h e^n = \nabla \phi^n + \mathrm{curl}\, \chi^n, \tag{5}$$

where $\phi^n \in H^1_0(\Omega) = \{v \in H^1(\Omega);\ v = 0 \text{ on } \partial\Omega\}$ is the solution of the problem

$$\int_\Omega \nabla\phi^n \cdot \nabla v\, dx = \int_\Omega \nabla_h e^n \cdot \nabla v\, dx \quad \forall v \in H^1_0(\Omega),$$

$\chi^n \in H(\mathrm{curl}, \Omega) = \{v \in (L^2(\Omega))^k;\ \mathrm{curl}\, v \in (L^2(\Omega))^d\}$ ($k = 1$ for $d = 2$ and $k = 3$ for $d = 3$). Moreover, the following holds: $\|\nabla_h e^n\|^2_\Omega = \|\nabla\phi^n\|^2_\Omega + \|\mathrm{curl}\,\chi^n\|^2_\Omega$. The orthogonality of the splitting is crucial because it suffices to estimate each part of the error independently. A proof of the above assertions can be found in [5].

Furthermore, we recall some fundamental properties presented in [6].

**Lemma 1.** *Let $v_h \in S^n_{hp} \cap H^1_0(\Omega)$, $\phi \in H^1_0(\Omega)$ and $\chi \in (H^1(\Omega))^k$ ($k = 1$ for $d = 2$ and $k = 3$ for $d = 3$) be arbitrary. The error $e^n$ satisfies*

$$\sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_K \nabla e^n \cdot \nabla v_h\, dx = \int_\Omega \frac{e^{n-1} - e^n}{\tau_n} v_h\, dx + \theta \sum_{\Gamma\in\mathcal{F}^I_{h,n}} \int_\Gamma \langle \nabla v_h \cdot \boldsymbol{n}\rangle [u^n_h]\, dS, \tag{6}$$

$$\begin{aligned}\sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_K \nabla e^n \cdot \nabla\phi\, dx &= \int_\Omega (f^n - \frac{u^n - u^{n-1}}{\tau_n})\phi\, dx - \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_{\partial K} \nabla u^n_h \cdot \boldsymbol{n}\phi\, dS \\ &\quad + \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_K \Delta u^n_h \phi\, dx, \end{aligned}\tag{7}$$

$$\sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_K \nabla e^n curl\chi\, dx = - \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_{\partial K} u^n_h curl\chi \cdot \boldsymbol{n}\, dS. \tag{8}$$

**Definition 3.** *Let $n \geq 1$. We define the local spatial error indicator by*

$$\eta_K^n = h_K \left\| f^n + \Delta u_h^n - \frac{u_h^n - u_h^{n-1}}{\tau_n} \right\|_K + h_K^{1/2} \|\nabla u_h^n \cdot \boldsymbol{n}\|_{\partial K} + \|u_h^n\|_{H^{1/2}(\partial K)}$$
$$+ \sum_{\Gamma \in \mathcal{F}_{h,n} \cap \mathcal{F}_K} \left( h_\Gamma^{-1/2} \|[u_h^n]\|_\Gamma + h_\Gamma^{1/2} \|[u_h^n]\|_\Gamma \right),$$

*where $\mathcal{F}_K$ denotes the set of all edges or faces of a triangle or of a tetrahedron $K$, respectively, and $||\cdot||_K$ stands for the $L^2(K)$-norm. The global spatial error estimator is defined by $\eta^n = (\sum_{K \in \widetilde{\mathcal{T}}_{h,n}} (\eta_K^n)^2)^{1/2}$.*

Now, we state the main result, an upper bound on the error.

**Theorem 1.** *Let $\{u^n\}_{1 \leq n \leq \bar{N}}$ be the semi-discrete solution and $\{u_h^n\}_{1 \leq n \leq \bar{N}}$ be the discrete solution of* (1)*. Let $1 \leq N \leq \bar{N}$. Then the error $e^n$ satisfies*

$$\sum_{K \in \widetilde{\mathcal{T}}_{h,N}} \|e^N\|_K^2 + \sum_{n=1}^{N} \tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \|\nabla e^n\|_K^2 \leq \sum_{K \in \widetilde{\mathcal{T}}_{h,1}} \|e^0\|_K^2 + \sum_{n=1}^{N} C(\eta^n)^2 (1 + \max\{h_n^2, \tau_n\}),$$

*where a constant $C$ is independent of the mesh parameter and the time step.*

**Sketch of the proof:** According to (5), we can write

$$\tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \|\nabla e^n\|_K^2 = \tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \int_K \nabla e^n \cdot \nabla \phi^n \, dx$$
$$+ \tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \int_K \nabla e^n \operatorname{curl} \chi^n \, dx. \tag{9}$$

Denoting $\psi_1$ and $\psi_2$ the two terms on the right-hand side of (9), setting $\phi = \phi^n$ in (7), $\chi = \chi^n$ in (8) and multiplying both inequalities by $\tau_n$ yield

$$\psi_1 = \tau_n \int_\Omega (f^n - \frac{u^n - u^{n-1}}{\tau_n}) \phi^n \, dx - \tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \int_{\partial K} \nabla u_h^n \cdot \boldsymbol{n} \phi^n \, dS$$
$$+ \tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \int_K \Delta u_h^n \phi^n \, dx,$$
$$\psi_2 = -\tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \int_{\partial K} u_h^n \operatorname{curl} \chi^n \cdot \boldsymbol{n} \, dS.$$

Now, we modify the expression $\psi_1$. Adding $\tau_n$ multiple of (6) with $v_h = I_{h,n}^0 \phi^n$ to $\psi_1$ and expressing term $-\tau_n \sum_{K \in \widetilde{\mathcal{T}}_{h,n}} \int_K \nabla e^n \cdot \nabla I_{h,n}^0 \phi^n \, dx$ according to identity (7), we obtain

$$
\begin{aligned}
\psi_1 \;=\; & \tau_n \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_K \Big(f^n + \Delta u_h^n - \frac{u^n - u^{n-1}}{\tau_n}\Big)(\phi^n - I_{h,n}^0 \phi^n)\, dx \\
& -\tau_n \int_\Omega \frac{e^{n-1} - e^n}{\tau_n}\, I_{h,n}^0 \phi^n \, dx - \tau_n \sum_{K\in\widetilde{\mathcal{T}}_{h,n}} \int_{\partial K} \nabla u_h^n \cdot \boldsymbol{n} (\phi^n - I_{h,n}^0\phi^n)\, dS \\
& +\tau_n \theta \sum_{\Gamma\in\mathcal{F}_{h,n}^I} \int_\Gamma \langle \nabla I_{h,n}^0 \phi^n \cdot \boldsymbol{n}\rangle [u_h^n]\, dS.
\end{aligned} \qquad (10)
$$

By adding and subtracting suitable terms in (10), estimating all terms in $\psi_1$ and $\psi_2$ using approximation properties of $I_{h,n}^0$, trace inequalities, inverse inequality, and well known inequalities such as Hölder's, Young's, etc., we finally come to the assertion of Theorem 1.

## 4 Conclusion

We derived the error upper bound for the heat conduction equation discretized by the high order discontinuous Galerkin finite element method in space and the backward Euler scheme in time. Analogously to [4], the Helmholtz decomposition was used to overcome difficulties arising due to the nonconformity of the DGFEM.

## References

[1] Dolejší V., Feistauer M., Kučera V., and Sobotíková V.: $L^\infty(L^2)$-error estimates for the DGFEM applied to convection-diffusion problems on nonconforming meshes. J. Numer. Math. **17** (2009), 45–65.

[2] Dolejší V. and Vlasák M.: Analysis of a BDF – DGFE scheme for nonlinear convection-diffusion problems. Numer. Math. **110** (2008), 405–447.

[3] Karakashian O. A. and Pascal F.: A posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems. SIAM J. Numer. Anal. **41** (2003), 2374–2399.

[4] Nicaise S. and Soualem N.: A posteriori error estimates for a nonconforming finite element discretization of the heat equation. M2AN Math. Model. Numer. Anal. **39** (2005), 319–348.

[5] Dari E., Duran R., Padra C., and Vampa V.: A posteriori error estimators for nonconforming finite element methods. M2AN Math. Model. Numer. Anal. **30** (1996), 385–400.

[6] Šebestová, I.: *A posteriori error estimates of the discontinuous Galerkin method for convection-diffusion equations.* Master thesis, Charles University in Prague, 2009.

# A COMPARISON OF SOME A POSTERIORI ERROR ESTIMATES FOR FOURTH ORDER PROBLEMS*

Karel Segeth

**Abstract**

A lot of papers and books analyze analytical a posteriori error estimates from the point of view of robustness, guaranteed upper bounds, global efficiency, etc. At the same time, adaptive finite element methods have acquired the principal position among algorithms for solving differential problems in many physical and technical applications. In this survey contribution, we present and compare, from the viewpoint of adaptive computation, several recently published error estimation procedures for the numerical solution of biharmonic and some further fourth order problems including computational error estimates.

## 1 Introduction

In the *hp*-adaptive finite element method, there are two possibilities to assess the error of the computed solution a posteriori: to construct an *analytical error estimate* or to obtain, by the same procedure as the approximate solution, a *computational error estimate.* In the latter case, a *reference solution* is computed in a systematically refined mesh and, at the same time, with polynomial degree of all elements increased by 1 (see, e.g., [4], [9]).

In the paper, we are concerned with several formulations of the biharmonic problem and a general 4th order elliptic problem on a 2D domain. We present analytical a posteriori error estimates of different nature found in literature for these problems. We are primarily concerned with the computability of the right-hand parts of the estimates. In conclusion, we assess the advantages and drawbacks of the analytical as well as computational estimates in general.

We use common notation based primarily on the book [3]. For the lack of space, we sometimes only refer to the notation introduced in the papers quoted. The complete hypotheses of the theorems presented should be also looked for there. A more detailed version of the paper should appear elsewhere.

## 2 Dirichlet and second problems for biharmonic equation

### 2.1 Dirichlet problem

Let $\Omega \subset R^2$ have a polygonal boundary $\Gamma$. We consider the two dimensional biharmonic problem

*This research was supported by the Grant Agency of the Academy of Sciences of the Czech Republic under Grant IAA100190803 and by the Academy of Sciences of the Czech Republic under Research Plan AV0Z10190503 of the Institute of Mathematics.

$$\Delta^2 u = f \quad \text{in} \quad \Omega, \tag{2.1}$$

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \Gamma \tag{2.2}$$

with $f \in L_2(\Omega)$ that models, e.g., the vertical displacement of the mid-surface of a clamped plate subject to bending.

We use the standard formulation of the weak solution $u \in X = H_0^2(\Omega)$ and approximate solution $u_h \in X_h$ written in the form $\langle F(u), v\rangle = 0$ and $\langle F_h(u_h), v_h\rangle = 0$. Denote by $k$, $k \geq 1$, the maximum degree of polynomials in $X_h$. Further, put $f_h = \sum_{T\in\mathcal{T}_h} \pi_{l,T} f$, where $T$ is a triangle of the triangulation $\mathcal{T}_h$, $\mathcal{E}_h$ is the set of all its edges, $P_l$, $l \geq 0$ fixed, is the space of polynomials of degree at most $l$ and $\pi_{l,S}$, $S \in \mathcal{T}_h \cup \mathcal{E}_h$, is the $L_2$ projection of $L_1(S)$ onto $P_{l|S}$.

Put $\varepsilon_T = \|f - f_h\|_{0;T}$. Let $h_T$ be the diameter of the triangle $T$ and $h_E$ the length of the edge $E$, $\mathcal{E}(T)$ the set of all edges of the triangle $T$, and $\mathcal{E}_{h,\Omega}$ the set of all inner edges of $\mathcal{T}_h$. Denote by $n_E$ the normal to the edge $E$ and by $[q]_E$ the jump of the function $q$ over the edge $E$. Defining the *local residual a posteriori error estimator*

$$\eta_{\mathrm{V},T} = \left( h_T^4 \|\Delta^2 u_h - f_h\|_{0;T}^2 + \sum_{E\in\mathcal{E}(T)\cap\mathcal{E}_{h,\Omega}} (h_E \|[\Delta u_h]_E\|_{0;E}^2 + h_E^3 \|[n_E \cdot \nabla \Delta u_h]_E\|_{0;E}^2) \right)^{1/2}$$

for all $T \in \mathcal{T}_h$, we have the following theorem [11].

**Theorem 2.1** *Let $u \in X$ be the unique weak solution of the problem* (2.1), (2.2) *and let $u_h \in X_h$ be an approximate solution of the corresponding discrete problem. Then we have the a posteriori estimates*

$$\|u - u_h\|_2 \leq c_1 \left( \sum_{T\in\mathcal{T}_h} \eta_{\mathrm{V},T}^2 \right)^{1/2} + c_2 \left( \sum_{T\in\mathcal{T}_h} h_T^4 \varepsilon_T^2 \right)^{1/2} + c_3 \|F(u_h) - F_h(u_h)\| + c_4 \|F_h(u_h)\|$$

*and*

$$\eta_{\mathrm{V},T} \leq c_5 \|u - u_h\|_{2;\omega_T} + c_6 \left( \sum_{T'\subset\omega_T} h_{T'}^4 \varepsilon_{T'}^2 \right)^{1/2}$$

*for all $T \in \mathcal{T}_h$. The quantities $c_1, \dots, c_6$ depend only on $h_T/\rho_T$, and the integers $k$ and $l$. Here $\omega_T$ is the set of all neighbors of the triangle $T$ and $\rho_T$ the diameter of the circle inscribed to $T$.*

The proof is given in [11].

### 2.2 Dirichlet problem in mixed formulation

Let $\Omega \subset R^2$ be a convex polygon with boundary $\Gamma$. Again, we consider the biharmonic problem (2.1), (2.2) with $f \in H^{-1}(\Omega)$. The problem is concerned in practice with both linear plate analysis and incompressible flow simulation.

We employ the Ciarlet-Raviart weak formulation of the problem (2.1) and (2.2) for the solution $\{w = \Delta u, u\}$ and the corresponding conforming second order approximate solution $\{w_h, u_h\}$. Let us put $f_h = \Pi_h f$ where $\Pi_h$ denotes the $L_2$ orthogonal projection on the set of piecewise constant functions on triangles.

The local residuals $\mathcal{P}_T$, $\mathcal{R}_T$, $\mathcal{P}_E$, and $\mathcal{R}_E$ are defined in [2]. Denoting the area of the triangle $T$ by $|T|$, we introduce the *local residual a posteriori error estimators*

$$\eta^2_{\mathrm{C},T} = |T| \|\mathcal{P}_T(u_h)\|^2_{0;T} + \tfrac{1}{2} \sum_{E \in \mathcal{E}(T)} h_E \|\mathcal{P}_E(u_h)\|^2_{0;E}$$

and $\widetilde{\eta}_{\mathrm{C},T}$ computed from $\mathcal{R}_T$ and $\mathcal{R}_E$. We put $e_h(u) = u - u_h$ and $e_h(w) = w - w_h$. Then the following theorem holds [2].

**Theorem 2.2** *Let $\{w, u\}$ be the unique mixed weak solution of the problem* (2.1) *and* (2.2)*, and let $\{w_h, u_h\}$ be an approximate solution of the corresponding discrete problem. For $T \in \mathcal{T}_h$ we then have the a posteriori estimates*

$$\|e_h(u)\|_1 + h\|e_h(w)\|_0 \leq C_1 \left( \left( \sum_{T \in \mathcal{T}_h} \eta^2_{\mathrm{C},T} \right)^{1/2} + h^2 \left( \sum_{T \in \mathcal{T}_h} \widetilde{\eta}^2_{\mathrm{C},T} \right)^{1/2} \right),$$

$$\eta_{\mathrm{C},T} + h^2 \widetilde{\eta}_{\mathrm{C},T} \leq C_2 \left( |e_h(u)|_{1;\omega_T} + h_T \|e_h(w)\|_{0;\omega_T} + h_T^3 \sum_{T' \subset \omega_T} \varepsilon_{T'} \right)$$

*with some positive constants $C_1$ and $C_2$ independent of $h$.*

The proof is given in [2].

### 2.3 Second problem in mixed formulation

Let $\Omega \subset R^2$ be a convex polygon with boundary $\Gamma$. We consider the two dimensional second biharmonic problem for the equation (2.1) with the boundary condition

$$u = \Delta u = 0 \quad \text{on} \quad \Gamma \tag{2.3}$$

with $f \in L_2(\Omega)$ that models the deformation of a simply supported thin elastic plate.

Again, we employ the Ciarlet-Raviart weak formulation of the problem (2.1) and (2.3). We introduce the quantities $\varepsilon_1$, $\varepsilon_2$, the gradient recovery operator $Gv_h$, and the *gradient recovery a posteriori error estimator* $\eta_{\mathrm{L}}$ like in [6]. Then the following theorem holds.

**Theorem 2.3** *Let $\{w, u\}$ be the unique weak solution of the problem* (2.1) *and* (2.3), *and let $\{w_h, u_h\}$ be an approximate solution of the corresponding discrete problem. Then we have the a posteriori estimates*

$$c\eta_{\mathrm{L}}^2 - C_2\varepsilon_2^2 \le |w - w_h|_1^2 + |u - u_h|_1^2 \le C\eta_{\mathrm{L}}^2 + C_1\varepsilon_1^2$$

*with some positive constants $c$, $C$, $C_1$, and $C_2$ independent of $h$.*

The proof is given in [6].

### 2.4 Kirchhoff plate bending problem

We consider the bending problem of an isotropic linearly elastic plate. We employ the Kirchhoff plate bending model for the deflection $u \in H_0^2$ of the plate in the weak formulation [1]. The nonconforming finite element approximation of the problem is done in the *discrete Morley space* $W_h$ of second degree piecewise polynomial functions on $\mathcal{T}_h$ [1].

Let us introduce the norm $|\!|\!|w|\!|\!|_h$ in $W_h \cup H^2$ and define the *local a posteriori error estimator* $\eta_{\mathrm{M},T}$ like in [1]. Then the following theorem holds.

**Theorem 2.4** *Let $u \in H_0^2$ be the unique weak solution of the Kirchhoff plate bending problem and let $u_h \in W_h$ be an approximate solution of the corresponding discrete problem. Then we have the a posteriori estimates*

$$|\!|\!|u - u_h|\!|\!|_h \le C\left(\sum_{T\in\mathcal{T}_h}\eta_{\mathrm{M},T}^2 + \sum_{T\in\mathcal{T}_h} h_T^4\varepsilon_T^2\right)^{1/2} \quad \textit{and} \quad \eta_{\mathrm{M},T} \le |\!|\!|u - u_h|\!|\!|_{h;T} + h_T^2\varepsilon_T$$

*with some positive constant $C$ independent of $h$ and for all $T \in \mathcal{T}_h$.*

The proof is given in [1].

## 3 Dirichlet problem for fourth order elliptic equation

**3.1.** Let $\mathrm{D}^2u$ denote the Hessian matrix of a function $u : \Omega \to R$, $u \in H^2(\Omega)$. Let the matrix-valued function $\Lambda = [\lambda_{ik}]$, $\Lambda : \Omega \times R^{n\times n} \to R^{n\times n}$ be measurable and bounded with respect to the variable $x \in \Omega$ and of class $C_2$ with respect to the matrix variable $\Theta \in R^{n\times n}$.

Let $\Omega \subset R^n$ have a piecewise $C_1$ boundary. We consider the fourth order problem

$$\mathrm{div}^2\Lambda(x, \mathrm{D}^2u) = f \quad \text{in} \quad \Omega \tag{3.1}$$

with the boundary condition (2.2) and $f \in L_2(\Omega)$.

We assume that $\Lambda'$ is positive definite with constants $0 < m \le M$. We introduce the weak solution $u \in H_0^2(\Omega)$ in the usual way.

Let $\bar{u}$ be an arbitrary function from $H_0^2(\Omega)$ considered as an approximation of the solution $u$. We measure the error of $\bar{u}$ by the functional $E(\bar{u})$ depending on $\Lambda$, $\mathrm{D}^2$, and $f$ [5].

For an arbitrary matrix-valued function $\Psi \in H(\mathrm{div}^2, \Omega) \cap L_\infty(\Omega, R^{n\times n})$ and an arbitrary scalar-valued function $w \in H_0^2(\Omega)$, define the *global a posteriori error estimator* $\eta_\mathrm{K}(\Psi, w, \bar{u})$ depending on $m$, $M$, the constant from the Friedrichs inequality for $\mathrm{D}^2$ on $H_0^2(\Omega)$, and the Lipschitz continuity constant of $\Lambda'$ [5]. Then the following theorem holds.

**Theorem 3.1** *Let $u \in H_0^2(\Omega)$ be the unique weak solution of the problem* (3.1), (2.2) *and $\bar{u} \in W^{2,\infty}(\Omega)$ an arbitrary function. Then*

$$E(\bar{u}) \leq \eta_\mathrm{K}(\Psi, w, \bar{u}) \tag{3.2}$$

*for any $\Psi \in H(\mathrm{div}^2, \Omega) \cap L_\infty(\Omega, R^{n\times n})$ and $w \in H_0^2(\Omega)$.*

The proof of the theorem is based on a more general statement proven in [5]. An analogous result is obtained there for a similar error estimator easier to compute. There is an interesting question of optimizing the inequality (3.2) with respect to $\Psi$ and $w$. Moreover, it is proven in [5] that the estimator $\eta_\mathrm{K}$ is sharp.

**3.2.** Let $\Omega \in R^n$ be a bounded connected domain and $\Gamma$ its Lipschitz continuous boundary. We consider the 4th order elliptic problem for a scalar-valued function $u$,

$$\mathrm{div}\,\mathrm{Div}(\gamma\nabla\nabla u) = f \quad \text{in} \quad \Omega, \tag{3.3}$$

with the boundary condition (2.2) and $f \in L_2(\Omega)$, $\gamma = [\gamma_{ijkl}]_{i,j,k,l=1}^n$ and $\gamma_{ijkl} = \gamma_{jikl} = \gamma_{klij} \in L_\infty(\Omega)$.

We define the energy norm $|\!|\!|\Phi|\!|\!|$ in $L_2(\Omega, R^{n\times n})$ and the *global a posteriori error estimator* $\eta_\mathrm{R}(\beta, \Phi, \bar{u})$ like in [8], where $\beta$ is an arbitrary positive real number and $\Phi$ an arbitrary smooth matrix-valued function. The estimator depends on the constant from the Friedrichs inequality for $\nabla\nabla$ on $H_0^2(\Omega)$. We then have the following theorem [8].

**Theorem 3.2** *Let $u \in H_0^2(\Omega)$ be the weak solution of the problem* (3.3), (2.2) *and $\bar{u} \in H_0^2(\Omega)$ an arbitrary function. Then*

$$|\!|\!|\nabla\nabla(\bar{u} - u)|\!|\!|^2 \leq \eta_\mathrm{R}(\beta, \Phi, \bar{u}) \tag{3.4}$$

*for any positive number $\beta$ and any matrix-valued function $\Phi \in H(\mathrm{div}\,\mathrm{Div}, \Omega)$.*

The proof of the theorem is based on a more general statement proven in [8]. There is an interesting question of optimizing the inequality (3.4) with respect to $\beta$ and $\Phi$. To avoid possible smoothness difficulties we can introduce a further global error estimator and prove the same statement as in Theorem 3.2 [8].

## 4 Conclusion

In the paper, we have presented several analytical a posteriori error estimators that appear in inequalities, usually with some unknown constants on the right-hand part. The quantitative properties of the estimators cannot be easily assessed and compared analytically. Only numerical experiment can be the means for this purpose. There are, however, global analytical error estimates for some classes of problems (see, e.g., [5], [7], [8]) that require as few unknown constants as possible. Some papers provide for the estimation of these constants. Analytical estimates are usually efficient in practice if they are asymptotically exact. The a posteriori estimates with unknown constants, however, are not optimal for the practical computation.

Exceptionally, there are analytical estimates containing really no unknown constants (see, e.g., [10] for a 2D linear 2nd order elliptic problem).

The paper is closely connected with the automatic *hp*-adaptivity that gives many $h$ as well as $p$ possibilities for the next step of the solution process. A single number provided by the local analytical a posteriori error estimator for each mesh element need not be enough information for the decision. This is the reason for using the computational error estimate (reference solution). The computation of the reference solution is rather time-consuming but it is obtained by the same software that is used to compute the approximate solution. We use reference solutions as robust error estimators with no unknown constants to control the adaptive strategies in the most complex finite element computations.

## References

[1] Beirão da Veiga, L., Niiranen, J., and Stenberg, R.: A posteriori error estimates for the Morley plate bending element. Numer. Math. **106** (2007), 165–179.

[2] Charbonneau, A., Dossou, K., and Pierre, R.: A residual-based a posteriori error estimator for the Ciarlet-Raviart formulation of the first biharmonic problem. Numer. Methods Partial Differential Equations **13** (1997), 93–111.

[3] Ciarlet, P.G.: *The finite element method for elliptic problems.* North Holland, Amsterdam, 1978.

[4] Demkowicz, L.: *Computing with hp-adaptive finite elements*, vol. 1, 2. Chapman & Hall/CRC, Boca Raton, FL, 2007, 2008.

[5] Karátson, J. and Korotov, S.: Sharp upper global a posteriori error estimates for nonlinear elliptic variational problems. Appl. Math. **54** (2009), 297–336.

[6] Liu, K. and Qin, X.: A gradient recovery-based a posteriori error estimators for the Ciarlet-Raviart formulation of the second biharmonic equations. Appl. Math. Sci. **1** (2007), 997–1007.

[7] Neittaanmäki, P. and Repin, S.: *Reliable methods for computer simulation: Error control and a posteriori estimates.* Elsevier, Amsterdam, 2004.

[8] Repin, S.: *A posteriori estimates for partial differential equations.* Walter de Gruyter, Berlin, 2008.

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

[10] Vejchodský, T.: Guaranteed and locally computable a posteriori error estimate. IMA J. Numer. Anal. **26** (2006), 525–540.

[11] Verfürth, R.: *A review of a posteriori error estimation and adaptive mesh refinement techniques.* John Wiley & Sons, Chichester, and B. G. Teubner, Stuttgart, 1996.

# NUMERICAL MODELING OF NEUTRON TRANSPORT – FINITE VOLUME METHOD, RESIDUAL DISTRIBUTION SCHEMES[*]

Martina Smitková, Marek Brandner

## 1 Introduction

Thanks to nuclear renaissance, numerical modeling of reactor physics has become an important field of study. This contribution deals with methods for numerical solving of the neutron transport equation. For its angular discretization we use the $P_N$ approximation, then we discuss two approaches to the spatial discretization – the Finite Volume Method and the Residual Distribution Schemes. Finally we present numerical results.

## 2 The neutron transport equation

Time-dependent transport of all neutral particles can be described by the one energy group Boltzmann transport equation [1]

$$\begin{aligned}\frac{1}{v}\frac{\partial}{\partial t}\psi(\mathbf{x},\mathbf{\Omega},t) + \mathbf{\Omega}\cdot\nabla\psi(\mathbf{x},\mathbf{\Omega},t) + \Sigma_t\psi(\mathbf{x},\mathbf{\Omega},t) = \\ = \frac{\Sigma_s}{4\pi}\int_{4\pi}\psi(\mathbf{x},\mathbf{\Omega}',t)\mathrm{d}\mathbf{\Omega}' + Q(\mathbf{x},\mathbf{\Omega},t),\end{aligned} \tag{1}$$

where $\psi(\mathbf{x},\mathbf{\Omega},t)$ is the unknown function angular flux, $\mathbf{x}$ is the position, $\mathbf{\Omega}$ is the particle direction, $t$ is time, $\Sigma_s$ is the isotropic scattering cross section, $\Sigma_t$ is the total cross section ($\Sigma_t = \Sigma_s + \Sigma_a$, where $\Sigma_a$ is the absorption cross section), $v$ is the neutron speed, which we set to $v = 1$ for convenience, and $Q$ is the independent or external source ($Q = 0$ in the sequel).

This formulation is basis for a time-dependent problem. We can either seek for a time-dependent solution, or in some cases the basic goal is to find a steady (stationary) solution and the time-dependent solution has just the role of an iterative process.

The sought function is a function of spatial variables, angular variables and time (function of 6 variables). We distinguish discretization of direction (for example: $P_N$, $S_N$ approximation), space and time.

[*] This work was supported by the specific research project of UWB and MSM 4977751301.

## 3 The $P_N$ approximation

This approximation is based on expanding the angular flux $\psi(\mathbf{\Omega})$ as a linear combination of the spherical harmonics as $\psi(\mathbf{x},\mathbf{\Omega},t)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\psi_l^m(\mathbf{x},t)Y_l^m(\mathbf{\Omega})$ [1]. This expansion is exact, but in order to make practical use of it, the series must be truncated. The $P_N$ approximation is based on the assumption that all $\psi_l^m=0$ for $l>N$. Then we solve a system of partial differential equations for the moments $\psi_l^m$.

The $P_N$ equations can be written in the matrix form, in two dimensions as

$$\mathbf{q}_t+\mathbf{A}_x\mathbf{q}_x+\mathbf{A}_y\mathbf{q}_y=\mathbf{S}\mathbf{q} \tag{2}$$

and in one dimension as

$$\mathbf{q}_t+\mathbf{A}\mathbf{q}_x=\mathbf{S}\mathbf{q}, \tag{3}$$

where $\mathbf{q}$ is vector of the unknown moments $\psi_l$ and $\psi_l^m$ respectively. Matrices$\mathbf{A},\mathbf{A}_x,\mathbf{A}_y$ (for their particular form see [1]) are diagonalizable, thus we are dealing with nonhomogeneous linear hyperbolic systems of partial differential equations.

The basic goal is to construct an efficient solver applicable to both stationary and time-dependent problems with arbitrary geometry.

## 4 Spatial discretization

Up-to-date numerical methods for solving hyperbolic partial differential equations are various types of the finite volume method (FVM) – such as upwind methods, central methods, based on many approximate Riemann solvers (Roe, HLL, HLLE...) and different reconstruction methods (TVD, ENO, WENO), several limiter functions etc. Next we have novel methods such as Residual Distribution Schemes (RDS), Streamline Upwind Petrov–Galerkin method (SUPG) or Discontinuous Galerkin Finite Element Method (DGFEM).

We will discuss the Finite Volume Method and the Residual Distribution Schemes.

### 4.1 Finite volume method

We focus on analysing one-dimensional problems. Multidimensional problems will be treated simply as multiple, independent, one-dimensional problems. But this approach can cause problems and it means an important drawback of this method.

We begin by dividing the $x$ axis into cells $\mathcal{C}_i=\langle x_{i-1/2},x_{i+1/2}\rangle$ (see Fig. 1) with uniform widths $\Delta x=x_{i+1/2}-x_{i-1/2}$ and edges at $x_{i+1/2}$. We introduce space-averaged data in cell $i$ at time $t$ as

$$\mathbf{q}_i(t)=\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}}\mathbf{q}(x,t)\mathrm{d}x. \tag{4}$$

The following equation is a consequence of the more general integral form of (3):

$$\frac{\partial\mathbf{q}_i}{\partial t}+\frac{\mathbf{F}_{i+1/2}-\mathbf{F}_{i-1/2}}{\Delta x}=\mathbf{S}\mathbf{q}_i. \tag{5}$$

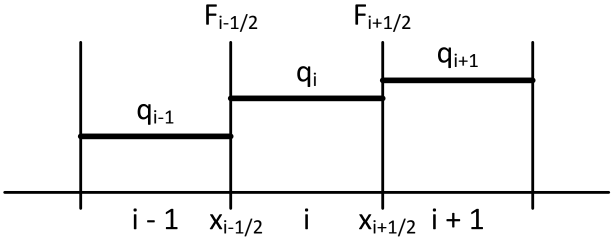


**Fig. 1:** *Notation of cells, cell averages and fluxes on interfaces in FVM.*

Here $\mathbf{F}_{i\pm1/2}$ denote the numerical fluxes at the cell boundaries, which we express using the Roe-type Riemann solver (the exact solution of the Riemann problem for the related linear homogeneous problem) as

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\mathbf{A}(\mathbf{q}_l + \mathbf{q}_r) - \frac{1}{2}|\mathbf{A}|(\mathbf{q}_r - \mathbf{q}_l), \tag{6}$$

where $|\mathbf{A}| = \sum_k \mathbf{r}_k|\lambda_k|\mathbf{l}_k$ ($\lambda_k$ are the eigenvalues of the matrix $\mathbf{A}$ from Eq. (3), $\mathbf{r}_k$ and $\mathbf{l}_k$ the right and left eigenvectors) and $\mathbf{q}_l$, $\mathbf{q}_r$ is obtained using the values of $\mathbf{Q}_i$ "to the left" and "to the right" of the cell $i$ ($\mathbf{Q}_i \approx \mathbf{q}_i$).

If we at the interface $i + 1/2$ naturally set $\mathbf{q}_l = \mathbf{Q}_i$ a $\mathbf{q}_r = \mathbf{Q}_{i+1}$, we get a method that is first order accurate in space. To get a higher-order method we reconstruct the approximate solution using linear interpolation within a cell to have a better estimate of the solution at the cell boundary. We seek to prevent the introduction of artificial oscillations into the solution, hence a nonlinear method must be used to calculate the slope within a cell to achieve better than first order accuracy. This is a statement of Godunov's Theorem. For example, the Van Leer's method and the minmod method can be used [3].

As we already mentioned, the multidimensional finite volume method is based on multiple one-dimensional problems, which brings some drawbacks, such as significant numerical diffusion, inability to tackle the real multidimensionality (no physical reasoning), wide stencil of the higher order schemes, rectangular mesh – disadvantage for problems with arbitrary geometry.

For the time integration we can use both the explicit and implicit Euler method, in this contribution just the explicit method was used. Another option are the TVD Runge–Kutta methods (suitable for computing the time-dependent solutions).

### 4.2 Residual distribution schemes

The residual distribution schemes have been developed on ideas borrowed from both the finite volume and finite element approaches and have become an attractive alternative to either one. The compact discretization stencil allows for the development of efficient implicit iterative solution strategies and for an easy parallelisation [2].

### 4.2.1 One–dimensional case

Consider scalar conservation law with source term $q_t + [f(q)]_x = s(q, x, t)$.

The solution is approximated by a continuous piecewise linear function $q(x,t) \approx \sum_i q_i(t) N_i(x)$, where $q_i(t)$ is the value of $q$ at node $i$, and $N_i$ the linear shape function equal to unity at $x_i$ and equal to zero outside the interval $\langle x_{i-1}, x_{i+1}\rangle$ (see Fig. 2).

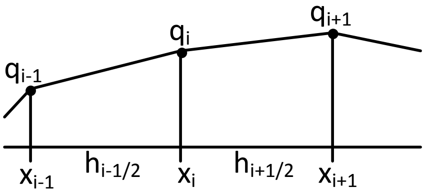


**Fig. 2:** *Data representation for RDS, using P1 elements.*

We define the cell residual as

$$\phi^{i+\frac{1}{2}} = \int_{x_i}^{x_{i+1}} (f_x - s)\mathrm{d}x = f_{i+1} - f_i - \frac{s_i + s_{i+1}}{2} h_{i+\frac{1}{2}}. \tag{7}$$

The nodal equation for node $i$ is then formed by distributing the cell residual to the two nodes of the cell. Gathering the contributions of the two elements at node $i$ we obtain for the steady state equation $\beta_i^{i-\frac{1}{2}}\phi^{i-\frac{1}{2}} + \beta_i^{i+\frac{1}{2}}\phi^{i+\frac{1}{2}} = 0$ where the distribution coefficients $\beta$ sum to one for a given cell (conservativity condition), $\beta_i^{i+\frac{1}{2}} + \beta_{i+1}^{i+\frac{1}{2}} = 1$. The coefficients $\beta$ can be specified so as to satisfy certain properties of monotonicity and accuracy in the solution, while maintaining the compact stencil. We formally define the distributed residuals as $\phi_i^{i-\frac{1}{2}} = \beta_i^{i-\frac{1}{2}}\phi^{i-\frac{1}{2}}$.

**Time discretization** For the higher order time-accurate time-dependent solution it is mandatory to use the consistent time discretization (for details see [2]). For the steady solution it is common to use the inconsistent time discretization (here using the explicit Euler method)

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{h_i}\left(\beta_i^{i-\frac{1}{2}}\phi^{i-\frac{1}{2}} + \beta_i^{i+\frac{1}{2}}\phi^{i+\frac{1}{2}}\right), \tag{8}$$

where $h_i = \frac{1}{2}(h_{i-\frac{1}{2}} + h_{i+\frac{1}{2}})$ is the volume of the median dual cell surrounding node $i$.

### 4.2.2 Two–dimensional case for systems

Consider the system of conservation laws $\mathbf{q}_t + \nabla\cdot\mathbf{F} = \mathbf{0}$ to be solved on an arbitrary triangulation of the domain. The solution is approximated by a continuous function, varying linearly over each triangle, $\mathbf{q}(x, y, t) \approx \sum_i \mathbf{q}_i(t) N_i(x, y)$. The residual in triangle $T$ is defined as

$$\Phi^T = -\iint_T \mathbf{q}_t \mathrm{d}x = \oint_{\partial T} \mathbf{F}\cdot \mathrm{d}\vec{n}_{ext}. \tag{9}$$

The Residual Distribution method consists of distributing fractions of this residual to the surrounding nodes. Starting from the inconsistent formulation and an Euler explicit time integration, we obtain the following update scheme

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n - \frac{\Delta t}{S_i} \sum_T \beta_i^T \Phi^T = \mathbf{q}_i^n - \frac{\Delta t}{S_i} \sum_T \Phi_i^T, \tag{10}$$

where $S_i$ is the area of the median dual cell around node $i$, i. e. 1/3 of the area of all triangles meeting at node $i$ (see Fig. 3). The residual $\Phi^T$ is now a vector, while the $\beta_i^T$ have become distribution matrices.

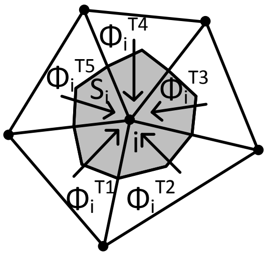


**Fig. 3:** *Node i and median dual cell $S_i$, with surrounding cells and updates $\Phi_i^T$.*

## 5 Results and conclusion

For clarity and brevity and regarding to the extent of this text, we used $q_t + 0.5q_x + 0.5q_y = -0.1q$ (a "special case" of (2)) as a simple test equation with initial condition a unit pulse in the center of the square domain. For the $P_N$ system the results would be analogous.

