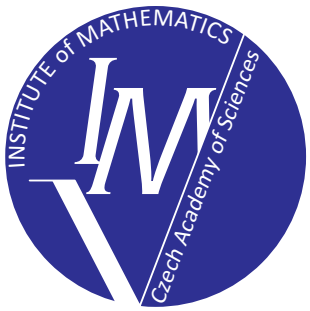
Programs and Algorithms of Numerical Mathematics 18

Janov nad Nisou, June 19–24, 2016

Proceedings of Seminar

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



Institute of Mathematics
Czech Academy of Sciences
Prague 2017

ISBN 978-80-85823-67-7
Matematický ústav AV ČR, v. v. i.
Praha 2017

Contents

Preface

This book comprises papers that originated from the invited lectures, survey lectures, short communications, and posters presented at the 18th seminar Programs and Algorithms of Numerical Mathematics (PANM) held in Janov nad Nisou, Czech Republic, June 19–24, 2016. All the papers have been peer-reviewed.

The seminar was organized by the Institute of Mathematics of the Czech Academy of Sciences under the auspices of EU-MATHS-IN.cz, Czech Network for Mathematics in Industry. It continued the previous seminars on mathematical software and numerical methods held (biennially, with only one exception) in Alšovice, Bratříkov, Janov nad Nisou, Kořenov, Lázně Libverda, Dolní Maxov, and Prague in the period 1983–2014. The objective of this series of seminars is to provide a forum for presenting and discussing advanced theoretical as well as practical topics in numerical analysis, computer implementation of algorithms, new approaches to mathematical modeling, and single- or multi-processor applications of computational methods.

More than 50 participants from the field took part in the seminar, most of them from Czech universities and from institutes of the Czech Academy of Sciences, and also from abroad. The participation of a significant number of young scientists, PhD students, and also some undergraduate students is an established tradition of the PANM seminar and it was observed this year, too. We do believe that those, who took part in the PANM seminar for the first time, have found the milieu of the seminar friendly and stimulating, and are going to join the PANM community.

The organizing committee consisted of Jan Chleboun, Pavel Kůs, Petr Přikryl, Karel Segeth, Jakub Šístek, and Tomáš Vejchodský. Ms 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 the papers. A photograph of the seminar participants in front of Achát (Agate) Hotel is included, too.

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.

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

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

A. C. CLARKE'S SPACE ODYSSEY AND NEWTON'S LAW OF GRAVITY

Stanislav Bartoň, Lukáš Renčín

Mendel University in Brno
Faculty of AgriSciences, Department of Technology and Automobile Transportation
Zemědělská 1, 61300 Brno, Czech Republic
barton@mendelu.cz

Abstract: In his famous tetralogy, Space Odyssey, A. C. Clarke called the calculation of a motion of a mass point in the gravitational field of the massive cuboid a classical problem of gravitational mechanics. This article presents a proposal for a solution to this problem in terms of Newton's theory of gravity. First we discuss and generalize Newton's law of gravitation. We then compare the gravitational field created by the cuboid — monolith, with the gravitational field of the homogeneous sphere. This is followed by the calculation of the shape of free fall trajectories and the solving of Newton's equations of motion, defining the motion of the mass point in the monolith's gravitational field for general initial conditions. The final section describes the procedures for calculating the shape of the monolith's equipotential surfaces. Due to the complexity of the problems, all calculations are performed in the Maple program. The results of the calculations are illustrated using both 2D and 3D graphs.

Keywords: law of gravity, gravitational force, potential, power, acceleration, equations of motion, trajectory, Maple

MSC: 68W30, 70F15, 34A34, 65Z05, 65D30

1. Introduction

In A. C. Clarke's Space Odyssey, namely [3] and [4], he dealt, among other things, with the motion of a spacecraft in the gravitational field of a cuboid with dimensions in the ratio of 1:4:9. He even calls the calculation of the trajectory a classical problem of gravitational mechanics. Let us try to solve this problem using classical mechanics and Newton's equations of motion.

2. Newton's law of gravitation

Let us take two mass pointes m_1 and m_2. The first point is at the origin of the coordinate system, and the second point is on the coordinates $[x, y, z]$. Accord-

DOI: 10.21136/panm.2016.01

ing to Newton's law of gravitation, [5], the two points attract each other with a force $\vec{F}$:

$$\vec{F} = \frac{\kappa\, m_1\, m_2}{x^2+y^2+y^2}\,\vec{e}_{12}, \quad \text{where} \quad \begin{array}{rcl} \kappa & = & \text{Newton's gravitational constant} \\ \vec{e}_{12} & = & \text{the unit vector of the line connecting } m_1 \text{ and } m_2 \end{array} \tag{1}$$

This law can be generalized with a complex equation for the calculation of the gravitational pull of two homogeneous spheres of identical mass.

First, let us assume a gravitational pull of the homogeneous sphere with a mass M_1, and radius R, with a mass point m_2 located on the coordinates $[X, 0, 0]$, $X > R$. Interestingly, it is impossible to achieve the expected result $|F| = \kappa\, m_2\, M_1\, X^{-2}$ by integration of the $|F| = \kappa\, m_2 \int_{M_1} X^{-2}\, dM_1$; it is necessary to calculate the potential energy of the mass point m_2 in the gravitational field of the sphere M_1.

2.1. Potential energy

The potential energy of mass point m_2 in the gravitational field generated by mass point m_1, is equal to the work required to move it from its current position to infinity. Let us assume that mass point m_2 is moved along a general, parametrically defined spatial curve $S = [x(p), y(p), z(p)]$, where p = parameter. If we move point m_2 by $d\vec{S}$, then it is possible using equation (1) to determine the corresponding element of work dW, according to the relationship $dW = \vec{F} \cdot d\vec{S}$,

$$dW = \kappa\, m_1\, m_2 \int_p^{\infty} \frac{\frac{\mathrm{d}\, R(p)}{\mathrm{d}p}}{R(p)^2}\,\mathrm{d}p \;\Rightarrow\; W = \frac{\kappa\, m_1\, m_2}{R(p)}, \quad \text{where } R(p)=\sqrt{x(p)^2+y(p)^2+z(p)^2}. \tag{2}$$

If we know the potential energy W, then it is possible to determine the gravitational force using the relation $\vec{F} = -\nabla\,(W)$.

2.2. Gravitational force between a mass point and a homogeneous sphere

Now we can calculate the potential energy of mass point dM, on coordinates $[X, Y, Z]$ in the gravitational field of the homogeneous sphere with a radius R and mass m, which is located at the origin of the coordinate system. The position of the mass element of the sphere is entered in cylindrical coordinates as shown in Fig. 1, the density of the sphere mass $\rho = \frac{3\,m}{4\,\pi\,R^3}$.

$$W = \kappa\,\mathrm{d}M \int_{-R}^{R}\left(\int_0^{\sqrt{R^2-x^2}}\left(\int_0^{2\pi} \frac{\rho\, r}{\sqrt{(X-x)^2+r^2}}\,\mathrm{d}\phi\right)\mathrm{d}r\right)\mathrm{d}x \Rightarrow F = -\frac{\kappa\, m\,\mathrm{d}M}{X^2}. \tag{3}$$

From Eq. (3) we can easily prove that the gravitational interaction between the homogeneous sphere and the mass point is determined by the same law as the gravitational interaction between two mass pointes, see Eq. (1). The generalization of two homogeneous spheres is based on the repetition of the above process. If mass point dM can move freely in the sphere's gravitational field, then the gravitational force will grant its acceleration $\vec{A} = \frac{\mathrm{d}\vec{F}}{\mathrm{d}M}$.

2.3. Gravitational field inside the homogeneous sphere

Let us suppose that the point P is located inside the homogeneous sphere. It is then possible to divide all the mass of the sphere that is at a greater distance from the center of the sphere S than from the point P into concentric spherical shells of elementary thickness $\mathrm{d}r$. We can then pass a line through the point P which is the axis of an elementary cone with apex $\mathrm{d}\phi$. Intersection points of this cone with a shell can be considered as two mass points, and we can prove that their gravitational forces on the point P cancel each other out. As a result, the only gravitational force affecting the point P comes from the spherical mass that is closer to the center of the sphere than the distance of the point P. The detailed computation can be found in [1]. The gravitational acceleration A_s inside and outside the homogeneous sphere with a radius R_s and mass M, equal to the mass of the monolith, is given by the following equation:

$$A_s = \frac{\kappa\, d\, M}{R_s^3} \text{ for } d \le R_s,\ A_s = \frac{\kappa\, M}{R_s^2} \text{ for } d \ge R_s, \text{ where } \begin{array}{l} R_s = \frac{3T}{\sqrt[3]{\pi}}, \\ T = \begin{array}{l}\text{length of the shortest} \\ \text{edge of the monolith}\end{array} \end{array} \tag{4}$$

3. Gravitational field of the monolith from Space Odyssey

We will calculate the monolith's gravitational acceleration $\vec{A} = [A_x, A_y, A_z]$ in the program Maple 13. To save space, we will only show the calculation for the acceleration coordinate A_x, calculations for Ay and A_z are very similar.

3.1. Monoliths gravitational force

If the acceleration vector is to be in units SI, or $[A]_{SI} = \mathrm{m\,s^{-2}}$, then we must multiply it by the density of the material; let us assume that $\rho = 2000\,\mathrm{kg\,m^{-3}}$ and Newton's gravitational constant $\kappa = 6.6710^{-11}\,\mathrm{m^3\,s^{-2}\,kg^{-1}}$, that is $1.33410^{-7}\,\mathrm{s^{-2}}$. Since this value is very small and it is multiplied by all members of the acceleration vector, it is not necessary to perform this multiplication, but all resulting acceleration and speed values must be multiplied by this constant in comparison with the actual acceleration values. The method also changes the speed or length of time steps. Given that this is a relative comparison of individual accelerations, velocities and displacements, these facts can be ignored.

```
> restart; with(plots): with(LinearAlgebra): R:=sqrt((X-x)^2+(Y-y)^2+(Z-z)^2):
> Lx:=9*T/2; Ly:=2*T; Lz:=T/2; Ix1:=Int((X-x)/R^3,x): Ix1:=simplify(Eval(Ix1,x=Lx)-Eval(Ix1,x=-Lx)):
> Ix1:=value(Ix1): Ix2:=Int(Ix1,y=-Ly..Ly); Ix2:=combine(value(Ix2),ln,symbolic);
```

$$Ix2 = \ln\left(\frac{(\%1+\%4)(\%2+\%5)}{(\%1+\%5)(\%2+\%4)}\right), \text{ where } \begin{array}{l} \%1 = -2Y-4T \\ \%2 = -2Y+4T \\ \%3 = 4X^2+97T^2+4Y^2+16YT+4Z^2+4z^2-8Zz \\ \%4 = \sqrt{\%3+36XT} \\ \%5 = \sqrt{\%3-36XT} \end{array} . \tag{5}$$

The final calculation of the acceleration component A_x cannot be performed by integration according to z, because this integral does not have an analytical solution. We will therefore create a procedure for A_x that will perform a numerical integration

for the coordinates $[X, Y, Z]$ on which the mass point m is located, with an accuracy of 10 significant figures. For the integration, it necessary to enter the length of the shortest edge of the monolith, let us suppose it's $T = 10km$, as exact values are not listed in any one of the books [1-4].

```
> T:=1e5; Ax:=(a,b,c)->evalf(Int(subs(X=a,Y=b,Z=c,Ix2),z=-Lz..Lz,epsilon=10),10);
```

Now we can plot the course of the monolith's gravitational force and compare it with the gravitational field of the homogeneous sphere with the same mass and material density. The gravitational force in the direction of axis X is plotted with a red line, in the direction of axis Y it is plotted with a blue line, in the direction of axis Z it is plotted with a green line, and the homogeneous sphere is plotted with a black line. The graph in Fig. 2 shows that the course of the gravitational force surrounding the monolith is significantly different from the spheres gravitational field.

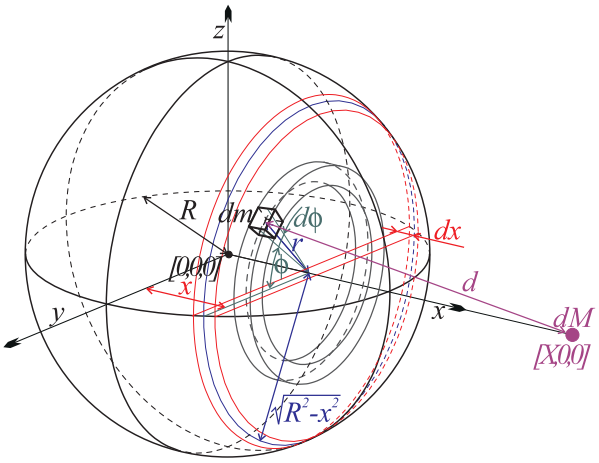


Figure 1: Gravitational interaction between sphere and dM.