The test domain consists of 49 × 49 cells, $\Delta x = \Delta y = 0.32$ (FVM), for the RDS we used the Delaunay triangulation for the centres of the cells. The time interval is $T = 8$ and the time step $\Delta t = 0.25$.

From the Residual Distribution Schemes we chose the N (Narrow) scheme with $\phi_i^{T,N} = -\frac{k_i^+}{\sum_j k_j^+} \sum_j k_j^- (u_i^n - u_j^n)$ (monotone linear first order) and the LDA (Low Diffusion A) scheme with $\beta_i^{LDA} = \frac{k_i^+}{\sum_j k_j^+}$ (linear second order). The scalars $k_i$, termed the inflow parameters, defined as $k_i = \frac{1}{2}\vec{\lambda}\cdot\vec{n}_i$, allows to distinguish between inflow and outflow faces, and upstream and downstream nodes of the triangle. The vectors $\vec{n}_i$ are defined as the interior normals to the triangle, scaled by their respective lengths, $\vec{\lambda}$ is the vector of advection coefficients (see [2] for details).

At the figures 4 and 5 we can see results of the first-order Finite Volume Method and of the N scheme and LDA scheme. According to their theoretical properties, the LDA scheme gives most accurate results, however not keeping the solution positive. The positivity requirement is satisfied by the N scheme.

For the future work we want to focus on blending these two schemes together to gain better accuracy in smooth regions while maintaining positivity in transient areas. Another objective is to see the PDE as a whole, define a space-time residual and introduce the space-time Residual Distribution Schemes.

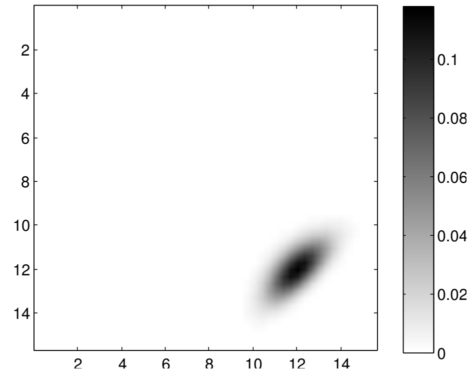


**Fig. 4:** *Result of the Finite Volume Method.*

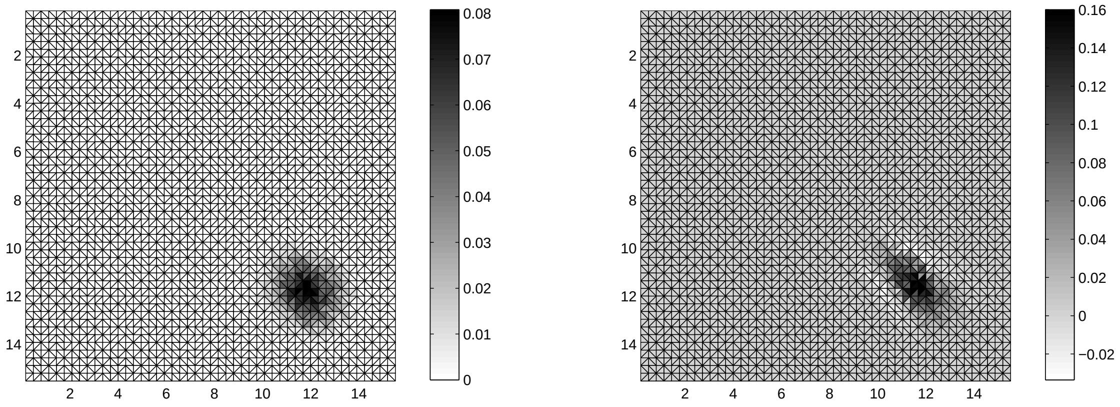


**Fig. 5:** *Result of the N (left) and LDA (right) scheme (RDS).*

## References

[1] Brunner, T.: *Riemann solvers for time-dependent transport based on the maximum entropy and spherical harmonics closures.* Ph.D. thesis, The University of Michigan, 2000.

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

[3] LeVeque, R.: *Finite volume methods for hyperbolic problems.* 1. edition, Cambridge University Press, Cambridge, 2002.

# ON A DYNAMICS IN RATIONAL POLYGONAL BILLIARDS[*]

Martin Soukenka

## 1 Introduction

A polygonal billiard table is a planar simply connected compact polygon $P$. The billiard flow $\{T_t\}_{t\in R}$, in $P$ is generated by the free motion of a mass-point subject to the elastic reflection in the boundary. This means that the point moves along a straight line in $P$ with a constant speed until it hits the boundary. At a smooth boundary point the billiard ball reflects according to the well known law of geometrical optics: the angle of incidence equals to the angle of reflection. If the billiard ball hits a corner, (a non-smooth boundary point), its further motion is not defined. Additionally to a corner, the billiard trajectory is not defined for a direction tangent to a side. By $D$ we denote the group generated by the reflections in the lines through the origin, parallel to the sides of the polygon $P$. It is known that the group $D$ is finite, when all the angles of $P$ are of the form $\pi m_i/n_i$ with distinct coprime integers $m_i, n_i$. In this case $D = D_N$ the dihedral group is generated by the reflections in lines through the origin that meet at angles $\pi/N$, where $N$ is the least common multiple of $n_i$'s and a trajectory changes its direction by $2N$ directions of the group $D_N$. In this case the polygon is called *rational.*

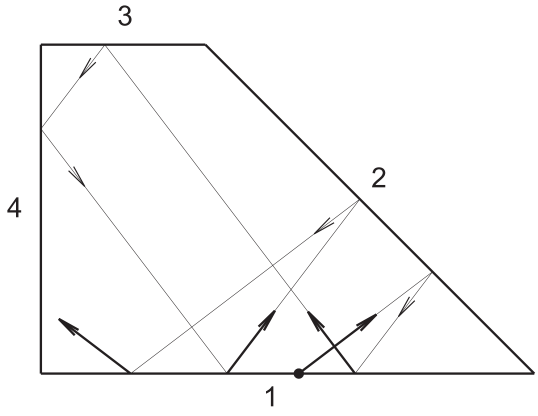


**Fig. 1:** *Elements of the group $D_4$ - directions of trajectory.*

[*]This work was supported by grant SGS 10-800020 of the Czech Technical University in Prague and research project of the Institute of Thermomechanics No AV0Z20760514.

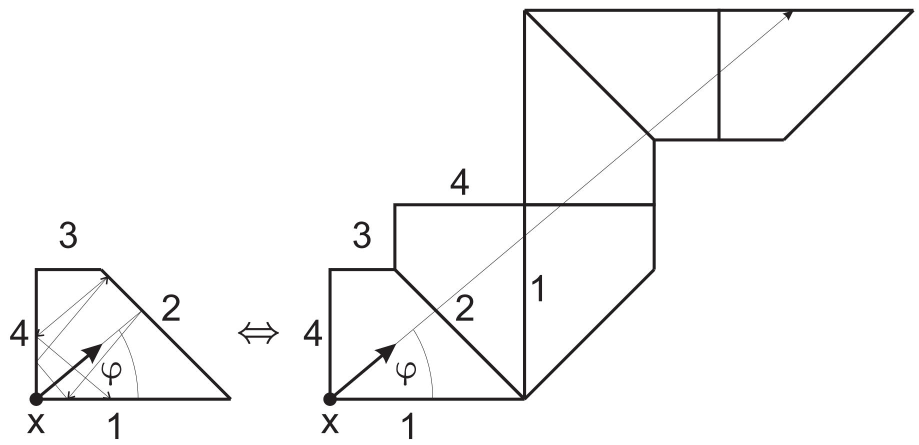


**Fig. 2:** *Unfolding of trajectory of* $(x, \varphi)$ *with symbolic itinerary* $1, 2, 1, 4, 2, \ldots$

**Example 1.** Consider a polygon from Figure 1 with angles (counterclockwise) $\pi/2$, $\pi/4$, $3\pi/4$, $\pi/2$, hence $N = 4$, so the group contains eight directions (see seven of them in Figure 1).

The aim of this text is to illustrate to non-experts some open problems in rational billiards and to point out that even in theoretical problems usage of numerical experiments can be at the beginning of the exact theoretical solution. We note that there are many open problems in rational billiards, most of which are studied by both nuclear and theoretical physicists and mathematicians [2], [1]. Let us first give some notations and basic facts.

For a simply connected polygon $P$ with $k$ sides consider counterclockwise orientation of its boundary $\partial P$ and denote by $[p_i, p_{i+1}]$ a closed arc with outgoing endpoint $p_i$ and incoming endpoint $p_{i+1}$ where $p_i$ is $i$-th corner of $P$. Let us denote by $1 := [p_1, p_2], 2 := [p_2, p_3], \ldots, k := [p_k, p_1]$ the sides of $P$ (Figure 1). Consider a pair $(x, \varphi)$, where $x \in \partial P$ (so called a *foot point*) and $\varphi$ is a direction. By a *symbolic itinerary* $a_0, a_1, a_2, \ldots$ of $(x, \varphi)$ we mean the sequence of visited numbers $a_i \in \{1, 2, \ldots, k\}$ of sides of $\partial P$ by trajectory $T_t x$. There is a simple way how to visualize a trajectory of a given $(x, \varphi)$, see Figure 2 - draw the straight line starting at $x \in \partial P$ under the angle $\varphi$ from the side $\partial P \ni x$ and reflect in successive steps the polygon by the sides crossing the line. This is called *unfolding* of a trajectory in $P$.

## 2 Dynamics in hammer polygons

In the rest of this text we consider very special shaped polygon $P$ from Figure 1 and Example 1 - a *hammer polygon* (shortly $h$-polygon). Let us now present two open problems in $h$-polygons. In what follows by $f^n$ of a map $f$ we mean $f(f^{n-1})$.

### 2.1 Is dynamics in $h$-polygons Li-York chaotic?

Let us start with the famous problem in polygonal billiards - an existence of so called *Li-York chaos.*

**Definition 1.** Let $X$ be a compact metric space with metric $\varrho$ and $f : X \to X$ continuous map. A pair $(x, y)$ of points $x, y \in X, x \neq y$ is called *Li-York pair*, if

$$\liminf_{n\to\infty} \varrho(f^n x, f^n y) = 0 \quad \text{and} \quad \limsup_{n\to\infty} \varrho(f^n x, f^n y) > 0.$$

If there is a Li-York pair, then the dynamical system $(X, f)$ is called *Li-York chaotic.*

It can be shown that an existence of Li-York pair in billiard dynamics can be formulated as follows: let $P$ be a fixed $h$-polygon and $u := (x, \varphi)$, $x \in \partial P$, resp. $v := (y, \psi)$, $y \in \partial P$, $u \neq v$ both generate infinite symbolic itineraries $a_0, a_1, a_2, \ldots$, resp. $b_0, b_1, b_2, \ldots$ in $P$. Then $(u, v)$ is Li-York pair, if lengths of blocks of the same numbers $a_i, b_i$ are (non-monotonically) increasing to infinity as $i \to \infty$, see Figure 3.

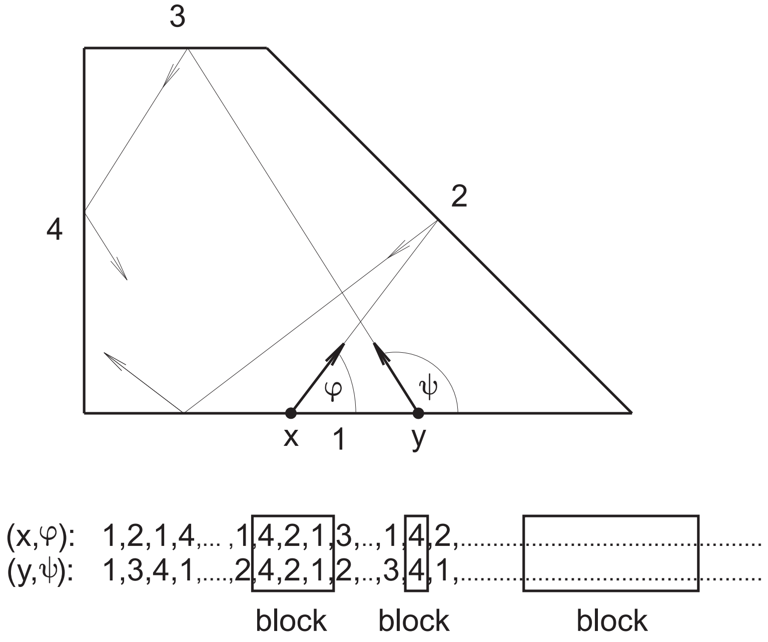

3
2
4
φ
ψ
x
1
y
(x,φ): 1,2,1,4,... ,1,4,2,1,3,..,1,4,2,...........................................
(y,ψ): 1,3,4,1,.....,2,4,2,1,2,..,3,4,1,...........................................
block
block
block

**Fig. 3:** *Blocks of the same numbers $a_i, b_i$ of lengths* $3, 1, \ldots$

**Problem 1.** *Is billiard dynamical system in h-polygon Li-York chaotic?*

For $h$-polygon $P$ of a fixed size we have been trying to find a Li-York pair in $P$. Numerical results of a number of the blocks of various lengths (denoting by $L_{\text{block}}$) bigger than 5 for some $(u, v)$ in $10^9$ iterations (thus in sequences $\{a_i\}, \{b_i\}$ for $i = 0, 1, \ldots, 10^9$) are listed below. Notice the extremely nonuniform frequencies

| $L_{\text{block}} > 5$ | frequency in $10^9$ iter |
|---|---|
| 186 | 1 |
| 360 | 1152 |
| 361 | 480 |
| 363 | 2040 |
| 364 | 1008 |
| 365 | 1056 |
| 366 | 1824 |
| 367 | 3704 |
| 368 | 12828 |
| 369 | 1966 |
| 370 | 4580 |
| 371 | 23954 |
| 373 | 43441 |
| 375 | 10066 |
| 376 | 14180 |
| 377 | 27073 |
| 378 | 20446 |
| 379 | 5636 |
| 380 | 2416 |
| 382 | 3580 |
| 383 | 12364 |
| 385 | 5798 |
| 386 | 136 |
| 388 | 1152 |
| 389 | 652 |
| 393 | 176 |
| 395 | 68 |

**Tab. 1:** *A number of the blocks of lengths bigger than 5 of the same numbers $a_i, b_i$ in sequences $\{a_i\}, \{b_i\}$ for $i = 0, 1, \ldots, 10^9$.*

of lengths: there is no block of length bigger than 5 and less than 360 except one exception - value 186 occurs due to "initial conditions" - it represents first 186 iterates at all. What are the next values after 395 as increasing a number of iterates nobody knows.

Theoretical explanation of these mysterious behavior is a subject of our further (theoretical) research. However, detailed analysis, which goes out of the purpose of this text, allow us to postulate the following conjecture.

**Conjecture 1.** *Billiard dynamical system in h-polygon is Li-York chaotic.*

## 2.2 What is the maximal length of the same itinerary in quasisimilar $h$-polygons?

As it is a frequent situation in physics and mathematics, one can ask how the outcome (for instance, a solution of ODE's or PDE's) changes if the income (the right hand side of a given ODE's or PDE's) changes by a some (small) perturbation. According to this we shall consider so called *quasisimilar* $h-$polygons and ask how the dynamics can change in these polygons. Let us first give some notations.

Let $P$ be $h$-polygon. For a point $x \in k \subset \partial P$ and direction $\varphi$ we say that a pair $(x, \varphi)$ is *minimal* for $P$ if the first 7 reflections of the trajectory of $(x, \varphi)$ generate 4 different directions of the group on side $k$ (for instance, the pair "$(\bullet, \nearrow)$" in Figure 1 is minimal while that of $(x, \varphi)$ in Figure 2 with $\varphi = \pi/2 - \varepsilon$ is not).

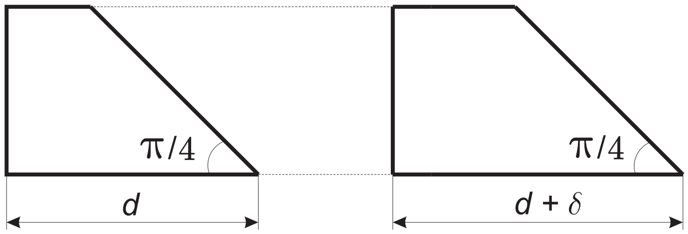


**Fig. 4:** *Quasisimilar h-polygons $P$, $P_\delta$ for perturbation $\delta > 0$.*

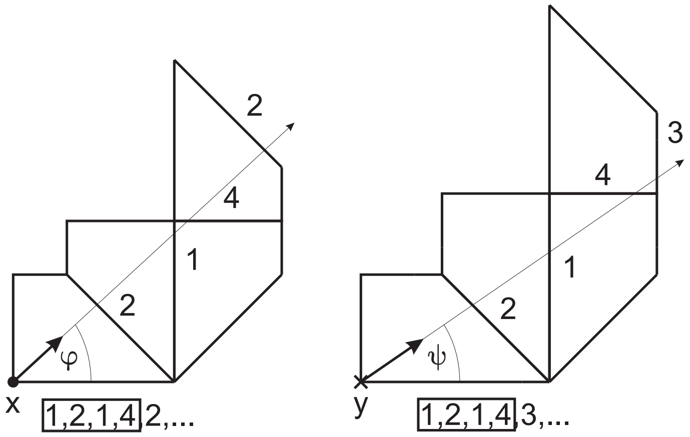


**Fig. 5:** *Unfoldings of trajectories of $u = (x, \varphi)$, $v = (y, \psi)$ with $L_u(\delta, v) = 4$.*

We consider *quasisimilar* to $P$ polygon as in Figure 4 for perturbation $\delta > 0$, that is a polygon $P_\delta$ with the same angles as $P$ including ordering but with different sizes of the sides.

For a symbolic itinerary $a_0, a_1, a_2, \ldots$ of $u := (x, \varphi)$ in $h$-polygon $P$ and a symbolic itinerary $b_0, b_1, b_2, \ldots$ of $v := (y, \psi)$ in $P_\delta$ for some $\delta > 0$ we denote by

$$L_u(\delta, v) := \max_i \{i + 1\,; a_j - b_j = 0, j = 0, 1, \ldots i\}$$

**Example 2.** Consider pair $u := (x, \varphi)$ in $P$, resp. $v := (y, \psi)$ in $P_\delta$ as in Figure 5. The itineraries are

$$\begin{aligned} \{a_n\}_{n=0}^{\infty} : &\quad 1, 2, 1, 4, 2, \ldots \\ \{b_n\}_{n=0}^{\infty} : &\quad 1, 2, 1, 4, 3, \ldots \end{aligned}$$

Then $L_u(\delta, v) = \max\{1, 2, 3, 4\} = 4$.

**Problem 2.** *Consider a point $x \in \partial P$ as a corner of $P$ and take $\varphi$ such that $u = (x, \varphi)$ is fixed minimal pair in $P$ generating an infinite symbolic itinerary*

*$SI : a_0, a_1, a_2, \ldots$ in $P$. Let $\delta > 0$. Consider $y \in \partial P_\delta$ as a corresponding fixed corner to that of $x$ (that is, corners $x$, $y$ have the same counterclockwise number, see Figure 5) and a symbolic itinerary of $(y, \psi)$ in $P_\delta$. What is the value of $L_{\max}(\delta) := \max_\psi L_u(\delta, \psi)$?*

The answer to Problem 2 is far from trivial. However, detailed theoretical (geometrical) analysis allows us to count values $L_{\max}(\delta) = \max_\psi L_u(\delta, \psi)$ numerically. Numerical experiments show, that there are big jumps of $L_{\max}(\delta)$ for a fixed $u$ in $P$ when varying a value of a perturbation $\delta$. A typical illustration of this jump for some fixed size of $P$ and $u$, depending on $\delta$, is listed bellow (compare $\delta = 0.50$ and $\delta = 0.51$). Theoretical explanation of these empirical results remains open.

| Perturbation $\delta$ | $L_{\max}(\delta)$ | Perturbation $\delta$ | $L_{\max}(\delta)$ |
|---|---|---|---|
| 0.47 | 1260 | 0.52 | 130 |
| 0.48 | 1254 | 0.53 | 85 |
| 0.49 | 1248 | 0.54 | 65 |
| 0.50 | 1242 | 0.55 | 55 |
| 0.51 | 255 | 0.56 | 45 |

**Tab. 2:** *Jumps of values of $L_{\max}(\delta)$ from Problem 2 when varying $\delta > 0$.*

## 3 Remark on numerical computation

As far as computing, all the codes we have used are written in Fortran 90 using the library for multiple precision computing FMLIB which allows one to set a number of significant digits. We have set 60 resp. 70 for controlling computation.

## References

[1] Bobok, J. and Troubetzkoy, S.: Does a billiard orbit determine its (polygonal) table? Accepted to Fundamenta Mathematicae, 2010, 17 pp.

[2] Masur, H. and Tabachnikov, S.: Rational billiards and flat surfaces. In: B. Hasselblatt and A. Katok (Eds.), *Handbook of Dynamical Systems*, vol. 1A, Elsevier Science B.V., 2002.

# NUMERICAL APPROXIMATION OF FLOW IN A SYMMETRIC CHANNEL WITH VIBRATING WALLS[*]

Petr Sváček, Jaromír Horáček

**Abstract**

In this paper the numerical solution of two dimensional fluid-structure interaction problem is addressed. The fluid motion is modelled by the incompressible unsteady Navier-Stokes equations. The spatial discretization by stabilized finite element method is used. The motion of the computational domain is treated with the aid of Arbitrary Lagrangian Eulerian (ALE) method. The time-space problem is solved with the aid of multigrid method.

The method is applied onto a problem of interaction of channel flow with moving walls, which models the air flow in the glottal region of the human vocal tract. The pressure boundary conditions and the effects of the isotropic and anisotropic mesh refinement are discussed. The numerical results are presented.

## 1 Introduction

This paper is concerned with numerical simulation of unsteady viscous incompressible flow in a simplified model of the glottal region of the human vocal tract with the aid of the finite element method (FEM). The main attention is paid to the efficient computation of the flow field. For the robust and efficient solver both the advanced stabilization (as streamline upwind/Petrov Galerkin stabilizations, cf. [6], [7]) and solution methods (as multigrid and/or domain decomposition, cf. [19], [9], [10], [13]) have to be employed.

FEM is well known as a general discretization method for partial differential equations. It can handle easily complex geometries and also boundary conditions employing derivatives. However, straightforward application of FEM procedures often fails in the case of incompressible Navier-Stokes equations. The reason is that momentum equations are of advection-diffusion type with dominating advection. The Galerkin FEM leads to unphysical solutions if the grid is not fine enough in regions of strong gradients (e.g. boundary layer). In order to obtain physically admissible correct solutions it is necessary to apply suitable mesh refinement (e.g. anisotropically refine mesh, cf. [5]) combined with a stabilization technique, cf. [7], [3], [18], [16].

Furthermore, the time and space discretized linearized problem of the arising large system of linear equations needs to be solved in fast and efficient manner. The

[*] This research was supported under the Project OC 09019 “Modelling of voice production based on biomechanics” within the program COST of the Ministry of Education of the Czech Republic, under grant No. 201/08/0012 of the Grant Agency of the Czech Republic and the Research Plan MSM 6840770003 of the Ministry of Education of the Czech Republic.

application of direct solvers as UMFPACK (cf. [4]) leads to robust method, where different stabilizations procedures can be easily applied even on anisotropically refined grids. However, the application of direct solver for system of equations with more than approximately $10^5$ unknowns becomes unfeasible in many cases (depending on computer CPU and memory).

In that case the application of multigrid (cf. [19]) or domain decomposition methods is an option, cf. [13]. In this paper a simplified version of multigrid method is shortly described together with a choice of finite elements and stabilization procedures. Even when the method is simplified, it was found to be efficient and robust enough.

The developed method is applied to the numerical solution of a channel flow modelling the glottis region of the human vocal tract including the vibrating vocal folds. The vibrations of the channel wall are prescribed, see [14]. Further, in order to obtain physically relevant results the pressure drop boundary conditions are employed, cf. [8].

First the mathematical model consisting of time dependent computational domain and incompressible flow model. Further, in Section 3 the time and space discretization is described and Section 4 describes the application of a simple multigrid version. Section 5 shows the numerical results.

## 2 Mathematical model

The model problem consists of flow model, which describes the fluid motion in the time-dependent computational domain $\Omega_t$, i.e. in a channel with moving walls, see Fig. 1. For the description and the approximation on moving meshes the Arbitrary Lagrangian-Eulerian (ALE) method is employed, cf. [12]. The geometry of the channel is chosen according [14], where a different distance between the moving walls, i.e. the gap $g(t)$, was considered. Further, on the outlet part of the channel a modification of do-nothing boundary condition was applied in order to allow the vortices flow smoothly out of the computational domain. On the inlet either the Dirichlet boundary condition for velocity is prescribed or preferably we use the pressure drop formulation, similarly as in cf. [8]. The presented mathematical model (and also its numerical approximation) is a slight modification of the mathematical model applied to the numerical simulation of flow induced airfoil vibrations in our previous works, cf. [18].

### 2.1 Arbitrary Lagrangian Eulerian method

In order to treat the fluid flow on moving domains, the so-called Arbitrary Lagrangian Eulerian method is used. We assume that $\mathcal{A} = \mathcal{A}(\xi, t) = \mathcal{A}_t(\xi)$ is an ALE mapping defined for all $t \in (0, T)$ and $\xi \in \Omega_0$, which is smooth enough and continuously differentiable mapping of $\Omega_0$ onto $\Omega_t$. We define the *domain velocity* $\mathbf{w}_D : \mathcal{M} \to R$ satisfies

$$\mathbf{w}_D(\mathcal{A}(\xi, t), t) = \frac{\partial \mathcal{A}}{\partial t}(\xi, t) \qquad \text{for all } \xi \in \Omega_0 \text{ and } t \in (0, T). \tag{1}$$

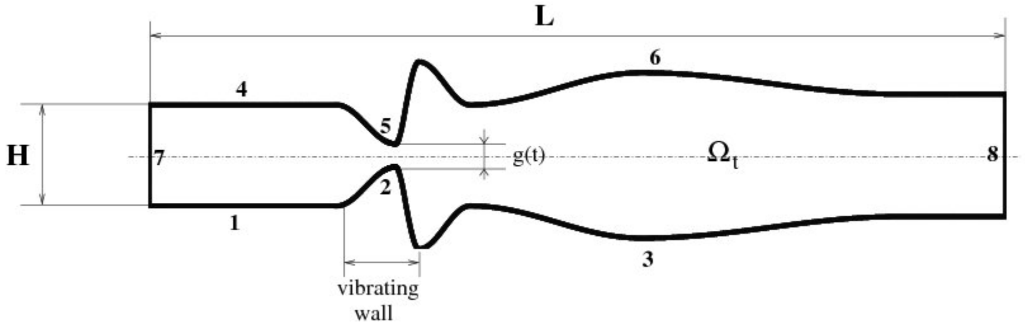


**Fig. 1:** *Computational domain and boundary parts: The inlet part of the boundary $\Gamma_I$ (number 7), the outlet part of the boundary $\Gamma_O$ (number 8), the fixed walls $\Gamma_D$ (numbers 1,4,3,6) and vibrating walls $\Gamma_{Wt}$ (numbers 2, 5).*

Furthermore the symbol $D^{\mathcal{A}}/Dt$ denotes the ALE derivative, i.e. the time derivative with respect to the reference configuration. The ALE derivative satisfies (cf. [18], [11])

$$\frac{D^{\mathcal{A}} f}{Dt}(x,t) = \frac{\partial f}{\partial t}(x,t) + \mathbf{w}_D(x,t) \cdot \nabla f(x,t). \tag{2}$$

In the present paper the ALE mapping can be analytically prescribed, but in the future this mapping will be a part of solution similary as in cf. [18].

### 2.2 Flow model

Let us consider the following system of the incompressible Navier-Stokes equations in a bounded time-dependent domain $\Omega_t \subset R^2$ written in ALE form

$$\begin{aligned} \frac{D^{\mathcal{A}}\mathbf{v}}{Dt} - \nu \triangle \mathbf{v} + ((\mathbf{v} - \mathbf{w}_D) \cdot \nabla)\mathbf{v} + \nabla p = 0, \qquad & \text{in } \Omega_t, \\ \nabla \cdot \mathbf{v} = 0, \qquad & \text{in } \Omega_t, \end{aligned} \tag{3}$$

where $\mathbf{v} = \mathbf{v}(x,t)$ is the flow velocity, $p = p(x,t)$ is the kinematic pressure (i.e. pressure divided by the constant fluid density $\rho_\infty$) and $\nu$ is the kinematic viscosity.

The boundary of the computational domain $\partial\Omega_t$ consists of mutually disjoint parts $\Gamma_D$ (wall), $\Gamma_I$ (inlet), $\Gamma_O$ (outlet) and the moving part $\Gamma_{Wt}$ (oscillating wall). The following boundary conditions are prescribed

$$\begin{array}{lll} \text{a)} & \mathbf{v}(x,t) = \mathbf{0} & \text{for } x \in \Gamma_D, \\ \text{b)} & \mathbf{v}(x,t) = \mathbf{w}_D(x,t) & \text{for } x \in \Gamma_{Wt}, \\ \text{c)} & -(p - p^o_{ref})\mathbf{n} + \frac{1}{2}(\mathbf{v}\cdot\mathbf{n})^- \mathbf{v} + \nu \frac{\partial \mathbf{v}}{\partial \mathbf{n}} = 0, & \text{on } \Gamma_O, \\ \text{d)} & -(p - p^i_{ref})\mathbf{n} + \frac{1}{2}(\mathbf{v}\cdot\mathbf{n})^- \mathbf{v} + \nu \frac{\partial \mathbf{v}}{\partial \mathbf{n}} = 0, & \text{on } \Gamma_I, \end{array} \tag{4}$$

where $\mathbf{n}$ denotes the unit outward normal vector, the constants $p^i_{ref}, p^o_{ref}$ denotes the reference pressure values, and $\alpha^-$ denotes the negative part of a real number $\alpha$. In computations the condition (4d) can be replaced by the condition

$$\text{e)}\quad \mathbf{v}(x,t) = \mathbf{v}_D \qquad \text{for } x \in \Gamma_I. \tag{5}$$

Finally, we prescribe the initial condition

$$\mathbf{v}(x,0) = \mathbf{v}^0(x) \qquad \text{for } x \in \Omega_0.$$

## 3 Numerical approximation

In this section the numerical approximation of the mathematical model given in Section 2 is shown. As already mentioned the presented numerical approximation is a slight modification of our previous works, cf. [18], [17]. Nevertheless there are several significant differences, which were found to be important for the numerical approximation: boundary conditions used on the inlet/outlet part of the computational and its weak formulation, a modified Galerkin/Least-Squares (GLS) scheme employed for stable pair of finite elements, and the choice of stabilizing parameters. The space discretization and its stabilization is briefly desribed for the sake of clarity and completeness.

### 3.1 Time discretization

We consider a partition $0 = t_0 < t_1 < \cdots < T,\ t_k = k\Delta t$, with a time step $\Delta t > 0$, of the time interval $(0,T)$ and approximate the solution $\mathbf{v}(\cdot,t_n)$ and $p(\cdot,t_n)$ (defined in $\Omega_{t_n}$) at time $t_n$ by $\mathbf{v}^n$ and $p^n$, respectively. For the time discretization we employ a second-order two-step scheme using the computed approximate solution $\mathbf{v}^{n-1}$ in $\Omega_{t_{n-1}}$ and $\mathbf{v}^n$ in $\Omega_{t_n}$ for the calculation of $\mathbf{v}^{n+1}$ in the domain $\Omega_{t_{n+1}} = \Omega_{n+1}$. We write

$$\frac{\partial \mathbf{v}}{\partial t}(x,t^{n+1}) \approx \frac{3\mathbf{v}^{n+1} - 4\hat{\mathbf{v}}^n + \hat{\mathbf{v}}^{n-1}}{2\Delta t} \qquad \text{where } x \in \Omega_{n+1}, \tag{6}$$

where $\hat{\mathbf{v}}^n$ and $\hat{\mathbf{v}}^{n-1}$ are the approximate solutions $\mathbf{v}^n$ and $\mathbf{v}^{n-1}$ defined on $\Omega_n$ and $\Omega_{n-1}$, respectively, and transformed onto $\Omega_{n+1}$ with the aid of ALE mapping, i.e. $\hat{\mathbf{v}}^i(x) = \mathbf{v}^i(\mathcal{A}_{t_i}(\xi))$ where $x = \mathcal{A}_{t_{n+1}}(\xi) \in \Omega_{n+1}$. Further, we approximate the domain velocity $\mathbf{w}_D(x,t_{n+1})$ by $\mathbf{w}_D^{n+1}$, where

$$\mathbf{w}_D^{n+1}(x) = \frac{3\mathcal{A}_{t_{n+1}}(\xi) - 4\mathcal{A}_{t_n}(\xi) + \mathcal{A}_{t_{n-1}}(\xi)}{2\Delta t}, \qquad x = \mathcal{A}_{t_{n+1}}(\xi),\ x \in \Omega_{n+1}.$$

Then the time discretization leads to the following problem in domain $\Omega_{n+1}$

$$\frac{3\mathbf{v}^{n+1} - 4\hat{\mathbf{v}}^n + \hat{\mathbf{v}}^{n-1}}{2\Delta t} - \nu\triangle\mathbf{v}^{n+1} + \left((\mathbf{v}^{n+1} - \mathbf{w}_D^{n+1})\cdot\nabla\right)\mathbf{v}^{n+1} + \nabla p^{n+1} = 0, \tag{7}$$
$$\nabla\cdot\mathbf{v}^{n+1} = 0,$$

equipped with boundary conditions (4a-d) and the initial condition.

### 3.2 Weak formulation

For solution of the problem by finite element method, the time-discretized problem (7) is reformulated in a weak sense. The following notation is used: By $W = \mathbf{H}^1(\Omega_{n+1})$ the velocity space is defined, by $X$ the space of test functions is denoted

$$X = \{\varphi \in W : \varphi = 0 \text{ on } \Gamma_{Wt_{n+1}} \cap \Gamma_D\},$$

and by $Q = L^2(\Omega_{n+1})$ the pressure space is denoted. Using the standard approach, cf. [18], the solution $\mathbf{v} = \mathbf{v}^{n+1}$ and $p = p^{n+1}$ of problem (7) satisfies

$$a(U, V) = f(V), \qquad U = (\mathbf{v}, p) \tag{8}$$

for any $V = (\mathbf{z}, q) \in X \times Q$, where

$$\begin{aligned}
a(U,V) &= \left(\frac{3}{2\Delta t}\mathbf{v}, \mathbf{z}\right) + \nu\left(\nabla\mathbf{v}, \nabla\mathbf{z}\right) + \mathcal{B}(\mathbf{v}, \mathbf{z}) + c_n(\mathbf{v}; \mathbf{v}, \mathbf{z}) - (p, \nabla\cdot\mathbf{z}) + (\nabla\cdot\mathbf{v}, q)\,,\\
c_n(\mathbf{w}, \mathbf{v}, \mathbf{z}) &= \int_{\Omega_{n+1}} \left(\frac{1}{2}(\mathbf{w}\cdot\nabla\mathbf{v})\cdot\mathbf{z} - \frac{1}{2}(\mathbf{w}\cdot\nabla\mathbf{z})\cdot\mathbf{v}\right) dx - \Big((\mathbf{w}_D^{n+1}\cdot\nabla)\mathbf{v}, \mathbf{z}\Big),\\
\mathcal{B}(\mathbf{v}, \mathbf{z}) &= \int\limits_{\Gamma_I\cup\Gamma_O} \frac{1}{2}(\mathbf{v}\cdot\mathbf{n})^+\mathbf{v}\cdot\mathbf{z}\, dS, \qquad (9)\\
f(V) &= \frac{1}{2\Delta t}\left(4\hat{\mathbf{v}}^n - \hat{\mathbf{v}}^{n-1}, \mathbf{z}\right) - \int_{\Gamma_I} p^i_{ref}\mathbf{v}\cdot\mathbf{n} dS - \int_{\Gamma_O} p^o_{ref}\mathbf{v}\cdot\mathbf{n} dS,
\end{aligned}$$

and by $(\cdot,\cdot)$ we denote the scalar product in the space $L^2(\Omega_{n+1})$.

### 3.3 Spatial discretization

Further, the weak formulation (8) is approximated by the use of FEM: we restrict the couple of spaces $(X, M)$ to finite element spaces $(X_h, M_h)$. First, the computational domain $\Omega_t$ is assumed to be polygonal and approximated by an admissible triangulation $\mathcal{T}_h$, cf. [2]. Based on the triangulation $\mathcal{T}_h$ the Taylor-Hood finite elements are used, i.e.

$$\begin{aligned}
\mathcal{H}_h &= \{v \in C(\overline{\Omega_{n+1}}); v|_K \in P_2(K) \text{ for each } K \in \mathcal{T}_h\},\\
\mathcal{W}_h &= [\mathcal{H}_h]^2\,, \qquad\qquad X_h = \mathcal{W}_h \cap \mathcal{X}, \qquad (10)\\
\mathcal{M}_h &= \{v \in C(\overline{\Omega_{n+1}}); v|_K \in P_1(K) \text{ for each } K \in \mathcal{T}_h\}.
\end{aligned}$$

The couple $(X_h, M_h)$ satisfy the Babuška-Brezzi inf-sup condition, which guarantees the stability of a scheme, cf. [20].

**Problem 1** (Galerkin approximations)**.** *Find $U_h = (\mathbf{v}_h, p_h) \in (X_h, M_h)$ such that $\mathbf{v}_h$ satisfy boundary conditions (4a,b) and*

$$a(U_h, V_h) = f(V_h), \tag{11}$$

*for all $\mathbf{z}_h \in X_h$ and $q_h \in M_h$.*

The Galerkin approximations are unstable in the case of high Reynolds numbers, when the convection dominates. In that case a stabilized method needs to be applied.

### 3.4 Stabilization

In order to overcome the above mentioned instability of the scheme, modified Galerkin Least Squares method is applied, cf. ([7]). We start with the definition of the local element rezidual terms $\mathcal{R}^a_K$ and $\mathcal{R}^f_K$ defined on the element $K \in \mathcal{T}_h$ by

$$\mathcal{R}^a_K(\tilde{\mathbf{w}};\mathbf{v},p) = \frac{3\mathbf{v}}{2\Delta t} - \nu\triangle\mathbf{v} + (\tilde{\mathbf{w}}\cdot\nabla)\,\mathbf{v} + \nabla p, \qquad \mathcal{R}^f_K(\hat{\mathbf{v}}_n,\hat{\mathbf{v}}_{n-1}) = \frac{4\hat{\mathbf{v}}_n - \hat{\mathbf{v}}_{n-1}}{2\Delta t}. \quad (12)$$

Further, the stabilizing terms are defined for $U^* = (\mathbf{v}^*,p^*)$, $U = (\mathbf{v},p)$, $V = (\mathbf{z},q)$ by

$$\begin{aligned} \mathcal{L}_{GLS}(U^*;U,V) &= \sum_{K\in T_h} \delta_K \Big(\mathcal{R}^a_K(\tilde{\mathbf{w}};\mathbf{v},p), (\tilde{\mathbf{w}}\cdot\nabla)\,\mathbf{z} + \nabla q\Big)_K, \\ \mathcal{F}_{GLS}(V_h) &= \sum_{K\in T_h} \delta_K \Big(\mathcal{R}^f_K(\hat{\mathbf{v}}_n,\hat{\mathbf{v}}_{n-1}), (\tilde{\mathbf{w}}\cdot\nabla)\,\mathbf{z} + \nabla q\Big)_K, \end{aligned} \quad (13)$$

where the function $\widetilde{\mathbf{w}}$ stands for the transport velocity, i.e. $\widetilde{\mathbf{w}} = \mathbf{v}^* - \mathbf{w}_D^{n+1}$. The additional grad-div stabilization terms read

$$\mathcal{P}_h(U,V) = \sum_{K\in\mathcal{T}_h} \tau_K(\nabla\cdot\mathbf{v},\nabla\cdot\mathbf{z})_K.$$

In the case of bounded convection velocity the choice of parameters according [7] for BB stable pair of FE (reduced scheme) would be possible. However, in order to obtain a fast and efficient multigrid method, the following choice of the parameters $\delta_K$ and $\tau_K$ is used

$$\tau_K = \nu\left(1 + Re^{loc} + \frac{h_K^2}{\nu\,\Delta t}\right), \qquad \delta_K = \frac{h_K^2}{\tau_K},$$

where the local Reynolds number $Re^{loc}$ is defined as $Re^{loc} = \frac{h\|\mathbf{v}\|_K}{2\nu}$.

**Problem 2** (Galerkin Least Squares stabilized approximations)**.** *We define the discrete problem to find an approximate solution $U_h = (\mathbf{v}_h,p_h) \in \mathcal{W}_h \times \mathcal{Q}_h$ such that $\mathbf{v}_h$ satisfies approximately conditions (4a,b) and the identity*

$$a(U_h,V_h) + \mathcal{L}_{GLS}(U_h;U_h,V_h) + \mathcal{P}_h(U_h,V_h) = f(V_h) + \mathcal{F}_{GLS}(V_h), \quad (14)$$

*for all $V_h = (\mathbf{z}_h,q_h) \in \mathcal{X}_h \times \mathcal{Q}_h$.*

### 4 Multigrid solution of the linear system

The space-time discretized system (14) needs to be solved by some linearization scheme, e.g. by Oseen linearization procedure described e.g. in [18] or [19]. The solution of the linearized system (14) leads to the the solution of a modified saddle point system

$$S\underline{\mathbf{v}} + B\underline{p} = \underline{f}, \qquad \tilde{B}^T\underline{\mathbf{v}} + \tilde{A}\underline{p} = 0, \quad (15)$$

where $\underline{\mathbf{v}}$ and $\underline{p}$ is the finite-dimensional representation of the finite element approximations of velocity and pressure, respectively. Let us mention that for the non-stabilized system (i.e. in the case of $\delta_K \equiv \tau_K \equiv 0$) we have $\tilde{A} = 0$ and $\tilde{B} = B$.

From the system of equations (15) the pressure degrees of freedoms can be formally eliminated by formally multiplying the first equation of (15) by $\tilde{B}^T S^{-1}$ from the left, i.e. we get the system of equations

$$\left(\tilde{B}^T S^{-1} B - \tilde{A}\right) \underline{p} = \tilde{B}^T S^{-1} \underline{f}, \tag{16}$$

or with notation $A_p = \tilde{B}^T S^{-1} B - \tilde{A}$ and $g = \tilde{B}^T S^{-1} f$ we have

$$A_p \underline{p} = g,$$

which can be solved by the Richardson iterative method

$$\underline{p}^{(l+1)} = \underline{p}^{(l)} + C^{-1}(g - A_p \underline{p}^{(l)}), \tag{17}$$

where $C$ is a suitable preconditioner, see e.g. [19]. Nevertheless the choice of the preconditioner $C$ is complicated in the case of convection dominated flows and the convergence of the scheme (17) is in this case slow. Moreover the stabilizing terms also badly influences the convergence rates.

In many cases and for small number of unknowns, the system can be solved with the aid of a direct solver, which yields fast, efficient and robust scheme. We refer to direct solver UMFPACK, cf. [4], which in the cases studied by the authors up to now [18] was efficient for number of unknowns less then approximately $10^5$. However, with further increase of the number of unknowns the memory and CPU requirements grows too fast, so that the fast and efficient solution becomes impossible. One possibility is to use the parallel implementation of multi-frontal method, cf. [1].

Here, the solution of the system (15) is carried out by a simplified version of multi-grid method. Only single mesh and two levels of solution (coarse and fine grid levels) are used. The fine grid is represented by the used higher order finite elements (here Taylor-Hood finite elements, i.e. P2/P1 approximations for velocity/pressure). The coarse grid is considered as lower order finite elements (i.e. equal order P1/P1 approximations for velocity/pressure) The solution on the coarse grid can be obtained with the aid of direct solver UMFPACK, which was found to be fast enough in the studied cases. On the fine grid the multiplicative Vanka-type smoother is used, cf. [9], [10]. This approach (i.e. the direct solver on coarse grid and Vanka-type smoother on fine grid) resulted in an efficient and fast method, which can be easily implemented. The performance of the multigrid method was found to be excellent for the isotropic grids. In the case of anisotropic mesh refinement, the convergence rates nevertheless become worse. The proper solution in this case is subject of a further study.

## 5 Numerical results

In this section the numerical results for air flow in a symmetric two-dimensional channel are presented. The channel geometry described in [14] is employed here, see also Fig. 1.

### 5.1 Stationary solution

First, we consider the non-moving computational domain $\Omega$, where the influence of isotropically and anisotropically refined meshes is studied, see Fig. 2.

The following constants were used in the computations: fluid density $\rho_\infty = 1.225$ kg m$^{-3}$ and kinematic viscosity $\nu = 1.5 \times 10^{-5}$ m$^2$/s, the width of the inlet part of the channel is $H = 0.0176$ m, the total length of the channel $L = 0.16$ m, and the constant gap width $g \equiv 4.4$ mm.

The boundary condition (4d) in the presented computations is replaced by the condition (5), where the constant flow velocity is prescribed $\mathbf{v}_D(x,t) = (U_\infty, 0)^T$ at the inlet part of boundary $\Gamma_I$, and $U_\infty$ was chosen in the range $[0.01, 0.05]\mathrm{m\,s^{-1}}$. The numerical results for stationary solution and different Reynolds numbers ($Re = \frac{1}{8}LU_\infty/\nu$) are presented in Figs. 3-4, where the isolines of the magnitude of velocity are shown. The results computed on both meshes for same Reynolds numbers show that even for low Reynolds numbers several stationary symmetric and nonsymmetric solutions exist. Fig. 3 (left) shows the symmetric solution obtained on both meshes for $Re = 20$. For $Re = 40$ and $Re = 50$ in Figs. 3-4 on isotropic mesh the non-symmetric solution was obtained , whereas on the anisotropical symmetric mesh the solution remains symmetric. For higher Reynolds number $Re > 50$ both solutions become non-symmetric.

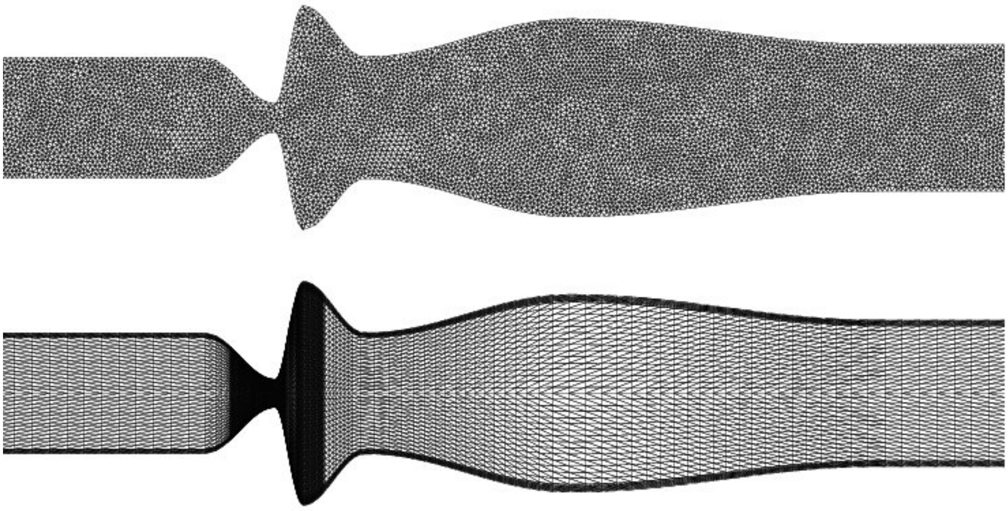


**Fig. 2:** *The employed grids: the isotropic non-symmetric mesh (upper part) with 12219 vertices and 23709 elements and approximately* $8 \times 10^4$ *unknowns for flow problem, and the anisotropic axisymmetric mesh (lower part) with 8241 vertices and 16000 elements (resulting in* $6 \times 10^4$ *unknowns).*

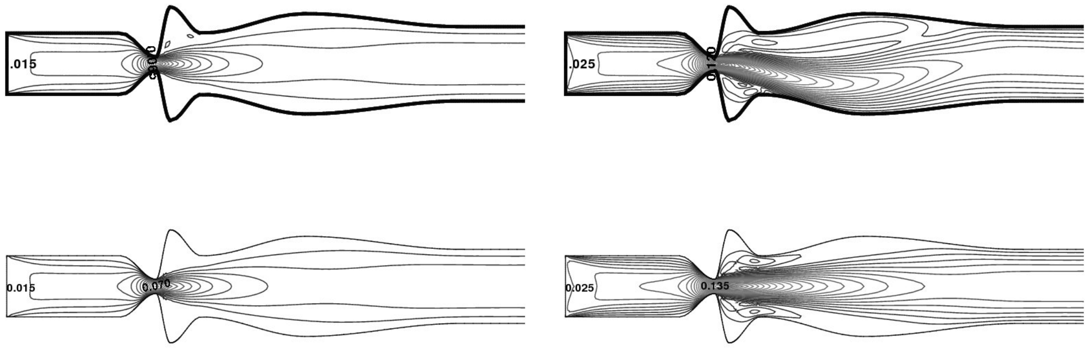


**Fig. 3:** *The isolines of flow velocity magnitudes for Reynolds number 20 (left) and 40 (right) on isotropic mesh(upper part) and anisotropic mesh (lower part).*

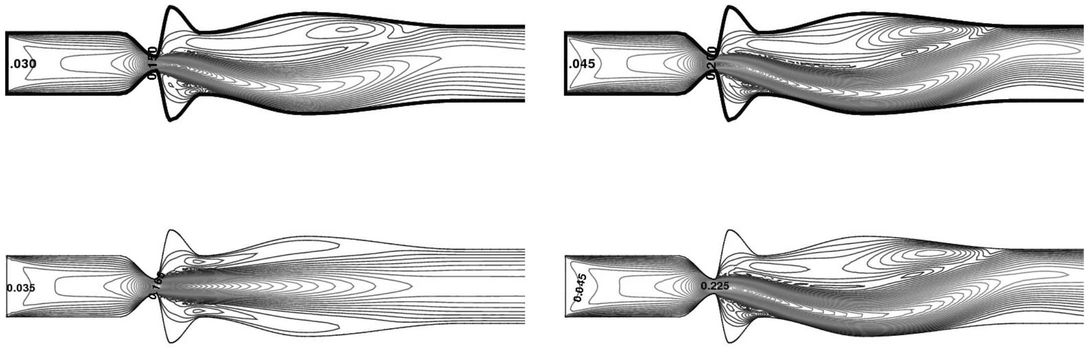


**Fig. 4:** *The isolines of flow velocity magnitudes for Reynolds number 50 (left) and 70 (right) on isotropic mesh(upper part) and anisotropic mesh (lower part).*

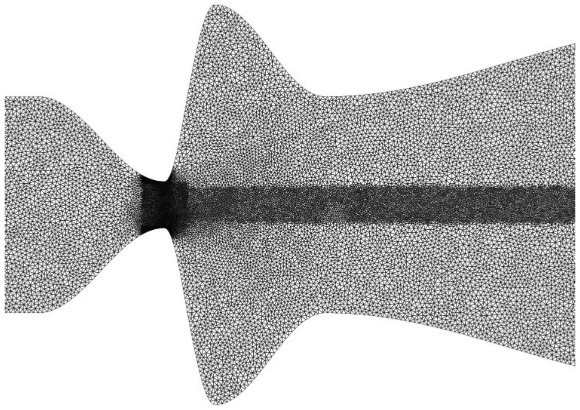


**Fig. 5:** *A detail of isotropic mesh used for multigrid solution with 42576 vertices and 84078 elements yielding approximately* $4 \times 10^5$ *unknowns for the flow problem.*

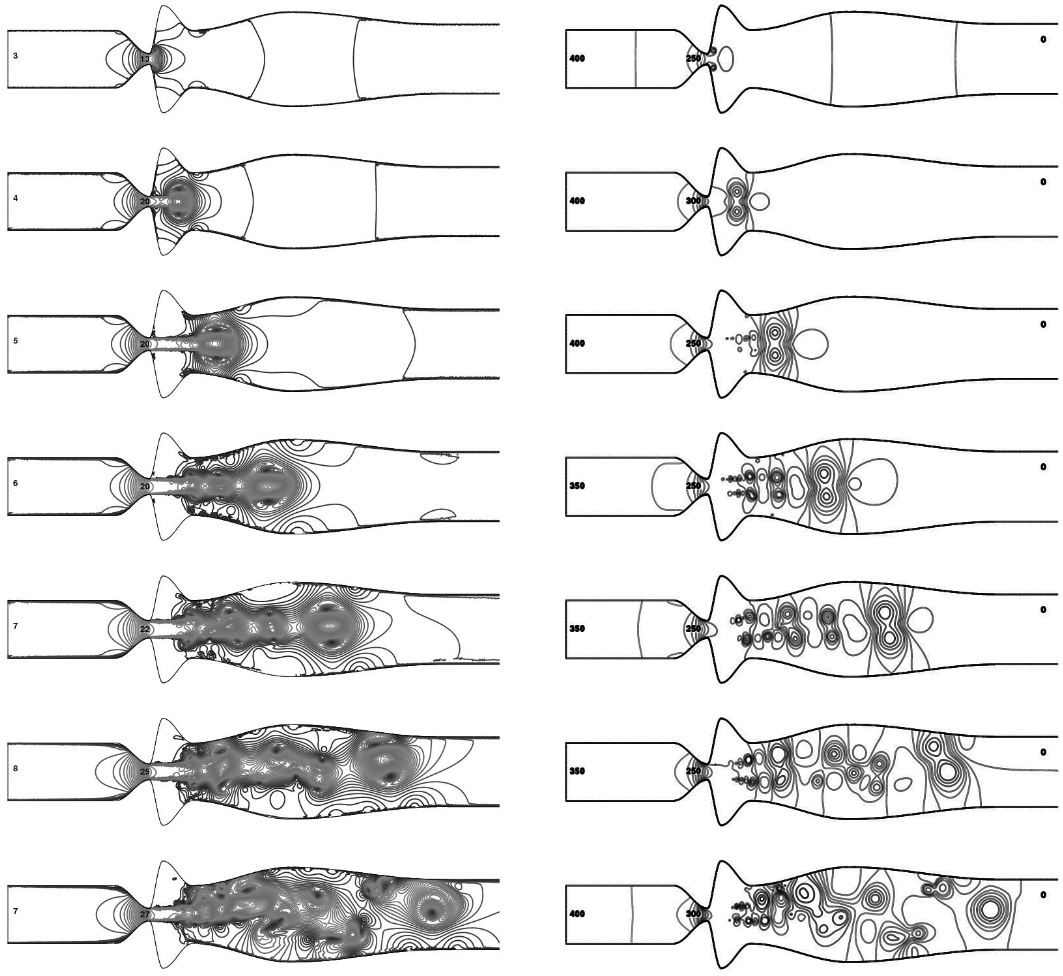


**Fig. 6:** *The isolines of velocity magnitude (left) and pressure (right) in a sequence of time instance (from top to bottom, Part 1).*

### 5.2 Flow in channel with vibrating vocal folds

The numerical results for flow in vibrating channel are presented for physically relevant pressure drop, inlet flow velocity, frequency of vibrations and width of the channel, which leads to the Reynolds numbers in the range $Re = 1000 - 3000$.

The computations were carried out for the pressure drop of 400 Pa, i.e $p^i_{ref} =$ 400 Pa and $p^o_{ref} = 0$ Pa. The initial condition was chosen as $\mathbf{v}^0 \equiv 0$ and the isotropically refined mesh was used, cf. Fig. 5. The gap oscillates harmonically around the mean gap value $\overline{g} = 4.4$ mm in the interval $g(t) \in [3.2\text{ mm}, 5.6\text{ mm}]$ with frequency $f = 100$ Hz .

The results are shown in Figs. 6-7 for the time instants marked in Fig. 8. The sudden expansion in the modelled glottal region leads to the faster flow in the vibrating narrowest part of the computational domain and to complicated flow structures in the outlet part of the channel. Similar effects were observed experimentally in [15].

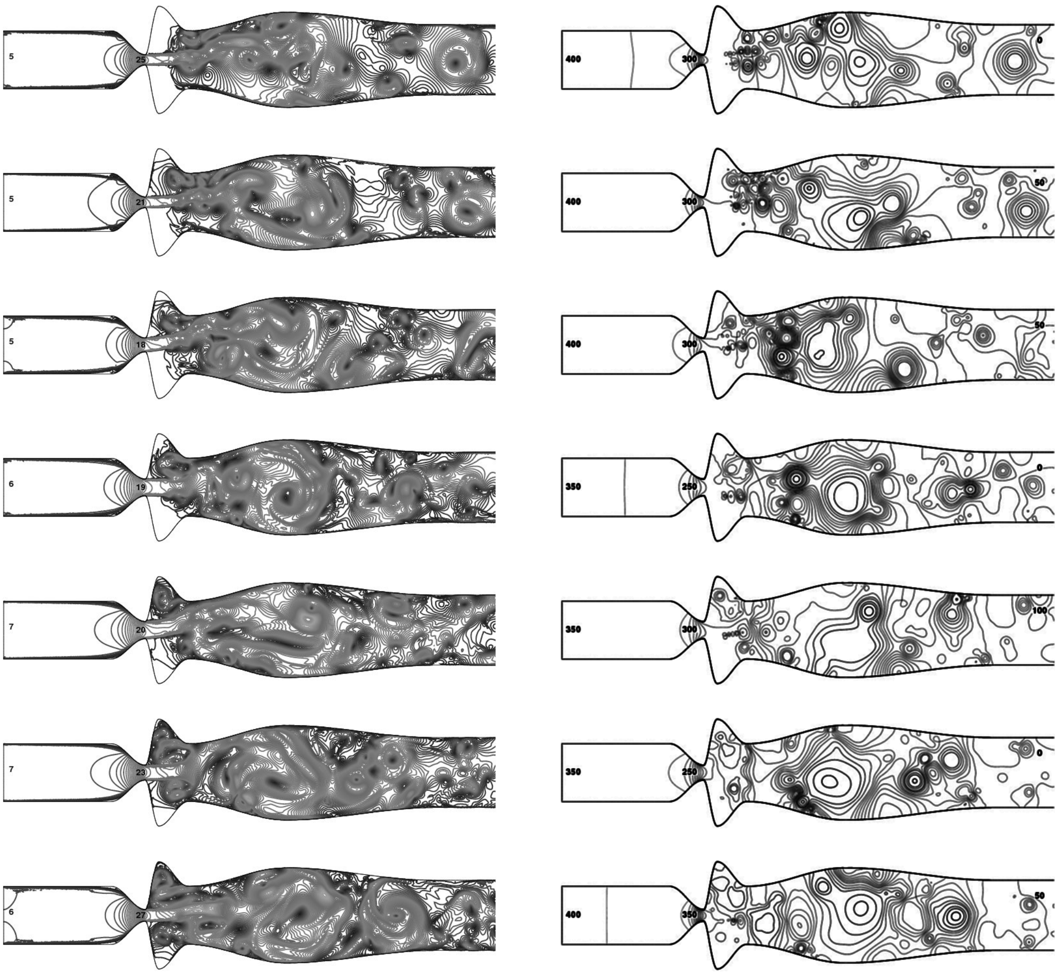


**Fig. 7:** *The isolines of velocity magnitude (left) and pressure (right) in a sequence of time instance (from top to bottom, Part 2).*

The inlet flow velocity and the flow velocity at on the axis of symmetry at the narrowest part of the channel are shown in Fig. 8. The both values oscillates with a similar frequency as the prescribed motion of the wall. However, the graphs are noisy partially due to the complicated flow structures downstream.

## 6 Conclusion

The paper presents the developed mathematical method and applied numerical technique for solution of fluid-structure problems encountered in biomechanics of voice production. The method consists of the advanced stabilization of the finite element method applied considering the moving domain. In order to obtain fast solution of the discretized problem a simplified multigrid method was applied, which allowed solution of significantly larger system of equations compared to the previously used approach, see e.g. [18].

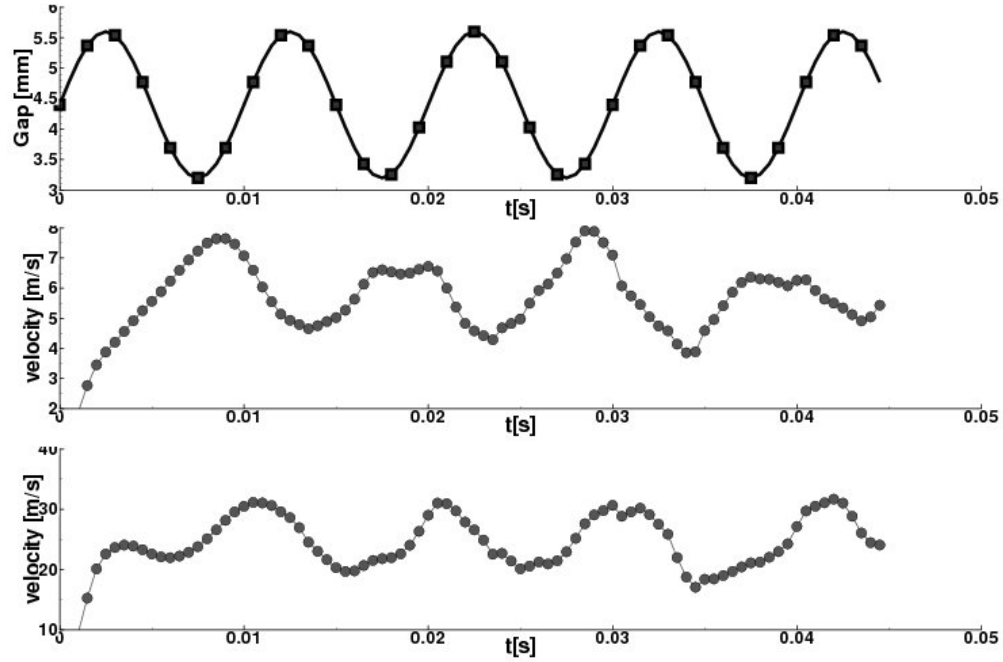


**Fig. 8:** *The gap oscillations $g(t)$ (upper graph), the computed flow velocity at the inlet (middle), and the computed flow velocity in the glottal orifice (lower graph).*

The influence of the isotropic and anisotropic meshes was studied and the multigrid technique was applied on a challenging problem of flow in symmetric channel with vibrating walls. The numerical results were presented showing the Coanda effect and complicated structure of small vortices and large size eddies generated at the glottal region by vibrating vocal fold. Similar vortex flow structures and Coanda effects were identified experimentally in [15].

## References

[1] Amestoy, P.R., Guermouche, A., L'Excellent, J.Y., and Pralet, S.: Hybrid scheduling for the parallel solution of linear systems. Parallel Computing **32** (2006), 136–156.

[2] Ciarlet, P.G.: *The finite element methods for elliptic problems.* North-Holland Publishing, 1979.

[3] Codina, R.: Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. Computational Method in Applied Mechanical Engineering **190** (2000), 1579–1599.

[4] Davis, T.A. and Duff, I.S.: A combined unifrontal/multifrontal method for unsymmetric sparse matrices. ACM Transactions on Mathematical Software **25** (1999), 1–19.

[5] Dolejší, V.: Anisotropic mesh adaptation technique for viscous flow simulation. East-West Journal of Numerical Mathematics **9** (2001), 1–24.

[6] Feistauer, M.: *Mathematical methods in fluid dynamics.* Longman Scientific & Technical, Harlow, 1993.

[7] Gelhard, T., Lube, G., Olshanskii, M.A., and Starcke, J.H.: Stabilized finite element schemes with LBB-stable elements for incompressible flows. Journal of Computational and Applied Mathematics **177** (2005), 243–267.

[8] Heywood, J.G., Rannacher, R., and Turek, S.: Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. Int. J. Numer. Math. Fluids **22** (1992), 325–352.

[9] John, V.: Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. Int. J. Num. Meth. Fluids **40** (2002), 775–798.

[10] John, V. and Tobiska, L.: Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations. Int. J. Num. Meth. Fluids **33** (2000), 453–473.

[11] Nobile, F.: *Numerical approximation of fluid-structure interaction problems with application to haemodynamics.* Ph.D. thesis, Ecole Polytechnique Federale de Lausanne, 2001.

[12] Nomura, T. and Hughes, T.J.R.: An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body. Computer Methods in Applied Mechanics and Engineering **95** (1992), 115–138.

[13] Otto, F.C. and Lube, G.: Non-overlapping domain decomposition applied to incompressible flow problems. Contemporary Mathematics **218** (1998), 507–514.

[14] Punčochářová, P., Fürst, J., Kozel, K., and Horáček, J.: Numerical simulation of compressible flow with low mach number through oscillating glottis. In: Zolotarev, I., and Horáček, J. (Eds.), *Flow Induced Vibrations.* Institute of Thermomechanics, CAS, Prague, 2008 .

[15] Šidlof, P.: *Fluid-structure interaction in human vocal folds.* Ph.D. thesis, Charles University, Faculty of Mathematics and Physics, 2007.

[16] Sváček, P. and Feistauer, M.: Application of a stabilized FEM to problems of aeroelasticity. In: *Numerical Mathematics and Advanced Application.* Springer, Berlin, 2004 pp. 796–805.

[17] Sváček, P., Feistauer, M., and Horáček, J.: Numerical simulation of a flow induced airfoil vibrations. In: *Proceedings Flow Induced Vibrations*, vol. 2. Ecole Polytechnique, Paris, 2004 pp. 57–62.

[18] Sváček, P., Feistauer, M., and Horáček, J.: Numerical simulation of flow induced airfoil vibrations with large amplitudes. Journal of Fluids and Structure **23** (2007), 391–411.

[19] Turek, S.: *Efficient solvers for incompressible flow problems: an algorithmic and computational approach.* Springer, Berlin, 1999.

[20] Verfürth, R.: Error estimates for mixed finite element approximation of the Stokes equations. R.A.I.R.O. Analyse numérique/Numerical analysis **18** (1984), 175–182.

# INVERSE PROBLEMS OF HEAT TRANSFER[*]

Jiří Vala

## 1 Introduction

Well-posed problems of heat transfer, much-favoured by most mathematicians, as an important class of simplified mathematical formulations of real physical processes, based on the conservation principles of classical mechanics, exploited in mechanical, electrical, civil etc. engineering, require the complete setting of i) initial conditions, ii) boundary conditions (prescribed temperature or heat flux everywhere) and iii) material characteristics. However, in engineering applications some data of types i), ii) or iii) are uncertain, inaccurate or missing. The remedy, coming from their reconstruction from some additional information, obtained from temperature or heat flux measurements, generate various classes of ill-posed problems with specific difficulties: even the apparently simple one-dimensional linearized model of heat propagation in a rod [6] needs non-trivial a priori estimates (valid under some additional regularity assumptions), combined truncation and regularization methods, to be able to apply the Schauder fixed point theorem.

The theoretical, experimental and computational analysis of inverse problems of heat transfer and related physical processes of the last decades has its own history: from different points of view it is monitored in [1], [3], [4] and [9]. In this paper we shall pay special attention to the missing data iii) in the analysis of insulation and accumulation properties of building materials (typically with a microscopically porous irregular structure), i.e. to the reliable identification of their basic macroscopic material characteristics. Following the Czech and European technical standards, we shall work with the thermal conductivity $\lambda$, heat capacity $c$ and material density $\rho$, constant at least within certain reasonable temperature range, in an isotropic medium. Whereas the experimental setting of $\rho$ is easy, the stationary measurements of $\lambda$ and $c$ do not give, according to the required measurement time, good results. The conventional non-stationary measurement equipments are expensive, use strange sets of calibration materials and their applicability to non-classical materials is limited. The development of alternative non-stationary identification methods (the frequency-domain method, the step-heating method, the hot-strip / hot-wire method, the infrared photography access, etc.) is documented in [1]. A class of primary inexpensive measurement devices, introduced in this paper, combines the hot-wire approach with the MATLAB-based numerical and computational support.

[*] This research is supported the project of specific research of Brno University of Technology No. FAST-S-10-17, by the Ministry of Education, Youth and Sports of Czech Republic No. MSM 0021630511 and by the OP Education for Competitiveness No. CZ.1.07/2.2.00/07.0410.

## 2 Physical and mathematical preliminaries

Let us consider the 3-dimensional Euclidean space, supplied by the Cartesian coordinate system $x = (x_1, x_2, x_3)$, and a domain $\Omega$ in this space, occupied by a (macroscopically) homogeneous and isotropic material with unknown characteristics $a = \lambda/(c\rho)$ and $b = 1/\lambda$, whose (sufficiently smooth) boundary $\partial\Omega$ involves some parts $\Gamma_D$ with prescribed boundary conditions of Dirichlet type

$$T(x,t) = T_*(x) \qquad \forall\, t \in I \;\; \forall\, x \in \Gamma_D \tag{1}$$

and $\Gamma_N$ with those of Neumann type

$$\nabla T(x,t) \cdot \nu(x) + bq(t) = 0 \qquad \forall\, t \in I \;\; \forall\, x \in \Gamma_N \tag{2}$$

where $\nu(x)$ refers to a local unit outside normal vector.

Since any real measurement device consists of a finite number $n$ of further material layers $\Omega_i$ with $i \in \{1, \ldots, n\}$ (cf. *Illustrative example*), the analogous notation can be applied to each $i$-th materials with prescribed characteristics $a_i$ and $b_i$. Moreover, (2) holds also with the heat flux $q(t)$, occurring on some part $\Gamma \subseteq \partial\Omega \setminus \Gamma_N$ of the union of interfaces $\Omega \cap \Omega_1, \ldots, \Omega \cap \Omega_n$. Similar heat fluxes are present on mutual interfaces of $\Omega_1, \ldots, \Omega_n$. All such fluxes are not known explicitly, being determined from contact conditions; here we shall consider only perfect contacts with continuous temperature distributions.

Following Chap. 3 of [2], the principle of conservation of energy together with the empirical constitutive Fourier law gives

$$\dot{T}(x,t) - a\nabla^2 T(x,t) = 0 \tag{3}$$

where the dot symbol is reserved for a derivative with respect to $t$ and $\nabla^2(\cdot)$ means $\mathrm{div}(\mathrm{grad}(\cdot))$ briefly; for $t = 0$ we shall consider $T(x,0) = T_e$ with the constant environmental temperature $T_e$ and the same, i.e. $T(\cdot,t) = T_e$, should be true in any time $t \in I$ on all outer surfaces of the layered measurement device to guarantee a physically closed measurement system.

It is natural to search for $T(x,t)$ in the space of abstract functions $L^2(I,V)$, mapping $I$ into some appropriate subspace $V$ of the Sobolev space $W^{1,2}(\Omega)$, although better regularity results can be obtained – see [4, p. 256]. In the direct formulation with given material characteristics $a$ and $b$ the solvability of (3) with boundary conditions (1) and (2) and convergence properties of sequences of approximate solutions in finite-dimensional spaces follow, at least for the most frequently discussed case $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$, from the Lax–Milgram theorem. Unfortunately, the inverse formulations with unknown $a$ and/or $b$, or, alternatively, with partially unknown initial or boundary conditions, result typically in ill-conditioned mathematical problems and unstable numerical algorithms, as discussed in [4, p. 21].

For simplicity, let us assume that just both $a$ and $b$ are unknown, consequently infinitely many solution of (3) with boundary conditions (1) and (2) may exist.

Let us introduce the following notations of scalar products: $(\cdot,\cdot)$ in $L^2(\Omega)$ and in $[L^2(\Omega)]^3$, $\langle\cdot,\cdot\rangle$ in $L^2(\partial\Omega)$, $\langle\cdot,\cdot\rangle_D$ in $L^2(\Gamma_D)$, $\langle\cdot,\cdot\rangle_N$ in $L^2(\Gamma_N)$, $\langle\cdot,\cdot\rangle_I$ in $L^2(I, L^2(\Gamma))$. Applying any test function $\Phi$ (usually) from $V$ to (3), the Green–Ostrogradskii theorem (at least in sense of distributions) gives $a^{-1}(\Phi,\dot{T}) - (\Phi,\nabla^2 T) = 0$, $a^{-1}(\Phi,\dot{T}) + (\nabla\Phi,\nabla T) - \langle\Phi,\nabla T\cdot\nu\rangle = 0$ and $a^{-1}(\Phi,\dot{T}) - (\nabla^2\Phi,T) - \langle\Phi,\nabla T\cdot\nu\rangle + \langle\nabla\Phi\cdot\nu,T\rangle = 0$; thus by (1) and (2) we have

$$a^{-1}(\Phi,\dot{T}) - B(\Phi,T) = b\langle\Phi,q\rangle_N - \langle\nabla\Phi\cdot\nu,T_*\rangle_D\,. \tag{4}$$

for $B(\Phi,T) := (\nabla^2\Phi,T) + \langle\Phi,\nabla T\cdot\nu\rangle_D - \langle\nabla\Phi\cdot\nu,T\rangle_N$.

The ideal final aim is to find $T \in L^2(I,V)$, together with real constants $a$ and $b$, satisfying (4) and (2). More realistic approaches try to satisfy (2) (or rarely (4)) in some weaker (inaccurate) sense – for the detailed overview see [1]. We shall apply the least squares technique, minimizing

$$F(a,b) = \frac{1}{2}\langle bq + \nabla T(a,b)\cdot\nu, bq + \nabla T(a,b)\cdot\nu\rangle_I\,; \tag{5}$$

$F$ here is only a real function of two variables $a$ and $b$, $T(x,t)$ from (4) depend on parameters $a$ and $b$, thus we have $T(x,t,a,b)$ now, omitting the first two variables $x$ and $t$ for brevity; the rectangular quadrature rule on $I$ in $m+1$ nodes $t = jh$ for $j \in \{0,1,\ldots,m\}$ and $h := \tau/m$ is needed in practice. The first and second derivatives of $F$,with respect to $a$ and $b$, i.e. $F_{,a}(a,b)$, $F_{,b}(a,b)$, $F_{,aa}(a,b)$, $F_{,ab}(a,b)$ and $F_{,bb}(a,b)$, can be then evaluated from the first and second temperature derivatives $T_{,a}(a,b)$, $T_{,b}(a,b)$, $T_{,aa}(a,b)$, $T_{,ab}(a,b)$ and $T_{,bb}(a,b)$. In the case of lack of boundary data (when $F$ becomes a more general functional), some (rather complicated) iterative procedures are available, e.g. that based on the conjugate gradient algorithm applied to direct, adjoint and sensitivity problems in [9, p. 21].

## 3 Computational algorithm

If some reasonable estimate of the characteristics $a$ and $b$ is available, we can apply the Newton algorithm to obtain their improved values $a^\star$ and $b^\star$ in the well-known form

$$\begin{bmatrix} F_{,aa}(a,b) & F_{,ab}(a,b) \\ F_{,ab}(a,b) & F_{,bb}(a,b) \end{bmatrix} \cdot \begin{bmatrix} a^\star - a \\ b^\star - b \end{bmatrix} = - \begin{bmatrix} F_{,a}(a,b) \\ F_{,b}(a,b) \end{bmatrix}\,. \tag{6}$$

The derivatives included in this formula should be as simple as possible.

Namely if $\partial\Omega = \Gamma_D = \Gamma$ and $\Gamma_N = \emptyset$ then $T$ is independent of $b$ and all derivatives of $F$ vanish or simplify substantially. For the discretization on $\Omega$ the finite element technique using the Hermite polynomials and the set of discrete unknown variables $\psi := (T,\nabla T)$ is available. Applying the Crank–Nicholson scheme, for any $j \in \{1,\ldots,m\}$ we obtain

$$\frac{1}{ah}N(\psi_j - \psi_{j-1}) - \frac{1}{2}K(\psi_j + \psi_{j-1}) = \frac{1}{2}(g_j + g_{j-1}) \tag{7}$$

with certain real symmetric sparse square matrices $N$ and $K$ and corresponding real vectors $g_0, g_1, \dots g_m$ (dependding on the choice of finite element mesh on $\Omega$), i.e. briefly $S\psi_j = Q\psi_{j-1} + \gamma_j$ with $M := N/h$, $S := a^{-1}M - K/2$, $Q := a^{-1}M + K/2$ and $\gamma_j := (g_j + g_{j-1})/2$. Consequently we receive $S\psi_{j,a} = Q\psi_{j-1,a} + a^{-2}M(\psi_j - \psi_{j-1})$ and $S\psi_{j,aa} = Q\psi_{j-1,aa} + 2a^{-2}M(\psi_{j,a} - \psi_{j-1,a}) - 2a^{-3}M(\psi_j - \psi_{j-1})$. Thus (7) enables us to evaluate all $a$-derivatives of $T$ and $\nabla T$ required in (6).