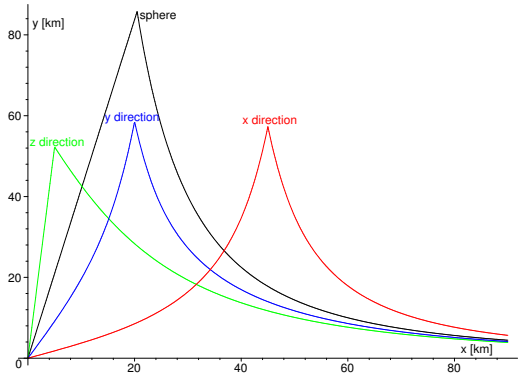


Figure 2: Gravitational force of the monolith and sphere.

4. Movement in the monolith's gravitational field

The movement of an object in the monolith's gravitational field can be solved using Newton's equations of motion written in the standard form, [5]:

$$\frac{\mathrm{d}^2\,\vec{P}\,(t)}{\mathrm{d}t^2} = \vec{A}\,(t)\,, \quad \vec{P}\,(0) = [X_0, Y_0, Z_0]\,, \quad \left.\frac{\mathrm{d}\,\vec{P}\,(t)}{\mathrm{d}t}\right|_{t=0} = [V_{x_0}, V_{y_0}, V_{z_0}]\,. \tag{6}$$

This is a system of non-linear second order differential equations, which have no analytical solution in this case. Although the Maple program is equipped with an extensive library of numerical solvers, they cannot be used because it is necessary to calculate the gravitational acceleration in each step - see Eq. (5) and the following

text. Because this vector must be calculated by numerical integration, we cannot use the direction command `dsolve` in Maple. First, we will convert Eq. (6) to a system of first order differential equations corresponding with the fact that $\frac{d\,\vec{P}(t)}{dt} = \vec{V}\,(t)$ and $\frac{\mathrm{d}\,\vec{V}(t)}{\mathrm{d}t} = \vec{A}\,(t)$.

Now we can create a procedure `Step`, based on the Runge-Kutta method, which will create a numerical solution to Eq. (6) with regard to the initial conditions. This procedure is very simillar to procedure described in [2]. The procedure checks the length of the spatial step. If the step is two short or long compared to the default value `DL`, the time step changes the `dt`.

```
> FV:=(t,X,Y,Z,Vx,Vy,Vz)->[Vx,Vy,Vz]; FA:=(t,a,b,c,Vx,Vy,Vz)->-
   [evalf(Int(subs(X=a,Y=b,Z=0,Ix2),z=Lz..Lz,epsilon=10),10),
    evalf(Int(subs(X=a,Y=b,Z=0,Iy2),x=-Lx..Lx,epsilon=10),10),
    evalf(Int(subs(X=a,Y=b,Z=c,Iz2),y=-Ly..Ly,epsilon=10),10)];
> Step:=proc(nu) local pv1,pv2,pv3,pv4,vv1,vv2,vv3,vv4,DPV,dl; global n,t,dt,Pos,PV,W,VV,Tau;
    pv1:=dt*evalf(FV(t,Pos[],W[])): vv1:=dt*evalf(FA(t,Pos[],W[])):
    pv2:=dt*evalf(FV(t+dt/2,(Pos+pv1/2)[],(W+vv1/2)[])):
    vv2:=dt*evalf(FA(t+dt/2,(Pos+pv1/2)[],(W+vv1/2)[])):
    pv3:=dt*evalf(FV(t+dt/2,(Pos+pv2/2)[],(W+vv2/2)[])):
    vv3:=dt*evalf(FA(t+dt/2,(Pos+pv2/2)[],(W+vv2/2)[])):
    pv4:=dt*evalf(FV(t+dt,(Pos+pv3)[],(W+vv3)[])):
    vv4:=dt*evalf(FA(t+dt,(Pos+pv3)[],(W+vv3)[])): #.......................... New Lines for StepPE
    DPV:=1/6*(pv1+2*pv2+2*pv2+pv4): dl:=sqrt(add(w^2,w=DPV));
    if dl>DL then dt:=dt/2; elif dl*8<DL then dt:=dt*2;
      else n:=n+1; t:=t+dt; Pos:=Pos+DPV; PV:=[PV[],Pos];
           W:=W+1/6*(vv1+2*vv2+2*vv3+vv4): VV:=[VV[],W];Tau:=[Tau[],t];
  end if: end proc;
```

Let us suppose we let a small object fall from the resting state of $Z = 0$, from a distance of 4 times the length of the edge L_y, from places where the position angle is changed by 10°. Each time step will be stored in variable `TTau`, velocity vectors in variable `TVV` and position vectors in variable `TPV`.

```
> Nu:=9; TPV:=[]: TVV:=[]: TTau:=[]:
> for i from 0 to Nu do;
    Pos:=[4*Ly*cos(pi*i/2/Nu),4*Ly*sin(pi*i/2/Nu),0]; W:=[0,0,0]; #...............Initial Conditions
    t:=0; dt:=0.125; DL:=1000; n:=0; PV:=[Pos]; VV:=[W]; Tau:=[0];
    while not(abs(Pos[1])<Lx and abs(Pos[2])<Ly and Pos[3]<Lz) do; Step(); end do:
    TPV:=[TPV[],PV]: TVV:=[TVV[],VV]; TTau:=[TTau[],Tau];
  end do:
```

The free fall trajectory for the remaining planes can be calculated in a similar manner $X = 0$ a $Y = 0$, see Fig. 3. The plotting commands are not listed here. The free fall trajectories are marked red, the trajectory tangents in the initial point of the fall are marked green, and the free fall trajectory in central gravitational field is marked brown. The figure shows that motion in the monolith's gravitational field and in the sphere's gravitational field varies considerably.

The equations of motion (6) can also be used for a general case of motion in the monolith's gravitational field. It only differs from the free fall in the choice of initial conditions. For calculating the trajectories we can use procedure `Step`. Let us suppose the motion in a plane $Z = 0$, with an initial position of $[Lx, Ly, 0]$ and the

velocity vector oriented in the direction of axis X; we will perform the calculations for 8 velocities that are constantly increasing. Remaining Maple commands are not listed.

```
> Pos:=[Lx,Lx,0]; W:=[-i*6000,0,0]; #..............Initial Conditions .......
```

As Fig. 4 shows, the body either hits the monolith or flies around it on the equivalent of a hyperbolic orbit. We can therefore assume that finding stable orbits around the monolith can be a problem that is difficult to solve. This assumption is supported by other calculations for different initial conditions. In the previous calculation, we can simply replace the line marked as `#Initial Conditions`, with the following line:

```
> Pos:=[Lx,Lx,3*Lz]; W:=[(-10600-i*200)/sqrt(2),0,(10600+i*200)/sqrt(2)]; #.....Initial Conditions
```

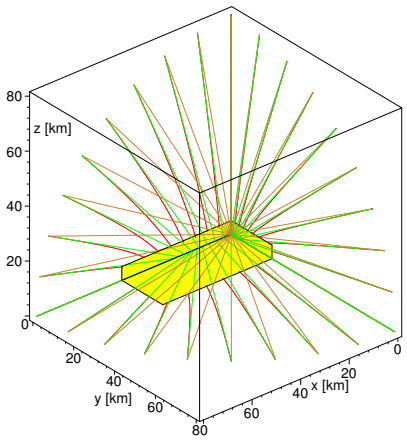


Figure 3: Free fall in the monolith's gravitational field.

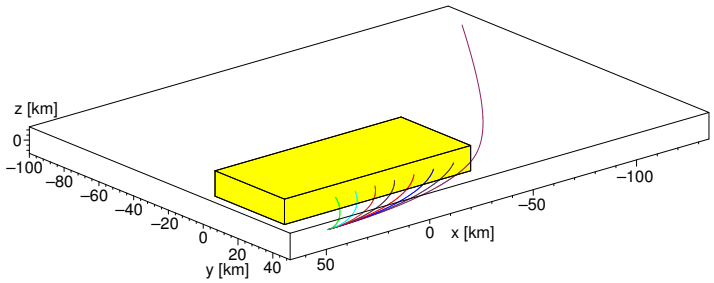


Figure 4: Motion in the plane $Z = 0$.

As Fig. 5 shows, the body moves along quite unusual trajectories, which ultimately lead it to fall on the monolith. If the initial velocity is slightly changed, the corresponding trajectories are close to each other, which is confirmed by the correctness of the calculations above.

5. Equipotential surfaces

The calculation of the shape of equipotential surfaces can be based on Eq. (2), which implies that the gravitational force is always perpendicular to the equipotential surface. We will therefore perform calculations of the shape of equipotential surfaces for the plane of symmetry of the monolith; the intersection of the plane of symmetry with the equipotential surface defines the curve that can be easily depicted thanks to the validity of Eq. (2). First, we must determine the distance of the equipotential surface from the center of the monolith. For this we can use the law of conservation

of energy. From the center of the monolith - the beginning of the coordinate system, we shoot a mass point in the direction of each axis at a known speed. The point at which its motion stops will correspond with the equipotential surface with a potential of $U = v^2/2$. The coordinates of this point can be found using the procedure `StepPE`. Since it is necessary to determine the point at which the mass point stops, it is necessary to change the lines of procedure `Step`, that follow `# New Lines for StepPE`, because it is not possible to optimize the length of the spatial step when it stops. The corresponding line of procedure `StepPE` is:

```
Pos:=Pos+1/6*(pv1+2*pv2+2*pv2+pv4): W:=W+1/6*(vv1+2*vv2+2*vv3+vv4):
```

At the stopping point, we can determine the direction of the gravitational field using numerical integration of Eq. (2), and in the plane of symmetry we can determine the direction that is perpendicular to this direction. There will be another point of the equipotential surface in this direction. This is basically determining the tangent trajectory of the direction field - solving the system of first order differential equations. A numerical solution can be found using the procedure `StepEQ`:

```
> StepEQ:=proc(0) local pv1, pv2, pv3, pv4, DPV, dl; global n, t, dt, Pos, PV, Tau;
   pv1:=dt*evalf(FA(t,Pos[])): pv2:=dt*evalf(FA(t+dt/2,(Pos+pv1/2)[])):
   pv3:=dt*evalf(FA(t+dt/2,(Pos+pv2/2)[])): pv4:=dt*evalf(FA(t+dt,(Pos+pv3)[])):
   DPV:=1/6*(pv1+2*pv2+2*pv2+pv4): dl:=sqrt(add(w^2,w=DPV));
   if dl>DL then dt:=dt/2; elif dl*8<DL then dt:=dt*2;
     else n:=n+1; t:=t+dt; DPV:=map(u->`if`(u=0,0,signum(u)*DPV[abs(u)]),0);
        Pos:=Pos+DPV; PV:=[PV[],Pos]; Tau:=[Tau[],t];
  end if: end proc;
```

The coordinates of points on equipotential surfaces can then be calculated using the following commands:

```
> TPV:=[]: i:='i':
> for i from 25 to 50 do;
    W:=[0,i*1000,0]; Pos:=[0,0,0]; VV:=[W];  t:=0; dt:=0.005; n:=0; PV:=[Pos]; Tau:=[0]; DL:=1500;
      while W[2]>0 do; StepPE(); end do:
      while abs(dt)>1e-5 do; dt:=-W[2]/FA(t,Pos[],W[])[2]; StepPE(); end do:
      PV:=[Pos];dt:=0.025; while Pos[2]>0 do; StepEQ([-2,1,0]): end do:
      TPV:=[TPV[],map(u->u[1..2],PV)];
  end do:
```

The plotting commands are not listed in order to save space. The commands for calculating equipotential surfaces corresponding with remaining planes of symmetry are not listed for the same reason. The equipotential surfaces are plotted in Fig. 6.

6. Conclusion and discussion

The results of our calculations are seemingly contrary to what we know about Earth's gravitational field. Especially the free fall trajectories in Fig. 3 and the unstable orbit trajectories in Fig. 5 are quite strange. These differences can be explained quite simply. In the sphere's gravitational field the free fall trajectory is identical to the local vertical, which does not apply to the monolith.

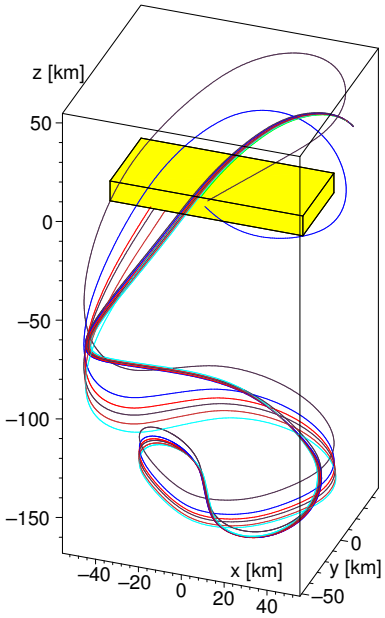


Figure 5: General orientation of the initial velocity.