The 2- and 3-dimensional configurations typically do not admit $\Gamma_N = \emptyset$. Moreover, the heat fluxes $q(t)$ in the modified (2) are available (unlike those from the original (2)) only indirectly, from solutions of (4) on $\Omega_i$ with $i \in \{1, \dots, n\}$ instead of $\Omega$; this will be highlighted using the index $i$. Let us assume $\Gamma = \Gamma_D$ and $\Gamma_N = \Gamma \setminus \Gamma_D \neq \emptyset$. Then (4) gets the form $a^{-1}(\Phi, \dot{T}) - B(\Phi, T) = b\langle \Phi, q\rangle_N - \langle \nabla\Phi \cdot \nu, T_*\rangle_D$, $a_i^{-1}(\Phi, \dot{T})_i - B(\Phi, T)_i = b_i\langle \Phi, q\rangle_{Ni} - \langle \nabla\Phi \cdot \nu, T_*\rangle_{Di}$ for all $i \in \{1, \dots, n\}$. Thanks to the identity of heat fluxes on all nonempty sets $\partial\Omega \cup \partial\Omega_i$ and $\partial\Omega_i \cup \partial\Omega_k$ with $i, k \in \{1, \dots, n\}$ all remaining $q(t)$ can be eliminated to receive $\psi_1, \dots, \psi_n$ from the analogy of (7) as functions of $a$ and linear functions of $b$ (not only functions of $a$ as in the preceding case); for the detailed structure of corresponding linear systems (involving both thermal conduction and convection) see [3, p. 116]. This approach enables us to define $S$, $Q$, $\gamma_j$, etc., in the same way as from (7) again; their linear dependence on $b$ guarantees that the formulae for the evaluation of derivatives of $F$ (using the numerical quadrature on $\Gamma_D \times I$) with respect to $a$ and $b$ do not disturb the efficiency of the algorithm (6).

The same algorithm offers the possibility of quick evaluation of changes of $a$ and $b$ forced by the modified input data. The variance-based sensitivity analysis (the construction of Sobol indices) by [5] can be then useful to study the effect of stochastic uncertainty on the resulting $a$ and $b$. However, the general approach considers the variables $q(x, t, \theta)$, $T(x, t, \theta)$, etc. also as functions of parameters $\theta$ from the sample space $\Theta$ of elementary events; such sample space must be supplied by the minimal $\sigma$-algebra on $\Theta$ and by certain probability measure $P$. Then it is possible to replace $F(a, b)$ from (5) by

$$F(a, b) = \frac{1}{2} \int_\Theta \langle bq + \nabla T(a, b) \cdot \nu, bq + \nabla T(a, b) \cdot \nu\rangle_I \, \mathrm{d}P \tag{8}$$

and apply some uncertainty representation technique to (8), as the Karhunen-Loève or polynomial chaos expansions by [9, p. 10], or, alternatively, a Bayesian approach by [9, p. 25].

## 4 Illustrative example

The basic configuration of the measurement device, suggested originally in [8], consists of the following layers: 1. thick insulation layer (polystyrene), 2. active heating plate (aluminium), 3. material specimen (with unknown material characteristics), 4. passive additional plate (aluminium), 5. thick insulation layer (polystyrene). The interfaces 1./2. and 4./5. contain two sets of temperature sensors recording the temperature $T_*(t)$ at (in practice discrete) times $t$ from the time interval $I = [0, \tau]$ of a given length $\tau$. The interface 1./2. hides also a carefully controlled built-in generator of time-variable heat flux $q(t)$ for the same times $t$. However, such configuration is not acceptable e.g. for the measurements of maturing silicate mixtures in massive structures in situ: the remedy is to remove 4. and 5., considering the real massive structure (nearly the half-space) instead of 3.

The above sketched special geometrical configuration is typical just for such one-dimensional simplified systems with parallel layers, here especially $n = 4$. Unlike the formally complicated algorithm of [7], coming from the a priori known temperatures on the boundary of 1. and 5. and from the temperatures and heat fluxes at the left side of 2. and right side of 4., we are able to prescribe the temperature at the whole boundary of 3.

Fig. 1 and Fig. 2 show the results of identification of $a$ and $b$ from the experiment, lasting $\tau = 300\,\mathrm{s}$. The first half of both Fig. 1 and Fig. 2 refers to the new building material specimen, tested at the Faculty of Civil Engineering of Brno University of Technology (resulting $a = 1.09377 \cdot 10^{-6}\,\mathrm{m}^2/\mathrm{s}$, $b = 2.05909 \cdot 10^1\,\mathrm{m{\cdot}K/W}$), the second one to the mineral wool, whose properties are similar to polystyrene (resulting $a = 6.55382 \cdot 10^{-7}\,\mathrm{m}^2/\mathrm{s}$, $b = 1.13620\,\mathrm{m{\cdot}K/W}$), as the test of algorithm robustness only: the strongly insulated heating device from both sides causes the low accuracy of recorded temperature differences. The experimental heating was very special: constant for $t \in [0, 300]\,\mathrm{s}$, zero for $t \in [300, 600]\,\mathrm{s}$. Fig. 1 shows the redistribution of temperature, its gradient and heat flux in the whole measurement system in time: full lines for $t \in [0, 300]\,\mathrm{s}$, dotted lines otherwise. Fig. 2 demonstrates the least-squares-based fitting of computed interface values of heat flux with corresponding experimental data. The complete original software code has been written in MATLAB (without any additional packages).

## 5 Conclusions

The paper presents the mathematical preliminaries and the computational support for a rather general class of heat transfer problems, especially in building materials. An illustrative example demonstrates the MATLAB-based support for the identification of material characteristics, i.e. for the missing information iii) from *Introduction*. This approach is open to further generalization: to the analysis of anisotropic material ($\lambda$ becomes a real square symmetrical matrix), interface heat convection (new material characteristics of interfaces occur), temperature-dependent material characteristics, etc.

The proper mathematical analysis, including both the existence of solutions and the convergence of sequences of approximate solutions in finite-dimensional function spaces, constructed from the algorithm of above sketched type, contains still open questions. However, the aim of sufficient generalization of the results of type [6] seems to be realistic. The relevant analysis in probabilistic measures (instead of standard Lebesgue ones) is needed, too, to handle the evaluation of uncertainty of identified characteristics.

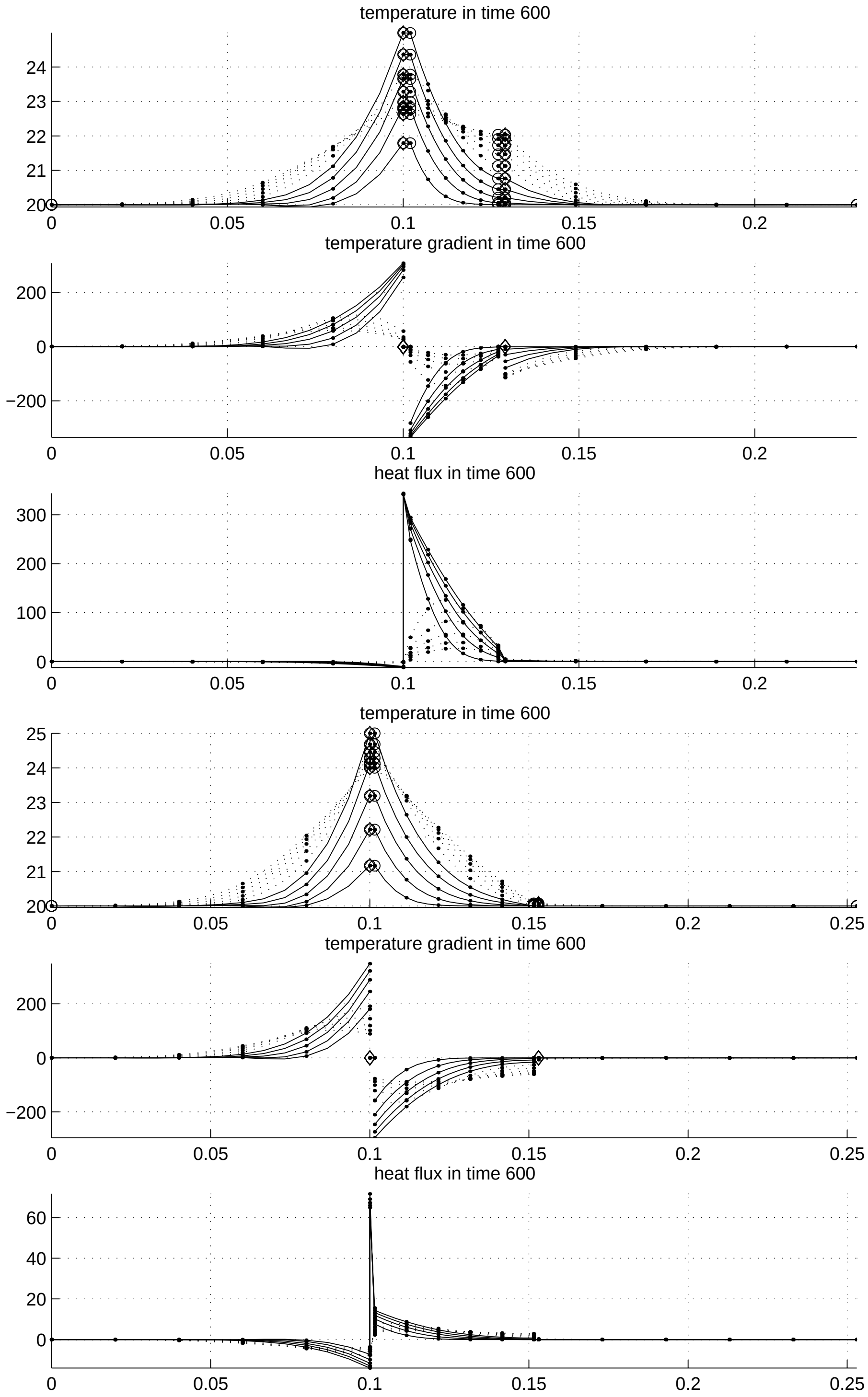


**Fig. 1:** *Temperature, its gradient and heat flux x-redistribution in time.*

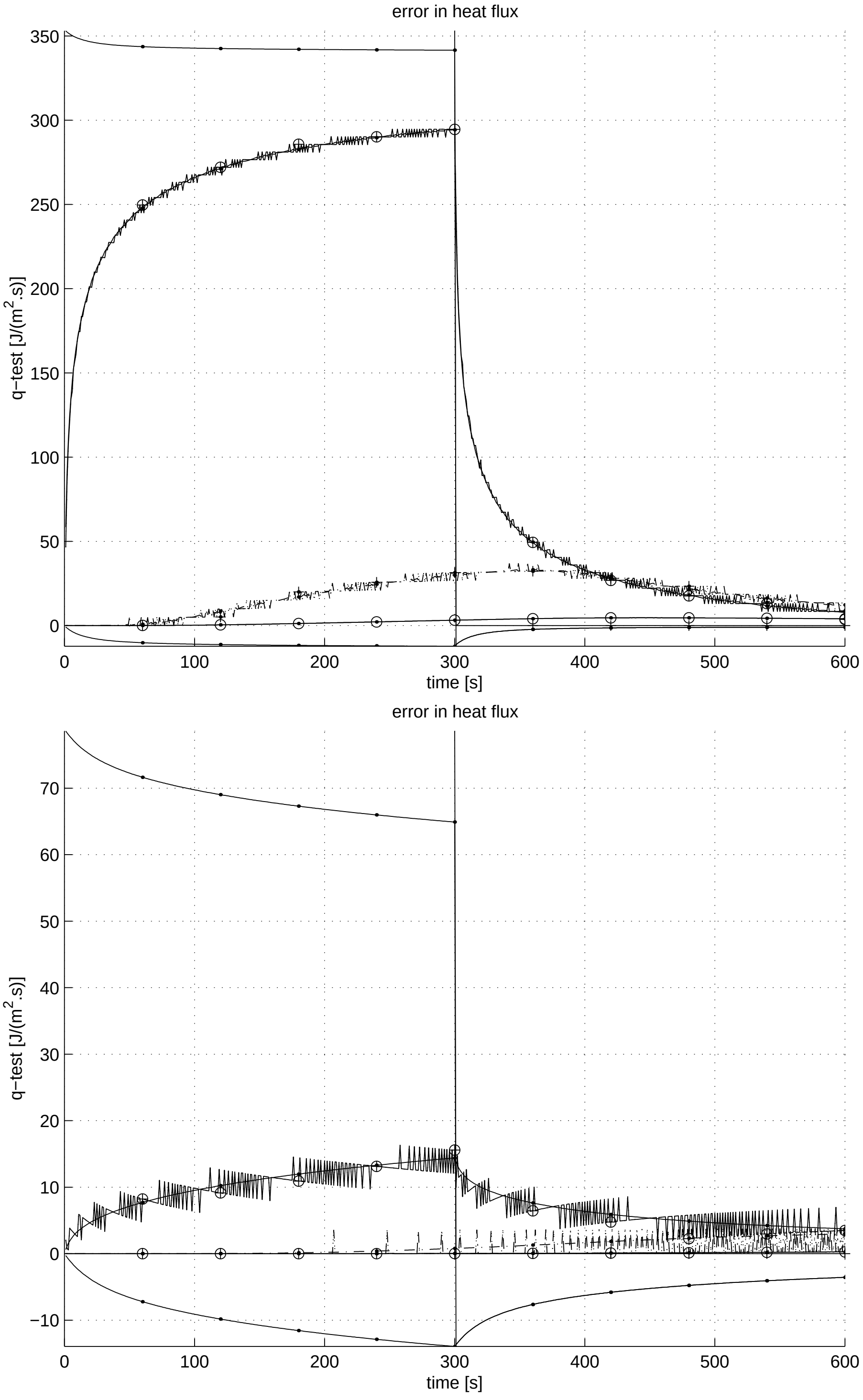


**Fig. 2:** *Fitting of computed interface values of heat flux with experimental data.*

## References

[1] Colaço, M., Orlande, H.R.B., and Dulikravich, G.S.: Inverse and optimization problems in heat transfer. Journal of the Brazilian Society of Mechanical Sciences and Engineering **28** (2006), 1–24.

[2] Davies, M.G.: *Building heat transfer.* J. Wiley & Sons, 2004.

[3] Duda, P.: Solution of multidimensional inverse heat conduction problem. Heat and Mass Transfer **40** (2003), 115–122.

[4] Isakov, V.: *Inverse problems for partial differential equations.* Springer, 2006.

[5] Kala, Z.: Stability problems of steel structures in the presence of stochastic and fuzzy uncertainty. Thin-Walled Structures **45** (2007), 861–865.

[6] Kozhanov, A.I.: Solvability of the inverse problem of finding thermal conductivity. Siberian Mathematical Journal **46** (2005), 841–856.

[7] Šťastník, S., Vala, J., and Kmínová, H.: Identification of basic thermal technical characteristics of building materials. Kybernetika **47** (2007), 561–576.

[8] Šťastník, S., Vala, J., and Kmínová, H.: Identification of thermal technical characteristics from the measurement of non-stationary heat propagation in porous materials. Forum Statisticum Slovacum **2** (2006), 203–210.

[9] Zabaras, N.: Inverse problems in heat transfer. In: W.J. Minkowycz, E.M. Sparrow and J. Y. Murthy (Eds.), *Handbook on Numerical Heat Transfer*, Chap. 17, J. Wiley & Sons, 2004.

# COMPLEMENTARITY – THE WAY TOWARDS GUARANTEED ERROR ESTIMATES*

Tomáš Vejchodský

**Abstract**

This paper presents a review of the complementary technique with the emphasis on computable and guaranteed upper bounds of the approximation error. For simplicity, the approach is described on a numerical solution of the Poisson problem. We derive the complementary error bounds, prove their fundamental properties, present the method of hypercircle, mention possible generalizations and show a couple of numerical examples.

## 1 Introduction

Reliability of numerical schemes is a crucial topic in the scientific and technical computing. There is a general agreement that an approximate solution alone is not sufficient as an output of the computations. The user needs certain information about its accuracy.

An ultimate goal of numerical algorithms is to provide an approximate solution with accuracy within a prescribed tolerance in an efficient way. In the framework of numerical methods for partial differential equation this goal can be achieved. The needed tool is an *adaptive algorithm* equipped with an efficient and reliable error indicator for mesh refinements and with a *computable guaranteed upper bounds on the error* for the stopping criterion.

In this contribution we concentrate on the guaranteed upper bounds on the error in the context of the finite element method for linear elliptic problems. As a model problem we use the Poisson equation with homogeneous Dirichlet boundary conditions. We point out that the complementary approach is not limited to finite elements only and can be used for arbitrary numerical method.

To illustrate the adaptive approach we introduce certain notation motivated by the finite element method. The finite element approximation $u_h$ of an exact solution $u$ is typically constructed on a finite element mesh $\mathcal{T}_h$. In order to employ the adaptive algorithm, we need an error indicator $\eta_K$ which estimates a suitable norm of the error $(u - u_h)|_K$ in the element $K \in \mathcal{T}_h$. In order to fulfill the goal and provide an approximation which is guaranteed to be under the user prescribed tolerance TOL, it is necessary to use certain guaranteed upper bound $\eta$ on a suitable norm of the

*This work was supported by grant No. IAA100760702 of the Grant Agency of the Academy of Sciences of the Czech Republic and by the institutional research plan no. AV0Z10190503 of the Academy of Sciences of the Czech Republic.

error. The error bound $\eta$ is said to be the *guaranteed upper bound* of the error if $\|u - u_h\| \le \eta$. Let us remark that the error bound $\eta$ is often computed from the error indicators as $\eta^2 = \sum_{K \in \mathcal{T}_h} \eta_K^2$. Now, we can present the general adaptive algorithm:

1. *Initialize*: Construct the initial mesh $\mathcal{T}_h$.
2. *Solve*: Find approximate solution $u_h$ on $\mathcal{T}_h$.
3. *Indicators*: Compute error indicators $\eta_K$ for all $K \in \mathcal{T}_h$.
4. *Estimator*: Compute error estimator $\eta$.
5. *Stop*: If $\eta \le \mathrm{TOL}$ then STOP.
6. *Mark*: If $\eta_K \ge \Theta \max\limits_{K \in \mathcal{T}_h} \eta_K$ then mark $K$.
7. *Refine*: Refine the marked elements and build the new mesh $\mathcal{T}_h$.
8. Go to 2.

The parameter $\Theta \in (0, 1)$ in Step 6 is given by the user and determines the fraction of elements to be refined in each adaptive cycle.

In this adaptive algorithm we can clearly distinguish the different roles of error indicators $\eta_K$ and the error estimator $\eta$. If the estimator $\eta$ provides guaranteed upper bound of the error and the algorithm stops in Step 5 then $\|u - u_h\| \le \eta \le \mathrm{TOL}$ and the goal is fulfilled – the error is below the prescribed tolerance.

The computation of fully computable guaranteed upper bounds on the error seems to be a more difficult problem than construction of local error indicators $\eta_K$. The guaranteed error bounds can be successfully obtained by the *complementary approach.* The idea is fairly old. It goes back to the method of hypercircle from 1950's [21]. Further development came in 1970's and 1980's with the dual (or equilibrium) finite elements, see e.g. [4, 6, 8, 9, 11, 22, 28]. Later, the idea was worked out even further in the approach of error majorants, see e.g. [3, 13, 14, 17, 18, 16, 19]. Anyway, the idea can be traced in many other works, see e.g. [2, 5, 27].

In the rest of the paper we would like to give a brief review of the complementary approach for the Poisson problem. The emphasis is put on the derivation, properties, and practical implementation of computable guaranteed upper bounds on the energy norm of the error. The organization is the following. Section 2 introduces the classical and weak formulation of the Poisson equation. Section 3 contains derivation of the complementary guaranteed error bound and Section 4 presents the corresponding complementary problem and the theoretical properties of the complementary solution including the method of hypercircle. Section 5 briefly describes the concurrent approach of error majorants. Section 6 provides hints for practical evaluation of the obtained error bounds. Section 7 briefly lists possible generalizations of the complementary approach to various especially non-elliptic problems. Section 8 presents numerical examples to show the practical implementation and to compare the described variants of the error bounds mainly by their accuracy. Section 9 contains concluding remarks.

## 2 Model problem

Let us consider a domain $\Omega \subset \mathbb{R}^d$ with Lipschitz continuous boundary. The classical formulation of the Poisson problem reads: find $u \in C^2(\Omega) \cap C(\overline{\Omega})$ such that

$$
\begin{aligned}
-\Delta u &= f \quad \text{in } \Omega, && (1)\\
u &= 0 \quad \text{on } \partial\Omega. && (2)
\end{aligned}
$$

In order to introduce the complementary approach, it is advantageous to formulate problem (1)–(2) in the weak sense. Therefore, we consider the standard Sobolev space $V = H_0^1(\Omega)$ of square-integrable functions with square-integrable derivatives and vanishing traces on the boundary $\partial\Omega$.

The weak formulation of problem (1)–(2) reads: find $u \in V$ such that

$$
\mathcal{B}(u, v) = \mathcal{F}(v) \quad \forall v \in V. \tag{3}
$$

The bilinear form $\mathcal{B}$ and the linear functional $\mathcal{F}$ are given as

$$
\mathcal{B}(u, v) = (\boldsymbol{\nabla} u, \boldsymbol{\nabla} v) \quad \text{and} \quad \mathcal{F}(v) = (f, v),
$$

where

$$
(\boldsymbol{v}, \boldsymbol{w}) = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{w} \, \mathrm{d}x \quad \text{and} \quad (v, w) = \int_\Omega v w \, \mathrm{d}x
$$

stand for the vector and scalar version of the $L^2(\Omega)$ inner product. We point out that within the paper we denote the vector quantities by bold symbols.

The following lemma presents a simple observation. It says that the gradient $\boldsymbol{\nabla} u$ of the weak solution of (3) lies in $\mathbf{H}(\mathrm{div}, \Omega)$. For the reader's convenience, we recall the definition

$$
\mathbf{H}(\mathrm{div}, \Omega) = \left\{ \boldsymbol{y} \in [L^2(\Omega)]^d : \mathrm{div}\, \boldsymbol{y} \in L^2(\Omega) \right\}, \tag{4}
$$

where the divergence is understood in the sense of distributions.

**Lemma 1.** *Let $u \in V$ be the weak solution given by* (3)*. If the corresponding right-hand side $f$ is in $L^2(\Omega)$ then $\boldsymbol{\nabla} u \in \mathbf{H}(\mathrm{div}, \Omega)$.*

*Proof.* The divergence $\mathrm{div}\, \boldsymbol{y}$ is in $L^2(\Omega)$ in the sense of distributions if and only if there exists $z \in L^2(\Omega)$ such that $(v, z) = -(\boldsymbol{\nabla} v, \boldsymbol{y})$ for all $v \in C_0^\infty(\Omega)$. Thus, putting $z = -f$, we immediately conclude that $\boldsymbol{y} = \boldsymbol{\nabla} u$ lies in $\mathbf{H}(\mathrm{div}, \Omega)$ whenever $f$ lies in $L^2(\Omega)$, see definitions (4) and (3). □

We remind that the $z \in L^2(\Omega)$ from the above proof is called the distributional divergence of $\boldsymbol{y} \in [L^2(\Omega)]^d$ and we put $\mathrm{div}\, \boldsymbol{y} = z$.

## 3 Derivation of the complementary error estimate

The complementary error estimates can be easily derived using the divergence theorem:

$$\int_\Omega v \operatorname{div} \boldsymbol{y} \,\mathrm{d}x + \int_\Omega \boldsymbol{y} \cdot \boldsymbol{\nabla} v \,\mathrm{d}x - \int_{\partial\Omega} v \boldsymbol{y} \cdot \boldsymbol{n} \,\mathrm{d}x = 0 \quad \forall v \in H^1(\Omega),\ \forall \boldsymbol{y} \in \mathbf{H}(\operatorname{div}, \Omega), \tag{5}$$

where $\boldsymbol{n}$ stands for the unit outward normal vector to the boundary $\partial\Omega$.

The definition of the weak solution (3) together with the divergence theorem yields the following identity for arbitrary $u_h \in V$, $v \in V$, and $\boldsymbol{y} \in \mathbf{H}(\operatorname{div}, \Omega)$:

$$\begin{aligned} \mathcal{B}(u - u_h, v) &= (f, v) - (\boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v) + (v, \operatorname{div} \boldsymbol{y}) + (\boldsymbol{y}, \boldsymbol{\nabla} v) \\ &= (f + \operatorname{div} \boldsymbol{y}, v) + (\boldsymbol{y} - \boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v). \end{aligned} \tag{6}$$

This is the main trick. Subsequent derivation of the complementary error estimates is based on more or less standard technical steps. Crucial point is the handling of the term $(f + \operatorname{div} \boldsymbol{y}, v)$. There are at least two possibilities. The first one is to restrict the set of admissible $\boldsymbol{y}$ such that this term vanishes. The second one is based on the Friedrichs' inequality. We postpone the second possibility to Section 5 and start with the first one.

We introduce an affine space

$$\boldsymbol{Q}(f) = \{\boldsymbol{y} \in \mathbf{H}(\operatorname{div}, \Omega) : (\boldsymbol{y}, \boldsymbol{\nabla} v) = (f, v) \quad \forall v \in V\}. \tag{7}$$

This is a set of vector fields $\boldsymbol{y} \in \mathbf{H}(\operatorname{div}, \Omega)$ satisfying $-\operatorname{div} \boldsymbol{y} = f$ in the weak sense. Below, we will use the consistent notation $\boldsymbol{Q}(0)$ for the space of divergence-free (solenoidal) vector fields.

Using identity (6) for any $\boldsymbol{y} \in \boldsymbol{Q}(f)$ and the Cauchy-Schwarz inequality, we immediately obtain

$$\mathcal{B}(u - u_h, v) = (\boldsymbol{y} - \boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v) \le \|\boldsymbol{y} - \boldsymbol{\nabla} u_h\|_0 \|\boldsymbol{\nabla} v\|_0, \tag{8}$$

where $\|\boldsymbol{w}\|_0^2 = (\boldsymbol{w}, \boldsymbol{w})$ is the norm in $[L^2(\Omega)]^d$. Introducing the energy norm $|||v|||^2 = \mathcal{B}(v, v) = \|\boldsymbol{\nabla} v\|_0^2$ and substituting $v = u - u_h$ into (8) yields finally the guaranteed upper bound on the approximation error of $u_h$:

$$|||u - u_h||| \le \eta(u_h, \boldsymbol{y}) \qquad \forall u_h \in V,\ \forall \boldsymbol{y} \in \boldsymbol{Q}(f), \tag{9}$$

where the complementary error estimate is given by

$$\eta(u_h, \boldsymbol{y}) = \|\boldsymbol{y} - \boldsymbol{\nabla} u_h\|_0. \tag{10}$$

We point out that the error bound (9) holds true for arbitrary conforming approximation $u_h \in V$ of the weak solution $u$, regardless what numerical method has been used for it. In addition, the bound (9) is valid for any $\boldsymbol{y} \in \boldsymbol{Q}(f)$. Hence, practically, any vector field from $\boldsymbol{Q}(f)$ used in (9) provides guaranteed upper bound on the energy norm of the approximation error. However, this $\boldsymbol{y} \in \boldsymbol{Q}(f)$ must be chosen with care, otherwise the value $\eta(u_h, \boldsymbol{y})$ overestimates the error too much. A practical choice of a suitable $\boldsymbol{y} \in \boldsymbol{Q}(f)$ is discussed below in Section 6.

Let us conclude this section by summarizing the main statement into a theorem.

**Theorem 2.** *If $u \in V$ is the exact solution of problem* (3) *and $u_h \in V$ its arbitrary approximation then estimate* (9)–(10) *holds true for any $\boldsymbol{y} \in \boldsymbol{Q}(f)$.*

## 4 The complementary problem

Let the approximation $u_h \in V$ be fixed. Since $\eta(u_h, \boldsymbol{y})$ is an upper bound of its error, it is natural to ask, what $\boldsymbol{y} \in \boldsymbol{Q}(f)$ minimizes this error bound. The problem of minimization of $\eta(u_h, \boldsymbol{y})$ with respect to $\boldsymbol{y} \in \boldsymbol{Q}(f)$ is called the complementary problem and its solution the complementary solution. It turns out that this problem can be formulated in several equivalent forms. Before, we state a theorem about this equivalence, let us introduce certain notation. Let us define the complementary bilinear form $\mathcal{B}^*(\boldsymbol{y}, \boldsymbol{w}) = (\boldsymbol{y}, \boldsymbol{w})$, the complementary energy norm $|||\boldsymbol{w}|||_*^2 = \mathcal{B}^*(\boldsymbol{w}, \boldsymbol{w})$, and the functional of the complementary energy $J^*(\boldsymbol{w}) = \frac{1}{2}\mathcal{B}^*(\boldsymbol{w}, \boldsymbol{w})$.

**Theorem 3.** *The following problems are equivalent*

$$\text{find } \boldsymbol{y} \in \boldsymbol{Q}(f): \quad \eta(u_h, \boldsymbol{y}) \le \eta(u_h, \boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{Q}(f), \tag{11}$$

$$\text{find } \boldsymbol{y} \in \boldsymbol{Q}(f): \quad J^*(\boldsymbol{y}) \le J^*(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{Q}(f), \tag{12}$$

$$\text{find } \boldsymbol{y} \in \boldsymbol{Q}(f): \quad \mathcal{B}^*(\boldsymbol{y}, \boldsymbol{w}^0) = 0 \quad \forall \boldsymbol{w}^0 \in \boldsymbol{Q}(0). \tag{13}$$

*Proof.* First, we prove the equivalence of (11) and (12). Using (10) in (11), and utilizing the fact that $(\boldsymbol{y}, \boldsymbol{\nabla} u_h) = (\boldsymbol{w}, \boldsymbol{\nabla} u_h) = (f, u_h)$ for any $\boldsymbol{y} \in \boldsymbol{Q}(f)$ and $\boldsymbol{w} \in \boldsymbol{Q}(f)$, we can perform the following chain of simple equivalent adjustments:

$$\begin{gathered}
\eta(u_h, \boldsymbol{y}) \le \eta(u_h, \boldsymbol{w}), \\
\|\boldsymbol{y} - \boldsymbol{\nabla} u_h\|_0^2 \le \|\boldsymbol{w} - \boldsymbol{\nabla} u_h\|_0^2, \\
\|\boldsymbol{y}\|_0^2 - 2(\boldsymbol{y}, \boldsymbol{\nabla} u_h) + \|\boldsymbol{\nabla} u_h\|_0^2 \le \|\boldsymbol{w}\|_0^2 - 2(\boldsymbol{w}, \boldsymbol{\nabla} u_h) + \|\boldsymbol{\nabla} u_h\|_0^2, \\
\|\boldsymbol{y}\|_0^2 \le \|\boldsymbol{w}\|_0^2, \\
J^*(\boldsymbol{y}) \le J^*(\boldsymbol{w}).
\end{gathered}$$

Second, we prove that any solution of problem (12) is a solution of (13). Let $\boldsymbol{y} \in \boldsymbol{Q}(f)$ be a solution of (12) and let $\boldsymbol{w}^0 \in \boldsymbol{Q}(0)$ be arbitrary. Then $\boldsymbol{y} + t\boldsymbol{w}^0$ lies in $\boldsymbol{Q}(f)$ for any $t \in \mathbb{R}$ and the real function $\varphi(t) = \|\boldsymbol{y} + t\boldsymbol{w}^0\|_0^2$ has the global minimum at $t = 0$. If we compute the derivative of $\varphi(t)$ at $t = 0$ by definition, we obtain

$$\varphi'(0) = \lim_{t \to 0} \frac{\|\boldsymbol{y} + t\boldsymbol{w}^0\|_0^2 - \|\boldsymbol{y}\|_0^2}{t} = \lim_{t \to 0} \frac{2t(\boldsymbol{y}, \boldsymbol{w}^0) + t^2 \|\boldsymbol{w}^0\|_0^2}{t} = 2\mathcal{B}^*(\boldsymbol{y}, \boldsymbol{w}^0).$$

Hence, the derivative exists and since $\varphi(t)$ has the minimum at $t = 0$, the derivative has to vanish: $\varphi'(0) = 0$. This proves that $\boldsymbol{y}$ solves (13).

Finally, we prove that any solution of (13) is a solution of (12). Let $\boldsymbol{y} \in \boldsymbol{Q}(f)$ be the solution of (13) and let $\boldsymbol{w} \in \boldsymbol{Q}(f)$ be arbitrary. Let us set $\boldsymbol{w}^0 = \boldsymbol{w} - \boldsymbol{y}$. Clearly, $\boldsymbol{w}^0 \in \boldsymbol{Q}(0)$. Since $(\boldsymbol{y}, \boldsymbol{w}) = (\boldsymbol{y}, \boldsymbol{y} + \boldsymbol{w}^0) = \|\boldsymbol{y}\|_0^2$, see (13), we easily conclude that

$$0 \leq \|\boldsymbol{w} - \boldsymbol{y}\|_0^2 = \|\boldsymbol{w}\|_0^2 - 2(\boldsymbol{y}, \boldsymbol{w}) + \|\boldsymbol{y}\|_0^2 = \|\boldsymbol{w}\|_0^2 - \|\boldsymbol{y}\|_0^2 .$$

This proves that $\boldsymbol{y}$ solves (12). □

Formulation (11) of the complementary problem is natural to derive. It is a straightforward minimization of $\eta$. Formulation (12) is variational. It is a minimization of a simple quadratic functional – the complementary energy $J^*$. Variant (13) is a weak formulation using the complementary bilinear form $\mathcal{B}^*$. Notice that in the simple case of Poisson equation (3), the complementary problem is just a problem of orthogonal projection. The following theorem finds the complementary solution and states that it is unique.

**Theorem 4.** *Let $u \in V$ be the exact solution of problem* (3)*. Then $\boldsymbol{y} = \boldsymbol{\nabla} u$ lies in $\boldsymbol{Q}(f)$ and it is the unique solution of complementary problems* (11)–(13)*.*

*Proof.* Lemma 1 implies that $\boldsymbol{\nabla} u$ lies in $\mathbf{H}(\mathrm{div}, \Omega)$. Weak formulation (3) guarantees that $\boldsymbol{\nabla} u$ is in $\boldsymbol{Q}(f)$. Substituting $\boldsymbol{y} = \boldsymbol{\nabla} u$ into (13) and using the definition (7) of $\boldsymbol{Q}(0)$ we immediately find that

$$(\boldsymbol{y}, \boldsymbol{w}^0) = (\boldsymbol{\nabla} u, \boldsymbol{w}^0) = 0 \quad \forall \boldsymbol{w}^0 \in \boldsymbol{Q}(0).$$

Thus, $\boldsymbol{y} = \boldsymbol{\nabla} u$ is a solution of problem (13).

To prove the uniqueness, we consider two solutions $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \boldsymbol{Q}(f)$ of problem (13). Then of course $(\boldsymbol{y}_1 - \boldsymbol{y}_2, \boldsymbol{w}^0) = 0$ for all $\boldsymbol{w}^0 \in \boldsymbol{Q}(0)$. Since $\boldsymbol{y}_1 - \boldsymbol{y}_2 \in \boldsymbol{Q}(0)$, we can set $\boldsymbol{w}^0 = \boldsymbol{y}_1 - \boldsymbol{y}_2$ and obtain $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_0 = 0$. Thus, $\boldsymbol{y}_1 = \boldsymbol{y}_2$.

Theorem 3 finishes the proof. □

Sometimes, we call problem (3) the primal problem, in order to distinguish it from the complementary problem. Notice that this primal problem can also be equivalently formulated as energy minimization. The corresponding functional of primal energy is $J(v) = \frac{1}{2}\mathcal{B}(v,v) - \mathcal{F}(v)$. Interestingly, if we sum up the functionals of primal and complementary energy evaluated at the exact primal and complementary solutions $u$ and $\boldsymbol{y} = \boldsymbol{\nabla} u$, we obtain zero:

$$J(u) + J^*(\boldsymbol{y}) = -\frac{1}{2}\mathcal{B}(u,u) + \frac{1}{2}\mathcal{B}^*(\boldsymbol{\nabla} u, \boldsymbol{\nabla} u) = 0.$$

The next theorem provides an interesting result. It reminds the Pythagoras' theorem and it is based on the orthogonality of the spaces $\boldsymbol{Q}(0)$ and $\boldsymbol{\nabla} V$, i.e.,

$$(\boldsymbol{w}^0, \boldsymbol{\nabla} v) = 0 \quad \forall \boldsymbol{w}^0 \in \boldsymbol{Q}(0),\ \forall v \in V, \tag{14}$$

see (7). Figure 1 (left) illustrates this orthogonality.

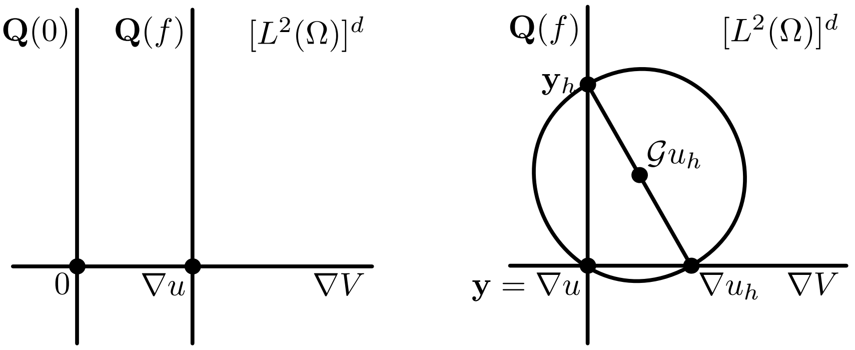


**Fig. 1:** *Orthogonality of spaces $\boldsymbol{Q}(f)$ and $\boldsymbol{\nabla} V$ in $[L^2(\Omega)]^d$ (left). An illustration of the method of hypercircle (right).*

**Theorem 5.** *Let $u \in V$ and $\boldsymbol{y} \in \boldsymbol{Q}(f)$ be exact solutions of problems* (3) *and* (11)–(13), *respectively. Then*

$$\eta^2(u, \boldsymbol{y}_h) + \eta^2(u_h, \boldsymbol{y}) = \eta^2(u_h, \boldsymbol{y}_h) \quad \forall u_h \in V, \ \forall \boldsymbol{y}_h \in \boldsymbol{Q}(f). \tag{15}$$

*Proof.* We use the fact that $\boldsymbol{y} = \boldsymbol{\nabla} u$, see Theorem 4, and the orthogonality (14) in the form $(\boldsymbol{y}_h - \boldsymbol{\nabla} u, \boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h) = 0$ to compute

$$\begin{aligned} \eta^2(u_h, \boldsymbol{y}_h) &= \|\boldsymbol{y}_h - \boldsymbol{\nabla} u + \boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h\|_0^2 \\ &= \|\boldsymbol{y}_h - \boldsymbol{\nabla} u\|_0^2 + \|\boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h\|_0^2 = \eta^2(u, \boldsymbol{y}_h) + \eta^2(u_h, \boldsymbol{y}). \end{aligned}$$

□

Notice that using definition (10) and Theorem 4, equality (15) can be stated in the form

$$\|\boldsymbol{y}_h - \boldsymbol{y}\|_0^2 + \|\boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h\|_0^2 = \|\boldsymbol{y}_h - \boldsymbol{\nabla} u_h\|_0^2 . \tag{16}$$

This relates the error in the complementary problem and the error in the primal problem with the computable difference of the approximate primal and complementary solutions. Consequently, the error estimate $\eta(u_h, \boldsymbol{y}_h)$ is also a guaranteed upper bound on the complementary energy norm of the error in the complementary problem:

$$|||\boldsymbol{y} - \boldsymbol{y}_h|||_* \leq \eta(u_h, \boldsymbol{y}_h) \quad \forall u_h \in V, \ \forall \boldsymbol{y}_h \in \boldsymbol{Q}(f).$$

The final result of this section is called the method of hypercircle. It is a remarkable result in the field of the a posteriori error estimates, because it provides an approximation whose error is known exactly. More precisely, the arithmetic average of $\boldsymbol{\nabla} u_h$ (the gradient of the approximate primal solution) and $\boldsymbol{y}_h$ (the approximate complementary solution) yields an approximation $\boldsymbol{\mathcal{G}} u_h = (\boldsymbol{y}_h + \boldsymbol{\nabla} u_h)/2$ of $\boldsymbol{\nabla} u$ (the gradient of the exact solution). The error of $\boldsymbol{\mathcal{G}} u_h$ measured in the complementary energy norm can be computed exactly from the knowledge of $u_h$ and $\boldsymbol{y}_h$. See Figure 1 (right) for an illustration.

**Theorem 6** (Method of hypercircle)**.** *Let $u \in V$ be the exact solution of problem* (3)*. Consider arbitrary $u_h \in V$ and $\boldsymbol{y}_h \in \boldsymbol{Q}(f)$ and set $\boldsymbol{\mathcal{G}} u_h = (\boldsymbol{y}_h + \boldsymbol{\nabla} u_h)/2$. Then*

$$\left|\!\left|\!\left| \boldsymbol{\nabla} u - \boldsymbol{\mathcal{G}} u_h \right|\!\right|\!\right|_* = \frac{1}{2}\eta(u_h, \boldsymbol{y}_h).$$

*Proof.* Using the fact that $\boldsymbol{\nabla} u \in \boldsymbol{Q}(f)$ and again the orthogonality (14) in the form $(\boldsymbol{\nabla} u - \boldsymbol{y}_h, \boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h) = 0$, the statement follows from (16) by direct computations:

$$\begin{aligned} 4 \left\| \boldsymbol{\nabla} u - \boldsymbol{\mathcal{G}} u_h \right\|_0^2 &= \left\| \boldsymbol{\nabla} u - \boldsymbol{y}_h + \boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h \right\|_0^2 \\ &= \left\| \boldsymbol{\nabla} u - \boldsymbol{y}_h \right\|_0^2 + \left\| \boldsymbol{\nabla} u - \boldsymbol{\nabla} u_h \right\|_0^2 = \left\| \boldsymbol{y}_h - \boldsymbol{\nabla} u_h \right\|_0^2 . \end{aligned}$$

□

## 5 Error majorants

As we announced above in Section 3, there is also another possibility how to derive a guaranteed upper bound from (6). It is based on *Friedrichs' inequality*:

$$\|v\|_0 \leq C_\Omega \left\| \boldsymbol{\nabla} v \right\|_0 \quad \forall v \in V,$$

see e.g. [15]. The optimal constant $C_\Omega$ is known as the *Friedrichs' constant.* Its determination is a difficult task and its exact value is known in exceptional cases only. However, various upper bounds for Friedrichs' constant $C_\Omega$ are known. For example, in [12] we can find the estimate

$$C_\Omega \leq \frac{1}{\pi} \left( \frac{1}{|a_1|^2} + \cdots + \frac{1}{|a_d|^2} \right)^{-1/2}, \tag{17}$$

where $|a_1|, \ldots, |a_d|$ are lengths of sides of a $d$-dimensional box, the domain $\Omega$ is contained in.

Using the Cauchy-Schwarz and the Friedrichs' inequality in (6), we obtain

$$\mathcal{B}(u - u_h, v) \leq \left( C_\Omega \left\| f + \operatorname{div} \boldsymbol{y} \right\|_0 + \left\| \boldsymbol{y} - \boldsymbol{\nabla} u_h \right\|_0 \right) |\!|\!| v |\!|\!|.$$

Substitution $v = u - u_h$ yields the error estimate

$$|\!|\!| u - u_h |\!|\!| \leq \widehat{\eta}(u_h, \boldsymbol{y}) \quad \forall u_h \in V, \ \forall \boldsymbol{y} \in \mathbf{H}(\operatorname{div}, \Omega), \tag{18}$$

where

$$\widehat{\eta}(u_h, \boldsymbol{y}) = C_\Omega \left\| f + \operatorname{div} \boldsymbol{y} \right\|_0 + \left\| \boldsymbol{y} - \boldsymbol{\nabla} u_h \right\|_0 . \tag{19}$$

This is another guaranteed upper bound on the energy norm of error. The advantage of $\widehat{\eta}(u_h, \boldsymbol{y})$ in comparison with $\eta(u_h, \boldsymbol{y})$ given by (10) is that the estimate (19) is valid for any $\boldsymbol{y} \in \mathbf{H}(\operatorname{div}, \Omega)$ and the set $\boldsymbol{Q}(f)$, which might be difficult to handle in practice – see Section 6, is not needed. On the other hand evaluation of $\widehat{\eta}(u_h, \boldsymbol{y})$ requires the knowledge of the Friedrichs' constant $C_\Omega$ or of its upper bound.

The error bound $\widehat{\eta}(u_h, \boldsymbol{y})$ – as well as $\eta(u_h, \boldsymbol{y})$ – is sharp in the sense that the gradient of the exact solution $\boldsymbol{y} = \boldsymbol{\nabla} u$ yields the error exactly: $\widehat{\eta}(u_h, \boldsymbol{\nabla} u) = \|\!| u - u_h |\!\|$. Notice that the term with $C_\Omega$ in $\widehat{\eta}(u_h, \boldsymbol{y})$ vanishes for $\boldsymbol{y} = \boldsymbol{\nabla} u$. It means that the error bound $\widehat{\eta}(u_h, \boldsymbol{y})$ can provide sharp results even if the Friedrichs' constant $C_\Omega$ is estimated very roughly.

However, from the point of the theory, the results of Theorems 3–6 are not valid for $\widehat{\eta}(u_h, \boldsymbol{y})$, in general. Moreover, the quantity $\widehat{\eta}^2(u_h, \boldsymbol{y})$ is not a quadratic functional in $\boldsymbol{y}$, any more. Nevertheless, there is a way how to transform it into a quadratic one. Introducing a real parameter $\beta > 0$, we can estimate $\widehat{\eta}^2(u_h, \boldsymbol{y})$ in an elementary way as

$$\widehat{\eta}(u_h, \boldsymbol{y}) \le \widehat{\eta}_\beta(u_h, \boldsymbol{y}) \quad \forall \beta > 0,\ \forall u_h \in V,\ \forall \boldsymbol{y} \in \mathbf{H}(\mathrm{div}, \Omega),$$

where

$$\widehat{\eta}^2_\beta(u_h, \boldsymbol{y}) = \left(1 + \beta^{-1}\right) C_\Omega^2 \left\| f + \operatorname{div} \boldsymbol{y} \right\|_0^2 + (1 + \beta) \left\| \boldsymbol{y} - \boldsymbol{\nabla} u_h \right\|_0^2$$

is already quadratic in $\boldsymbol{y}$. Notice that there is always a suitable value of $\beta$ such that $\widehat{\eta}_\beta(u_h, \boldsymbol{y}) = \widehat{\eta}(u_h, \boldsymbol{y})$.

In principle we should minimize $\widehat{\eta}_\beta(u_h, \boldsymbol{y})$ simultaneously with respect to $\boldsymbol{y}$ and $\beta$. This general nonlinear minimization problem might be difficult to solve. Anyway, for fixed $u_h \in V$ and fixed $\beta > 0$ the quadratic functional $\widehat{\eta}^2_\beta(u_h, \boldsymbol{y})$ can be minimized in a standard way. If we consider the minimization problem

$$\text{find } \boldsymbol{y} \in \mathbf{H}(\mathrm{div}, \Omega) : \quad \widehat{\eta}_\beta(u_h, \boldsymbol{y}) \le \widehat{\eta}_\beta(u_h, \boldsymbol{w}) \quad \forall \boldsymbol{w} \in \mathbf{H}(\mathrm{div}, \Omega),$$

we find as above that it is equivalent to the problem

$$\text{find } \boldsymbol{y} \in \mathbf{H}(\mathrm{div}, \Omega) : \quad \widehat{\mathcal{B}}(\boldsymbol{y}, \boldsymbol{w}) = \widehat{\mathcal{F}}(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \mathbf{H}(\mathrm{div}, \Omega), \tag{20}$$

where

$$\begin{aligned} \widehat{\mathcal{B}}(\boldsymbol{y}, \boldsymbol{w}) &= (\operatorname{div} \boldsymbol{y}, \operatorname{div} \boldsymbol{w}) + \beta C_\Omega^{-2} (\boldsymbol{y}, \boldsymbol{w}), \\ \widehat{\mathcal{F}}(\boldsymbol{w}) &= (-f, \operatorname{div} \boldsymbol{w}) + \beta C_\Omega^{-2} (\boldsymbol{\nabla} u_h, \boldsymbol{w}). \end{aligned}$$

Notice that the upper bound $\widehat{\eta}(u_h, \boldsymbol{y})$ is more general than the upper bound $\eta(u_h, \boldsymbol{y})$ in the sense that $\eta(u_h, \boldsymbol{y})$ can be derived from $\widehat{\eta}(u_h, \boldsymbol{y})$. Indeed, $\widehat{\eta}(u_h, \boldsymbol{y}) = \eta(u_h, \boldsymbol{y})$ for all $\boldsymbol{y} \in \boldsymbol{Q}(f)$.

## 6 Practical computation of the complementary solution

The practical handling of the affine space $\boldsymbol{Q}(f)$ defined in (7) might be difficult in general. Here, we present a possible approach from [11]. For simplicity, we consider two dimensions only, i.e., $d = 2$.

First of all, we exploit the affine structure of $\boldsymbol{Q}(f)$. Any vector field $\boldsymbol{w} \in \boldsymbol{Q}(f)$, can be expressed as $\boldsymbol{w} = \overline{\boldsymbol{q}} + \boldsymbol{w}^0$, where $\overline{\boldsymbol{q}} \in \boldsymbol{Q}(f)$ is fixed and $\boldsymbol{w}^0$ lies in a linear space $\boldsymbol{Q}(0)$ of divergence-free vector fields. If an antiderivative of $f = f(x_1, x_2)$ with respect to one of its variables is known, we can construct $\overline{\boldsymbol{q}}$ for example as

$$\overline{\boldsymbol{q}}(x_1, x_2) = -\left(\int_0^{x_1} f(s, x_2)\,\mathrm{d}s, 0\right)^T. \tag{21}$$

Further, if the domain $\Omega$ is simply connected, then for any $\boldsymbol{w}^0 \in \boldsymbol{Q}(0)$ exists $v \in H^1(\Omega)$ such that $\boldsymbol{w}^0 = \mathbf{curl}\, v$, where $\mathbf{curl}\, v = (\partial v/\partial x_2, -\partial v/\partial x_1)^T$ is understood in the weak sense, see e.g. [11]. All together, any $\boldsymbol{w} \in \boldsymbol{Q}(f)$ can be expressed as $\boldsymbol{w} = \overline{\boldsymbol{q}} + \mathbf{curl}\, v$ for a $v \in H^1(\Omega)$. In terms of spaces, we can write

$$\boldsymbol{Q}(f) = \overline{\boldsymbol{q}} + \mathbf{curl}\, H^1(\Omega).$$

This structure enables to reformulate the complementary problem (13) as follows:

$$\text{find } z \in H^1(\Omega): \quad \mathcal{B}^*(\mathbf{curl}\, z, \mathbf{curl}\, v) = -\mathcal{B}^*(\overline{\boldsymbol{q}}, \mathbf{curl}\, v) \quad \forall v \in H^1(\Omega). \tag{22}$$

The corresponding complementary solution is then $\boldsymbol{y} = \overline{\boldsymbol{q}} + \mathbf{curl}\, z$. If we notice that $\mathcal{B}^*(\mathbf{curl}\, z, \mathbf{curl}\, v) = \mathcal{B}(z, v)$, problem (22) actually turns into the Poisson problem:

$$\text{find } z \in H^1(\Omega): \quad \mathcal{B}(z, v) = -\mathcal{B}^*(\overline{\boldsymbol{q}}, \mathbf{curl}\, v) \quad \forall v \in H^1(\Omega). \tag{23}$$

Let us remark that in contrast to (3), where we have prescribed the Dirichlet boundary conditions, problem (23) is equipped with Neumann boundary conditions. It is a consistent pure Neumann problem. Thus, it has infinitely many solutions and these solutions differ by a constant. Notice, that the actual value of this constant is irrelevant, because we are only interested in $\mathbf{curl}\, z$.

Problem (23) can be approximately solved by any standard numerical method for Poisson equation. For example, we can use the same method as we have used for the approximate solution of (3).

## 7 Generalizations

The complementary approach seems to be quite special. From this point of view it might be surprising that it can be generalized to a wide variety of problems. However, for more complicated problems the complementary upper bounds loose some of their properties, we presented in Theorems 3–6.

Generalization of the complementary approach for diffusion-reaction equation

$$-\Delta u + \kappa^2 u = f$$

is of particular interest, because it requires an additional idea, see e.g. [2, 5, 10, 20, 23, 24, 25]. We will not describe it here in detail, we only introduce the resulting upper bound:

$$|||u - u_h||| \le \eta(u_h, \boldsymbol{y}) \quad \forall u_h \in V,\ \forall \boldsymbol{y} \in \mathbf{H}(\mathrm{div}, \Omega),$$

where

$$\eta(u_h, \boldsymbol{y})^2 = \|\boldsymbol{y} - \boldsymbol{\nabla} u_h\|_0^2 + \left\|\kappa^{-1}(f - \kappa^2 u_h + \operatorname{div} \boldsymbol{y})\right\|_0^2. \tag{24}$$

We point out that this upper bound cannot be used for the Poisson problem, i.e. for $\kappa = 0$. However, in the singularly perturbed case, i.e., for large values of $\kappa$, this upper bound provides very sharp results. In addition, for the upper bound (24) we can prove analogues of Theorems 3–6, see [25].

The presented complementary approaches (10), (19), and (24) can be generalized in more or less straightforward way to general linear elliptic problems with anisotropic diffusion, convection and reaction terms, equipped with a combination of Dirichlet, Neumann, and Robin boundary conditions. They can be generalized even to systems of such elliptic equations [26].

Nevertheless, the complementary approach is not limited to elliptic problems only. It has been generalized to linear elasticity [14], to system of thermo-elasticity [13], to stationary Navier-Stokes problem [17], to variational inequalities [7], to certain nonlinear problems [18], to equations with the curl operator [3], etc.

The complementary approach of error majorants for most of these problems is well described in the book [19]. The book [16] is devoted more to the general theory and derivation of the complementary error bounds based on the calculus of variation.

## 8 Numerical examples

In this section we present a few numerical examples showing the performance of the variants of the complementary upper bounds in the finite element method.

In these experiments, we consider the two-dimensional case (i.e. $d = 2$), polygonal domain $\Omega$, a triangular finite element mesh $\mathcal{T}_h$ in $\Omega$, and a space of continuous and piecewise linear functions in $\Omega$:

$$V_h = \{v_h \in V : v_h|_K \in P^1(K),\ \forall K \in \mathcal{T}_h\},$$

where $P^1(K)$ stands for the space of linear functions on the triangle $K$. The finite element solution of (3) is then $u_h \in V_h$ such that

$$\mathcal{B}(u_h, v_h) = \mathcal{F}(v_h) \quad \forall v_h \in V_h.$$

First, we use the error estimate $\eta(u_h, \boldsymbol{y}_h)$ given by (9)–(10). The approximate complementary solution $\boldsymbol{y}_h$ is computed as $\boldsymbol{y}_h = \overline{\boldsymbol{q}} + \mathbf{curl}\, z_h$, where $\overline{\boldsymbol{q}}$ is given by (21) and $z_h$ is obtained as the finite element solution of problem (23). More precisely, we introduce a space

$$X_h = \{v_h \in H^1(\Omega) : v_h|_K \in P^p(K),\ \forall K \in \mathcal{T}_h\}$$

of piecewise polynomials of degree at most $p$ over the same mesh $\mathcal{T}_h$ and define $z_h \in X_h$ such that

$$\mathcal{B}(z_h, v_h) = -\mathcal{B}^*(\overline{\boldsymbol{q}}, \mathbf{curl}\, v_h) \quad \forall v_h \in X_h.$$

In the experiments presented below we compare the values of $\eta(u_h, \boldsymbol{y}_h)$ for $p = 1$, $p = 2$, and $p = 3$.

As an alternative, we use the error bound $\widehat{\eta}(u_h, \widehat{\boldsymbol{y}}_h)$ given by (18)–(19). The corresponding approximate complementary solution $\widehat{\boldsymbol{y}}_h$ is computed as an approximate solution $\widehat{\boldsymbol{y}}_h$ of problem (20). The best results are obtained for small values of $\beta$, because the smaller the value of $\beta$ is, the more the complementary solution is enforced to satisfy $-\operatorname{div} \boldsymbol{y}_h = f$. In the example, we use $\beta = C_\Omega^2 \, 10^{-4}$. To solve the complementary problem (20) approximately, we use the Raviart-Thomas finite elements of degree $\widehat{p} = 1$ and $\widehat{p} = 2$ on the same mesh $\mathcal{T}_h$.

Finally, for comparison, we present results of $\eta(u_h, \boldsymbol{y}_h^{\text{expl}})$, see (9)–(10), where $\boldsymbol{y}_h^{\text{expl}}$ is obtained by a quite complicated but explicit formula from [2]. This formula is based on the so-called equilibrated residuals [1] and the approach utilizes the trick of so-called data oscillations.

In particular, we consider two specific examples. In the first example, the Poisson problem (1)–(2) is defined in a square $\Omega = (-1/2, 1/2)^2$ with the right-hand side $f(x_1, x_2) = \cos(\pi x_1)\cos(\pi x_2)$. The corresponding exact solution is then $u = \cos(\pi x_1)\cos(\pi x_2)/(2\pi^2)$. To use the error bound $\widehat{\eta}(u_h, \widehat{\boldsymbol{y}}_h)$, we estimate the Friedrichs' constant by (17) as $C_\Omega = 1/(\pi\sqrt{2})$. The finite element mesh is shown in Figure 2 (left).

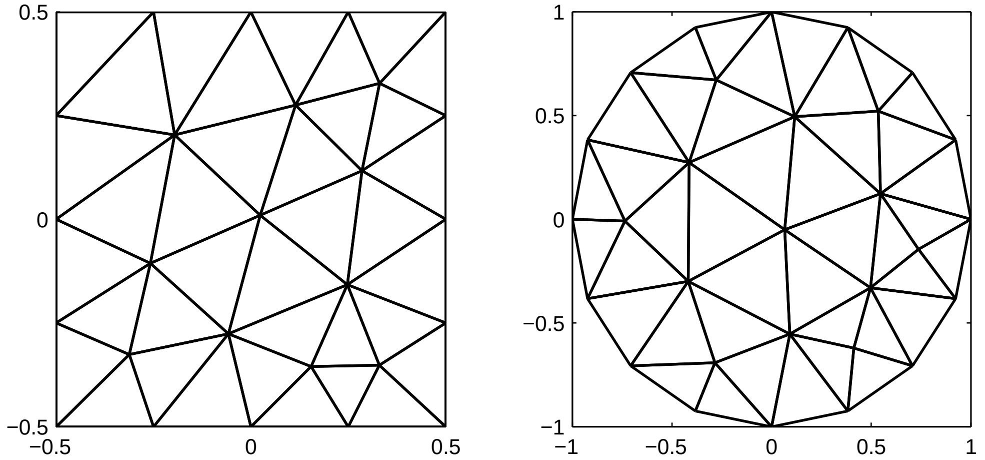


**Fig. 2:** *The finite element mesh used in the first (left) and in the second (right) example.*

In the second example, we solve also the Poisson problem (1)–(2). In this case, the domain $\Omega$ is the unit disk $\Omega = \{(x_1, x_2) : r < 1\}$, where $r^2 = x_1^2 + x_2^2$. The right-hand side is $f = 1$ and the corresponding exact solution is $u = (1 - r^2)/4$. The Friedrichs' constant is estimated as $C_\Omega = \sqrt{2}/\pi$. Figure 2 (right) sketches the used finite element mesh.

Tables 1–2 present the indices of effectivity $I_{\text{eff}}$. It is the ratio of the estimate and the true value of the estimated quantity, for example $I_{\text{eff}} = \widehat{\eta}(u_h, \widehat{\boldsymbol{y}}_h)/|||u - u_h|||$. The first row corresponds to the mesh shown in Figure 2. The subsequent rows correspond to the subsequent uniform refinements of this mesh.

First of all, we do not see any substantial dependence of the values on the mesh refinement. This confirms the correctness of the approach and the correctness of the numerical implementation. Further, we observe that if the complementary problems are solved with the same orders of accuracy, i.e. with $p = 1$ and $\widehat{p} = 1$, then the complementary error bounds provide fair but not absolutely sharp results. They overestimate the error roughly by 40–80 %.

We point out that the number of degrees of freedom (DOFs) needed to compute $z_h$ in the case $p = 1$ is comparable to the number of DOFs needed to compute $u_h$ (i.e. to solve the primal problem). On the other hand, the number of DOFs needed for $\widehat{\boldsymbol{y}}_h$ is roughly six times higher. (There are two DOFs per edge and there is roughly three times more edges than vertices in triangular meshes.)

If we invest more DOFs into the solution of the complementary problem and use quadratic or even cubic finite elements, we obtain almost exact results. However, the solution of the complementary problem then requires much more computational time and such approach is not very practical. A remedy is presented in the last columns of Tables 1–2. They show the results obtained by a fast and explicit approach from [2]. The number of needed arithmetic operations is proportional to the number of DOFs in the primal problem. This is quite sharp and fast alternative.

The kind reader already noticed that certain values in Table 2 are less than one. This seems as a contradiction with Theorem 2 which states that the error estimate is an upper bound on the energy norm of the error. However, Theorem 2 assumes

| | $\eta(u_h, \overline{\boldsymbol{q}} + \mathbf{curl}\, z_h)$ | | | $\widehat{\eta}(u_h, \widehat{\boldsymbol{y}}_h)$ | | $\eta(u_h, \boldsymbol{y}_h^{\mathrm{expl}})$ |
|---|---|---|---|---|---|---|
| | $p = 1$ | $p = 2$ | $p = 3$ | $\widehat{p} = 1$ | $\widehat{p} = 2$ | |
| $h$ | 1.410 | 1.008 | 1.000 | 1.789 | 1.099 | 1.419 |
| $h/2$ | 1.419 | 1.002 | 1.000 | 1.791 | 1.052 | 1.405 |
| $h/4$ | 1.422 | 1.001 | 1.000 | 1.791 | 1.027 | 1.406 |
| $h/8$ | 1.424 | 1.000 | 1.000 | 1.790 | 1.013 | 1.407 |
| $h/16$ | 1.424 | 1.000 | 1.000 | 1.790 | 1.007 | 1.408 |

**Tab. 1:** *Indices of effectivity obtained in the first example.*

| | $\eta(u_h, \overline{\boldsymbol{q}} + \mathbf{curl}\, z_h)$ | | | $\widehat{\eta}(u_h, \widehat{\boldsymbol{y}}_h)$ | | $\eta(u_h, \boldsymbol{y}_h^{\mathrm{expl}})$ |
|---|---|---|---|---|---|---|
| | $p = 1$ | $p = 2$ | $p = 3$ | $\widehat{p} = 1$ | $\widehat{p} = 2$ | |
| $h$ | 1.708 | 1.000 | 0.978 | 1.000 | 0.978 | 1.047 |
| $h/2$ | 1.692 | 1.000 | 0.990 | 1.000 | 0.990 | 1.128 |
| $h/4$ | 1.686 | 1.000 | 0.995 | 1.000 | 0.995 | 1.153 |
| $h/8$ | 1.683 | 1.000 | 0.998 | 1.000 | 0.998 | 1.158 |
| $h/16$ | 1.683 | 1.000 | 0.999 | 1.000 | 0.999 | 1.158 |

**Tab. 2:** *Indices of effectivity obtained in the second example.*

both $u$ and $u_h$ to be defined in the same domain $\Omega$, but in the second example we actually approximate the circular domain $\Omega$ by a polygon $\Omega_h$. Thus, strictly speaking the assumptions of Theorem 2 are not satisfied. Anyway, if we refine the mesh and use more precise approximation of the circular domain, we should obtain sharper results. In Table 2, we observe that this is indeed the case.

## 9 Conclusions

In this paper we surveyed the complementary approach yielding the computable and guaranteed upper bounds of the energy norm of error. A straightforward implementation of the complementary error bounds is computationally too expensive for practical purposes. However, there are fast approaches providing sufficiently accurate results.

From the point of view of reliability of numerical computations, the complementary approach is invaluable for its ability to provide computable and guaranteed upper bounds on the error. These errors bounds used in an adaptive algorithm enable to solve the given problem with prescribed accuracy. Up to the author's knowledge there is no available software, capable to solve for instance linear elliptic problems with guaranteed accuracy. The complementary framework provides theoretical background for the creation of such software.

## References

[1] Ainsworth, M. and Oden, J.T.: *A posteriori error estimation in finite element analysis.* John Wiley & Sons, New York, 2000.

[2] Ainsworth, M. and Vejchodský, T.: Fully computable robust a posteriori error bounds for singularly perturbed reaction–diffusion problems. Numer. Math. (2010). Submitted.

[3] Anjam, I., Mali, O., Muzalevsky, A., Neittaanmäki, P., and Repin, S.: A posteriori error estimates for a Maxwell type problem. Russian J. Numer. Anal. Math. Modelling **24** (2009), 395–408.

[4] Aubin, J.P. and Burchard, H.G.: Some aspects of the method of the hypercircle applied to elliptic variational problems. In: *Numerical Solution of Partial Differential Equations, II (SYNSPADE 1970) (Proc. Sympos., Univ. of Maryland, College Park, Md., 1970)*, pp. 1–67. Academic Press, New York, 1971.

[5] Cheddadi, I., Fučík, R., Prieto, M.I., and Vohralík, M.: Guaranteed and robust a posteriori error estimates for singularly perturbed reaction–diffusion problems. M2AN Math. Model. Numer. Anal. **43** (2009), 867–888.

[6] Haslinger, J. and Hlaváček, I.: Convergence of a finite element method based on the dual variational formulation. Apl. Mat. **21** (1976), 43–65.

[7] Hlaváček, I., Haslinger, J., Nečas, J., and Lovíšek, J.: *Solution of variational inequalities in mechanics*, *Applied Mathematical Sciences*, vol. 66. Springer-Verlag, New York, 1988.

[8] Hlaváček, I.: Some equilibrium and mixed models in the finite element method. In: *Mathematical models and numerical methods (Papers, Fifth Semester, Stefan Banach Internat. Math. Center, Warsaw, 1975)*, *Banach Center Publ.*, vol. 3, pp. 147–165. PWN, Warsaw, 1978.

[9] Hlaváček, I. and Křížek, M.: Internal finite element approximations in the dual variational method for second order elliptic problems with curved boundaries. Apl. Mat. **29** (1984), 52–69.

[10] Korotov, S.: Two-sided a posteriori error estimates for linear elliptic problems with mixed boundary conditions. Appl. Math. **52** (2007), 235–249.

[11] Křížek, M.: Conforming equilibrium finite element methods for some elliptic plane problems. RAIRO Anal. Numér. **17** (1983), 35–65.

[12] Mikhlin, S.G.: *Constants in some inequalities of analysis.* John Wiley & Sons., 1986.

[13] Muzalevskiĭ, A.V. and Repin, S.I.: On error estimates for approximate solutions in problems of the linear theory of thermoelasticity. Izv. Vyssh. Uchebn. Zaved. Mat. (2005), 64–72.

[14] Muzalevsky, A.V. and Repin, S.I.: On two-sided error estimates for approximate solutions of problems in the linear theory of elasticity. Russian J. Numer. Anal. Math. Modelling **18** (2003), 65–85.

[15] Nečas, J.: *Les méthodes directes en théorie des équations elliptiques.* Masson et Cie, Éditeurs, Paris, 1967.

[16] Neittaanmäki, P. and Repin, S.: *Reliable methods for computer simulation*, *Studies in Mathematics and its Applications*, vol. 33. Elsevier Science B.V., Amsterdam, 2004.

[17] Repin, S.: On a posteriori error estimates for the stationary Navier-Stokes problem. J. Math. Sci. (N. Y.) **150** (2008), 1885–1889.

[18] Repin, S. and Valdman, J.: Functional a posteriori error estimates for problems with nonlinear boundary conditions. J. Numer. Math. **16** (2008), 51–81.

[19] Repin, S.: *A posteriori estimates for partial differential equations*, *Radon Series on Computational and Applied Mathematics*, vol. 4. Walter de Gruyter GmbH & Co. KG, Berlin, 2008.

[20] Repin, S. and Sauter, S.: Functional a posteriori estimates for the reaction-diffusion problem. C. R. Math. Acad. Sci. Paris **343** (2006), 349–354.

[21] Synge, J.L.: *The hypercircle in mathematical physics: a method for the approximate solution of boundary value problems.* Cambridge University Press, New York, 1957.

[22] Vacek, J.: Dual variational principles for an elliptic partial differential equation. Apl. Mat. **21** (1976), 5–27.

[23] Vejchodský, T.: Fast and guaranteed a posteriori error estimator. In: J. Chleboun, P. Přikryl, and K. Segeth (Eds.), *Programs and Algorithms of Numerical Mathematics 12*, pp. 257–272. Mathematical Institute, Academy of Sciences, Czech Republic, Prague, 2004.

[24] Vejchodský, T.: Guaranteed and locally computable a posteriori error estimate. IMA J. Numer. Anal. **26** (2006), 525–540.

[25] Vejchodský, T.: Complementarity based a posteriori error estimates and their properties. Math. Comput. Simulation (2010). Submitted.

[26] Vejchodský, T.: Complementary error bounds for elliptic systems and applications. Appl. Math. Comput. (2010). Submitted.

[27] Vohralík, M.: A posteriori error estimation in the conforming finite element method based on its local conservativity and using local minimization. C. R. Math. Acad. Sci. Paris **346** (2008), 687–690.

[28] Weisz, J.: On a posteriori error estimate of approximate solutions to a mildly nonlinear nonpotential elliptic boundary value problem. Math. Nachr. **153** (1991), 231–236.

# DERIVATION OF BDF COEFFICIENTS FOR EQUIDISTANT TIME STEP*

Miloslav Vlasák, Zuzana Vlasáková

**Abstract**

We present the derivation of the explicit formulae of BDF coefficients for equidistant time step.

## 1 Introduction

In this paper we deal with the coefficients of *Backward Differential Formulae* (BDF). BDF represent very important scheme for solving stiff ODE's (see [2] and [3]) which can arise from a lot of important practical tasks, see e.g. [1]. For survey on solving stiff problems see [5]. In this paper we present the order conditions for the coefficients of BDF which can be viewed as some linear system of equations, formulate the explicit relations for the BDF coefficients and show that these relations for BDF coefficients represent the solution of this system for arbitrary order. Advantage of our approach is that we use only simple arithmetic means without differentiation. In fact, the differentiation is hidden in derivation of the order conditions, where Taylor expansions are used.

## 2 BDF and order conditions

We consider $y \in C^1(0,T)$ the solution of ordinary differential equation

$$\begin{aligned} y'(t) &= F(t, y(t))), \quad \forall t \in (0,T), \\ y(0) &= A \in R. \end{aligned} \tag{1}$$

We assume the equidistant partition $t_m = m\tau$, $m = 0, \dots, r$ of interval $[0,T]$ with discretization step $\tau = T/r$. We denote $y^m$ the approximation to the exact solution $y(t_m)$. The difference equation

$$\sum_{v=0}^{k} \alpha_v y^{m+v} = \tau F(t_{m+k}, y^{m+k}), \tag{2}$$

where $\alpha_v$ are some suitable real constants, we call Backward Differential Formula (BDF). We call the method (2) $k$–step BDF, if $\alpha_k \neq 0$ and $\alpha_0 \neq 0$.

*This work was supported by grant No. 10209/B-MAT/MFF of the Grant Agency of Charles University.

Now we formulate the order conditions. We say that BDF has order $p \geq 0$, if

$$\sum_{v=0}^{k} \alpha_v v^s = s k^{s-1} \tag{3}$$

for $s = 0, \ldots, p$. The order conditions for general linear multistep method including the proof can be found in e.g. [4].

**Theorem 1.** *Let $k \geq 1$. Then there exists only one $k$-step BDF of order $k$ and this method has the coefficients*

$$\alpha_v \equiv (-1)^{k-v} \binom{k}{v} \frac{1}{k-v}, \quad v = 0, \ldots, k-1, \tag{4}$$

$$\alpha_k \equiv \sum_{v=1}^{k} \frac{1}{v}. \tag{5}$$

## 3 Proof of Theorem

It is simple to see that (3) represents linear system of Vandermonde type (which is obviously nonsingular), if $p = k$.