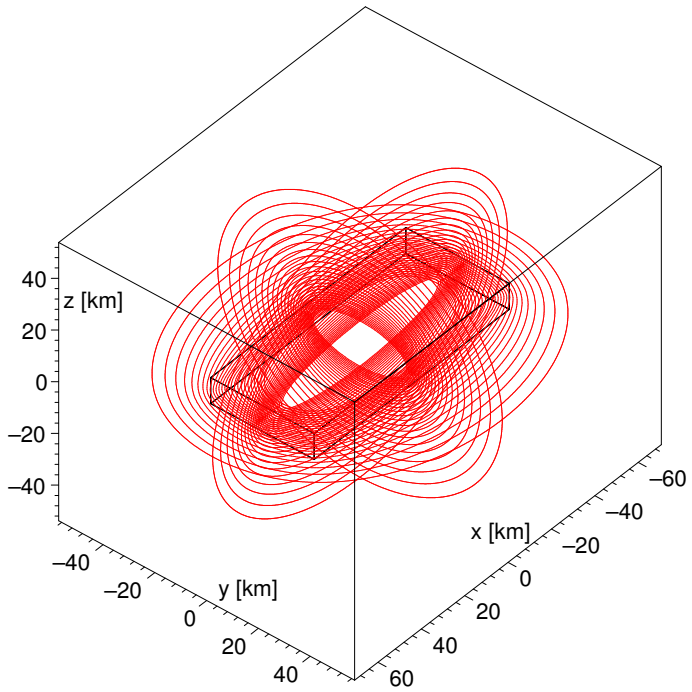


Figure 6: Equipotential levels.

Moreover, the gravitational force of the sphere's gravitational field is always directed towards the center of the object that is the source of gravity. The monolith's gravitational field does not meet this condition, which is why an object moving away from the monolith, e.g. in the direction of axis Z, can oscillate in all directions perpendicular to this axis, which is quite evident from Fig. 5.

Acknowledgements

The presented work has been prepared with the support of IGA MENDELU IP 10/2016 Verification of the model force action in a three-point hitch.

References

[1] Bartoň, S.: *Fyzika I v řešených příkladech.* In Czech. Mendel University, Brno, 94–98, 2011.

[2] Bartoň, S. and Krumpholc, T: Driver's influence on kinematics of articulated bus rear axle. In: J. Chleboun, K. Segeth, J. Šístek, and T. Vejchodský (Eds.), *Programs and Algorithms of Numerical Mathematics 16.* Institute of Mathematics, Prague, pp. 9–14., 2013.

[3] Clarke, A. C.: *A space odyssey.* Hutchinson, UK, 224p., 1969.

[4] Clarke, A. C.: *Odyssey two.* Granada Publishing, UK, 291 p., 1982.

[5] Newton, I.: *Philosopiae naturalis principia mathematica.* 1687, English translation 1729.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

GOAL ORIENTED A POSTERIORI ERROR ESTIMATES FOR THE DISCONTINUOUS GALERKIN METHOD

Vít Dolejší, Filip Roskovec

Charles University, Faculty of Mathematics and Physics
Sokolovská 83, Czech Republic
dolejsi@karlin.mff.cuni.cz, roskovec@gmail.com

Abstract: This paper is concerned with goal-oriented a posteriori error estimates for discontinous Galerkin discretizations of linear elliptic boundary value problems. Our approach combines the Dual Weighted Residual method (DWR) with local weighted least-squares reconstruction of the discrete solution. This technique is used not only for controlling the discretization error, but also to track the influence of the algebraic errors. We illustrate the performance of the proposed method by numerical experiments.

Keywords: discontinuous Galerkin method, aposteriori error estimates, goal-oriented estimates, algebraic error

MSC: 65M60, 65M70

1. Introduction

Usually a posteriori error estimation techniques measure the error in a norm which is connected with the mathematical formulation of the problem being solved. The concern in practical application may be quite different. The main purpose of the computation may be to calculate a quantity of interest, expressed in the mathematical language as a functional applied to the solution of the solved problem (e.g. drag or lift in the airflow simulations). The dual weighted residual (DWR) method first proposed by Rannacher et al., (for a survey, see e.g. [2]), suggests a way how to connect the error of the target quantity with the solved problem. This is enabled by solving the so-called dual (or adjoint) problem.

Our main goal is to employ the DWR method for designing an efficient adaptive algorithm for solving stationary partial differential equations. We focus on the Poisson problem with Dirichlet boundary conditions in this paper, but most of the work can be extended even to nonlinear problems. Even though the DWR method can be combined with any discretization technique based on the variational formulation, we focus mainly on the discontinuous Galerkin (DG) method, which is based on discontinuous piece-wise polynomial approximation.

DOI: 10.21136/panm.2016.02

Solving the additional dual problem may lead to an increase in the computational effort. Moreover, the dual solution needs to be in a space $V_h^+ \supsetneq V_h$, where V_h is the original discrete space. On the other hand, the ability of measuring directly the error of the target quantity reduces the computational efforts compared to other estimation techniques.

There are several possibilities for discretization of the dual problem. One can solve the dual problem on a globally refined mesh with higher polynomial degree, see e.g. [8]. This approach gives very precise results, but the computational effort used to solve the dual problem exceeds the cost of the original problem dramatically. Therefore, we present an algorithm based on higher-order reconstruction presented originally in [5], which can be computed locally and therefore much more efficiently.

Further, the presented method naturally allows to integrate estimation of the algebraic errors arising from inexact solution of both the primal and dual problems. We present estimates enabling to keep the discretization and algebraic errors in balance.

Finally, we compare the performance of the presented goal-oriented error estimation method with a classical (not goal-oriented) a posteriori error estimate and we examine the influence of the algebraic errors by a numerical experiment.

2. Problem description

Let $\Omega \in \mathbb{R}^2$ be a bounded polygonal domain. We consider the Poisson equation

$$-\Delta u = f \quad \text{in } \Omega, \tag{1a}$$

$$u = u_D \quad \text{on } \partial\Omega \tag{1b}$$

where $u : \Omega \to \mathbb{R}$ is an unknown scalar function defined on Ω.

We use the standard notation for the Lebesgue spaces $L^p(\Omega)$, Sobolev spaces $W^{k,p}(\Omega)$, $H^k(\Omega) = W^{k,2}(\Omega)$ and $H_0^1(\Omega)$ for the subspace of $H^1(\Omega)$ containing functions with vanishing traces of $\partial\Omega$. Moreover, the space of polynomial functions up to the degree k defined on a domain $M \subset \mathbb{R}^2$ is denoted by $P^k(M)$. We assume that $f \in L^2(\Omega)$ and u_D is trace of some $u^* \in H^1(\Omega) \cap L^\infty(\Omega)$ on $\partial\Omega$.

We say that function $u \in H^1(\Omega)$ is the *weak solution* of problem (1) if it satisfies

$$\begin{aligned} &u - u^* \in H_0^1(\Omega), \\ &a(u, \varphi) = l(\varphi) \quad \forall \varphi \in H_0^1(\Omega), \end{aligned} \tag{2}$$

where $a(u, \varphi) := \int_\Omega \nabla u \cdot \nabla \varphi \, \mathrm{d}x$, $l(\varphi) := \int_\Omega f \varphi \, \mathrm{d}x$, $u, \varphi \in H^1(\Omega)$.

3. Discretization of the problem

Let $\mathcal{T}_h$ be a partition covering $\overline{\Omega}$ consisting of finite number of closed triangles K with mutually disjoint interiors. The boundary of the element $K \in \mathcal{T}_h$ will be denoted by ∂K and its diameter by $h_K = \operatorname{diam} K$.

We assume that there exists $h_0 > 0$ such that $\{\mathcal{T}_h\}_{h\in(0,h_0)}$ is a system of triangulations is *shape-regular* and *locally quasi-uniform*. We do not require the conforming properties known from finite element methods. Therefore, the triangulations $\mathcal{T}_h$ could contain so called *hanging nodes*. Over the triangulation $\mathcal{T}_h$ we define the so-called *broken Sobolev space* over the triangulation $\mathcal{T}_h$ as $H^s(\Omega, \mathcal{T}_h) = \{v \in L^2(\Omega),\, v\big|_K \in H^s(K),\, \forall K \in \mathcal{T}_h\}$.

The DWR method can be combined with any discretization technique based on the variational formulation of the solved problem such as finite element method, finite volumes method or discontinuous Galerkin.

Here we focus only on the discontinuous Galerkin (DG) method. Since the DG method is very convenient for *hp*-adaptation, we assign to each $K \in \mathcal{T}_h$ its local polynomial degree p_K. Then we define vector $\mathtt{p} := \{p_K;\, K \in \mathcal{T}_h\}$ and as the finite dimensional discrete space we use

$$S_h^{\mathtt{p}} = \{v \in L^2(\Omega);\, v\big|_K \in P^{p_K}(K)\, \forall K \in \mathcal{T}_h\}. \tag{3}$$

We say that the function $u_h \in S_h^{\mathtt{p}}$ is the approximate solution of (2) if

$$a_h(u_h, \varphi_h) = l_h(\varphi_h) \quad \forall \varphi_h \in S_h^{\mathtt{p}}, \tag{4}$$

where $a_h(\cdot,\cdot)$ is a bilinear form resulting from the DG discretization of the problem (2) and l_h is a linear form representing the right-hand side of the equation enriched by some terms resulting from the DG method.

The Dirichlet boundary condition (1b) is not enforced directly, but it is integrated in the forms a_h and l_h by a penalty term. Detailed introduction of this method is not necessary for this paper, so we only stress out the important properties of the DG method when needed. Precise definitions of the forms and description of the properties of the method can be found in the monograph [3].

The crucial requirement on this method needed in this article is to be *consistent,* i.e. the exact solution u of problem (2) also satisfies

$$a_h(u, \varphi) = l_h(\varphi) \quad \forall \varphi \in H^2(\Omega, \mathcal{T}_h). \tag{5}$$

For the purpose of the higher-order reconstruction presented in Section 5 we also define the space $S_h^{\mathtt{p}+1} := \{v \in L^2(\Omega);\, v\big|_K \in P^{p_k+1}(K)\, \forall K \in \mathcal{T}_h\}$. Obviously $S_h^{\mathtt{p}} \subset S_h^{\mathtt{p}+1} \subset H^2(\Omega, \mathcal{T}_h)$.

4. Dual weighted residual method

Our goal is to estimate the error $J(u) - J(u_h)$, where $J : V \cup S_h^{\mathtt{p}} \to \mathbb{R}$ is a linear functional representing the so-called quantity of interest. Typically, this functional represents a regularized value of the solution (or its derivative) in a given point in Ω or an integral over a part of Ω or its boundary $\partial\Omega$.