**Lemma 1.** *The system of equations (3) is equivalent to the system:*

$$\sum_{v=0}^{k} \alpha_v = 0 \tag{6}$$

$$\sum_{v=0}^{k} \alpha_v (k-v) = -1 \tag{7}$$

$$\sum_{v=0}^{k} \alpha_v (k-v)^s = 0 \tag{8}$$

*for $s = 2, \ldots, k$.*

*Proof.* We can see that matrix represented by the system (6)–(8) is nonsingular, because similarly as in the previous case the matrix is of Vandermonde type. At first we will prove that (7) and (8) follows from (3). First equation (6) is one of the equations of (3). Second equation (7) we can divide into two parts and enumerate them by (3):

$$\sum_{v=0}^{k} \alpha_v (k-v) = k \sum_{v=0}^{k} \alpha_v - \sum_{v=0}^{k} \alpha_v v = 0 - 1 = -1. \tag{9}$$

The remaining equations (8) are proved using (3):

$$\sum_{v=0}^{k}\alpha_v(k-v)^s=\sum_{v=0}^{k}\alpha_v\sum_{i=0}^{s}\binom{s}{i}(-1)^i k^{s-i}v^i=\sum_{i=0}^{s}(-1)^i\binom{s}{i}k^{s-i}\sum_{v=0}^{k}\alpha_v v^i \tag{10}$$
$$=\sum_{i=0}^{s}(-1)^i\binom{s}{i}k^{s-i}ik^{i-1}=k^{s-1}\sum_{i=0}^{s}(-1)^i\frac{s!}{i!(s-i)!}i$$
$$=k^{s-1}\sum_{i=1}^{s}(-1)^i\frac{s!}{i!(s-i)!}i=-k^{s-1}\sum_{i=1}^{s}(-1)^{i-1}\frac{s(s-1)!}{(i-1)!(s-1-(i-1))!}$$
$$=-sk^{s-1}\sum_{i=0}^{s-1}\binom{s-1}{i}(-1)^i=-sk^{s-1}(1-1)^{s-1}=0$$

for $s\geq 2$. Now we will prove that (3) follows from (6), (7) and (8). The equation (3) for $s=0$ is exactly (6). The rest we will prove by induction. As first step we will prove that (3) holds for $s=1$.

$$-\sum_{v=0}^{k}\alpha_v(k-v)=-k\sum_{v=0}^{k}\alpha_v+\sum_{v=0}^{k}\alpha_v v=\sum_{v=0}^{k}\alpha_v v=1=sk^{s-1} \tag{11}$$

Then let us assume that (3) holds for $i=1,\ldots,s-1$. Then

$$0=\sum_{v=0}^{k}\alpha_v(k-v)^s=\sum_{v=0}^{k}\alpha_v\sum_{i=0}^{s}(-1)^i\binom{s}{i}k^{s-i}v^i \tag{12}$$
$$=\sum_{i=0}^{s}(-1)^i\binom{s}{i}k^{s-i}\sum_{v=0}^{k}\alpha_v v^i=(-1)^s\sum_{v=0}^{k}\alpha_v v^s+\sum_{i=0}^{s-1}(-1)^i\binom{s}{i}k^{s-i}\sum_{v=0}^{k}\alpha_v v^i$$

With the induction assumptions we will get from the second term:

$$\sum_{i=0}^{s-1}(-1)^i\binom{s}{i}k^{s-i}\sum_{v=0}^{k}\alpha_v v^i=\sum_{i=0}^{s-1}(-1)^i\binom{s}{i}k^{s-i}ik^{i-1} \tag{13}$$
$$=k^{s-1}\sum_{i=0}^{s-1}(-1)^i\binom{s}{i}i=-(-1)^s sk^{s-1}+k^{s-1}\sum_{i=0}^{s}(-1)^i\binom{s}{i}i$$

Now it is sufficient to show that

$$\sum_{i=0}^{s}(-1)^i\binom{s}{i}i=\sum_{i=1}^{s}(-1)^i\binom{s}{i}i=\sum_{i=1}^{s}(-1)^i\frac{s!}{i!(s-i)!}i \tag{14}$$
$$=-s\sum_{i=1}^{s}(-1)^{i-1}\frac{(s-1)!}{(i-1)!(s-1-(i-1))!}=-s\sum_{i=1}^{s}(-1)^{i-1}\binom{s}{i}$$
$$=-s\sum_{i=0}^{s-1}(-1)^i\binom{s-1}{i}=-s(1-1)^{s-1}=0$$

for $s\geq 2$. From (12), (13) and (14) follows (3). □

**Lemma 2.** *Let $\alpha_v$ are the coefficients of $k$-step BDF of order $k \geq 2$. Then*

$$\alpha_v = (-1)^{k-v} \binom{k}{v} \frac{1}{k-v} \tag{15}$$

*for $v = 0, \ldots, k-1$.*

*Proof.* Because we have shown in Lemma 1 that system of equations (3) is equivalent to system (6), (7) and (8) and since $\alpha_k$ depends only on (6) we can prove our lemma by substituting to (7) and (8). When we substitute to (7) we get by binomial theorem

$$\begin{aligned} \sum_{v=0}^{k} \alpha_v (k-v) = \sum_{v=0}^{k-1} \alpha_v (k-v) = \sum_{v=0}^{k-1} (-1)^{k-v} \binom{k}{v} \frac{1}{k-v} (k-v) \\ = \sum_{v=0}^{k-1} (-1)^{k-v} \binom{k}{v} = \sum_{v=0}^{k} (-1)^{k-v} \binom{k}{v} - 1 = (1-1)^k - 1 = -1, \end{aligned} \tag{16}$$

for $k \geq 1$. Now we will prove the rest by induction. We denote $\alpha_i^j$ the coefficient $\alpha_i$ of BDF of order $j$. As the first step we will prove that our $\alpha_v^j$ satisfies (8) for $s = 2$, $2 \leq j \leq k$:

$$\begin{aligned} \sum_{v=0}^{j} \alpha_v^j (j-v)^s = \sum_{v=0}^{j-1} \alpha_v^j (j-v)^2 = \sum_{v=0}^{j-1} (-1)^{j-v} \binom{j}{v} \frac{1}{j-v} (j-v)^2 \\ = \sum_{v=0}^{j-1} -(-1)^{j-1-v} \frac{j(j-1)!}{v!(j-1-v)!} \\ = -j \sum_{v=0}^{j-1} (-1)^{j-1-v} \binom{j-1}{v} = -j(1-1)^{j-1} = 0 \end{aligned} \tag{17}$$

Now let us assume that $\alpha_v^j$ satisfies (8) for $j = 2, \ldots, k-1$. Now we want to prove that $\alpha_v^k$ satisfies (8) for $2 \leq s \leq k$. We know that it holds for $s = 2$. We will assume that it holds for $s - 1$. From this follows

$$\begin{aligned} \sum_{v=0}^{k} \alpha_v^k (k-v)^s = k \sum_{v=0}^{k} \alpha_v^k (k-v)^{s-1} - \sum_{v=0}^{k} \alpha_v^k (k-v)^{s-1} v \\ = 0 - \sum_{v=1}^{k-1} \alpha_v^k (k-v)^{s-1} v = - \sum_{v=1}^{k-1} (-1)^{k-v} \binom{k}{v} (k-v)^{s-2} v \\ = - \sum_{v=1}^{k-1} (-1)^{k-1-(v-1)} \frac{k(k-1)!}{(v-1)!(k-1-(v-1))!} (k-1-(v-1))^{s-2} \\ = -k \sum_{v=0}^{k-2} (-1)^{k-1-v} \frac{(k-1)!}{v!(k-1-v)!} (k-1-v)^{s-2} \\ = -k \sum_{v=0}^{k-2} \alpha_v^{k-1} (k-1-v)^{s-1} = 0 \end{aligned} \tag{18}$$

□

**Lemma 3.** *Let $\alpha_v$ are the coefficients of $k$-step BDF of order $k \geq 2$. Then*

$$\alpha_k = \sum_{v=1}^{k} \frac{1}{v} \tag{19}$$

*Proof.* We will use the notation $\alpha_v^j$ for $\alpha_v$ of the BDF of order $j$. It is easy to compute that $\alpha_1^1$ and $\alpha_2^2$ satisfy our lemma. Now we want to show that $\alpha_{k+1}^{k+1} - \alpha_k^k = \frac{1}{k+1}$, which proves our lemma. From (6) follows

$$\begin{aligned}
\alpha_{k+1}^{k+1} &= -\sum_{v=0}^{k} \alpha_v^{k+1} = -\sum_{v=0}^{k} (-1)^{k+1-v} \binom{k+1}{v} \frac{1}{k+1-v} \qquad (20)\\
&= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=1}^{k} (-1)^{k+1-v} \binom{k+1}{v} \frac{1}{k+1-v}\\
&= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=1}^{k} (-1)^{k-(v-1)} \frac{1}{v} \frac{(k+1)k!}{(v-1)!(k-(v-1))!} \frac{1}{k-(v-1)}\\
&= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=0}^{k-1} (-1)^{k-v} \binom{k}{v} \frac{1}{k-v} \frac{k+1}{v+1}\\
&= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=0}^{k-1} \alpha_v^k \frac{k+1}{v+1}
\end{aligned}$$

Now we can compute $\alpha_{k+1}^{k+1} - \alpha_k^k$. From (20) and $\alpha_k^k = -\sum_{v=0}^{k-1} \alpha_v^k$ we get

$$\begin{aligned}
\alpha_{k+1}^{k+1} - \alpha_k^k &= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=0}^{k-1} \alpha_v^k \left(\frac{k-v}{v+1}\right) \qquad (21)\\
&= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=0}^{k-1} (-1)^{k-v} \binom{k}{v} \frac{1}{k-v} \left(\frac{k-v}{v+1}\right)\\
&= -(-1)^{k+1}\frac{1}{k+1} - \sum_{v=0}^{k-1} (-1)^{k-v} \frac{k!}{(v+1)!(k-v)!}\\
&= -\frac{1}{k+1}\left((-1)^{k+1} + \sum_{v=0}^{k-1} (-1)^{k+1-(v+1)} \frac{(k+1)!}{(v+1)!(k+1-(v+1))!}\right)\\
&= -\frac{1}{k+1}\left((-1)^{k+1} + \sum_{v=1}^{k} (-1)^{k+1-v} \binom{k+1}{v}\right)\\
&= -\frac{1}{k+1} \sum_{v=0}^{k} (-1)^{k+1-v} \binom{k+1}{v} = \frac{1}{k+1} - \frac{1}{k+1} \sum_{v=0}^{k+1} (-1)^{k+1-v} \binom{k+1}{v}\\
&= \frac{1}{k+1} - \frac{1}{k+1}(-1+1)^{k+1} = \frac{1}{k+1}
\end{aligned}$$

□

We can verify by simple calculation that Lemma 2 and Lemma 3 hold for $k = 1$, too.

## References

[1] Dolejší, V. and Vlasák, M.: Analysis of a BDF-DGFE scheme for nonlinear convection-diffusion problems. Numer. Math. **110** (2008), 405–447.

[2] Gear, C.W.: The automatic integartionof ordinary differential equations. Communications of the ACM **14** (1971), 176–179.

[3] Gear, C.W.: *Numerical initial value problems in ordinary differential equations.* Prentice–Hall, Inc., Englewood Cliffs, N.J., 1971.

[4] Hairer, E., Norsett, S.P., and Wanner, G.: Solving ordinary differential equations I, Nonstiff problems. *Number 8 in Springer Series in Computational Mathematics.* Springer Verlag, 2000.

[5] Hairer, E. and Wanner, G.: *Solving ordinary differential equations II, Stiff and differential-algebraic problems.* Springer Verlag, 2002.

# LIMITED-MEMORY VARIABLE METRIC METHODS THAT USE QUANTITIES FROM THE PRECEDING ITERATION*

Jan Vlček, Ladislav Lukšan

## 1. Introduction

In this contribution, a new family of globally convergent limited-memory (LM) variable metric (VM) line search methods for unconstrained minimization is presented. Numerical results indicate that the new methods can save computational time substantially for certain problems in comparison with the well-known L-BFGS method, see [3], [8].

VM or quasi-Newton line search methods, see [2], [4], start with an initial point $x_0 \in \mathcal{R}^N$ and generate iterations $x_{k+1} \in \mathcal{R}^N$ by the process $x_{k+1} = x_k + s_k$, $s_k = t_k d_k$, $k \ge 0$, where $d_k$ is the direction vector and $t_k > 0$ is a stepsize.

It is assumed that the problem function $f : \mathcal{R}^N \to \mathcal{R}$ is differentiable and stepsize $t_k$ is chosen in such a way that

$$f_{k+1} - f_k \le \varepsilon_1 t_k g_k^T d_k, \qquad g_{k+1}^T d_k \ge \varepsilon_2 g_k^T d_k, \tag{1}$$

$k \ge 0$, where $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$ and $d_k = -H_k g_k$ with a symmetric positive definite matrix $H_k$; usually $H_0$ is a multiple of $I$ and $H_{k+1}$ is obtained from $H_k$ by a rank-two VM update to satisfy the quasi-Newton condition $H_{k+1} y_k = s_k$ (see [2], [4]), where $y_k = g_{k+1} - g_k$, $k \ge 0$. For $i \ge 0$ we denote

$$b_i = s_i^T y_i, \qquad V_i = I - (1/b_i) s_i y_i^T$$

(note that $s_i^T y_i > 0$ for $g_i \ne 0$ by (1)). To simplify the notation we frequently omit index $k$ and replace index $k+1$ by symbol $+$ and index $k-1$ by symbol $-$.

The L-BFGS method (LM variant of the well-known BFGS method, see [3], [8]) is based on the following quasi-product form of the BFGS update

$$H_+ = (1/b) s s^T + V H V^T. \tag{2}$$

The advantage of this form consists in the fact that only the last $\tilde{m}+1 = \min(k+1, m)$ couples $\{s_i, y_i\}_{i=k-\tilde{m}}^k$, where $m \ge 1$ is a given parameter, are stored to compute the direction vector $d_{k+1} = -H_{k+1} g_{k+1}$ by the Strang recurrences, see [8]. Matrices $H_{k+1}$ are not computed, only defined by $H_{k+1} = H_{\tilde{m}+1}^{k+1}$, $k \ge 0$, where

$$\begin{aligned} H_0^{k+1} &= (b_k/|y_k|^2) I, & (3)\\ H_{i+1}^{k+1} &= (1/b_j) s_j\, s_j^T + V_j H_i^{k+1} V_j^T, \quad j = k - \tilde{m} + i, \quad 0 \le i \le \tilde{m}\,. & (4)\end{aligned}$$

*This work was supported by the Grant Agency of the Czech Republic, project No. 201/09/1957, and the Institutional research plan No. AV0Z10300504.

Note that matrix $H_k$, which satisfies $d_k = -H_k g_k$, is different from matrix $H_{\tilde{m}}^{k+1}$ in the last update (4) in general; among others since matrix $H_k$ is created by updating of matrix $H_0^k = (b_{k-1}/|y_{k-1}|^2)I$, not $H_0^{k+1} = (b_k/|y_k|^2)I$. Thus $H_{\tilde{m}}^{k+1} g_k \neq -d_k$ generally.

The Strang recurrences cannot be used directly for other updates from the Broyden class (see [2], [4]) than for the BFGS update (but another efficient approach is possible, see [6]). Some generalizations of the L-BFGS method are investigated in [9]. Here we focus on the approach which uses quantities from the preceding iteration.

Note that our methods do not belong to the Broyden class and has some common features with the multi-step quasi-Newton methods (see e.g. [7]).

We describe the new class of VM updates in Section 2 and the corresponding algorithm in Section 3; global convergence is treated in Section 4 and numerical results are reported in Section 5. Details and proofs of assertions can be found in [9].

## 2. The new class of methods

The Broyden class updates except for the BFGS update need calculate vector $Hy$ in every iteration. This drawback can be eliminated by utilization of the quasi-Newton condition $Hy_- = s_-$. Although it is not satisfied in LM case, in this way we can construct efficient methods that use the same number of stored vectors and matrix by vector multiplications as the L-BFGS method, see Section 3.

**Theorem 2.1.** *Let matrix $H$ be symmetric positive definite, $Hy_- = s_-$, $\sigma \in (-1, 1)$, $\bar{s} = s - \sigma\sqrt{b/b_-}\, s_-$, $\bar{y} = y - \sigma\sqrt{b/b_-}\, y_-$, $\bar{b} = \bar{s}^T y \neq 0$ and $\bar{\varrho} = (1 - \sigma^2)\, b/\bar{b}$. Then update $H_+^{NB}$ with parameter $\sigma$ given by*

$$H_+^{NB} = (\bar{\varrho}/\bar{b})\, \bar{s}\bar{s}^T + \bar{V} H \bar{V}^T, \qquad \bar{V} = I - (1/\bar{b})\, \bar{s}\bar{y}^T, \tag{5}$$

*is positive definite and satisfies the quasi-Newton condition $H_+^{NB} y = s$ (for $\sigma = 0$ we obtain the BFGS update and assumption $Hy_- = s_-$ can be omitted). If $\sigma = s^T y_-/\sqrt{bb_-}$ then $\bar{s}^T y_- = 0$, $\bar{b} = \bar{s}^T \bar{y}$ and if also $\sigma \in (-1, 1)$ and $\bar{b} > 0$, then (5) represents the generalized BFGS update with nonquadratic correction parameter $\bar{\varrho}$ (see [4]), with vectors $s$ and $y$ replaced by $\bar{s}$, $\bar{y}$. If $\sigma = s_-^T y/\sqrt{bb_-}$ then $s_-^T \bar{y} = 0$ and $\bar{\varrho} = 1$.*

Our numerical experiments indicate that convergence is significantly deteriorated when $|\sigma| \to 1$ and that all values $\sigma$ satisfying $|\sigma| \leq 1/2$ with a suitable sign (Theorem 2.1 and Lemma 2.1 motivate us to use the sign of $s^T y_-$) give very good results.

**Lemma 2.1.** *Let $Hy_- = s_-$ and $f$ be quadratic function $f(x) = \frac{1}{2}(x - x^*)^T G (x - x^*)$, $x^* \in \mathcal{R}^N$, with a symmetric positive definite matrix $G$. If vectors $s$, $s_-$ are linearly independent and update $H_+^{NB}$ of matrix $H$ is given by (5) then choice $\sigma = s^T y_-/\sqrt{bb_-}$ (or equivalently $\sigma = s_-^T y/\sqrt{bb_-}$) satisfies $\bar{b} > 0$, $\sigma \in (-1, 1)$, $\bar{\varrho} = 1$ and $H_+^{NB} y_- = s_-$.*

Note that we need not calculate value $s^T y_-$. We use only the sign of $s^T y_-$, therefore in view of the following lemma we can utilize the value $s_-^T g$, computed during the line search procedure, in spite of the fact that assumption $d = -Hg$ is not appropriate to LM updates, see Section 1. In Section 3 we describe a choice of the sign of $\sigma$ in details.

**Lemma 2.2.** *Let $H$ be nonsingular matrix, $Hy_- = s_-$. If $d = -Hg$ then $s^T y_- = -ts_-^T g$.*

Taking into account Theorem 2.1 and Lemma 2.1, we will choose such parameter $\sigma \in (-1,1)$ that corresponding $\bar{b}$ is positive and not too small in comparison with $b$ in a sense that $\bar{b} \equiv b(1 - \sigma\, s_-^T y/\sqrt{bb_-}\,) \geq b(1-\lambda)$, $\lambda \in (0,1)$, which is equivalent to $\sigma\, s_-^T y \leq \lambda\sqrt{bb_-}$. The following lemma shows that in case that $\bar{b} < b(1-\lambda)$ for some $\sigma \in (-1,1)$, we can replace this $\sigma$ by a more appropriate value.

**Lemma 2.3.** *Let $\sigma\, s_-^T y > \lambda\sqrt{bb_-}$ for some $\lambda \in (0,1)$. Then $s_-^T y \neq 0$ and value $\hat{\sigma} = \lambda\sqrt{bb_-}/|s_-^T y| > 0$ satisfies $\pm\hat{\sigma}\, s_-^T y \leq \lambda\sqrt{bb_-}$ (for both signs) and $\hat{\sigma} < |\sigma|$.*

## 3. Implementation

Here we give the procedure based on Section 2. We define matrices $H_0^{k+1}$ and $H_{k+1} = H_{\tilde{m}+1}^{k+1}$, $\tilde{m} = \min(k, m-1)$, $m \geq 1$, $k \geq 0$, by relations similar to (3), (4). Instead of matrices $H_k$, only $\tilde{m}+1 \leq m$ couples of vectors are stored here to compute the direction vector $d_{k+1} = -H_{k+1} g_{k+1}$, using the Strang recurrences, see [8], with a little modification - using transformed nonquadratic correction parameter $\bar{\varrho}$, see [4].

We choose the sign of $\sigma$ in accordance with the sign of $-ts_-^T g \approx s^T y_-$, see Lemma 2.2 and Theorem 2.1. Since $s^T y_- = s_-^T y$ for $f$ quadratic, see Lemma 2.1, we prefer the sign of $s_-^T y$ in case that $|ts_-^T g|$ is too small in comparison with $|s_-^T y|$ (constant 20 in Step 2 was found empirically). Using Lemma 2.3, we bound $|\sigma|$ to have $\bar{b}$ not too small, compared with $b$. For simplicity, we omit stopping criteria.

### Algorithm 3.1

*Data:* The number $m$ of VM updates per iteration, upper bound $\overline{\sigma} \in (0,1)$ for $|\sigma_k|$, safeguard parameter $\lambda \in (0,1)$ and line search parameters $\varepsilon_1$ and $\varepsilon_2$, $0 < \varepsilon_1 < \frac{1}{2}$, $\varepsilon_1 < \varepsilon_2 < 1$.

*Step 0:* *Initiation.* Choose the starting point $x_0 \in \mathcal{R}^N$, define direction vector $d_0 = -g_0$ and initiate iteration counter $k$ to zero.

*Step 1:* *Line search.* Compute $x_{k+1} = x_k + t_k d_k$, where $t_k$ satisfies (1), $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$ and $b_k$.

*Step 2:* *Update preparation.* If $|s_-^T y| > 20t|s_-^T g|$ set $\nu_k = \text{sgn}(s_-^T y)$, otherwise set $\nu_k = -\text{sgn}(s_-^T g)$. Choose parameter $\check{\sigma}_k \in [0, \overline{\sigma}]$ (for $k=0$ we choose $\check{\sigma}_k = 0$) and set $\sigma_k = \nu_k\, \check{\sigma}_k$. If $\sigma_k\, s_-^T y > \lambda\sqrt{bb_-}$ set $\sigma_k = \lambda\nu_k\sqrt{bb_-}/|s_-^T y|$. Using Theorem 2.1, compute $\bar{b}_k$, $\bar{s}_k$ and $\bar{\varrho}_k$ and define $\bar{V}_k$.

*Step 3:* *Update definition.* Set $\tilde{m} = \min(k, m-1)$ and define $H_0^{k+1} = (b_k/|y_k|^2)\, I$ and $H_{k+1} \equiv H_{\tilde{m}+1}^{k+1}$ by

$$H_{i+1}^{k+1} = (\bar{\varrho}_j/\bar{b}_j)\bar{s}_j\, \bar{s}_j^T + \bar{V}_j H_i^{k+1} \bar{V}_j^T, \quad j = k - \tilde{m} + i, \quad 0 \leq i \leq \tilde{m}\,. \tag{6}$$

*Step 4:* *Direction vector.* Set $k := k+1$ and compute $d_k = -H_k g_k$ by the modified Strang recurrences, using the definition of matrices $\{H_i^k\}_{i=0}^{\min(k,m)}$, and go to Step 1.

## 4. Global convergence

**Assumption 4.1.** *The objective function $f : \mathcal{R}^N \to \mathcal{R}$ is bounded from below and uniformly convex with bounded second-order derivatives (i.e. $0 < \underline{G} \le \underline{\lambda}(G(x)) \le \overline{\lambda}(G(x)) \le \overline{G} < \infty$, $x \in \mathcal{R}^N$, where $\underline{\lambda}(G(x))$ and $\overline{\lambda}(G(x))$ are the lowest and the greatest eigenvalues of the Hessian matrix $G(x)$).*

Since our new LM methods do not belong to the Broyden class, the usual approach must be modified. The following lemma before the main theorem plays basic role.

**Lemma 4.1.** *Let matrix $A$ be symmetric positive definite, $\vartheta > 0$, $\tau \neq 0$, $u \in \mathcal{R}^N$ and $v \in \mathcal{R}^N$. Then update $A_+$ given by $A_+ = \tau^2 \vartheta\, uu^T + \left(I - \tau\, uv^T\right) A \left(I - \tau\, vu^T\right)$ is positive definite and satisfies*

$$\begin{aligned} \mathrm{Tr}(A_+) &\le \tau^2 \vartheta |u|^2 + \mathrm{Tr}(A)\Big(1 + |\tau|(|u||v|)\Big)^2, && (7) \\ \mathrm{Tr}(A_+^{-1}) &\le \mathrm{Tr}(A^{-1}) + |v|^2/\vartheta. && (8) \end{aligned}$$

**Theorem 4.1.** *Let objective function $f$ satisfy Assumption 4.1. Then Algorithm 3.1 generates a sequence $\{g_k\}$ that either terminates with $g_k{=}0$ for some $k$ or $\lim\limits_{k\to\infty}|g_k|{=}0$.*

## 5. Numerical results

First we demonstrate the influence of parameter $\sigma$ on the number of evaluations and computational time, using the collection of sparse and partially separable test problems from [5] (Test 14, 22 problems) with $N = 1000$, $m = 10$, $\lambda = 1/2$ and the final precision $\|g(x^\star)\|_\infty \le 10^{-6}$.

Results are given in Table 1, where 'NFE' is the total number of function and also gradient evaluations over all problems, 'Time' the total computational time in seconds and $\phi$ is the arithmetic mean of values 'NFE' and 'Time' over all $\sigma$.

<table>
<tr><th>σ</th><th>NFE</th><th>Time</th><th>σ</th><th>NFE</th><th>Time</th></tr>
<tr><td>0.0</td><td>22522</td><td>8.36</td><td>0.3</td><td>19854</td><td>7.42</td></tr>
<tr><td>0.033</td><td>22185</td><td>8.25</td><td>0.333</td><td>19865</td><td>7.36</td></tr>
<tr><td>0.067</td><td>21121</td><td>7.80</td><td>0.367</td><td>20068</td><td>7.49</td></tr>
<tr><td>0.1</td><td>20751</td><td>7.72</td><td>0.4</td><td>21359</td><td>7.81</td></tr>
<tr><td>0.133</td><td>20940</td><td>7.82</td><td>0.433</td><td>21250</td><td>7.82</td></tr>
<tr><td>0.167</td><td>20929</td><td>7.77</td><td>0.467</td><td>20779</td><td>7.71</td></tr>
<tr><td>0.2</td><td>20144</td><td>7.55</td><td>0.5</td><td>19754</td><td>7.28</td></tr>
<tr><td>0.233</td><td>20579</td><td>7.62</td><td>0.533</td><td>20207</td><td>7.39</td></tr>
<tr><td>0.267</td><td>22064</td><td>8.08</td><td>$\phi$</td><td>20845</td><td>7.72</td></tr>
<tr><td colspan="6">L-BFGS: NFE = 22092 Time = 8.91</td></tr>
</table>

**Tab. 1:** *Influence of parameter $\sigma$ for Test 14.*

|  |  | NFV | Percentage increase of NFV for $\sigma =$ |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Problem | $N$ | L-BFGS | .05 | .10 | .15 | .20 | .25 | .30 | .35 | .40 | .45 | .50 |
| BDQRTIC | 5000 | 248 | -29 | -10 | -19 | -43 | -33 | -16 | -8 | -18 | 5 | -26 |
| BROYDN7D | 2000 | 3029 | -1 | -2 | -3 | -3 | -3 | -3 | -2 | 0 | 2 | 6 |
| CHAINWOO | 1000 | 515 | -8 | -13 | -19 | -14 | -18 | -13 | -20 | -17 | -15 | -14 |
| CURLY10 | 1000 | 5628 | 4 | 8 | 8 | 5 | -5 | 2 | -1 | 3 | -7 | -3 |
| CURLY20 | 1000 | 6852 | -6 | -7 | -6 | -9 | -9 | -7 | -10 | -9 | -7 | -10 |
| CURLY30 | 1000 | 7222 | -3 | -5 | -5 | -7 | -10 | -10 | -5 | -9 | -13 | -7 |
| DIXMAANE | 3000 | 249 | -4 | -3 | -4 | 6 | -4 | -4 | -10 | 2 | -11 | -10 |
| DIXMAANF | 3000 | 189 | 1 | 2 | 14 | 14 | 16 | 14 | 11 | -2 | -4 | 13 |
| DIXMAANG | 3000 | 188 | 11 | 17 | 9 | 5 | 10 | 6 | 13 | 6 | 6 | -7 |
| DIXMAANH | 3000 | 185 | 7 | 12 | 15 | 10 | 10 | 7 | -6 | 5 | -4 | 5 |
| DIXMAANI | 3000 | 881 | -9 | -12 | -17 | -14 | -27 | -33 | -40 | -64 | -77 | -35 |
| DIXMAANJ | 3000 | 317 | -3 | -3 | -4 | -5 | 0 | -9 | -6 | -16 | 17 | 20 |
| DIXMAANK | 3000 | 270 | 9 | -5 | -11 | -7 | 7 | 4 | 16 | 7 | 37 | 28 |
| DIXMAANL | 3000 | 263 | 0 | -8 | -10 | -3 | -10 | -13 | -9 | 8 | 8 | 14 |
| FLETCBV2 | 1000 | 944 | 28 | 1 | -6 | 26 | 35 | 35 | 23 | 54 | 37 | -4 |
| FMINSRF2 | 5625 | 305 | 5 | 1 | 2 | 2 | 2 | 1 | 2 | 8 | 6 | 3 |
| FMINSURF | 5625 | 460 | 0 | -2 | 4 | 13 | -6 | -18 | -4 | 3 | -3 | -13 |
| GENHUMPS | 1000 | 2223 | 8 | 26 | 14 | 17 | 41 | 19 | 27 | 47 | 52 | 48 |
| GENROSE | 1000 | 2393 | -2 | -2 | 0 | 0 | 2 | 3 | 5 | 8 | 10 | 13 |
| MOREBV | 5000 | 116 | 3 | 3 | -10 | -7 | -1 | -3 | -5 | -2 | 0 | 5 |
| MSQRTALS | 529 | 3622 | -22 | -9 | -22 | 3 | -7 | -10 | -4 | -12 | -27 | -12 |
| NCB20 | 1010 | 497 | 3 | 33 | 28 | 7 | 48 | 10 | -5 | 25 | 4 | 3 |
| NCB20B | 1000 | 1792 | -5 | -23 | -5 | -5 | -8 | -9 | -9 | -12 | -9 | -6 |
| NONCVXU2 | 1000 | 3902 | -11 | -17 | -4 | 4 | -2 | -13 | -9 | 0 | -16 | -39 |
| NONDQUAR | 5000 | 4244 | -17 | 3 | 1 | 3 | -1 | -11 | 3 | 13 | -16 | -10 |
| POWER | 500 | 110 | -5 | -7 | -7 | -5 | -12 | -13 | -14 | -13 | -11 | -13 |
| QUARTC | 5000 | 236 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SINQUAD | 5000 | 339 | 5 | 3 | 3 | -3 | 10 | 0 | 1 | 11 | -3 | 7 |
| SPARSINE | 1000 | 10680 | -10 | -8 | -8 | -4 | -12 | -9 | -11 | -15 | -26 | -19 |
| SPMSRTLS | 4999 | 224 | 1 | 0 | -1 | 0 | -5 | -2 | 1 | -2 | -2 | -3 |
| VAREIGVL | 500 | 168 | -3 | -4 | -3 | -10 | -10 | -15 | -5 | -8 | -9 | -11 |
| All problems |  | 58291 | -5.6 | -3.5 | -3.8 | -1.2 | -4.0 | -5.7 | -4.2 | -2.6 | -10.2 | -8.1 |

**Tab. 2:** *CUTE - Percentage increase of NFV against L-BFGS.*

For a better comparison with the L-BFGS method, we performed additional tests with problems from the widely used CUTE collection [1] with various dimensions $N$, $m = 10$, $\lambda = 1/2$ and the final precision $\|g(x^\star)\|_\infty \le 10^{-6}$. The percentage increase of NFV for various values of parameter $\sigma$ against NFV for the L-BFGS (negative values indicate that our results are better than for the L-BFGS) is given in Table 2, where NFV is the number of function and also gradient evaluations. In the last line, the total values over all problems and their percentage increase are given.

Our limited numerical experiments indicate that the suitable choice of parameter $\sigma$ can improve efficiency of limited-memory methods, substantially for some problems.

## References

[1] Bongartz, I., Conn, A.R., Gould, N., and Toint, P.L.: CUTE: constrained and unconstrained testing environment. ACM Transactions on Mathematical Software **21** (1995), 123–160.

[2] Fletcher, R.: *Practical methods of optimization.* John Wiley & Sons, Chichester, 1987.

[3] Liu, D.C. and Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Prog. **45** (1989), 503–528.

[4] Lukšan, L. and Spedicato, E.: Variable metric methods for unconstrained optimization and nonlinear least squares. J. Comput. Appl. Math. **124** (2000), 61–95.

[5] Lukšan, L. and Vlček, J.: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Report V-767, ICS AS CR, Prague, 1998.

[6] Lukšan, L. and Vlček, J.: A recursive formulation of limited memory variable metric methods from the Broyden class. Report V-1059, ICS AS CR, Prague, 2009.

[7] Moughrabi, I.A.: New implicit multistep quasi-Newton methods. Numerical Analysis and Applications **2** (2009), 154–164.

[8] Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comp. **35** (1980), 773–782.

[9] Vlček, J. and Lukšan, L.: Generalizations of the limited-memory BFGS method based on quasi-product form of update. Report V-1060, ICS AS CR, Prague, 2009.

# SIMULATION OF TRANSPORT COLUMN EXPERIMENTS – TRACING TESTS*

Vratislav Žabka, Jan Šembera

**Abstract**

This paper presents the modelling of tracing tests in column experiments. Program Transport was used for the simulation. Its main function is not to predict results of experiments but to compare influence of individual physical and chemical processes to the experiment results. The one-dimensional advection-diffusion model is based on Finite Volume Method; it includes the triple porosity concept, sorption, retardation, and chemical reactions simulated using connected program React from The Geochemist's Workbench package or PhreeqC.

The program Transport simulates not only the processes inside the column but also preparation of entering solutions and measurement methods of outgoing solution parameters.

Features of the program Transport allow a more precise simulation of the ongoing action and study the interaction of tested solutions and rocks. In the last chapter we introduce a simple example of using the program Transport to study a physical phenomenon inside column. It is sorption of sodium ions on colloidal particles of the quartz sand.

**Keywords**: column experiment; tracing test; groundwater modelling; sorption

## 1 Introduction

Importance of modelling is rising due to frequent contamination of the groundwater. In the last twenty years, the coupling of hydrologic transport and reactive chemistry has been fast developed. The observed influence of chemical and biochemical reaction to transport is the reason for the effort to coupling of hydrologic transport and reactive chemistry.

Column experiments are important for correct set up of 3D model parameters. With column experiment we evaluate properties of the tested rock (like sorption capacity, pore volume etc.). Good understanding of processes in the column is required for correct differentiation of individual processes. Thus, the most comprehensive model of the column is needed for accurate estimation of parameters.

The Transport program includes some innovations comparing to conventional models. These innovations can be divided in three groups: 1) more precise geometrical and physical model of the column experiment; 2) major attention to the reactive

[*] This result was realized under the state subsidy of the Czech Republic within the research and development project "Advanced Remedial Technologies and Processes Center" 1M0554 - Programme of Research Centers supported by Ministry of Education and by the Technical university in Liberec within the students' project "Study of chemical processes for groundwater remediation".

component of the process; and 3) communication between transport and reactive component of the process.

In paragraph three we are engaged in streamlining communication between programs. The Transport program, which is being developed by authors of this paper, first computes the transport processes. Then it sends the request to one of the geochemical software PhreeqC or The Geochemist's Workbench. Geochemical program calculates chemical equilibrium and sends data back to the Transport program. Both geochemical programs are commercial and we do not intervene in their code. Thermodynamic equilibrium is calculated by finding a minimum of Gibbs function.

Every group is important for the simulation but for the purposes of this work we describe especially geometrical and physical model innovations.

## 2 Geometrical and physical model innovation

The simplest model of column experiment is a cylinder consisting of porous medium (Figure 1). In our case, we have used one-dimensional model based on the Finite Volume Method. The inlet is modelled by boundary condition defining species concentrations as piecewise constant functions in time. The result of the simulation is the chemical composition of the solute outgoing from the column.

The real column experiment looks slightly different and some simplifications affect computed parameters. In this chapter we briefly describe correction of two simplifications that are most relevant: 1) input and output chamber of the column and 2) output flask.

By the terms input and output chamber of the column we mean the volumes ahead of the porous medium cylinder and behind this cylinder. Those parts of the

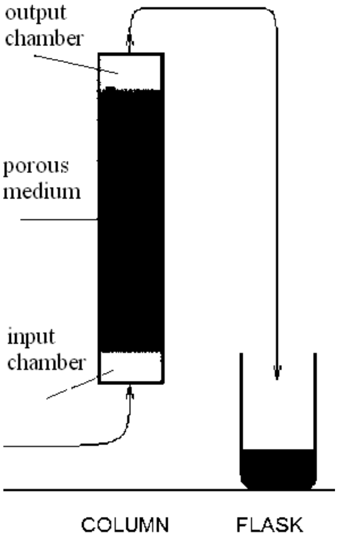


**Fig. 1:** *Scheme of laboratory column experiment.*

column are formed due to technology of column creation and they have various effects in various column experiments. Both chambers frequently contain another material than the column cylinder. When we do not include chambers into the model of column experiment, the parameters can be calibrated incorrectly. For better understanding we present the equation (3) for computation of concentration of one species in the input chamber. All these equations (2), (3), (4) are based on the principle of mass conservation and on the relation (1).

$$Q = \frac{C \cdot V}{t} \tag{1}$$

$$C_0(t + \Delta t) = C_0(t) + (C_{vst}(t) - C_0(t))\frac{\Delta t \cdot Q(t)}{V_0} \tag{2}$$

where $C_0$ [mg·l$^{-1}$] is concentration of one species in the input chamber, $t$ [s] is the actual time, $V_0$ [l] is volume of the input chamber, $C_{vst}$ [mg·l$^{-1}$] is the concentration of the input solution and $Q$ [l·s$^{-1}$] is the flow rate. Another type of the simplest model extension is the model of the output flask. The solute concentrations and properties in the real column experiment are measured in the output flask. The solution outflows from the output chamber into the output flask. At a certain time the flask is replaced and analyzed. Composition of the solution in the output flask is different than composition of the solute outflowing from the column computed by the simplest model. This is the reason why we include the computation of the output flask into the model. It is described by equations (3) and (4):

$$V_{N+2}(t + \Delta t) = V_{N+2}(t) + \Delta t \cdot Q(t) \tag{3}$$

$$C_{N+2}(t + \Delta t) = \frac{C_{N+2}(t) \cdot V_{N+2}(t) + \Delta t \cdot Q(t) \cdot C_{N+1}(t)}{V_{N+2}(t + \Delta t)} \tag{4}$$

where the index $N + 1$ refers to the output chamber, and the index $N + 2$ refers to the output flask.

## 3 Geochemical reaction modeling

Following [1], while the coupling of hydrologic transport and chemical reaction models is an active area of research, the development of chemical reaction batch models has received much less attention. Whereas reactive parameters setting is more difficult then setting of transport parameters. Reactive transport program cannot only compute with species concentrations. Information about other solute properties is important to include. Those properties are changing along the column experiment depending on current reactions and ambient conditions.

For example, setting of the solute and external atmosphere equilibrium is important for solute properties and inside chemical reactions. Otherwise setting of precipitation processes have an effect to solute composition and sometimes also transport properties. E.g. when column experiment takes only few days, it is not possible for hematite to precipitate; the mineral hematite is the final product of precipitation for solution including oxygen and iron but its precipitation needs at least hundreds of years and column experiments do not last as long, so we have to suppress this mineral in the thermodynamic equilibrium computations.

In the case of tracing tests, reactive component plays not such a significant role. Into the column filled with rocks whose pores are saturated with water, NaCl solution is injected. There are no chemical reactions which could significantly affect properties of the solution. But there are other physical phenomena that can influence solution properties. One of them is sorption.

## 4 Example of using the model – simulation of sorption

Our column experiments use quartz sand, which contain colloidal particles. According to [3], in natural waters with pH of 6–8 clay minerals are cationic. These minerals may exchange the crystal lattice ions for cations in solution because minerals surface charge is mostly negative (electrostatic sorption). Quartz clay may exchange calcite ions for sodium ions.

The effect of solute concentration on the adsorption is described by adsorption isotherms. In the simple case of equilibrium sorption, Langmuir isotherm corresponds with electrostatic adsorption the best way. To compute the concentration of absorbate, analytical solution of equation system is used in the program Transport. Except of Langmuir isotherm equation (5) system contains also the mass conservation equation (6).

$$C_s = C_{max} \frac{C_r K_L}{1 + C_r K_L} \tag{5}$$

$$C_{s0} + C_{r0} = M = C_s + C_r \tag{6}$$

where $C_s$ [mg·l$^{-1}$] is concentration of adsorbate adsorbed of the adsorbent, $C_{max}$ [mg·l$^{-1}$] is the maximal concentration of adsorbate adsorbed of the adsorbent. $K_L$ [mg·l$^{-1}$] is the adsorption constant, $C_r$ [mg·l$^{-1}$] is the concentration of adsorbate in liquid, $M$ [mg·l$^{-1}$] is total concentration of the substance.

Custom column experiment consists in injecting a certain amount of tracers (sodium chloride) in the column, a constant flow of fresh water column. At the exit from the column mainly monitors the solution conductivity and pH.

When we simulated the situation without sorption, computed pH is changing minimally (see pH-model in Figure 2). These results did not correspond to the measurement. But simulation of conductivity fit well. When sorption of Na$^+$ according to the Langmuir isotherm is included into calculatin, the calculated conductivity does not significantly change. The calculated pH in this case approaches the measured values much better then without sorption simulation (see pH-model with sorption in Figure 2).

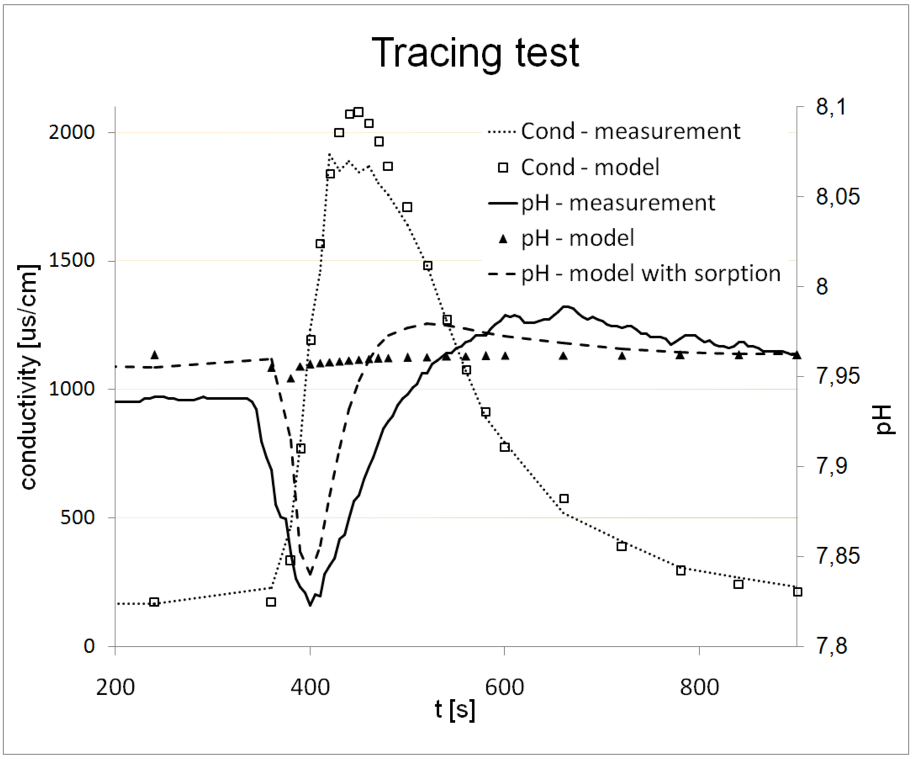


**Fig. 2:** *Tracing test – dependence conductivity and pH on time of the experiment. It was injected 10 ml of NaCl solution (20 g·l$^{-1}$).*

## 5 Conclusion

This paper presented changes in column experiment model that were done to improve understanding of reactive transport processes. Parameters of real column experiment can be more precisely estimated and better implemented to 3D groundwater simulation.

In the last section, possibilities of use of the program Transport for the study of physical and chemical phenomena that take place inside the column during the tracing tests were presented.

## References

[1] Fang, Y.: *Reactive chemical transport under multiphase system.* Ph.D. Dissertation, Dept. of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA, 2003.

[2] Gombos, L.: Laboratorní testy imobilizace kontaminantů. Etapa II. Dynamické průtočné testy – injektáž "slivu A". (Laboratory tests of contaminant imobilization. Stage II. Dynamical flow tests – grouting of "Solution A"). DIAMO, s. p., o. z. TÚU, Stráž p/R, 2006 (in Czech).

[3] Pitter, P.: *Hydrochemie (Hydrochemistry).* SNTL, Prague, 1990 (in Czech).

[4] Yeh, G., Zhang, F., Sun, J., Li, Y., Li, M., Siegel, M.D., and Fang, Y.: Numerical modeling of coupled fluid flow and thermal and reactive biogeochemical transport in porous and fractured media. Computational Geosciences **14** (2009), 149–170.

# NUMERICAL APPROACH TO A RATE-INDEPENDENT MODEL OF DECOHESION IN LAMINATED COMPOSITES*

Jan Zeman, Pavel Gruber

### Abstract

In this paper, we present a numerical approach to evolution of decohesion in laminated composites based on incremental variational problems. An energy-based framework is adopted, in which we characterize the system by the stored energy and dissipation functionals quantifying reversible and irreversible processes, respectively. The time-discrete evolution then follows from a solution of incremental minimization problems, which are converted to a fully discrete form by employing the conforming finite element method. Results of a benchmark problem suggest that the resulting model allows to describe both initiation and propagation of interfacial decohesion, with a low sensitivity to spatial discretization.

## 1 Introduction

The overall behavior of the vast majority of engineering materials and structures is significantly affected or even dominated by the presence of interfaces (i.e. internal boundaries). This is particularly true for composite materials, where interfaces provide weak spots from which damage initiates at different levels of resolution. Therefore, in the engineering community, considerable research efforts have been focused on the adequate description and simulation of interfacial behavior; see, e.g., a recent review [14] for additional details.

During the last decade, the cohesive zone concept has established itself to be a convenient tool to predict interfacial damage initiation and propagation, both from the modeling [19] and computational [3] viewpoints. In this framework, originally introduced for quasi-brittle material by Hillerborg et al. [8], behavior of the bulk material is assumed to be damage-free, whereas the interfacial response is described by means of an inelastic law formulated in terms of interfacial separation and cohesive tractions bridging the crack. Such description is also well-suited to treatment by methods of computational inelasticity, particularly when applied in the *quasi-static setting* (i.e. neglecting viscosity and inertia effects).

Under this modeling assumption, the delamination problem can be conveniently described by the theory of Energetic Rate-Independent Systems developed by Mielke and co-workers, see [11] for a general overview. In this framework, a mechanical system is characterized by a time-dependent *stored energy* functional $\mathcal{E}$ and a *dissipation*

*This work was supported by grant No. 106/08/1379 of the Czech Science Foundation and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS10/020/OHK1/1T/11.

*distance* $\mathcal{D}$, quantifying the reversible and irreversible processes in the system, respectively. When supplemented with suitable initial data, evolution of the system then follows from conditions of energetic stability and conservation of energy, formulated solely in terms of $\mathcal{E}$ and $\mathcal{D}$. This provides a mathematical basis to study a wide range of problems of inelastic solid mechanics in a unified way. Moreover, the framework naturally leads to the *time-incremental energy minimization* concept, thus providing a starting point for the subsequent numerical treatment by optimization methods.

In the context of delamination, the rate-independent approach was first employed by Kočvara et al. [9] to study systems with perfectly brittle interfaces and later extended even to fully rate-dependent systems subject to temperature changes [17]. In this contribution, the focus is on numerical and engineering aspects of the rate-independent setting. In Section 2, we introduce an energy-based delamination model of the Ortiz-Pandolfi type [15], characterized by a piecewise affine traction-separation law. For simplicity, the small-strain setting is adopted and the bulk material is assumed to be described by linear elasticity. In Section 3, we briefly review available existence results for the time-independent problem, which are used to construct fully discrete schemes based on the finite element method in Section 4. The paper is concluded by an illustrative example of flexural delamination.

## 2 The model setup

Let $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) be a bounded Lipschitz domain with boundary $\partial\Omega$ and let us consider its decomposition into a finite number of mutually disjoint Lipschitz subdomains $\Omega^{(i)}$, $i = 1, ..., N$. Further, for $N \geq j > i$, we denote by $\Gamma^{(ij)} = \partial\Omega^{(i)} \cap \partial\Omega^{(j)}$ the (possibly empty) common boundary between $\Omega^{(i)}$ and $\Omega^{(j)}$.

Kinematics of the system is described by independent domain displacement fields $\boldsymbol{u}^{(i)} : \Omega^{(i)} \to \mathbb{R}^d$. Local impenetrability is enforced by means of the Signorini condition, requiring

$$[\![u_n]\!]^{(ij)} \geq 0 \text{ on } \Gamma^{(ij)} \quad \text{where} \quad [\![u_n]\!]^{(ij)} = [\![\boldsymbol{u}]\!]^{(ij)} \cdot \boldsymbol{n}^{(ij)}, \tag{1}$$

Here, $\boldsymbol{n}^{(ij)}$ denotes the unit normal to $\Gamma^{(ij)}$ oriented from $\Omega^{(j)}$ to $\Omega^{(i)}$ and $[\![\boldsymbol{u}]\!]^{(ij)} = \boldsymbol{u}^{(i)}|_{\Gamma^{(ij)}} - \boldsymbol{u}^{(j)}|_{\Gamma^{(ij)}}$, $[\![\boldsymbol{u}]\!]^{(ij)} : \Gamma^{(ij)} \to \mathbb{R}^d$ denotes the interfacial displacement jump, with $\boldsymbol{u}^{(i)}|_{\Gamma^{(ij)}}$ being the trace of $\boldsymbol{u}^{(i)}$ on $\Gamma^{(ij)}$. We assume that the system is subject to a time-dependent boundary displacement $\boldsymbol{w}_{\mathrm{D}}(t)$, $t \in [0; T]$ imposed on the time-independent Dirichlet part of the boundary $\Gamma_{\mathrm{D}} \subset \partial\Omega$. As for the interfacial damage processes, these are quantified by the *damage* variable $\omega^{(ij)} : \Gamma^{(ij)} \to [0; 1]$, with $\omega^{(ij)}(\boldsymbol{x}) = 0$ and $\omega^{(ij)}(\boldsymbol{x}) = 1$ indicating a healthy and a fully damaged interfacial point $\boldsymbol{x} \in \Gamma^{(ij)}$, see Figure 1(a) for an illustration.

As indicated earlier, we shall characterize evolution of the system by means of certain energetic functionals. First, we introduce the spaces of admissible state variables in the form

$$
\begin{aligned}
\mathscr{U} &= \Big\{ \boldsymbol{u} \in L^2(\Omega;\mathbb{R}^d) : \boldsymbol{u}^{(i)} \in W^{1,2}(\Omega^{(i)};\mathbb{R}^d), \boldsymbol{u}^{(i)} = \boldsymbol{0} \text{ on } \partial\Omega^{(i)} \cap \Gamma_{\mathrm{D}}, \\
&\qquad [\![u_n]\!]^{(ij)} \geq 0 \text{ on } \Gamma^{(ij)} \Big\}, && (2) \\
\mathscr{Z} &= \Big\{ \omega \in L^\infty(\cup_{ij}\Gamma^{(ij)}) : \omega^{(ij)} \in L^\infty(\Gamma_{ij}) : 0 \leq \omega^{(ij)} \leq 1 \text{ on } \Gamma^{(ij)} \Big\}, && (3)
\end{aligned}
$$

and define the stored energy functional $\mathcal{E} : [0;T] \times \mathscr{U} \times \mathscr{Z} \to \mathbb{R}$ as

$$
\begin{aligned}
\mathcal{E}(t,\boldsymbol{u},\omega) &= \sum_{i=1}^{N} \tfrac{1}{2} \int_{\Omega_i} \boldsymbol{\varepsilon}\left(\boldsymbol{u}^{(i)} + \boldsymbol{u}_{\mathrm{D}}^{(i)}(t)\right) : \boldsymbol{C}^{(i)} : \boldsymbol{\varepsilon}\left(\boldsymbol{u}^{(i)} + \boldsymbol{u}_{\mathrm{D}}^{(i)}(t)\right) \mathrm{d}\Omega \\
&+ \sum_{i=1}^{N} \sum_{j=i+1}^{N} \int_{\Gamma^{(ij)}} e^{(ij)} \left([\![\boldsymbol{u}]\!]^{(ij)}, \omega^{(ij)}\right) \mathrm{d}S, && (4)
\end{aligned}
$$

where $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathsf{T}}) \in \mathbb{R}^{d\times d}$ denotes the small-strain tensor, $\boldsymbol{C}^{(i)} \in \mathbb{R}^{d\times d\times d\times d}$ is the positive-definite material stiffness tensor of the $i$-th domain and $e^{(ij)} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ denotes the density of stored interfacial energy presented later in Section 2.1. Further, $\boldsymbol{u}_{\mathrm{D}}^{(i)}(t)$ is the restriction of an extension $\boldsymbol{u}_{\mathrm{D}}(t)$ of the the time-dependent Dirichlet boundary conditions, i.e. $\boldsymbol{u}_{\mathrm{D}}(t)|_{\Gamma_{\mathrm{D}}} = \boldsymbol{w}_{\mathrm{D}}(t)$ with $\boldsymbol{u}_{\mathrm{D}}(t) \in W^{1,2}(\Omega;\mathbb{R}^d)$.

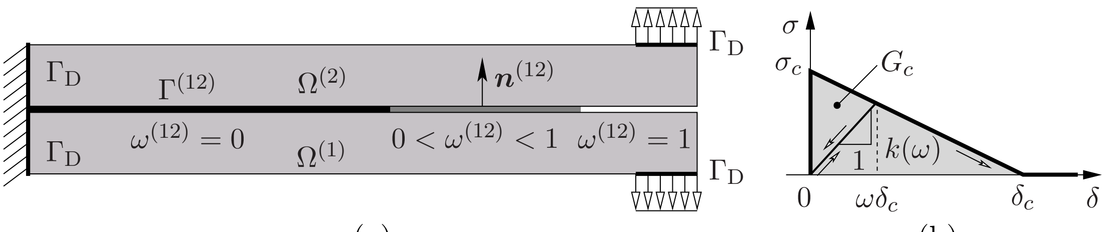
Since the domains are assumed to be elastic, the irreversible processes occur only at the interfaces. Therefore, the dissipation distance $\mathcal{D} : \mathscr{Z} \times \mathscr{Z} \to \overline{\mathbb{R}}$, quantifying the energy dissipated by changing the internal variable from $\omega_1$ to $\omega_2$, admits the expression

$$
\mathcal{D}(\omega_1,\omega_2) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \int_{\Gamma^{(ij)}} d^{(ij)}(\omega_1^{(ij)}, \omega_2^{(ij)})\, \mathrm{d}S, \tag{5}
$$

where $d^{(ij)} : \mathbb{R} \times \mathbb{R} \to \overline{\mathbb{R}}$ is the density of dissipated interfacial energy specified next.

### 2.1 Interfacial constitutive law

To introduce the cohesive zone model, we consider the following decomposition of interfacial displacement jumps (the superscript $\bullet^{(ij)}$ is dropped for the sake of brevity)



(a) (b)

**Fig. 1: (a)** *An example of the introduced notation and of* **(b)** *the traction-separation law.*

$$[\![\boldsymbol{u}]\!] = [\![u_n]\!]\boldsymbol{n} + [\![\boldsymbol{u}_s]\!], \tag{6}$$

where the normal displacement jump $u_n$ follows from Eq. (1) and $[\![\boldsymbol{u}_s]\!]$ denotes the tangential component. The vector of interfacial tractions $\boldsymbol{t} \in \mathbb{R}^d$ is decomposed analogously:

$$\boldsymbol{t} = \sigma_n \boldsymbol{n} + \boldsymbol{t}_s \quad \text{where} \quad \sigma_n = \boldsymbol{t} \cdot \boldsymbol{n}. \tag{7}$$

Following [15], we introduce the effective interfacial displacement jumps and tractions in the form

$$\delta([\![\boldsymbol{u}]\!])^2 = u_n^2 + \beta^2 \|\boldsymbol{u}_s\|^2, \qquad \sigma(\boldsymbol{t})^2 = \sigma_n^2 + \beta^{-2}\|\boldsymbol{t}_s\|^2, \tag{8}$$

where $\beta > 0$ is a mode mixity parameter, which needs to be determined from experiments. Due to the adopted linear traction-separation law, the perfect interface (with $\omega = 0$) is characterized by its strength $\sigma_c$ (in Pa) and maximal effective opening $\delta_c$ (in m), cf. Figure 1(b). The area under the traction-separation line gives the density of the dissipated energy $G_c = \frac{1}{2}\sigma_c\delta_c$ for $\omega_1 = 0$ and $\omega_2 = 1$. For general case, this yields the following expression for the stored and dissipated energies:[1]

$$\begin{aligned} e([\![\boldsymbol{u}]\!], \omega) &= \frac{1}{2}\frac{\sigma_c(1-\omega)}{\delta_c \omega}\delta^2([\![\boldsymbol{u}]\!]) = \frac{1}{2}k(\omega)\delta^2([\![\boldsymbol{u}]\!]), & (9) \\ d(\omega_1, \omega_2) &= \begin{cases} G_c(\omega_2 - \omega_1) & \text{for } \omega_2 \geq \omega_1, \\ +\infty & \text{otherwise.} \end{cases} & (10) \end{aligned}$$

Note that the '$+\infty$' term in Eq. (10) corresponds to the damage unidirectionality, i.e. the damage variable $\omega$ never decreases during the decohesion process.

## 3 Incremental energetic minimization

The evolution of the mechanical system will be described using a time-incremental approach, where each step corresponds to a variational minimization problem. To this goal, we discretize the time interval $[0; T]$ as $0 = t_0 < t_1 = t_0 + \Delta t < \cdots < t_M = T$ and abbreviate $u_k = \boldsymbol{u}(t_k)$ and $\omega_k = \omega(t_k)$. Then, given the initial condition $(\boldsymbol{u}_0, \omega_0) \in \mathscr{U} \times \mathscr{Z}$, the time-incremental solution is defined via

**Definition 1** (Time-incremental solution)**.** *For $k = 1, 2, \ldots, M$, find iteratively $(\boldsymbol{u}_k, \omega_k) \in \mathscr{U} \times \mathscr{Z}$ such that*

$$(\boldsymbol{u}_k, \omega_k) = \arg \min_{(\boldsymbol{u}, \omega) \in \mathscr{U} \times \mathscr{Z}} \mathcal{E}(t_k, u, \omega) + \mathcal{D}(\omega_{k-1}, \omega). \tag{11}$$

[1] Note that for $k \to \infty$ for $\omega \to 0_+$, which agrees with the assumption of perfect interface, but leads to numerical difficulties. Therefore, in the numerical experiments, the $\omega = 0$ case is replaced with $\omega = \omega^{\text{in}} < 1$.

The existence of the time-discrete solution to the delamination problem follows from the next proposition, proven in [9]:

**Proposition 1.** *Assume that* $\operatorname{meas}_{d-1}\left(\partial\Omega^{(i)} \cap \Gamma_{\mathrm{D}}\right) \neq 0$ *for* $i = 1, 2, \ldots, N$, $\boldsymbol{w}_{\mathrm{D}}(t_k) \in W^{1/2,2}(\Gamma_{\mathrm{D}}; \mathbb{R}^d)$ *for* $k = 1, 2, \ldots, M$ *and that*

$$(\boldsymbol{u}_0, \omega_0) = \arg \min_{(\boldsymbol{u},\omega) \in \mathscr{U} \times \mathscr{Z}} \mathcal{E}(0, u, \omega) + \mathcal{D}(\omega_0, \omega). \tag{12}$$

*Then for all* $k = 1, 2, \ldots, M$ *we have*

*i) existence of time-incremental solution* $(\boldsymbol{u}_k, \omega_k) \in \mathscr{U} \times \mathscr{Z}$,

*ii) stability of* $(\boldsymbol{u}_k, \omega_k)$*:*

$$\mathcal{E}(t_k, \boldsymbol{u}_k, \omega_k) \leq \mathcal{E}(t_k, \boldsymbol{u}, \omega) + \mathcal{D}(\omega_k, \omega) \tag{13}$$

*for all* $(\boldsymbol{u}, \omega) \in \mathscr{U} \times \mathcal{Z}$,

*iii) two-sided energy inequality*

$$\begin{aligned}\int_{t_{k-1}}^{t_k} \partial_t \mathcal{E}(t, \boldsymbol{u}_k, \omega_k)\,\mathrm{d}t \;&\leq\; \mathcal{E}(t_{k-1}, \boldsymbol{u}_{k-1}, \omega_{k-1}) + \mathcal{D}(\boldsymbol{u}_{k-1}, \omega_k) - \mathcal{E}(t_k, \boldsymbol{u}_k, \omega_k) \\ &\leq\; \int_{t_{k-1}}^{t_k} \partial_t \mathcal{E}(t, \boldsymbol{u}_{k-1}, \omega_{k-1})\,\mathrm{d}t. \end{aligned} \tag{14}$$

## 4 Numerical treatment

The developments presented up to this point provide a convenient framework for an implementable numerical scheme, obtained by discretizing the time-incremental formulation (11) in the space variables by the finite element method. In particular, we employ low-order discretizations of domain displacements $\boldsymbol{u}^{(i)}$ by $P^1$-continuous finite elements and of interfacial damage variables $\omega^{(ij)}$ by $P^0$ finite elements, as this choice is supported by convergence proofs for $h \to 0$ in [12].

To this goal, each domain $\Omega_i$ is triangulated using elements with a mesh size $h$. We assume that the discretization is conforming, i.e. that two interfacial nodes belonging to the adjacent domains $\Omega_i$ and $\Omega_j$ are geometrically identical, and that the same mesh is used to approximate variables $\boldsymbol{u}$ and $\omega$. Then, the finite element discretization with a suitable numbering of nodes yields a discrete incremental minimization problem in the form

$$\left.\begin{array}{ll} \text{minimize} & (\mathbf{u}, \mathbf{w}) \mapsto E(t_k, \mathbf{u}, \mathbf{w}) + D(\mathbf{w}_{k-1}, \mathbf{w}) \\ \text{subject to} & \mathbf{B}_{\mathrm{E}}\mathbf{u} = \mathbf{0}, \quad \mathbf{B}_{\mathrm{I}}\mathbf{u} \geq \mathbf{0}, \quad \mathbf{w}_{k-1} \leq \mathbf{w} \leq \mathbf{1}. \end{array}\right\} \tag{15}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ stores the nodal displacements for individual sub-domains and $\mathbf{w} \in \mathbb{R}^{n_\omega}$ designates the delamination parameters associated with interfacial element edges. The discretized stored energy functional $E \to [0; T] \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\omega} \to \mathbb{R}$ receives the form, cf. (4),

$$E(t, \mathbf{u}, \mathbf{w}) = \tfrac{1}{2}\left(\mathbf{u} + \mathbf{u}_{\mathrm{D}}(t)\right)^{\mathsf{T}} \mathbf{K} \left(\mathbf{u} + \mathbf{u}_{\mathrm{D}}(t)\right) + \tfrac{1}{2}[\![\mathbf{u}]\!]^{\mathsf{T}} \mathbf{k}(\mathbf{w}) [\![\mathbf{u}]\!], \tag{16}$$

where $\mathbf{K} = \mathrm{diag}(\mathbf{K}^{(1)}, \mathbf{K}^{(2)}, \ldots, \mathbf{K}^{(N)})$ is a symmetric positive semi-definite block-diagonal stiffness matrix of order $n_u$ (derived from $\boldsymbol{C}^{(i)}$), $[\![\mathbf{u}]\!] \in \mathbb{R}^{n_k}$ stores the displacement jumps at interfacial nodes, and $\mathbf{k}$ is a symmetric positive-definite interfacial stiffness matrix of order $n_k$, which depends non-linearly on $\mathbf{w}$ as follows from Eq. (9). The discrete dissipation distance is expressed by a linear function

$$D(\mathbf{w}_1, \mathbf{w}_2) = \mathbf{a}^{\mathsf{T}} \left( \mathbf{w}_2 - \mathbf{w}_1 \right), \tag{17}$$

where the entries of $\mathbf{a} \in \mathbb{R}^m$ store the amount of energy dissipated by the complete delamination of an interfacial element; see [7, 9] for additional details. The constraints in problem (15) consist of the homogeneous Dirichlet boundary conditions prescribed at nodes specified by a full-rank $m_E \times n_u$ Boolean matrix $\mathbf{B}_{\mathrm{E}}$, nodal interpenetration conditions specified by a full-rank matrix $\mathbf{B}_{\mathrm{I}} \in \mathbb{R}^{m_I \times n_u}$ storing the corresponding components of the normal vector, and the box constraints on the internal variable.

### 4.1 Alternating minimization algorithm

```
1. Require $\mathbf{w}_{(0)}$, set $j = 0$
2. Repeat
    (a) Set $j = j + 1$
    (b) Solve for $\mathbf{u}_{(j)}$:
            minimize     $\mathbf{u} \mapsto E(t_k, \mathbf{u}, \mathbf{w}_{(j-1)})$
            subject to   $\mathbf{B}_{\mathrm{E}}\mathbf{u} = \mathbf{0} \quad \mathbf{B}_{\mathrm{I}}\mathbf{u} \geq \mathbf{0}$          (18)
    (c) Solve for $\mathbf{w}_{(j)}$:
            minimize     $\mathbf{w} \mapsto E(t_k, \mathbf{u}_{(j)}, \mathbf{w}) + D(\mathbf{w}_{k-1}, \mathbf{w})$
            subject to   $\mathbf{w}_{k-1} \leq \mathbf{w} \leq \mathbf{1}$          (19)
    (d) Until $\|\mathbf{w}_{(j)} - \mathbf{w}_{(j-1)}\| \leq \eta$
3. Set $\mathbf{u}_k = \mathbf{u}_{(j)}$ and $\mathbf{w}_k = \mathbf{w}_{(j)}$
```

**Tab. 1:** *Conceptual implementation of the alternating minimization algorithm for the k-th time step and an initial guess* $\mathbf{w}_{(0)}$.

The discrete incremental problem (15) represents a large-scale non-convex program (due to the $\mathbf{k}(\mathbf{w})$-term), which is very difficult to solve using a monolithic approach. Nevertheless, it can be observed that the problem is separately convex with respect to variables $\mathbf{u}$ and $\mathbf{w}$. This directly suggests the concept of the *alternating minimization algorithm*, proposed by Bourdin et al. [4] for variational models of fracture. In the current context, the algorithm is briefly summarized in Table 1.

The individual sub-problems of the alternating minimization algorithm can be resolved using specialized solvers. In particular, step (18) now becomes a quadratic programming problem, which can be efficiently solved when employing recent developments in duality-based solvers for domains separated by imperfect interfaces [10] and for frictionless contact problems [6]. Owing to the piecewise constant approximation of the delamination parameters, problem (19) can be solved locally element-by-element in a closed form, see [7] for additional details.

### 4.2 Time-stepping strategy

Even though the alternating minimization algorithm performs well for a wide range of computational examples, it generally converges only to a local minimizer of the objective function (15), which can violate the two-sided energetic inequality (14). Exactly this observation was used in Mielke et al. [13] to propose a heuristic backtracking strategy summarized for the current problem in Table 2.

```
1. Set $k = 1$, $\mathbf{w}_0 = \mathbf{w}_{(0)} = \mathbf{0}$
2. Repeat
    (a) Determine $\mathbf{w}_k$ using the alternating minimization algorithm for time $t_k$ and initial value $\mathbf{w}_{(0)}$
    (b) If
        $\int_{t_{k-1}}^{t_k} \partial_t E(t, \mathbf{u}_k, \mathbf{w}_k)\,\mathrm{d}t \leq E(t_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) + D(\mathbf{w}_{k-1}, \mathbf{w}_k) - E(t_k, \mathbf{u}_k, \mathbf{w}_k)$
        $\leq \int_{t_{k-1}}^{t_k} \partial_t E(t, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})\,\mathrm{d}t \qquad (20)$
        set $\mathbf{w}_{(0)} = \mathbf{w}_k$ and $k = k + 1$
    (c) Else set $\mathbf{w}_{(0)} = \mathbf{w}_k$ and $k = k - 1$
    (d) Until $k > M$
```

**Tab. 2:** *Conceptual implementation of time-stepping strategy.*

The computational procedure proceeds as follows. At the $k$-th time level, the approximate solution is found using the alternate minimization algorithm, initiated with the solution $\mathbf{w}_{(0)}$ (Step 2(a)). If the pair of solutions $(\mathbf{u}_{k-1}, \mathbf{w}_{k-1})$ and $(\mathbf{u}_k, \mathbf{w}_k)$ satisfies the discretized energy inequality, $\mathbf{w}_k$ is certified as an initial guess for the next time level (Steps 2(b)). In the opposite case, the solution $(\mathbf{u}_k, \mathbf{w}_k)$ leads to a smaller value of the objective function (15) at time $t_{k-1}$ than the actual result

$(\mathbf{u}_{k-1}, \mathbf{w}_{k-1})$. Therefore, it is used as an initial guess at time $t_{k-1}$ (Step 2(c));[2] see also [2] for additional details and further discussion.

It should be emphasized that there is generally no guarantee that the algorithm will locate the global optimum of the objective function (15) for all time levels and that it will converge in a finite number of steps. Computational experiments nevertheless indicate that it is sufficiently robust and that it delivers solutions with (often substantially) lower energies than the basic alternating minimization scheme [2, 5, 13].

## 5 Example

The basic features of the model will be illustrated by means of the mixed-mode flexure test, adopted from [18]. The beam specimen consists of two non-symmetric aluminum layers, bonded together by a thin layer of resin adhesive. The beam is simply supported and the loading is imposed by a prescribed displacement at the mid-span, increasing linearly with time $t$ up to the final value of 1.5 mm for $t = T = 1$. The delamination is initiated by a pre-existing interfacial crack, see also Fig. 2 for an illustration.

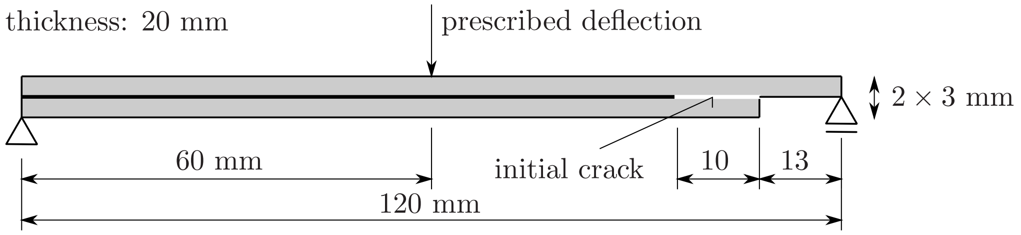
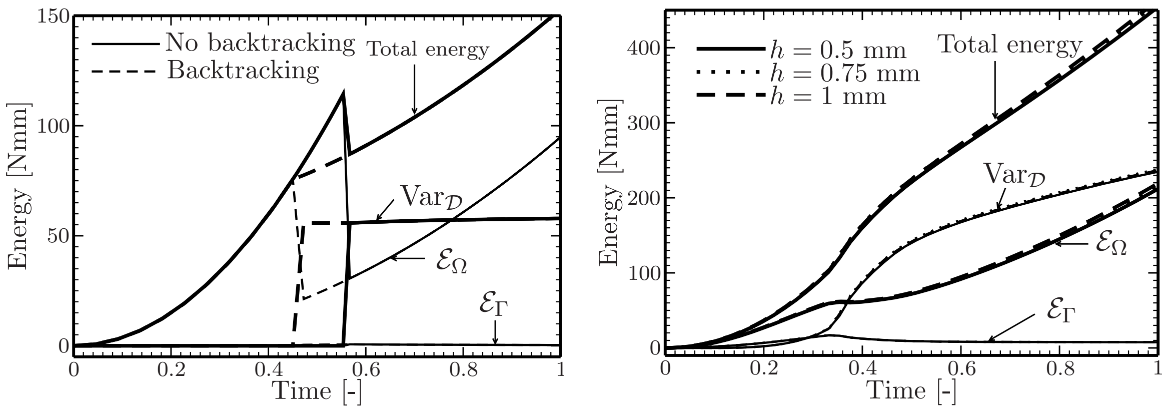


**Fig. 2:** *Setup of the mixed-mode flexure test.*

The material properties of the bulk material and the interface appear summarized in Table 3. Two different sets of interfacial properties are considered, one characterized by a higher value of fracture energy $G_c$ and a lower value of initial stiffness $k(\omega^{\text{in}})$ as defined by Eq. (9), whereas the brittle interface corresponds to a low fracture energy $G_c$ and high initial stiffness. The results below correspond to time step $\Delta t = 0.025$ and the value of termination tolerance of the alternating minimization algorithm set to $\eta = 10^{-6}$, recall Table 1. All simulations were performed with an in-house code implemented in MATLAB®.

The energetics of the delamination process for the brittle interface is shown in Figure 3, highlighting the difference between the local energy minimization (full lines) and the time back-tracking scheme (dashed lines). The local scheme predicts initially elastic behavior, followed by complete separation of the two layers at $t \approx 0.56$, resulting in the jump of the dissipated energy $\text{Var}_{\mathcal{D}}$. However, exactly at this step the two-sided inequality is violated, as detected by the back-tracking algorithm. Inductively using such solution as the initial guess of the alternating minimization scheme,

[2] Note that the stability of initial data (12) ensures that $k \geq 1$.



(a) (b)

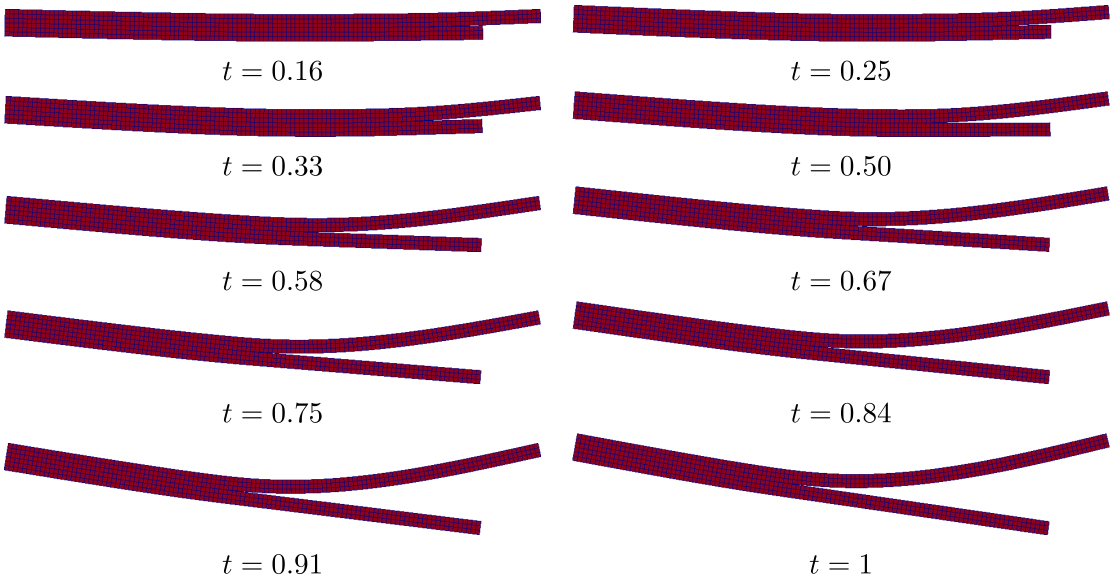
**Fig. 3:** *Energetics of the delamination process for (a) brittle and (b) ductile interfaces. $\mathcal{E}_{\Omega}$ = energy stored in the bulk, $\mathcal{E}_{\Gamma}$ = interfacial stored energy and $\mathrm{Var}_{\mathcal{D}}$ = energy dissipated during the whole process.*

the algorithm returns to the original elastic path, thereby predicting a response leading to a lower value of the total energy for $t \in [0.46, 0.56]$. During the whole time interval, the contribution of the stored interfacial energy $\mathcal{E}_{\Gamma}$ remains relatively small, owing to a large value of the interfacial stiffness.

The ductile interface shows a more gradual transition from the elastic response up to the fully debonded state. In this case, the two-sided inequality (20) remains valid during the whole loading program and no back-tracking is necessary. The interfacial delamination initiates first in the shearing mode, which corresponds to a rapid increase in the dissipated energy for $t \in [0.3; 0.4]$. Then it propagates mainly due to opening in the normal direction, which is manifested by the decrease of interfacial energy; see also Fig. 4 for an illustration. We observe that the response remains almost independent on the mesh size $h$, which is in agreement with theoretical convergence results at disposal. Moreover, no artificial oscillations, reported e.g. in [1], appear in the overall response for both variants of material data. This demonstrates suitability of the algorithm for engineering applications.

| **Material parameter** | **Ductile** | **Brittle** |
|---|---|---|
| *Bulk material* | | |
| Young's modulus, $E$ (GPa) | 75 | 75 |
| Poission's ratio, $\nu$ | 0.3 | 0.3 |
| *Interface* | | |
| Fracture energy, $G_c$ (Jm$^{-2}$) | 250 | 25 |
| Critical stress $\sigma_c$ (MPa) | 5 | 5 |
| Initial damage $\omega^{\mathrm{in}}$ | $10^{-1}$ | $10^{-3}$ |
| Mode mixity parameter $\beta$ | 1 | 1 |

**Tab. 3:** *Material parameters.*



$t = 0.16$ $t = 0.25$

$t = 0.33$ $t = 0.50$

$t = 0.58$ $t = 0.67$

$t = 0.75$ $t = 0.84$

$t = 0.91$ $t = 1$

**Fig. 4:** *Snapshots of delamination evolution (displacements depicted as magnified by a factor of* 5*).*

## 6 Conclusions

In this paper, we have presented a variational model for delamination phenomena based on incremental energy minimization. Its algorithmic treatment relies on the alternating minimization algorithm, complemented with a-posteriori two-sided energy estimates to test the energetic stability of the evolution. Results of the model problem indicate that the method is sufficiently robust for a wide range of material parameters and allows to capture the whole delamination process from the damage initiation up to the complete separation. Note that we have omitted the time-continuous model, obtained as $\Delta t \to 0$. This aspect, together with additional details and extensions, is available in a recent review [16].

**Acknowledgments** The authors would like to express their thanks to Martin Kružík, Tomáš Roubíček and Jaroslav Kruis for numerous discussions on the subject of this work. We also thank an anonymous referee for a number of constructive suggestions which helped us to improve the original version of the paper.

## References

[1] Alfano, G. and Crisfield, M.A.: Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues. Int. J. Numer. Methods Eng. **50** (2001), 1701–1736.

[2] Benešová, B.: Global optimization numerical strategies for rate-independent processes. J. Glob. Optim. (2010). Online First,
URL: http://dx.doi.org/10.1007/s10898-010-9560-6.

[3] de Borst, R.: Numerical aspects of cohesive-zone models. Eng. Fract. Mech. **70** (2003), 1743–1757.

[4] Bourdin, B., Francfort, G.A., and Marigo, J.J.: Numerical experiments in revisited brittle fracture. J. Mech. Phys. Solids **48** (2000), 797–826.

[5] Bourdin, B.: Numerical implementation of the variational formulation for quasistatic brittle fracture. Interface Free Bound. (2007), 411–430.

[6] Dostál, Z.: *Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities*, *Springer Optimization and Its Applications*, vol. 23. Springer Science+Business Media, New York, NY, 2009.

[7] Gruber, P., Zeman, J., and Kruis, J.: Numerical modeling of delamination via incremental energy minimization and duality-based solvers, 2010. In preparation.

[8] Hillerborg, A., Modeer, M., and Petersson, P.: Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. Cem. Concr. Res. **6** (1976), 773–781.

[9] Kočvara, M., Mielke, A., and Roubíček, T.: A rate-independent approach to the delamination problem. Math. Mech. Solids **11** (2006), 423–447.

[10] Kruis, J. and Bittnar, Z.: Reinforcement-matrix interaction modeled by FETI method. In: *Domain Decomposition Methods in Science and Engineering XVII*, pp. 567–573. Springer Science, 2007.

[11] Mielke, A.: Evolution in rate-independent systems (Ch. 6). In: C. Dafermos and E. Feireisl (Eds.), *Handbook of Differential Equations, Evolutionary Equations*, vol. 2, pp. 461–559. Elsevier B.V., Amsterdam, 2005.

[12] Mielke, A. and Roubíček, T.: Numerical approaches to rate-independent processes and applications in inelasticity. Math. Model. Numer. Anal. (M2AN) **43** (2009), 399–428.