The trick which enables to plug the functional J into the computation (similarly as the Aubin-Nietche trick used to prove the optimal rate of convergence in L^2-norm)

is the introduction of the so-called *dual* (or adjoint) *problem.* In addition to the original problem we introduce the dual problem:

$$\text{find } z \in V \text{ such that } a_h(\psi, z) = J(\psi) \quad \forall \psi \in H^2(\Omega, \mathcal{T}_h). \tag{6}$$

And also its discrete variant:

$$\text{find } z_h \in S_h^{\mathtt{p}} \text{ such that } a_h(\psi_h, z_h) = J(\psi_h) \quad \forall \psi_h \in S_h^{\mathtt{p}}. \tag{7}$$

Remark 1. *Problem* (6) *may look a bit artificial since it contains the bilinear form* $a_h(\cdot,\cdot)$ *from the discretized problem* (4)*. One could also consider directly the dual problem to the weak formulation* (2)*, i.e.* $a(\psi, z) = l(\psi)$, $\forall \psi \in V$*. If the discretization method is* dually consistent*, i.e. the discrete dual problem is a consistent discretization of the weak dual formulation, then these two definitions coincide. It has been shown in [6] that the dual consistency is essential in order to maintain the optimal convergence order of the method. Dual consistency is maintained only for the symmetric variant of DG method known as SIPG – symmetric interior penalty Galerkin, see [3] for details.*

The following manipulation gives us a relation between the error of the quantity of interest and the residual of the solved problem. Thanks to linearity of J, consistency of the DG scheme and the Galerkin orthogonality $a_h(u - u_h, \varphi_h) = a_h(\varphi_h, z - z_h) = 0 \, \forall \varphi_h \in S_h^{\mathtt{p}}$, we get

$$\begin{aligned} J(u) - J(u_h) &= J(u - u_h) = a_h(u - u_h, z) = a_h(u - u_h, z - \varphi_h) \\ &= l_h(z - \varphi_h) - a_h(u_h, z - \varphi_h) =: r_h(u_h)(z - \varphi_h) \quad \forall \varphi_h \in S_h^{\mathtt{p}} \end{aligned} \tag{8}$$

and very similarly

$$\begin{aligned} J(u) - J(u_h) &= a_h(u - u_h, z - z_h) = a_h(u - \psi_h, z - z_h) \\ &= J(u - \psi_h) - a_h(u - \psi_h, z_h) =: r_h^*(z_h)(u - \psi_h) \quad \forall \psi_h \in S_h^{\mathtt{p}}. \end{aligned} \tag{9}$$

Hence the residuals $r_h(u_h)(\cdot)$ and $r_h^*(z_h)(\cdot)$ are equivalent in the following way

$$r_h(u_h)(z - \varphi_h) = r_h^*(z_h)(u - \psi_h) \quad \forall \varphi_h, \psi_h \in S_h^{\mathtt{p}}. \tag{10}$$

Unfortunately, even the "exact" discrete solution satisfying (4) is not available in practical computations due to algebraic errors. Instead, we compute their approximation u_h^a and z_h^a typically resulting from a finite number of iterations of an iterative solver. In this case, Galerkin orthogonality property is violated and hence identities (8) and (9) need to be revised. Similarly to [1], by adding the algebraic error to (8), we get

$$\begin{aligned} J(u) - J(u_h^a) &= a_h(u - u_h^a, z - z_h^a) + a_h(u - u_h^a, z_h^a) \\ &= r_h(u_h^a)(z - z_h^a) + r_h(u_h^a)(z_h^a), \end{aligned} \tag{11}$$

and rewriting (9) gives

$$\begin{aligned} J(u) - J(u_h^a) &= a_h(u - u_h^a, z - z_h^a) + a_h(u - u_h^a, z_h^a) \\ &= a_h(u, z - z_h^a) - a_h(u_h^a, z - z_h^a) + a_h(u - u_h^a, z_h^a) \\ &= l_h(z) - a_h(u, z_h^a) - (J(u_h^a) - a_h(u_h^a, z_h^a)) + l_h(z_h^a) - a_h(u_h^a, z_h^a) \\ &= r_h^*(z_h^a)(u) - r_h^*(z_h^a)(u_h^a) + r_h(u_h^a)(z_h^a). \end{aligned} \tag{12}$$

Here, the expressions $r_h(u_h^a)(z - z_h^a)$ and $r_h^*(z_h^a)(u)$ represent the discretization error, while $r_h(u_h^a)(z_h^a)$ and $r_h^*(z_h^a)(u_h^a)$ represent the algebraic errors of the primal and dual problem, respectively. Unlike $r_h^*(z_h^a)(u_h^a)$ and $r_h(u_h^a)(z_h^a)$, expressions $r_h(u_h^a)(z - z_h^a)$ and $r_h^*(z_h^a)(u)$ are not computable and have to be further approximated.

5. Approximation of the exact solutions u and z

Except for a few very special examples (see e.g. [2, Chapter 3]) exact solution of the dual problem is not computable and has to be approximated. Since the residuals of the (algebraically exact) approximate solutions u_h and z_h equal to zero for all functions from $S_h^{\mathtt{p}}$, functions approximating u and z must be from a richer space than $S_h^{\mathtt{p}}$, otherwise the error estimates (8) and (9) would degenerate.

The standard approach is to compute the dual problem on a finer mesh and/or with higher polynomial degree. To avoid this costly procedure we exploit a higher order reconstruction of the discrete solutions u_h and z_h, which can be obtained locally and hence much faster. We use the weighted least-square reconstruction, firstly presented in [5].

Let $u_h \in S_h^{\mathtt{p}}$ be the approximate solution of (4). We compute the reconstruction $u_h^+ \in S_h^{\mathtt{p}+1}$ locally for each element $K \in \mathcal{T}_h$ by a weighted least square approximation from the elements sharing at least a vertex with K. We denote this patch of elements $\mathscr{D}_K = \{K' \in \mathcal{T}_h;\ K' \cap K \neq \emptyset\}$.

We compute the function $\boldsymbol{U}_K^+ \in P^{p_K+1}(\mathscr{D}_K)$ by

$$\boldsymbol{U}_K^+ = \underset{U_h \in P^{p_K+1}(\mathscr{D}_K)}{\arg\min} \sum_{K' \in \mathscr{D}_K} \omega_{K'} \|U_h - u_h\|^2_{H^1(K')}. \tag{13}$$

Then we assemble the higher-order reconstruction u_h^+ as an element-wise composition of $\boldsymbol{U}_K^+\big|_K$, i.e. $u_h^+ = \sum_{K \in \mathcal{T}_h} \boldsymbol{U}_K^+\big|_K$.

When choosing the values of the weights $\omega_{K'}$, we distinguish between elements sharing a face and elements having only a common vertex. We set $\omega_{K'} = 1$ if $K' = K$ or if K, K' share a face and $\omega_{K'} = \varepsilon$ if K, K' share only a vertex. The parameter $\varepsilon > 0$ is chosen to be considerably smaller than one.

The computation of z_h^+ is done alike, using function z_h.

6. Error estimates

Exploiting (11), (12) and the reconstructions defined in Section 5, we introduce two kinds of discretization error estimators

$$\eta_{\mathrm{S}} := r_h(u_h^a)(z_h^+ - z_h^a), \quad \eta_{\mathrm{S}}^* := r_h^*(z_h^a)(u_h^+). \tag{14}$$

Furthermore, we define algebraic error estimators

$$\eta_{\mathrm{A}} := r_h(u_h^a)(z_h^a), \qquad \eta_{\mathrm{A}}^* := r_h^*(z_h^a)(u_h^a), \tag{15}$$

which measure the influence of the algebraic errors arising from the inexact solution of the primal and the dual problem, respectively. Since we do not have in hands the true error, but only its approximation, we proceed with iterations of the Krylov solver until these algebraic estimators decrease significantly (10–1000 times) under the level of the discretization error.

Using the definitions of the error estimators (14) and (15) and the relations (11) and (12), we can write the error estimates

$$J(u) - J(u_h^a) \approx \eta_{\mathrm{S}} + \eta_{\mathrm{A}} \tag{16}$$

and

$$J(u) - J(u_h^a) \approx \eta_{\mathrm{S}}^* - \eta_{\mathrm{A}}^* + \eta_{\mathrm{A}}. \tag{17}$$

The functional J has not the properties of a norm and can attain both positive and negative values on different elements. Hence, we have to separate the estimate of the error, where we avoid overestimation, and the local error indicators that have to be positive at each element. Therefore, we define

$$\eta_{\mathrm{S,K}} = |r_h(u_h^a)((z_h^+ - z_h^a)\big|_K)|, \qquad \eta_{\mathrm{S,K}}^* = |r_h^*(z_h^a)(u_h^+\big|_K)|, \qquad K \in \mathcal{T}_h. \tag{18}$$

Either of those can be used as a local error indicator for mesh refinement. Although the primal and dual residuals are theoretically equivalent, see (10), localizations (18) can differ notably and hence may lead to differently refined meshes.

7. Numerical experiments

The problem we solve comes from [7]. We consider Poisson problem

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega = (0,1) \times (0,1) \\ u\big|_{\partial\Omega} &= 0, \end{aligned} \tag{19}$$

and we set the primal and dual right-hand sides

$$f(v) = -\int_{T_{\mathrm{f}}} \frac{\partial v}{\partial x_1}\,\mathrm{d}x, \qquad J(v) = -\int_{T_{\mathrm{g}}} \frac{\partial v}{\partial x_1}\,\mathrm{d}x, \tag{20}$$

where T_f and T_g are triangles with vertices $T_\mathrm{f} = \{(0,0),(0.5,0),(0,0.5)\}$ and $T_\mathrm{g} = \{(1,1),(0.5,1),(1,0.5)\}$, see Figure 1.

Both f and J are in $H^{-1}(\Omega)$ and the solutions have singularities along the lines connecting points $[0,0.5],[0.5,0]$ and $[0.5,1],[1,0.5]$ for the primal and dual problems, respectively. We discretized the problem by SIPG DG method with piece-wise quadratic polynomials. We compare the numerical results obtained by goal-oriented estimates with u_h^+, z_h^+ computed with the least-squares reconstruction (DWR) and with globally increased polynomial degree (DWR_P). The third approximation was computed with a classical (not goal-oriented) error estimation technique (RES). This method, first proposed in [4], estimates dual norm of the residual of the discrete solution.

Meshes after 25 steps of mesh adaptation are showed in Figure 1. In each adaptation step we refined 10% of elements with the largest local error indicators. In Figure 2, we compare the decrease of the error $J(u) - J(u_h^a)$ for all three algorithms on adaptively refined meshes. The adaptive RES technique does not take into account the singularity of the dual problem. For this reason it does not refine the mesh in the upper-right corner and it cannot decrease the error of the target quantity bellow the level 10^{-6}. The goal-oriented algorithm reduces the error more steadily. The computation with the least-squares reconstruction behaves comparably to the more expensive algorithm DWR_P. Our goal-oriented algorithm almost achieves the optimal theoretical rate of convergence $O((\#\mathcal{T}_h)^{-2})$ proved in [7].

In the second experiment, Figure 3, we compare the decrease of the true error of the quantity of interest $J(u) - J(u_h^a)$ with estimates $\eta_\mathrm{S}, \eta_\mathrm{S}^*$ of the discretization error and estimates $\eta_\mathrm{A}, \eta_\mathrm{A}^*$ of the algebraic errors given by (14)–(15). These results were obtained by the SIPG method with quadratic polynomials on fixed uniform mesh with 256 elements. At each step (outer iterations, which are marked on the horizontal axis in Figure 3) we simultaneously performed 8 iterations of the algebraic solver (GMRES with ILU preconditioning) for primal and 50 iterations for the dual problem, respectively. In other words, at the outer step i, the situation after $i \times 8$ and $i \times 50$ steps of the algebraic solver for the primal and the dual problem, respectively, is plotted.

Figure 3 nicely illustrates the relations (12) and (17). Since we perform more iterations of the algebraic solver in each (outer) step, thus both $|z_h - z_h^a|$ and the

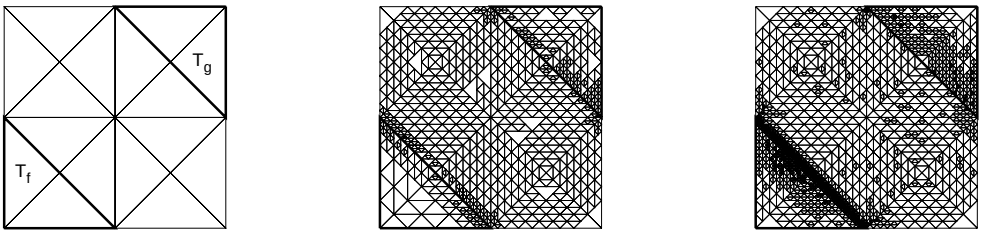


Figure 1: The initial mesh (left) and final meshes produced by the DWR method (center) and DWR_P method (right), respectively.

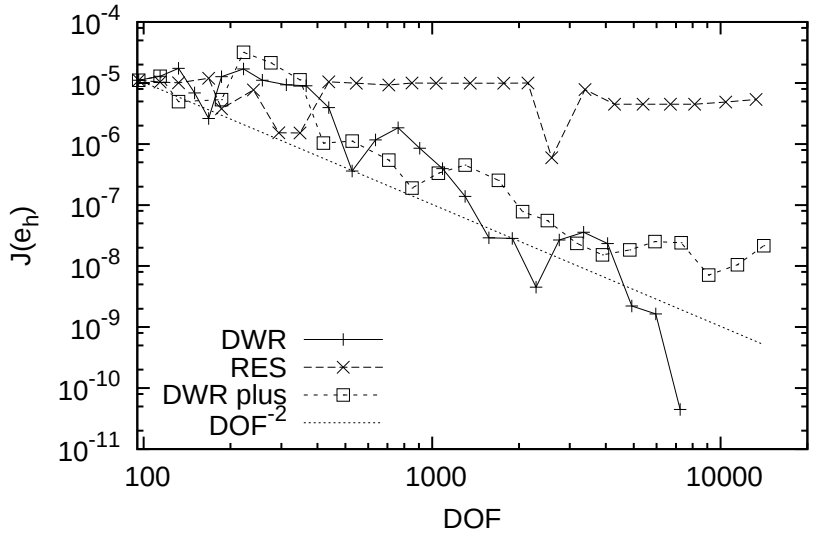


Figure 2: Error convergence of the RES, DWR and DWR_P methods compared to the theoretical rate.

algebraic error of the dual problem ($\approx \eta_{\mathrm{A}}^*$) is negligible already after a few steps. On the other hand, the algebraic error of the primal problem ($\approx \eta_{\mathrm{A}}$) decreases slower. Therefore, in the first 10 steps the error of the target quantity $J(u) - J(u_h^a)$ is mainly caused by the algebraic errors in the solution of the primal problem, hence its decrease corresponds to the decrease of η_{A}. Only when the estimate η_{A} decreases under the level of the discretization error, $J(u) - J(u_h^a)$ stops at the level of $J(u) - J(u_h) \approx \eta_{\mathrm{S}}^*$.

For example in the step 6 (i.e, after 48 and 300 iterations of the algebraic solver for the primal and dual problem, respectively) both the error $J(u) - J(u_h^a)$ and the estimate η_{S} are still strongly influenced by the algebraic errors of u_h^a. On the the contrary, the dual estimate η_{S}^* is already at the level of the exact discretization error $J(u) - J(u_h)$. In other words, the algebraical inexactness in the discrete solution u_h^a influences the primal estimate η_{S} more seriously than the dual estimate η_{S}^*.

Similar (but reversed) behavior was observed in the opposite case, when $u_h^a \approx u_h$ but z_h^a is far from z_h. This indicates that even quite inexact approximation z_h^a of z_h could be sufficient for the primal estimate η_{S}. On the other hand, if we knew that for some reason the dual algebraic problem was easier to solve, we should use the dual estimate η_{S}^* which can give better results for rough approximations of u_h. Finally, we note that this is possible only thanks to the equivalence (10) between the primal and the dual residual, which also implies that $J(u_h) = a_h(u_h, z_h) = l_h(z_h)$, hence we are able to obtain an approximation of $J(u)$ even without computing u_h at all.

Acknowledgements

The research of V. Dolejší was supported by Grant No.: 13-00522S of the Czech Science Foundation. The research of F. Roskovec was supported by the Charles University, project GA UK No. 92315.

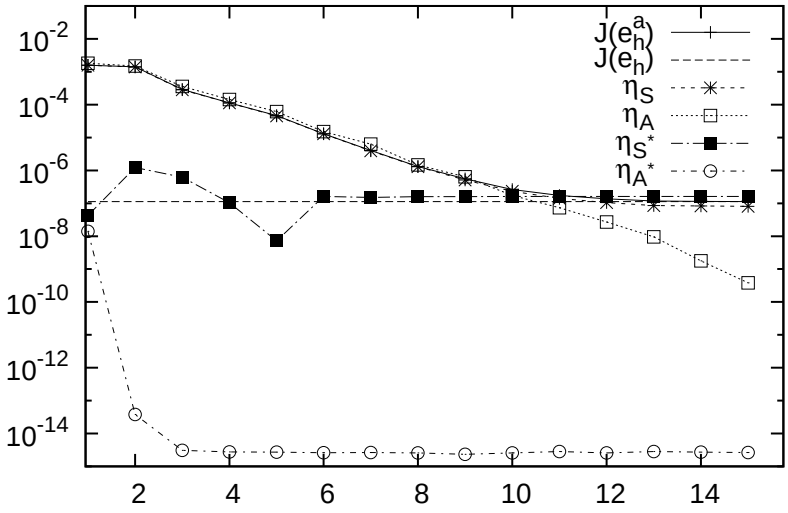


Figure 3: Decrease of error estimates during iterations of the algebraic solver.

References

[1] Arioli, M., Liesen, J., Miedlar, A., and Strakoš, Z.: Interplay between discretization and algebraic computation in adaptive numerical solution of elliptic PDE problems. GAMM-Mitteilungen **36** (2013), 102–129.

[2] Bangerth, W. and Rannacher, R.: *Adaptive Finite element methods for differential equations*. Lectures in Mathematics. ETH Zürich, 2003.

[3] Dolejší, V. and Feistauer, M.: *Discontinuous Galerkin Method – analysis and applications to compressible flow*. Springer Verlag, 2015.

[4] Dolejší, V.: *hp*-DGFEM for nonlinear convection-diffusion problems. Math. Comput. Simulation **87** (2013), 87–118.

[5] Dolejší, V. and Solin, P.: *hp*-discontinuous Galerkin method based on local higher order reconstruction. Appl. Math. Comput. **279** (2016), 219–235.

[6] Hartmann, R.: Adjoint consistency analysis of discontinuous galerkin discretizations. SIAM J. Numer. Anal. **45** (2007), 2671–2696.

[7] Mommer, M.S. and Stevenson, R.: A goal-oriented adaptive finite element method with convergence rates. SIAM J. Numer. Anal. **47** (2009), 861–886.

[8] Šolín, P. and Demkowicz, L.: Goal-oriented *hp*-adaptivity for elliptic problems. Comput. Methods Appl. Mech. Engrg. **193** (2004), 449–468.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

REMARKS ON INVERSE OF MATRIX POLYNOMIALS

Cyril Fischer, Jiří Náprstek

Institute of Theoretical and Applied Mechanics, CAS, v.v.i.
Prosecká 76, Prague 9, Czech Republic
fischerc@itam.cas.cz, naprstek@itam.cas.cz

Abstract: Analysis of a non-classically damped engineering structure, which is subjected to an external excitation, leads to the solution of a system of second order ordinary differential equations. Although there exists a large variety of powerful numerical methods to accomplish this task, in some cases it is convenient to formulate the explicit inversion of the respective quadratic fundamental system. The presented contribution uses and extends concepts in matrix polynomial theory and proposes an implementation of the inversion problem.

Keywords: matrix polynomial, polynomial eigenvalues, structural vibration

MSC: 65H10, 15A15, 15A18, 30C10, 34M03

1. Introduction

Solution of matrix differential equations is closely associated to the theory of matrix polynomials. The very important class of the second order matrix differential equations has a wide variety of applications, among others in vibration analysis in civil or mechanical engineering or in the analysis of oscillation circuits in electrical engineering. The motivation behind this contribution originates from the vibration analysis of non-classically damped engineering structures, which are subjected to a random external excitation. In case of non-stationary excitation, the numerical integration of the differential system gives only a very limited information on the stochastic character of the response. In such cases it is more convenient to formulate the exact or approximate analytical solution, if possible, and to use it for an assessment of the stochastic properties of the system response. Such a procedure is provided by, e.g., the spectral decomposition method [2].

The behaviour of the structure is described by a relation:

$$\mathbf{A}\ddot{\mathbf{U}}(\omega,t)+\mathbf{B}\dot{\mathbf{U}}(\omega,t)+\mathbf{C}\mathbf{U}(\omega,t)=\mathbf{f}(\omega,t) \tag{1}$$

where the coefficient matrices $\mathbf{A},\mathbf{B},\mathbf{C}\in\mathbb{R}^{n\times n}$ are considered to be constant, real and symmetric, $\mathbf{U}(\omega,t)$ is a deterministic function describing transformation of the

DOI: 10.21136/panm.2016.03

random excitation and $\mathbf{f}$ describes properties of the random excitation. The Laplace transform changes the differential system into an algebraic one

$$(\mathbf{A}p^2 + \mathbf{B}p + \mathbf{C})\mathbf{U}^\star(\omega, p) = \mathbf{Q}(p)\mathbf{U}^\star(\omega, p) = \mathbf{f}^\star(\omega, p)\,, \tag{2}$$

whose solution is given by

$$\mathbf{U}^\star(\omega, p) = \mathbf{Q}(p)^{-1}\mathbf{f}^\star(\omega, p)\,. \tag{3}$$

The inverse of the matrix polynomial $\mathbf{Q}(p)^{-1}$ can be written in a form of a sum [1]

$$\mathbf{Q}(p)^{-1} = \sum_{j=1}^{n} \left(\mathbf{S}_j \frac{1}{p - p_j} + \overline{\mathbf{S}}_j \frac{1}{p - \overline{p}_j} \right) \tag{4}$$

where p_j are the roots of $\det \mathbf{Q}(p)$ (generalized eigenvalues of $\mathbf{Q}$) and matrices $\mathbf{S}_j$ are rank 1 matrices related to the generalized eigenvectors of $\mathbf{Q}$. Solution of (1) is finally given as

$$\mathbf{U}(\omega, t) = \sum_{j=1}^{2n} \mathbf{S}_j \int_0^t \mathsf{e}^{p_j(t-\tau)} \mathbf{f}(\omega, \tau) \mathsf{d}\tau. \tag{5}$$

In the following section, the basics of matrix polynomial theory will be introduced according to the monograph by Gohberg et al. [1]. Sections 3 and 4 will be devoted to the lemma which leads to an advantageous formulation of matrices $\mathbf{S}_j$ in (4) and to a computational algorithm.

2. Basics of the matrix polynomials theory

Definition. *Let* $l > 0$ *and* $\mathbf{A}_j \in \mathbb{R}^{n\times n}$, $j = 0, \ldots, l$, $\mathbf{A}_l \neq 0$ *be square matrices. The* matrix polynomial $\mathbf{L}(\lambda)$ of degree l *is defined as*

$$\mathbf{L}(\lambda) = \sum_{j=0}^{l} \mathbf{A}_j \lambda^j \ . \tag{6}$$

An eigenvalue λ *of the matrix polynomial* $\mathbf{L}(\lambda)$ *is the solution of*

$$\mathbf{L}(\lambda) = 0 \quad \textit{or} \quad \det \mathbf{L}(\lambda) = 0\,, \tag{7}$$

whilst the corresponding (right) eigenvector $\mathbf{x}$ *and* left eigenvector $\mathbf{y}$ *is any non-zero solution of*

$$\mathbf{L}(\lambda)\mathbf{x} = \sum_{j=0}^{l} \mathbf{A}_j \lambda^j \mathbf{x} = 0 \quad \textit{resp.} \quad \mathbf{y}^\mathsf{T}\mathbf{L}(\lambda) = \sum_{j=0}^{l} \mathbf{y}^\mathsf{T}\mathbf{A}_j \lambda^j = 0\ . \tag{8}$$

Two matrix polynomials $\mathbf{M}(\lambda)$ *and* $\mathbf{N}(\lambda)$ *are* equivalent, $\mathbf{M}(\lambda) \simeq \mathbf{N}(\lambda)$, *if there exist two matrix polynomials* $\mathbf{E}(\lambda)$ *and* $\mathbf{F}(\lambda)$ *with constant determinants such that*

$$\mathbf{M}(\lambda) = \mathbf{E}(\lambda)\mathbf{N}(\lambda)\mathbf{F}(\lambda). \tag{9}$$

A linearization *of a matrix polynomial* $\mathbf{L}(\lambda)$ *of dimension* n *and degree* l *is a linear matrix polynomial* $\mathbf{E}\lambda - \mathbf{H}$ *of dimension* nl *where*

$$\mathbf{L}(\lambda) \simeq (\mathbf{E}\lambda - \mathbf{H})\,. \tag{10}$$

The linearization matrix $\mathcal{A} \in \mathbb{R}^{nl\times nl}$ *is such a matrix that* $\mathbf{L}(\lambda) \simeq \mathbf{I}\lambda - \mathcal{A}$.

The concept of linearization is traditionally used for computation of eigenvalues of a matrix polynomial using standard methods for the linear eigenvalue problem [4]. The linearization is not uniquely defined. However, all linearizations share the same set of eigenvalues. The commonly used linearization assumes $\mathbf{E} = \mathbf{I}$ and uses a block--matrix $\mathbf{H}$ consisting of terms $-\mathbf{A}_l^{-1}\mathbf{A}_i, i = 0, \ldots, l-1$ in the last row and identity matrices in positions of the first superdiagonal. However, there exist also other forms, suitable for particular purposes. One of the most interesting examples is the symmetric linearization, which assures symmetry of the matrices $\mathbf{E}$ and $\mathbf{H}$ due to symmetry in individual matrices $\mathbf{A}_i$, see [3].

Definition. *A* standard pair *of a matrix polynomial is a pair of matrices* $(\mathbf{X}, \mathbf{T})$, $\mathbf{X} \in \mathbb{C}^{n\times nl}, \mathbf{T} \in \mathbb{C}^{nl\times nl}$ *such that the matrix* $\mathbf{Z}$ *of dimension* $(nl \times nl)$, *where*

$$\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{X}\mathbf{T} \\ \vdots \\ \mathbf{A}_l\mathbf{X}\mathbf{T}^{l-1} \end{pmatrix}$$

is regular and

$$\sum_0^l \mathbf{A}_j\mathbf{X}\mathbf{T}^j = 0\ .$$

The standard pairs are not unique. However, if $\mathbf{T}$ is diagonal (or in a Jordan form in the case where some eigenvalues have higher multiplicity), the matrix $\mathbf{X}$ will be uniquely defined. Its columns will be formed by eigenvectors corresponding to the respective eigenvalues. Such a standard pair $(\mathbf{X}, \mathbf{T})$ is called a *Jordan pair*.