[13] Mielke, A., Roubíček, T., and Zeman, J.: Complete damage in elastic and viscoelastic media and its energetics. Comput. Meth. Appl. Mech. Eng. **199** (2010), 1242–1253.

[14] Orifici, A., Herszberg, I., and Thomson, R.: Review of methodologies for composite material modelling incorporating failure. Compos. Struct. **86** (2008), 194–210.

[15] Ortiz, M. and Pandolfi, A.: Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. Int. J. Numer. Methods Eng. **44** (1999), 1267–1282.

[16] Roubíček, T., Kružík, M., and Zeman, J.: *Delamination and adhesive contact models and their mathematical analysis and numerical treatment*, chap. Mathematical Methods and Models in Composites. Imperial College Press, 2010. 45 pages, submitted for publication.

[17] Roubíček, T. and Rossi, R.: Thermodynamics and analysis of rate-independent adhesive contact at small strains. Nonlinear Anal.-Theory Methods Appl. (2010). Submitted for publication, URL: `http://arxiv.org/abs/1004.3764`.

[18] Valoroso, N. and Champaney, L.: A damage-mechanics-based approach for modelling decohesion in adhesively bonded assemblies. Eng. Fract. Mech. **73** (2006), 2774–2801.

[19] Wisnom, M.R.: Modelling discrete failures in composites with interface elements. Compos. Pt. A-Appl. Sci. Manuf. **41** (2010), 795–805.

# AN IMPROVEMENT OF EUCLID'S ALGORITHM[*]

Jan Zítko, Jan Kuřátko

**Abstract**

The paper introduces the calculation of a greatest common divisor of two univariate polynomials. Euclid's algorithm can be easily simulated by the reduction of the Sylvester matrix to an upper triangular form. This is performed by using *c*-*s* transformation and $QR$-factorization methods. Both procedures are described and numerically compared. Computations are performed in the floating point environment.

## 1 Introduction

Euclid's algorithm and the corresponding manipulations with the Sylvester resultant matrix are two well-known methods for computing the greatest common divisor of two univariate polynomials. See the book [1] or the paper [4].

Theory has been developed in those papers and all practical examples included only low-degree polynomials in which the effect of computing in floating point arithmetic has not shown. That is why we have decided to work on computation of the greatest common divisor of two large-degree polynomials in this article. Many times the numerical experiments have yielded inaccurate or even wrong results caused for instance by the big differences in the absolute value of coefficients of polynomials which are calculated during Euclid's algorithm. Since the problems in real world have demanded the best possible precision on the coefficients of the greatest common divisor some of the ideas on the balancing the coefficients have been introduced in the article [5] and several others. We have developed an improvement of Euclid's algorithm in this paper, called *c*-*s* transformation, which is conducted by the transformation of Sylvester matrix. The above mentioned method, described in [2], has not been published yet and its rigorous analysis has been presented in this article.

Scalars $c$ and $s$ are computed from coefficients of polynomials in every step of Euclid's algorithm and are resembled to scalars used in Givens rotation. Detailed description is given in paragraph 2 where the classic and well known Euclid's algorithm has been compared with *c*-*s* transformation. Let us mention that structure of the Sylvester matrix is preserved by both methods.

We have decided to mention another interesting approach proposed in [7] which does not preserve the structure of the Sylvester matrix. This method is based on $QR$-factorization of the Sylvester matrix or a part of the Sylvester matrix. Coefficients of the greatest common divisor can be obtained from the last non-vanishing row of

[*] This work is a part of the research project MSM0021620839 financed by MSMT.

the upper triangular matrix $R$ obtaining by $QR$-factorization. Complete description of aforementioned algorithm and numerical experiments are given in paragraph 3.

Reasonable results can be obtained if we know the degree of the greatest common divisor. In that case we know exactly where coefficients of the greatest common divisor can be found in the matrix $R$. Algorithms for determining the degree of the greatest common divisor have been studied in [3] or [6]. Those methods are not and cannot be included in this article.

All test polynomials have been computed via convolution that is why the degree of the greatest common divisor is known and used in our examples.

In this article, all numerical experiments have been carried out in double precision. We have worked with polynomials having non-trivial greatest common divisor.

## 2 Euclid's algorithm and transformations of the Sylvester matrix

Let the symbol GCD $(f_0, f_1)$ denotes the greatest common divisor of polynomials $f_0$ and $f_1$ and $\deg(f_0)$ the degree of $f_0$. Let

$$f_0(x) = a_0x^m + a_1x^{m-1} + \cdots + a_{m-1}x + a_m, \tag{1}$$

$$f_1(x) = b_0x^n + b_1x^{n-1} + \cdots + b_{n-1}x + b_n, \tag{2}$$

where $m \geq n$, $a_0a_m \neq 0$, $b_0b_n \neq 0$, To illustrate the algorithm, let us consider the polynomials $f_0$ and $f_1$ of degrees 5 and 2 respectively:

$$f_0(x) = a_0x^5 + a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5,$$
$$f_1(x) = b_0x^2 + b_1x + b_2.$$

The Sylvester resultant matrix $S(f_0, f_1)$ for the polynomials $f_0$ and $f_1$ of degrees $m = 5$ and $n = 2$ is

$$S(f_0, f_1) = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & 0 \\ 0 & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & 0 & 0 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 & 0 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 \end{bmatrix}.$$

We will now formulate modified Euclid's algorithm which can scale down the big differences between the coefficients of $f_0$ and $f_1$. Let us define the division $f_0/f_1$ in the following form:

$$c_0 \underbrace{(a_0x^5 + a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5)}_{f_0(x)} + s_0 \underbrace{(b_0x^2 + b_1x + b_2)}_{f_1(x)} x^3$$
$$\underbrace{= 0 + \underbrace{(c_0a_1 + s_0b_1)}_{a_1^{(1)}} x^4 + \underbrace{(c_0a_2 + s_0b_2)}_{a_2^{(1)}} x^3 + \underbrace{c_0a_3}_{a_3^{(1)}} x^2 + \underbrace{c_0a_4}_{a_4^{(1)}} x + \underbrace{c_0a_5}_{a_5^{(1)}}}_{h_4(x) := a_1^{(1)}x^4 + a_2^{(1)}x^3 + a_3^{(1)}x^2 + a_4^{(1)}x + a_5^{(1)}}.$$

The numbers $c_0$ and $s_0$ are chosen to remove the leading coefficient of $f_0$. To define the corresponding transformation of the Sylvester matrix, let us define the matrix $G_0^{(1)}(c_0, s_0)$

$$G_0^{(1)}(c_0, s_0) = \begin{bmatrix} c_0 & 0 & s_0 & 0 & 0 & 0 & 0 \\ 0 & c_0 & 0 & s_0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Apparently

$$S^{(1)}(f_0, f_1) := G_0^{(1)}(c_0, s_0) S(f_0, f_1) = \begin{bmatrix} 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & a_4^{(1)} & a_5^{(1)} & 0 \\ 0 & 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & a_4^{(1)} & a_5^{(1)} \\ b_0 & b_1 & b_2 & 0 & 0 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 & 0 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 \end{bmatrix},$$

where

$$a_i^{(1)} = \begin{cases} c_0 a_i + s_0 b_i & \text{for} \quad i = 1, 2, \\ c_0 a_i & \quad\quad \text{otherwise.} \end{cases}$$

Let $a_1^{(1)} \neq 0$. Then $\deg(h_4) = 4$. In the opposite case, the process would be performed with the polynomial of degree less than 4. The Euclid's algorithm proceeds according to the following schema:

$$c_1 \underbrace{\left(a_1^{(1)} x^4 + a_2^{(1)} x^3 + a_3^{(1)} x^2 + a_4^{(1)} x + a_5^{(1)}\right)}_{h_4(x)} + s_1 \underbrace{\left(b_0 x^2 + b_1 x + b_2\right)}_{f_1(x)} x^2$$
$$\underbrace{= 0 + \underbrace{\left(c_1 a_2^{(1)} + s_1 b_1\right)}_{a_2^{(2)}} x^3 + \underbrace{\left(c_1 a_3^{(1)} + s_1 b_2\right)}_{a_3^{(2)}} x^2 + \underbrace{c_1 a_4^{(1)}}_{a_4^{(2)}} x + \underbrace{c_1 a_5^{(1)}}_{a_5^{(2)}}}_{h_3(x) := a_2^{(2)} x^3 + a_3^{(2)} x^2 + a_4^{(2)} x + a_5^{(2)}}.$$

The numbers $c_1$ and $s_1$ are again chosen to remove the coefficient of $x^4$. The corresponding matrix operation consists of the construction of the matrix $G_1^{(1)}(c_1, s_1)$, by analogy to the previous case, such that

$$S^{(2)}(f_0, f_1) := G_1^{(1)}(c_1, s_1) S^{(1)}(f_0, f_1) = \begin{bmatrix} 0 & 0 & a_2^{(2)} & a_3^{(2)} & a_4^{(2)} & a_5^{(2)} & 0 \\ 0 & 0 & 0 & a_2^{(2)} & a_3^{(2)} & a_4^{(2)} & a_5^{(2)} \\ b_0 & b_1 & b_2 & 0 & 0 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 & 0 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 \end{bmatrix},$$

where

$$a_i^2 = \begin{cases} c_1 a_i^{(1)} + s_1 b_{i-1} & \text{for} \quad i = 2, 3, \\ c_1 a_i^{(1)} & \text{otherwise.} \end{cases}$$

If $a_2^{(2)} = 0$ and $a_3^{(2)} \neq 0$, then instead of $h_3$ the polynomial $h_2$,

$$h_2(x) = a_3^{(2)} x^2 + a_4^{(2)} x + a_5^{(2)} \ ,$$

is considered. Moreover, if $a_3^{(2)} = 0$, then the first stage of Euclid's algorithm terminates. Let us assume that the degrees of all polynomials after division decrease by 1. Hence $a_2^2 \neq 0$. Let the numbers $c_2$, $s_2$ and then $c_3$ and $s_3$ are chosen to remove the coefficient of dominant power. The last two divisions yield the polynomials

$$\begin{array}{rclcl} h_2(x) & = & a_3^{(3)} x^2 + a_4^{(3)} x + a_5^{(3)} & = & c_2 h_3(x) + s_2 f_1(x) x, \\ h_1(x) & = & a_4^{(4)} x + a_5^{(4)} & = & c_3 h_2(x) + s_3 f_1(x), \end{array}$$

where $a_3^{(3)} \neq 0$ and $a_4^{(4)} \neq 0$. The matrices $G_2^{(1)}(c_2, s_2)$, $G_3^{(1)}(c_3, s_3)$ correspond to the last two divisions. Their construction is omitted because it is the same as in the previous steps. If we define

$$G_1 = G_3^{(1)}(c_3, s_3) G_2^{(1)}(c_2, s_2) G_1^{(1)}(c_1, s_1) G_0^{(1)}(c_0, s_0)$$

and

$$P_1 = [e_3, e_4, e_5, e_6, e_7, e_1, e_2] \ ,$$

then the first stage of Euclid's algorithm can be written in the matrix form as follows

$$P_1 G_1 S(f_0, f_1) = \left[ \begin{array}{cccc|ccc} b_0 & b_1 & b_2 & 0 & 0 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 & 0 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & 0 \\ \hline 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 \\ 0 & 0 & 0 & 0 & a_4^{(4)} & a_5^{(4)} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_4^{(4)} & a_5^{(4)} \end{array} \right] =: \left[ \begin{array}{cc} F_{1,1} & F_{1,2} \\ F_{2,1} & F_{2,2} \end{array} \right].$$

Note that we have obtained the coefficients of the polynomial $h_1$ in the last two rows. The formula

$$\underbrace{c_3 c_2 c_1 c_0 f_0(x)}_{\widetilde{f}_0(x)} = \underbrace{-(c_3 c_2 c_1 s_0 x^3 + c_3 c_2 s_1 x^2 + c_3 s_2 x + s_3)}_{\widetilde{q}_0(x)} \underbrace{f_1(x)}_{\widetilde{f}_1(x)} + \underbrace{h_1(x)}_{\widetilde{f}_2(x)}.$$

summarises the first stage of Euclid's algorithm. Hence we have

$$\widetilde{f}_0(x) = \widetilde{q}_0(x) \widetilde{f}_1(x) + \widetilde{f}_2(x).$$

The block $F_{2,2}$ is again the Sylvester matrix $S(\widetilde{f}_1, \widetilde{f}_2)$. We suppose that $\widetilde{f}_2(x) \neq 0$. If $\widetilde{f}_2(x) = 0$, then $\widetilde{f}_1(x) = GCD(f_0, f_1)$. The transformation of the Sylvester resultant matrix to an upper triangular matrix requires that the same procedure is applied to the matrix $S(\widetilde{f}_1, \widetilde{f}_2)$, and this corresponds to the second stage of Euclid's algorithm, that is, the division $\widetilde{f}_1/\widetilde{f}_2$. Analogously there exist matrices $G_2$ and $P_2$ such that

$$P_2G_2P_1G_1S(f_0, f_1) = \begin{bmatrix} b_0 & b_1 & b_2 & 0 & 0 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 & 0 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & 0 & 0 & a_4^{(4)} & a_5^{(4)} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_4^{(4)} & a_5^{(4)} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_2^{(2)} \end{bmatrix}.$$

It is $\widetilde{f}_3(x) = b_2^{(2)}$. Let us remark that $\widetilde{f}_2(x) = GCD(f_0, f_1)$ if $\widetilde{f}_3(x) = 0$. Otherwise $f_0$ and $f_1$ are coprime.

We will now demonstrate how to pick the numbers $c$ and $s$. If we take

$$c_0 = 1 \qquad \text{and} \qquad s_0 = -\frac{a_0}{b_0}\ ,$$

then the division in Euclid's algorithm has the following form

$$\underbrace{(a_0x^5 + a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5)}_{f_0(x)} - \underbrace{(b_0x^2 + b_1x + b_2)}_{f_1(x)}(\frac{a_0}{b_0})x^3$$
$$= 0 + \underbrace{\left(a_1 - \frac{a_0b_1}{b_0}\right)}_{a_1^{(1)}} x^4 + \underbrace{\left(a_2 - \frac{a_0b_2}{b_0}\right)}_{a_2^{(1)}} x^3 + \underbrace{a_3}_{a_3^{(1)}} x^2 + \underbrace{a_4}_{a_4^{(1)}} x + \underbrace{a_5}_{a_5^{(1)}}.$$

In the next step we have considered $c_1$ and $s_1$ in the form

$$c_1 = 1 \qquad \text{and} \qquad s_1 = -\frac{a_1^{(1)}}{b_0}\ ,$$

and analogously are defined the numbers $c_i$ a $s_i$ in the following steps. This choice forms Euclid's algorithm in the well known form.

The second possible choice of $c$ and $s$ is based on the idea of balance of the coefficients of $f_0$ and $f_1$. In the first step these numbers are defined as

$$c_0 = \frac{b_0}{\sqrt{a_0^2 + b_0^2}} \qquad \text{and} \qquad s_0 = -\frac{a_0}{\sqrt{a_0^2 + b_0^2}}\ ,$$

and analogously in the next steps. This form of Euclid's algorithm will be called $c$-$s$ transformation. Let us denote for the polynomials (1) and (2)

$$d_1 = \max_{i,j \in \{1,\ldots,m\}} \Big||a_i| - |a_j|\Big|, \quad d_2 = \max_{i,j \in \{1,\ldots,n\}} \Big||b_i| - |b_j|\Big|, \quad \text{diff}(f_0, f_1) = (d_1, d_2).$$

**Example 1.** Let $f_0$ and $f_1$ be two polynomials such that

$$\begin{aligned} f_0(x) &= (x-4.1)(x-3)(x-\sqrt{2})^2(x+1)(x+\sqrt{2})(x+5)^2, \\ f_1(x) &= (x-3)(x-\sqrt{3})(x-1)(x+1)(x+\sqrt{3})(x+8). \end{aligned}$$

Their greatest common divisor $u$ has the form

$$u(x) = (x-3)(x+1) = x^2 - 2x - 3.$$

Let $u_{Euc}$ and $u_{c\text{-}s}$ denotes the greatest common divisor computed by Euclid's algorithm and *c-s* transformation, respectively. We have obtained

$$\begin{aligned} u_{Euc}(x) &= x^2 - 1.99999999999994x - 2.99999999999993, \\ u_{c\text{-}s}(x) &= x^2 - 1.99999999999998x - 2.99999999999998, \\ \|u_{Euc} - u\|_2 = 9.26402450510883e{-}14, &\quad \|u_{c\text{-}s} - u\|_2 = 3.14269606124535e{-}14\ . \end{aligned}$$

Both procedures yielded practically the exact greatest common divisor. Let us remark, that for the first division $f_0/f_1$ in Euclid's algorithm we have obtained $\mathrm{diff}(f_0, f_1) = (1189.33794283234, 98)$.

**Example 2.** Let

$$\begin{aligned} f_0(x) &= (x-4)^2(x-\sqrt{5})^2(x-\sqrt{3})^2(x-\sqrt{2})^2(x+0.5)^2(x+1)^2, \\ f_1(x) &= (x-6.51)^2(x-5)^2(x-4)(x+0.5)^2(x+0.9)(x+1)^2. \end{aligned}$$

Their greatest common divisor $u$ has the form

$$u(x) = (x-4)(x+0.5)^2(x+1)^2 = x^5 - x^4 - 8.75x^3 - 11.5x^2 - 5.75x - 1\ .$$

We have obtained

$$\begin{aligned} &u_{Euc}(x) = x^5 - 0.9999999x^4 - 8.75000000x^3 - 11.5000000x^2 - 5.75000000x - 1.00000000, \\ &u_{c\text{-}s}(x) = x^5 - 0.9999999x^4 - 8.74999999x^3 - 11.49999999x^2 - 5.74999999x - 0.99999999, \\ &\|u_{Euc} - u\|_2 = 1.70205143034978e{-}11, \quad \|u_{c\text{-}s} - u\|_2 = 6.36839947245598e{-}12\ . \end{aligned}$$

We have obtained again a good result. Let us remark, that for the first division $f_0/f_1$ in Euclid's algorithm we have obtained $\mathrm{diff}(f_0, f_1) = (1116.93467622, 12607.8786650)$.

**Example 3.** Let $f_0$ and $f_1$ be the following polynomials:

$$\begin{aligned} f_0(x) &= (x-11)^2(x-8)^2(x-6)^2(x-1)^2(x+2)^2(x+3)^2, \\ f_1(x) &= (x-15)^2(x-8)(x-6)(x+5)^2(x+11)^2. \end{aligned}$$

Their greatest common divisor $u$ has the form $u(x) = (x-8)(x-6) = x^2 - 14x + 48$.

Let us compare the result which yields the modification of Euclid's algorithm by using *c-s* transformation with the result which yields the standard implementation represented by the m-file `poly_gcd.m`[1] denoted by $u_{Euc}$. We have obtained

$$\begin{aligned}
&u_{Euc}(x) = x^2 - 13.99999999946275x + 47.99999999538410,\\
&u_{c\text{-}s}(x) = x^2 - 14.00000000045505x + 48.00000000093540,\\
&\|u_{Euc} - u\|_2 = 4.64705900662480e{-}09,\\
&\|u_{c\text{-}s} - u\|_2 = 1.04021318800892e{-}09,\\
&\mathrm{diff}(f_0, f_1) = (11024639, 32669999)
\end{aligned}$$

More examples have been calculated and we have found out that Euclid's algorithm in matrix form and *c-s* transformation yield almost the same results for low-degree polynomials, in some cases Euclid's algorithm gives better results. If the degree of both polynomials gets larger, then the *c-s* transformation yields more accurate results.

## 3 QR factorization method for computing the greatest common divisor

The following idea described in [7] will be illustrated for the polynomials of degree $m = 4$ and $n = 3$. Let

$$\begin{aligned}
f_0(x) &= x^4 + a_1x^3 + a_2x^2 + a_3x + a_4,\\
f_1(x) &= b_0x^3 + b_1x^2 + b_2x + b_3.
\end{aligned}$$

A companion matrix $C_4$ associated with the polynomial $f$ has the form

$$C_4 = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -a_4 & -a_3 & -a_2 & -a_1 \end{bmatrix}.$$

It is assumed that the coefficient $a_0 = 1$. The matrix $f_1(C_4)$ is very important. The $\mathrm{GCD}(f_0, f_1)$ can be obtained very easily from the matrix $f_1(C_4)$. See the book [1]. Let the Sylvester matrix be split into the four blocks

$$S(f_0, f_1) = \left[\begin{array}{ccccccccc} 1 & a_1 & a_2 & | & a_3 & a_4 & 0 & 0\\ 0 & 1 & a_1 & | & a_2 & a_3 & a_4 & 0\\ 0 & 0 & 1 & | & a_1 & a_2 & a_3 & a_4\\ - & - & - & + & - & - & - & -\\ b_0 & b_1 & b_2 & | & b_3 & 0 & 0 & 0\\ 0 & b_0 & b_1 & | & b_2 & b_3 & 0 & 0\\ 0 & 0 & b_0 & | & b_1 & b_2 & b_3 & 0\\ 0 & 0 & 0 & | & b_0 & b_1 & b_2 & b_3 \end{array}\right] =: \begin{bmatrix} S_{1,1} & S_{1,2}\\ S_{2,1} & S_{2,2} \end{bmatrix}.$$

[1] http://www.mathworks.com/matlabcentral/fileexchange/20859-gcd-of-polynomials

It is clear that all blocks are Toeplitz matrices. It is easy to calculate the Schur complement $S_{2,2}^{(*)} = S_{2,2} - S_{2,1}S_{1,1}^{-1}S_{1,2}$ and according to the well known theory $S_{2,2}^{(*)} = J_4 f_1(C_4)J_4$. Moreover, there exists an orthogonal matrix Q such that

$$QJ_4f_1(C_4)J_4 = R,$$

where $J_4$ is a matrix with ones on the counter diagonal and $R$ is an upper-triangular matrix, the last nonzero row of which contains the coefficients of the GCD of $f_0$ and $f_1$. Let $\mathrm{GCD}(f_0, f_1) = d_0x^2 + d_1x + d_2$ in our special case. Then the matrix $R$ has the form

$$R = \begin{bmatrix} x & x & x & x \\ 0 & d_0 & d_1 & d_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where the $x$'s indicates elements whose values are unimportant.

In the following examples let $u_{Schu}$ and $u_{Hor}$ denote the greatest common divisor which was obtained using QR-factorization of $S_{2,2}^{(*)} = S_{2,2} - S_{2,1}S_{1,1}^{-1}S_{1,2}$ and from the $QR$-factorization of $J_mf_1(C_m)J_m$, where $f_1(C_m)$ was constructed by Horner's scheme, respectively. Let us remark that $m$ is the degree of the polynomial $f_0$.

**Example 4.** Let $f_0, f_1$ and $u$ be polynomials from Example 1. We have calculated

$$\begin{aligned} u_{Schu}(x) &= x^2 - 2.00000000001121x - 3.00000000001110, \\ u_{Hor}(x) &= x^2 - 2.00000000000175x - 3.00000000000208, \end{aligned}$$
$$\|u_{Schu} - u\|_2 = 1.57742694329152e{-}11, \quad \|u_{Hor} - u\|_2 = 2.71252314056005e{-}12\ .$$

**Example 5.** Let $f_0, f_1$ and $u$ be polynomials from Example 2. We have obtained

$$\begin{aligned} u_{Schu}(x) &= x^5 - 0.999999985809256x^4 - 8.749999957416437x^3 \\ &\quad -11.499999953695204x^2 - 5.749999978552083x - 0.999999996447228, \\ u_{Hor}(x) &= x^5 - 1.00000000005345x^4 - 8.75000000015926x^3 \\ &\quad -11.50000000017163x^2 - 5.75000000007888x - 1.00000000001314, \end{aligned}$$
$$\|u_{Schu} - u\|_2 = 6.80551732665222e{-}08, \quad \|u_{Hor} - u\|_2 = 2.53131658107744e{-}10\ .$$

**Example 6.** Let $f_0, f_1$ and $u$ be the same polynomials as in Example 3. Then the following results have been calculated.

$$\begin{aligned} u_{Schu}(x) &= x^2 - 14.00000000217399x + 48.00000001232704, \\ u_{Hor}(x) &= x^2 - 14.00000000138107x + 48.00000000802783, \end{aligned}$$
$$\|u_{Schu} - u\|_2 = 1.25172701840642e{-}08, \quad \|u_{Hor} - u\|_2 = 8.14575566893229e{-}09\ .$$

The coefficients of the greatest common divisor are obtained from the 10th row of matrix $R$. Let $R$ be a matrix from $QR$-factorization of the matrix $J_{12}f_1(C_{12})J_{12}$

where $f_1(C_{12})$ was constructed by Horner's scheme. The coefficients of the greatest common divisor can be obtained from the 10th row. Let us present the elements of $R$ in 9th–11th row.

row 9: $-8.32446551450895e{+}05$, $1.13733574278167e{+}07$, $-3.60249143760594e{+}07$, $-1.34829260320804e{+}07$

row 10: $-7.99170726641915e{+}05$, $1.11883901740905e{+}07$, $-3.83601948852275e{+}07$ ($u_{Hor}$ is obtained after transformation to monic form)

row 11: $5.13598466751748e{-}03$, $-2.90332235929244e{-}02$.

If the degree of greatest common divisor is not known, the following problem appears: Are the numbers in the 11th row zero? This difficult question is behind the topic of this short paper. For more details see [7].

## 4 Summary

Euclid's algorithm is composed from sequence of steps and division of two polynomials, whose degree is decreasing to zero, is represented in each particular step. New modification of Euclid's algorithm called *c-s* transformation has been introduced in this article. This modification produces better numerical results in comparison with classical Euclid's algorithm and it is conducted by transformation of the Sylvester matrix and structure of the Sylvester matrix is preserved. The algorithm based on $QR$-decomposition was mentioned and its numerical results were compared with *c-s* transformation. The second algorithm does not preserve the structure of Sylvester matrix.

**Acknowledgments.**

The work is a part of the research project MSM0021620839 financed by MSMT. The author thanks Bc. Ján Eliaš for careful reading of the manuscript and a referee for references leading to improvement of the whole paper.

## References

[1] Barnett, S.: *Polynomial and linear control system.* Marcel Dekker, New York, USA, 1983.

[2] Eliaš, J.: *Problems connected with the calculation of the GCD.* Bachelor thesis, Charles University, Faculty of Mathematics and Physics at Prague, 2009.

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

[4] Laidacker, M. A.: Another Theorem Relating Sylvester's Matrix and the Greatest Common Divisors. Mathematics Magazine **42** (1969), 126–128.

[5] Li, B., Yang, Z., and Zhi, L.: Fast low rank approximation of a Sylvester matrix by structured total least norm. J. Japan Soc. Symbolic and Algebraic Comp. **11** (2005), 165–174.

[6] Winkler, J. R. and Allan, J. D.: Structured total least norm and approximate GCDs of inexact polynomials. J. Comput. Appl. Math. **215** (2008), 1–13

[7] Zarowski, C. J., Ma, X., and Fairman, F. W.: QR- factorization method for computing the greatest common divisor of two polynomials with inexact coeffixients. IEEE Trans. Signal Processing **48**, No. 11 (2000), 3042–3051.

Programs and Algorithms of Numerical Mathematics 15

## List of Participants

Stanislav Bartoň, doc. RNDr., CSc.
Ústav základů techniky
Agronomická fakulta
Mendelova univerzita v Brně
Zemědělská 1, 613 00 Brno
e-mail: 3128@node.mendelu.cz

Marek Brandner, doc. Ing., Ph.D.
Katedra matematiky
Fakulta aplikovaných věd ZČU v Plzni
Univerzitní 22, 306 14 Plzeň
e-mail: brandner@kma.zcu.cz

Jan Brandts, Dr.
Korteweg-de Vries Institute for Math.
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam, the Netherlands
e-mail: janbrandts@gmail.com

Jan Březina, Mgr., Ph.D.
Ústav nových technologií a aplik. infor.
Fak. mechat., infor. a meziobor. studií
Technická univerzita v Liberci
Studentská 2, 461 17 Liberec
e-mail: jan.brezina@tul.cz

Martin Čermák, Ing.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: martin.cermak@vsb.cz

Marta Čertíková, RNDr., Ph.D.
Ústav technické matematiky
Fakulta strojní ČVUT v Praze
Karlovo náměstí 13, 121 35 Praha 2
e-mail: marta.certikova@fs.cvut.cz

Jan Česenek, Mgr.
Katedra numerické matematiky
Matematicko-fyzikální fak. UK v Praze
Sokolovská 83, 186 75 Praha 8
e-mail: jan.cessa@seznam.cz

Jan Chleboun, doc. RNDr., CSc.
Katedra matematiky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: chleboun@mat.fsv.cvut.cz

Pavol Chocholatý, doc. RNDr., CSc.
Kat. mat. analýzy a numer. mat.
Fakulta matematiky, fyziky a informat.
Univerzita Komenského v Bratislave
Mlynská dolina, 842 48 Bratislava
Slovensko
e-mail: chocholaty@fmph.uniba.sk

Josef Dalík, doc. RNDr., CSc.
Ústav matemat. a deskript. geometrie
Fakulta stavební VUT v Brně
Veveří 95, 602 00 Brno
e-mail: Dalik.j@fce.vutbr.cz

Jiří Egermaier, Ing., Ph.D.
Katedra matematiky
Fakulta aplikovaných věd ZČU v Plzni
Univerzitní 22, 306 14 Plzeň
e-mail: jirieggy@kma.zcu.cz

Dalibor Frydrych, doc. Ing., Ph.D.
Ústav nových technologií a aplik. infor.
Fak. mechat., infor. a meziobor. studií
Technická univerzita v Liberci
Studentská 2, 461 17 Liberec
e-mail: dalibor.frydrych@tul.cz

Milan Hanuš, Ing.
Katedra matematiky
Fakulta aplikovaných věd ZČU v Plzni
Univerzitní 22, 306 14 Plzeň
e-mail: mhanus@kma.zcu.cz

Petr Harasim, Ing., Ph.D.
Ústav geoniky AV ČR, v.v.i.
Studentská 1768, 708 00 Ostrava-Poruba
e-mail: petr.harasim@ugn.cas.cz

Milan Hokr, doc. Ing., Ph.D.
Ústav nových technologií a aplik. infor.
Fak. mechat., infor. a meziobor. studií
Technická univerzita v Liberci
Studentská 2, 461 17 Liberec
e-mail: milan.hokr@tul.cz

Martin Horák, Ing.
Katedra mechaniky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: martin.horak@fsv.cvut.cz

Jiří Hozman, Mgr., Ph.D.
Katedra matemat. a didakt. matemat.
Fak. přírodovědně-humanitní a pedagog.
Technická univerzita v Liberci
Studentská 2, 461 17 Liberec
e-mail: jiri.hozman@tul.cz

Radka Keslerová, Mgr., Ph.D.
Ústav technické matematiky
Fakulta strojní ČVUT v Praze
Karlovo náměstí 13, 121 35 Praha 2
e-mail: keslerov@marian.fsik.cvut.cz

Martin Kocurek, Mgr.
Katedra matematiky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: kocurek@mat.fsv.cvut.cz

Roman Kohut, RNDr., CSc.
Ústav geoniky AV ČR, v.v.i.
Studentská 1768, 708 00 Ostrava-Poruba
e-mail: kohut@ugn.cas.cz

Hana Kopincová, Ing.
Katedra matematiky
Fakulta aplikovaných věd ZČU v Plzni
Univerzitní 22, 306 14 Plzeň
e-mail: kopincov@kma.zcu.cz

Petr Kotas, Ing.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: petr.kotas@vsb.cz

Martin Kramář, Ing.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: martin.kramar@vsb.cz

Václav Kučera, RNDr., Ph.D.
Katedra numerické matematiky
Matematicko-fyzikální fak. UK v Praze
Sokolovská 83, 186 75 Praha 8
e-mail: vaclav.kucera@email.cz

Jan Kuřátko, Bc.
Katedra numerické matematiky
Matematicko-fyzikální fak. UK v Praze
Sokolovská 83, 186 75 Praha 8
e-mail: j.kuratko@gmail.com

Pavel Kůs, Mgr.
Matematický ústav AV ČR, v.v.i.
Žitná 25, 115 67 Praha 1
e-mail: pavel.kus@gmail.com

Martin Lanzendörfer, Mgr.
Ústav informatiky AV ČR, v.v.i.
Pod Vodárenskou věží 2, 182 07 Praha 8
e-mail: lanz@cs.cas.cz

Ladislav Lukšan, prof. Ing., DrSc.
Ústav informatiky AV ČR, v.v.i.
Pod Vodárenskou věží 2, 182 07 Praha 8
e-mail: luksan@cs.cas.cz

Josef Malík, doc. RNDr., CSc.
Ústav geoniky AV ČR, v.v.i.
Studentská 1768, 708 00 Ostrava-Poruba
e-mail: josef.malik@ugn.cas.cz

Ivo Marek, prof. RNDr., DrSc.
Katedra matematiky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: marek@mbox.ms.mff.cuni.cz

Jiří Maryška, prof. Dr. Ing., CSc.
Výzkumné centrum: Pokročilé sanační
technologie a procesy, TU v Liberci
Studentská 2, 461 17 Liberec
e-mail: jiri.maryska@tul.cz

Ctirad Matonoha, RNDr., Ph.D.
Ústav informatiky AV ČR, v.v.i.
Pod Vodárenskou věží 2, 182 07 Praha 8
e-mail: matonoha@cs.cas.cz

Petr Mayer, doc. RNDr., Dr.
Katedra matematiky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: pmayer@mat.fsv.cvut.cz

Martin Menšík, Ing.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: martin.mensik@vsb.cz

Lukáš Mocek, Ing.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: lukas.mocek@vsb.cz

Vratislava Mošová, RNDr., CSc.
Ústav exaktních věd
Moravská vysoká škola Olomouc, o.p.s.
Jeremenkova 42, 772 00 Olomouc
e-mail: vratislava.mosova@mvso.cz

Oto Přibyl, RNDr.
Ústav matemat. a deskript. geometrie
Fakulta stavební VUT v Brně
Veveří 95, 602 00 Brno
e-mail: Pribyl.o@fce.vutbr.cz

Jan Přikryl, Dr. Ing.
Ústav aplikované matematiky
Fakulta dopravní ČVUT v Praze
Na Florenci 25, 110 00 Praha
e-mail: prikryl@fd.cvut.cz

Petr Přikryl, prof. RNDr., CSc.
Matematický ústav AV ČR, v.v.i.
Žitná 25, 115 67 Praha 1
e-mail: prikryl@math.cas.cz

Ondřej Rokoš, Ing.
Katedra mechaniky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: ondrej.rokos@fsv.cvut.cz

Miroslav Rozložník, doc. Ing., Dr.
Ústav informatiky AV ČR, v.v.i.
Pod Vodárenskou věží 2, 182 07 Praha 8
e-mail: miro@cs.cas.cz

Ivana Šebestová, Mgr.
Katedra numerické matematiky
Matematicko-fyzikální fak. UK v Praze
Sokolovská 83, 186 75 Praha 8
e-mail: ivasebestova@seznam.cz

Karel Segeth, prof. RNDr., CSc.
Matematický ústav AV ČR, v.v.i.
Žitná 25, 115 67 Praha 1
e-mail: segeth@math.cas.cz

Jakub Šístek, Ing., Ph.D.
Matematický ústav AV ČR, v.v.i.
Žitná 25, 115 67 Praha 1
e-mail: sistek@math.cas.cz

Martina Smitková, Ing.
Katedra matematiky
Fakulta aplikovaných věd ZČU v Plzni
Univerzitní 22, 306 14 Plzeň
e-mail: smitkova@kma.zcu.cz

Vojtěch Sokol, Ing.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: sokol.vojtech@gmail.com

Pavel Šolín, RNDr., Ph.D.
Ústav termomechaniky AV ČR, v.v.i.
Dolejškova 5, 182 00 Praha 8
e-mail: solin@unr.edu

Martin Soukenka, Ing.
Katedra matematiky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
and
Ústav termomechaniky AV ČR, v.v.i.
Dolejškova 5, 182 00 Praha 8
e-mail: soukenkam@mat.fsv.cvut.cz

Petr Sváček, doc. RNDr., Ph.D.
Ústav technické matematiky
Fakulta strojní ČVUT v Praze
Karlovo náměstí 13, 121 35 Praha 2
e-mail: psvacek@gmail.com

Tomáš Svatoň, Ing.
Katedra matematiky
Fakulta aplikovaných věd ZČU v Plzni
Univerzitní 22, 306 14 Plzeň
e-mail: tomass@kma.zcu.cz

Petr Tichý, RNDr., Ph.D.
Ústav informatiky AV ČR, v.v.i.
Pod Vodárenskou věží 2, 182 07 Praha 8
e-mail: tichy@cs.cas.cz

Jiří Vala, prof. Ing., CSc.
Ústav matemat. a deskript. geometrie
Fakulta stavební VUT v Brně
Veveří 95, 602 00 Brno
e-mail: Vala.J@fce.vutbr.cz

Tomáš Vejchodský, RNDr., Ph.D.
Matematický ústav AV ČR, v.v.i.
Žitná 25, 115 67 Praha 1
e-mail: vejchod@math.cas.cz

Miloslav Vlasák, RNDr.
Katedra numerické matematiky
Matematicko-fyzikální fak. UK v Praze
Sokolovská 83, 186 75 Praha 8
e-mail: vlasakmila@gmail.com

Jan Vlček, prom. mat., CSc.
Ústav informatiky AV ČR, v.v.i.
Pod Vodárenskou věží 2, 182 07 Praha 8
e-mail: vlcek@cs.cas.cz

Vít Vondrák, doc. Mgr., Ph.D.
Katedra aplikované matematiky
Fakulta elektrotechniky a informatiky
VŠB–TU Ostrava
17. listopadu 15, 708 33 Ostrava-Poruba
e-mail: vit.vondrak@vsb.cz

Vratislav Žabka, Ing.
Ústav nových technologií a aplik. infor.
Fak. mechat., infor. a meziobor. studií
Technická univerzita v Liberci
Studentská 2, 461 17 Liberec
e-mail: vratislav.zabka@tul.cz

Jan Zeman, doc. Ing., Ph.D.
Katedra mechaniky
Fakulta stavební ČVUT v Praze
Thákurova 7, 166 29 Praha 6
e-mail: zemanj@cml.fsv.cvut.cz

Jan Zítko, doc. RNDr., CSc.
Katedra numerické matematiky
Matematicko-fyzikální fak. UK v Praze
Sokolovská 83, 186 75 Praha 8
e-mail: zitko@karlin.mff.cuni.cz