Definition A Jordan triple *is called a triple of matrices* $(\mathbf{X}, \mathbf{T}, \mathbf{Y})$, *where* $(\mathbf{X}, \mathbf{T})$ *is a Jordan pair and* $\mathbf{Y} \in \mathbb{C}^{nl\times n}$ *satisfies:*

$$\begin{aligned} \mathbf{X}\mathbf{T}^i\mathbf{Y} &= \mathbf{0} \quad i = 0, \ldots, l-1\ , \\ \mathbf{A}_l\mathbf{X}\mathbf{T}^{l-1}\mathbf{Y} &= \mathbf{I}\ . \end{aligned} \tag{11}$$

3. Inverse of matrix polynomial

Lemma 1. *Let all eigenvalues of the matrix polynomial* $\mathbf{L}(\lambda)$ *be non-zero and the leading coefficient matrix be regular. Then the rows of the matrix* $\mathbf{Y}$ *of the Jordan triple* $(\mathbf{X}, \mathbf{T}, \mathbf{Y})$ *form the left eigenvectors of* $\mathbf{L}(\lambda)$, *i.e.*

$$\sum_{j=0}^{k} \mathbf{T}^j\mathbf{Y}\mathbf{A}_j = 0\ . \tag{12}$$

Proof: Let $l = 2$. The proof assumes the linearization

$$\left[- \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix} \lambda + \begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{A}_0 & -\mathbf{A}_1 \end{pmatrix} \right].$$

Let $\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{XT} \end{pmatrix}$. Because $(\mathbf{X}, \mathbf{T})$ is a Jordan pair it holds that

$$\begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{A}_0 & -\mathbf{A}_1 \end{pmatrix} \mathbf{Z} = \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix} \mathbf{ZT}. \tag{13}$$

Right multiplication by $\mathbf{Z}^{-1}$ and further transformation leads to

$$\mathbf{Z}^{-1} \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{A}_2^{-1} \end{pmatrix} \begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{A}_0 & -\mathbf{A}_1 \end{pmatrix} = \mathbf{TZ}^{-1}. \tag{14}$$

Let $\mathbf{T} = \begin{pmatrix} \mathbf{T}_1 & 0 \\ 0 & \mathbf{T}_2 \end{pmatrix}$ and $\mathbf{Z}^{-1} = \begin{pmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 \\ \mathbf{Z}_3 & \mathbf{Z}_4 \end{pmatrix}$, where $\mathbf{Z}_i$ $i = 1, \ldots, 4$ are square blocks of dimension n. Expansion of the last expression (14) gives

$$\begin{pmatrix} -\mathbf{Z}_2\mathbf{A}_2^{-1}\mathbf{A}_0 & \mathbf{Z}_1 - \mathbf{Z}_2\mathbf{A}_2^{-1}\mathbf{A}_1 \\ -\mathbf{Z}_4\mathbf{A}_2^{-1}\mathbf{A}_0 & \mathbf{Z}_3 - \mathbf{Z}_4\mathbf{A}_2^{-1}\mathbf{A}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{T}_1\mathbf{Z}_1 & \mathbf{T}_1\mathbf{Z}_2 \\ \mathbf{T}_2\mathbf{Z}_3 & \mathbf{T}_2\mathbf{Z}_4 \end{pmatrix} . \tag{15}$$

Now, comparing first columns

$$\begin{aligned} \mathbf{Z}_1 &= -\mathbf{T}_1^{-1}\mathbf{Z}_2\mathbf{A}_2^{-1}\mathbf{A}_0 \ , \\ \mathbf{Z}_3 &= -\mathbf{T}_2^{-1}\mathbf{Z}_4\mathbf{A}_2^{-1}\mathbf{A}_0 \ , \end{aligned}$$

and substituting into the second columns (15) then writing in the matrix form leads to

$$-\mathbf{T}^{-1} \begin{pmatrix} \mathbf{Z}_2\mathbf{A}_2^{-1} \\ \mathbf{Z}_4\mathbf{A}_2^{-1} \end{pmatrix} \mathbf{A}_0 - \begin{pmatrix} \mathbf{Z}_2\mathbf{A}_2^{-1} \\ \mathbf{Z}_4\mathbf{A}_2^{-1} \end{pmatrix} \mathbf{A}_1 = \mathbf{T} \begin{pmatrix} \mathbf{Z}_2 \\ \mathbf{Z}_4 \end{pmatrix} .$$

Denoting $\mathbf{Y} = \mathbf{Z}^{-1} \begin{pmatrix} 0 \\ \mathbf{A}_2^{-1} \end{pmatrix} = \begin{pmatrix} \mathbf{Z}_2\mathbf{A}_2^{-1} \\ \mathbf{Z}_4\mathbf{A}_2^{-1} \end{pmatrix}$, and multiplying by the matrix $\mathbf{T}$ from the left hand side leads to

$$-\mathbf{YA}_0 - \mathbf{TYA}_1 = \mathbf{T}^2\mathbf{YA}_2. \quad \square$$

The proof for general l can be performed in a similar manner: the key step is the expansion of the $\mathbf{Z}^{-1} = \big(\mathbf{Z}_1 \ldots \mathbf{Z}_l\big)$, where $\mathbf{Z}_i$ $i = 1, \ldots, l$ are the column blocks.

In the next section, it will be supposed that $\mathbf{A}_l$ is regular. The inverse matrix polynomial can be written using its Jordan triple in the following form [1]:

$$(\mathbf{L}(\lambda))^{-1} = \mathbf{X}(\lambda\mathbf{I} - \mathbf{T})^{-1}\mathbf{Y} \tag{16}$$

If $\mathbf{T}$ is diagonal, e.g., if all eigenvalues λ_i are distinct, it holds

$$(\lambda\mathbf{I} - \mathbf{T})^{-1} = \operatorname{diag}\left(\frac{1}{\lambda - \lambda_i}\right)$$

and equation (16) can be rewritten as

$$(\mathbf{L}(\lambda))^{-1} = \sum_{j=1}^{ln} \frac{1}{\lambda - \lambda_j} \mathbf{x}_j \mathbf{y}_j^{\mathsf{T}} \tag{17}$$

where $\mathbf{x}_j$ are columns of $\mathbf{X}$ and $\mathbf{y}_j^{\mathsf{T}}$ are rows of $\mathbf{Y}$.

By respecting the character of the underlying physical problem, it is possible to assume that all matrices $\mathbf{A}_i$ are symmetrical and that $\mathbf{T}$ is diagonal and regular with distinct elements. The matrix $\mathbf{Y}$ is defined by the conditions (11). It remains to show that there exists a matrix $\mathbf{D}$ such that $\mathbf{Y}^{\mathsf{T}}\mathbf{D} = \mathbf{X}$.

$$\mathbf{0}^{\mathsf{T}} = \left(\sum \mathbf{A}_i \mathbf{Y}^{\mathsf{T}} \mathbf{D} \mathbf{T}^i\right)^{\mathrm{T}} \tag{18}$$

$$= \sum \mathbf{T}^i \mathbf{D} \mathbf{Y} \mathbf{A}_i \tag{19}$$

$$= \sum \mathbf{D}^{\mathsf{T}^i} \mathbf{T}^i \mathbf{D} \mathbf{Y} \mathbf{A}_i = \sum \mathbf{T}^i \mathbf{Y} \mathbf{A}_i \tag{20}$$

where the symbol $\mathbf{D}^{\mathsf{T}^i}$ means i-multiple transpositions.

The last equation (20) implies symmetry of $\mathbf{D}$, i.e. $\mathbf{DT} = \mathbf{TD}$ and thus for elements d_{ij} of $\mathbf{D}$ it holds: $d_{i,j} = 0 \Leftrightarrow \frac{t_i}{t_j} \neq 1$.

This means that if the diagonal elements of $\mathbf{T}$ are distinct, the matrix $\mathbf{D}$ is diagonal and regular. The same result can be reached using a different reasoning: due to Lemma 1 the third term of the Jordan triple is formed by the left eigenvectors. For symmetric matrices $\mathbf{A}_j$ the right and left generalized eigenvectors coincide. This means that the corresponding columns of $\mathbf{Y}^{\mathsf{T}}$ and $\mathbf{X}$ differ by multiplicative constants and so the matrix $\mathbf{D}$ has to be diagonal.

Under the assumptions introduced above, it is possible to find such eigenvectors $\mathbf{X}$ that $(\mathbf{X}, \mathbf{T}, \mathbf{X}^{\mathsf{T}})$ forms the Jordan triple. The conditions (11) attain the form:

$$\begin{aligned} \mathbf{X}\mathbf{T}^i\mathbf{X}\mathsf{T} &= 0 \quad i = 0, \ldots, l-1 \ , \\ \mathbf{A}_l\mathbf{X}\mathbf{T}^{l-1}\mathbf{X}^{\mathsf{T}} &= \mathbf{I} \ . \end{aligned} \tag{21}$$

The only unknown step in the procedure is selection of the proper scaling constants of the eigenvectors $\mathbf{X}$.

4. Formulation of the algorithm

The inverse of a matrix polynomial $\mathbf{L}(\lambda)$ can be formulated using the following procedure

1. Solve the linear eigenvalue problem with some linearization matrix to obtain a pair of matrices $(\widetilde{\mathbf{X}}, \mathbf{T})$.

2. Find the diagonal matrix $\mathbf{D}$ such that $\mathbf{X} = \widetilde{\mathbf{X}}\mathbf{D}$ and $(\mathbf{X}, \mathbf{T})$ satisfy (21). Its existence was proven before.
   According to (11) we now have

$$\begin{pmatrix} \widetilde{\mathbf{X}}\mathbf{D}\widetilde{\mathbf{X}}^{\mathsf{T}} \\ \mathbf{A}_2\widetilde{\mathbf{X}}\mathbf{T}\mathbf{D}\widetilde{\mathbf{X}}^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{I} \end{pmatrix} . \tag{22}$$

   Substituting $\Delta = \widetilde{\mathbf{X}}\mathbf{D}$ into (22) the equation transforms into

$$\begin{pmatrix} \widetilde{\mathbf{X}} \\ \mathbf{A}_2\widetilde{\mathbf{X}}\mathbf{T} \end{pmatrix} \Delta^{\mathsf{T}} = \begin{pmatrix} 0 \\ \mathbf{I} \end{pmatrix} . \tag{23}$$

   Because $(\Delta)_{ij} = x_{ij}d_j$ and x_{ij} are known, it is sufficient to solve the system (23) for only one column of Δ and the corresponding column of the right hand side. Selection of such a column depends on the distribution of non-zero elements of rows of the matrix $\widetilde{\mathbf{X}}$.
3. The diagonal elements of $\mathbf{D}$ are computed as ratios

$$d_{ii} = \Delta_{ji}/x_{ij} \ , \tag{24}$$

   supposing that the j-th column has been used. Finally, set $\mathbf{X} = \mathbf{Y}^{\mathsf{T}} = \widetilde{\mathbf{X}}\sqrt{\mathbf{D}}$.
4. The inverse of the matrix polynomial can be computed using relation (17) where both $\mathbf{x}_j$ and $\mathbf{y}_j^{\mathsf{T}}$ are columns of $\mathbf{X}$.

Acknowledgements

The kind support of the Czech Science Foundation project No. 15-01035S and the institutional support No. RVO 68378297 are gratefully acknowledged.

References

[1] Gohberg, I., Lancaster, P., and Rodman, L.: *Matrix polynomials*. SIAM Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[2] Náprstek, J. and Fischer, O.: A combined analytical-numerical method of solving the non-stationary random response of large systems excited by seismic movement. In: C. A. Brebbia and A. Cakmak (Eds.), *Proc. 6th International Conference — Soil Dynamics and Earthquake Engineering, WIT Transactions on the Built Environment*, vol. 3, pp. 667–681. Wessex Institute of Technology, Bath, UK, 1993.

[3] Tisseur, F. and Meerbergen, K.: The quadratic eigenvalue problem. SIAM Rev. **43** (2001), 235–286.

[4] Wilkinson, J. H.: *The algebraic eigenvalue problem*. Clarendon Press, Oxford, 1965.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

IMPLEMENTATION OF FULL LINEARIZATION IN SEMISMOOTH NEWTON METHOD FOR 2D CONTACT PROBLEM

Ladislav Foltyn, Oldřich Vlach

Department of Applied Mathematics, FEECS
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava, Czech Republic
ladislav.foltyn@vsb.cz, oldrich.vlach2@vsb.cz

Abstract: To solve the contact problems by using a semismooth Newton method, we shall linearize stiffness and mass matrices as well as contact conditions. The latter are prescribed by means of mortar formulation. In this paper we describe implementation details.

Keywords: semismooth Newton method, contact problem, active set strategy, mortar formulation

MSC: 74M15, 65N30

1. Introduction

After a finite element discretization, a mathematical model of the contact problem is a problem of quadratic programming with an equality and an inequality constraints in a special form, see for example [1]. If a system matrix isn't positive definite, we cannot use this approach. Positive definiteness of the system matrix may be impaired in a case of material nonlinearity and computing in increments. One of the possible solutions is the solution of a nonlinear equation system instead of a minimization problem. The inequality constraints can be also written as the equality constraints unfortunately compensated by the price of a nonsmooth function appearance in the formulation. This reformulated problem can be solved by the semismooth Newton method. Therefore, it is necessary to linearize the stiffness matrix and the contact matrices which is in detail described in [3]. The mortar contact topic is described in [4]. Our goal was to implement solution for the linearized contact problem using the semismooth Newton method in the framework of the MatSol library.

DOI: 10.21136/panm.2016.04

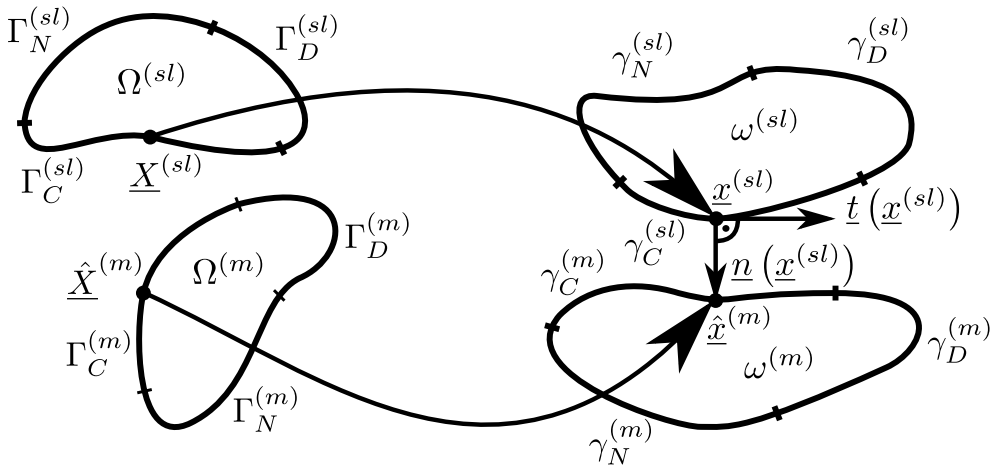


Figure 1: Reference and actual configuration.

2. Formulation of a contact problem

Consider a 2D contact problem with definite deformations of two elastic bodies (see Figure 1). Both bodies are represented as open sets $\Omega^{(sl)} \subset \mathbb{R}^2$ (a slave), $\Omega^{(m)} \subset \mathbb{R}^2$ (a master). Their boundary $\partial\Omega^{(sl)}, \partial\Omega^{(m)}$ can be divided into the following parts:

- the part with prescribed Neumann condition $\Gamma_N^{(i)}$,
- the part with prescribed Dirichlet condition $\Gamma_D^{(i)}$,
- and the part with contact boundary $\Gamma_C^{(i)}$,

where $i \in \{sl, m\}$. We will assume that all boundary parts $\Gamma_N^{(i)}$, $\Gamma_D^{(i)}$ and $\Gamma_C^{(i)}$ are mutually disjoint. We also distinguish two types of configuration in the contact problem, an actual configuration $(\omega^{(i)}, \gamma_D^{(i)}, \gamma_N^{(i)}, \gamma_C^{(i)}, \underline{x}^{(i)})$ and a reference configuration $(\Omega^{(i)}, \Gamma_D^{(i)}, \Gamma_N^{(i)}, \Gamma_C^{(i)}, \underline{X}^{(i)})$. Actual configuration of both bodies is described by a displacement vector

$$\underline{u}^{(i)} = \underline{X}^{(i)} - \underline{x}^{(i)}. \tag{1}$$

On contact boundary a gap function is introduced to define gap between the slave and the master body

$$g(\underline{X}^{(sl)}) = -\underline{n}\left(\underline{x}^{(sl)}(\underline{X}^{(sl)})\right) \cdot \left[\underline{x}^{(sl)}(\underline{X}^{(sl)}) - \hat{\underline{x}}^{(m)}(\hat{\underline{X}}^{(m)})\right], \tag{2}$$

where $\underline{n} = \underline{n}^{(sl)}$ is a normal vector of the slave surface $\gamma_C^{(sl)}$ in the actual configuration, $\hat{\underline{x}}^{(m)}$ is a projection of the slave node $\underline{x}^{(sl)}$ to the master surface $\gamma_C^{(m)}$ in the direction

of the normal vector and $\underline{X}$ is a corresponding point to $\underline{x}$ in the reference configuration. Using the gap function we are able to find a corresponding node on the master surface in the reference configuration to the slave node in the same configuration.

After a discretization process (for details see [5]), we can reformulate the contact problem with the KKT conditions to an algebraic form

$$\begin{aligned} \underline{\underline{\mathrm{K}}}\,\underline{\mathrm{d}} + \underline{\underline{\mathrm{D}}}^{\top}\underline{\mathrm{z}} - \underline{\underline{\mathrm{M}}}^{\top}\underline{\mathrm{z}} - \underline{\mathrm{f}} &= \underline{0} \\ \tilde{g}_j &\geq 0 \quad \text{(nonnegative gap between bodies)} \\ (z_n)_j &\geq 0 \\ (z_n)_j\,\tilde{g}_j &= 0 \quad \text{(the complementarity condition)} \\ (z_t)_j &= 0 \quad \text{(no friction)}, \end{aligned} \tag{3}$$

where $\underline{\underline{\mathrm{K}}}$ is a stiffness matrix, $\underline{\underline{\mathrm{D}}}$ and $\underline{\underline{\mathrm{M}}}$ are mortar contact matrices, $\underline{\mathrm{d}}$ is a displacement vector, $\underline{\mathrm{z}}$ is a vector of multiplicators and $\underline{\mathrm{f}}$ is a vector of volume and boundary forces.

If we want to calculate mortar matrices, we have to divide the contact elements to smaller parts called the contact segments (see Figure 2). It is necessary for the numerical integration of the master shape functions because one segment is connected with exactly one master and one slave element and the shape function formulas have nonchanging prescriptions. A segmentation process is based on the node projection from the slave (the master) surface to the other surface along the normal of the slave surface.

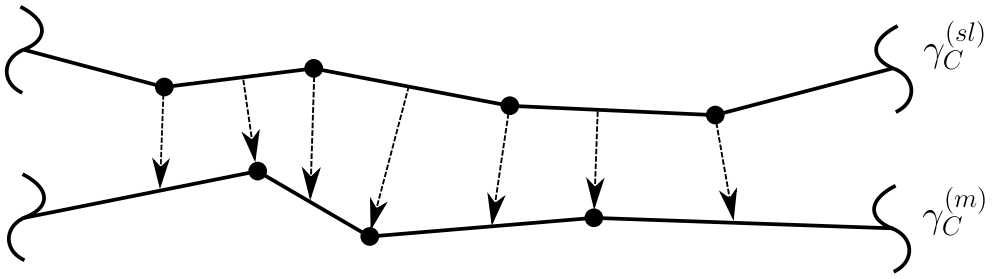


Figure 2: Surface segmentation.

3. Nonsmooth formulation for discrete problem

The semismooth Newton method solves

$$\mathbf{F}^{\circ}(\underline{\mathrm{x}}) = \underline{0}\,, \tag{4}$$

where $\mathbf{F}^{\circ}$ is a nonsmooth function, so it is necessary to transform all inequalities in (3) to equality. Moreover, the semismooth Newton method uses iterative prescription which is similar to continuous one

$$\mathbf{F}^{\circ}(\mathbf{x}^k)\Delta\mathbf{x}^k = -\mathbf{F}(\mathbf{x}^k), \quad \mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k, \tag{5}$$

but there is the nonsmooth function $\mathbf{F}^{\circ}$ and we reach only a superlinear convergence. For details see [2].

With the use of the active set strategy idea, it is possible to reformulate inequalities in (3) to equalities but there is no apriori information which nodes of the slave contact surface belong to an active set $\mathcal{A}$ or to an inactive set $\mathcal{I}$. It is known only if the solution is available. Luckily, the inequalities can be formulated as finding a zero level of the nonsmooth function. Therefore we introduce a so-called complementarity function C_j

$$C_j(\underline{\mathrm{z}}_j, \underline{\mathrm{d}}) = (\underline{\mathrm{z}}_n)_j - \max\left(0, (\underline{\mathrm{z}}_n)_j - c_n \tilde{g}_j\right) \quad \forall j \in \mathcal{S}, \tag{6}$$

where $\mathcal{S}$ denotes a set of all contact nodes of the slave surface. The zero level of the complementarity function C_j is equal with the KKT conditions

$$\left.\begin{aligned} \tilde{g}_j &\geq 0 \\ (z_n)_j &\geq 0 \quad \forall j \in \mathcal{S} \\ (z_n)_j \tilde{g}_j &= 0 \end{aligned}\right\} \Leftrightarrow C_j(\underline{\mathrm{z}}_j, \underline{\mathrm{d}}) = 0 \quad \forall j \in \mathcal{S}\ . \tag{7}$$

The function C_j is continuous but nonsmooth, we cannot determine a derivative at $(z_n)_j - c_n \tilde{g}_j = 0$. With use of the complementarity function C_j, we are able to reformulate (3) to

$$\begin{aligned} \underline{\mathrm{r}} = \underline{\underline{\mathrm{K}}}\,\underline{\mathrm{d}} + \underline{\underline{\mathrm{D}}}^{\top}\underline{\mathrm{z}} - \underline{\underline{\mathrm{M}}}^{\top}\underline{\mathrm{z}} - \underline{\mathrm{f}} = \underline{0}\ , \\ C_j\left(\underline{\mathrm{z}}_j, \underline{\mathrm{d}}\right) = 0 \quad \forall j \in \mathcal{S}\ , \\ \left(z_t\right)_j = 0 \quad \forall j \in \mathcal{S}\ . \end{aligned} \tag{8}$$

If we want to be able to linearize the function C_j, we have to define a generalized derivative of the $\max(a, x)$ function

$$f(x) = \max(a, x) \longrightarrow \Delta f(x) = \begin{cases} 0, & \text{for } x \leq a \\ 1, & \text{for } x > a \end{cases}. \tag{9}$$

Sets which are described bellow are used to determine which nodes of the slave contact surface belong to the active $\mathcal{A}_k$ or to the inactive $\mathcal{I}_k$ set in each step of the algorithm

$$\mathcal{I}_k = \left\{ j \in \mathcal{S} \middle| \left(\underline{\mathrm{n}}_j^k \cdot \underline{\mathrm{z}}_j^k - c_n \tilde{g}_j^k\right) \leq 0 \right\}\ , \tag{10}$$

$$\mathcal{A}_k = \left\{ j \in \mathcal{S} \middle| \left(\underline{\mathrm{n}}_j^k \cdot \underline{\mathrm{z}}_j^k - c_n \tilde{g}_j^k\right) > 0 \right\}\ . \tag{11}$$

Using these sets also allows us to use a block matrix notation which you can see below.

Complete linearization of the problem yields the system of linear equations

$$\begin{bmatrix} \underline{\underline{\mathrm{K}}}_{\mathcal{N}\mathcal{N}} & \underline{\underline{\mathrm{K}}}_{\mathcal{N}\mathcal{M}} & \underline{\underline{\mathrm{K}}}_{\mathcal{N}\mathcal{I}} & \underline{\underline{\mathrm{K}}}_{\mathcal{N}\mathcal{A}} & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{\mathrm{K}}}_{\mathcal{M}\mathcal{N}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{M}\mathcal{M}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{M}\mathcal{I}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{M}\mathcal{A}} & -\underline{\underline{\mathrm{M}}}_{\mathcal{I}}^{\top} & -\underline{\underline{\mathrm{M}}}_{\mathcal{A}}^{\top} \\ \underline{\underline{\mathrm{K}}}_{\mathcal{I}\mathcal{N}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{I}\mathcal{M}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{I}\mathcal{I}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{I}\mathcal{A}} & \underline{\underline{\mathrm{D}}}_{\mathcal{I}} & \underline{\underline{0}} \\ \underline{\underline{\mathrm{K}}}_{\mathcal{A}\mathcal{N}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{A}\mathcal{M}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{A}\mathcal{I}} & \underline{\underline{\tilde{\mathrm{K}}}}_{\mathcal{A}\mathcal{A}} & \underline{\underline{0}} & \underline{\underline{\mathrm{D}}}_{\mathcal{A}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{\mathrm{I}}}_{\mathcal{I}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\tilde{\mathrm{M}}}}_{\mathcal{A}} & \underline{\underline{\tilde{\mathrm{S}}}}_{\mathcal{A}\mathcal{I}} & \underline{\underline{\tilde{\mathrm{S}}}}_{\mathcal{A}\mathcal{A}} & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{\tilde{\mathrm{F}}}}_{\mathcal{A}\mathcal{I}} & \underline{\underline{\tilde{\mathrm{F}}}}_{\mathcal{A}\mathcal{A}} & \underline{\underline{0}} & \underline{\underline{\mathrm{T}}}_{\mathcal{A}} \end{bmatrix} \begin{bmatrix} \Delta\underline{\mathrm{d}}_{\mathcal{N}} \\ \Delta\underline{\mathrm{d}}_{\mathcal{M}} \\ \Delta\underline{\mathrm{d}}_{\mathcal{I}} \\ \Delta\underline{\mathrm{d}}_{\mathcal{A}} \\ \underline{\mathrm{z}}_{\mathcal{I}} \\ \underline{\mathrm{z}}_{\mathcal{A}} \end{bmatrix} = - \begin{bmatrix} \underline{\mathrm{r}}_{\mathcal{N}} \\ \underline{\mathrm{r}}_{\mathcal{M}} \\ \underline{\mathrm{r}}_{\mathcal{I}} \\ \underline{\mathrm{r}}_{\mathcal{A}} \\ \underline{0} \\ \tilde{\underline{\mathrm{g}}}_{\mathcal{A}} \\ \underline{0} \end{bmatrix}, \quad (12)$$

where elements with a tilde above are affected by a linearization process, $\mathcal{N}$ denotes all nodes which aren't on the contact surfaces, $\mathcal{M}$ denotes all nodes of the master contact surface, $\mathcal{A}$ denotes nodes of the slave contact surface which are actually in contact (active set) and $\mathcal{I}$ denotes all nodes of the slave contact surface which aren't actually in contact (inactive set).

We can also eliminate all multiplicators $\underline{\mathrm{z}}$ from the system above by using

$$\underline{\mathrm{z}}_{\mathcal{I}} = \underline{0}\,, \quad (13)$$

$$\underline{\mathrm{z}}_{\mathcal{A}} = \underline{\underline{\mathrm{D}}}_{\mathcal{A}}^{-1} \left(-\underline{\underline{\mathrm{K}}}_{\mathcal{A}\mathcal{N}} \Delta\underline{\mathrm{d}}_{\mathcal{N}} - \underline{\underline{\mathrm{K}}}_{\mathcal{A}\mathcal{M}} \Delta\underline{\mathrm{d}}_{\mathcal{M}} - \underline{\underline{\mathrm{K}}}_{\mathcal{A}\mathcal{I}} \Delta\underline{\mathrm{d}}_{\mathcal{I}} - \underline{\underline{\mathrm{K}}}_{\mathcal{A}\mathcal{A}} \Delta\underline{\mathrm{d}}_{\mathcal{A}} - \underline{\mathrm{r}}_{\mathcal{A}} \right). \quad (14)$$

4. Algorithm

The algorithm for solving the contact problem, which was formulated above, can be written in this way

1. In the step $k = 0$, set initial value of the vector $\begin{bmatrix} \Delta\underline{\mathrm{d}}^0 \\ \underline{\mathrm{z}}^0 \end{bmatrix}$.
2. Determine $\mathcal{A}_0$ and $\mathcal{I}_0$, where $\mathcal{A}_0 \cup \mathcal{I}_0 = \mathcal{S}$ and $\mathcal{A}_0 \cap \mathcal{I}_0 = \emptyset$.
3. Find primal-dual couple $\left(\Delta\underline{\mathrm{d}}^k, \underline{\mathrm{z}}^{k+1}\right)$ by solving the system of linear equations (12) (or the system with eliminated multiplicators).
4. Update $\underline{\mathrm{d}}^{k+1} = \underline{\mathrm{d}}^k + \Delta\underline{\mathrm{d}}^k$.
5. Determine $\mathcal{A}_{k+1}$ and $\mathcal{I}_{k+1}$

$$\mathcal{I}_{k+1} = \left\{ j \in \mathcal{S} \middle| \left(\underline{\mathrm{n}}_j^{k+1} \cdot \underline{\mathrm{z}}_j^{k+1} - c_n \tilde{g}_j^{k+1} \right) \leq 0 \right\},$$
$$\mathcal{A}_{k+1} = \left\{ j \in \mathcal{S} \middle| \left(\underline{\mathrm{n}}_j^{k+1} \cdot \underline{\mathrm{z}}_j^{k+1} - c_n \tilde{g}_j^{k+1} \right) > 0 \right\}.$$

6. If $\mathcal{A}_{k+1} = \mathcal{A}_k$, $\mathcal{I}_{k+1} = \mathcal{I}_k$ and $\|\underline{\mathrm{r}}_{tot}\| \leq \varepsilon_r$, then stop, else increment $k = k + 1$ and continue from the 3rd step.

ε_r represents accuracy of our calculation and a vector $\underline{\mathrm{r}}_{tot}$ contains vector of a residual force $\underline{\mathrm{r}}$ and the residual contact constraints.

5. Numerical experiment

The semismooth Newton method was tested on a static problem, in which the slave body was divided into 10×10 elements and the master body into 20×10 elements (see Figure 3). On the upper halves of sides of the slave boundary a Dirichlet condition was defined. The Dirichlet condition was also defined on the bottom of the master boundary. On the other parts of the boundaries, except the contact boundary, a Neumann condition was prescribed. We choose the calculation accuracy equal to $\varepsilon_r = 10^{-9}$.

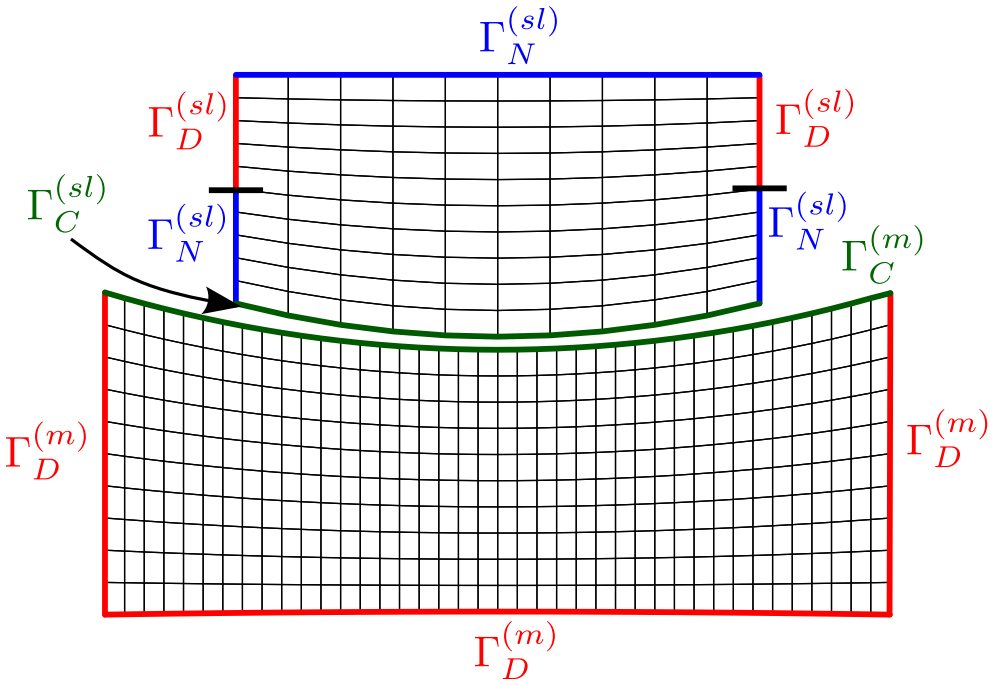


Figure 3: Static problem.

We compared both modifications of the semismooth Newton method (the first modification, denoted SSNM – Alg. 1, uses the system which contains multiplicators $\underline{z}$ and the second modification, denoted SSNM – Alg. 2, uses the system without multiplicators $\underline{z}$) with a fixed point problem. In each step of the algorithms the $\|\underline{r}_{tot}\|$ value was used to stop the algorithm. The individual $\|\underline{r}_{tot}\|$ values are shown in Table 1 listed below .

Individual matrices assembling was implemented in a C++ language with the use of the mex interface for ability to employ this code in MATLAB. Both of the mentioned algorithms (semismooth Newton method modification and fixed point algorithm) were implemented in MATLAB and were added to existing MatSol library which was developed at the Department of Applied Mathematics of the VŠB – Technical University of Ostrava.

k	SSNM – Alg. 1	SSNM – Alg. 2	fixed point
1	$6.13 \cdot 10^{1}$	$6.13 \cdot 10^{1}$	$6.00 \cdot 10^{4}$
2	$6.53 \cdot 10^{-1}$	$6.53 \cdot 10^{-1}$	$3.30 \cdot 10^{0}$
3	$5.62 \cdot 10^{-4}$	$5.62 \cdot 10^{-4}$	$2.97 \cdot 10^{0}$
4	$3.17 \cdot 10^{-7}$	$3.17 \cdot 10^{-7}$	$3.30 \cdot 10^{-3}$
5	$8.08 \cdot 10^{-10}$	$7.85 \cdot 10^{-10}$	$3.13 \cdot 10^{-4}$
6			$7.38 \cdot 10^{-6}$
7			$4.75 \cdot 10^{-7}$
8			$1.92 \cdot 10^{-8}$
9			$1.01 \cdot 10^{-9}$
10			$4.54 \cdot 10^{-11}$

Table 1: $\|\underline{r}_{tot}\|$ values.

Acknowledgements

This work was supported by grant SGS . SP2015/100, VŠB – Technical University of Ostrava, Czech Republic.

References

[1] Dostál, Z., Kozubek, T., Brzobohatý, T., Markopoulos, A., and Vlach, O.: Scalable TFETI with optional preconditioning by conjugate projector for transient frictionless contact problems of elasticity. Comput. Methods Appl. Mech. Engrg. **247–248** (2012), 37–50.

[2] Motyčková, K. and Kučera, R.: Semi-smooth Newton method for solving 2D contact problems with Tresca and Coulomb friction. Advances Electr. Electron. Engrg. **11** (2013), 218–226.

[3] Popp, A., Gee, M. W., and Wall, W.A.: A finite deformation mortar contact formulation using a primal-dual active set strategy. Internat. J. Numer. Methods Engrg. **79** (2009), 1354–1391.

[4] Wohlmuth, B. I.: Variationally consistent discretization schemes and numerical algorithms for contact problems. Acta Numer. **20** (2011), 569–734.

[5] Wriggers, P.: *Computational contact mechanics*. J. Wiley, Hoboken, NJ, 2002.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

NUMERICAL STUDIES OF GROUNDWATER FLOW PROBLEMS WITH A SINGULARITY

Milan Hokr, Aleš Balvín

Technical University of Liberec
Studentska 2, Liberec, 46117, Czech Republic
milan.hokr@tul.cz, ales.balvin@tul.cz

Abstract: The paper studies mesh dependent numerical solution of groundwater problems with singularities, caused by boreholes represented as points, instead of a real radius. We show on examples, that the numerical solution of the borehole pumping problem with point source (singularity) can be related to the exact solution of a regular problem with adapted geometry of a finite borehole radius. The radius providing the fit is roughly proportional to the mesh step. Next we define a problem of fracture-rock coupling, with one part equivalent to the singular point source problem and the second part with a uniform flow. It is a regularized problem, but with the mesh dependence similar to the radial flow, in a certain range of steps. The behavior is explained by comparing the numerical solution with the analytical solution of a simplified problem. It also captures the effects of varying physical parameters.

Keywords: finite elements, mesh dependence, borehole, radial flow

MSC: 35A20, 65N30, 76S05, 86A05

1. Introduction

Although not mentioned among the main challenges in groundwater modelling, the issue of singularity, related to boreholes represented by a single point (in 2D) or line (in 3D), is not fully resolved, also because it is studied differently in theoretical work and in practical applications or simulation software.

The problem of point source in groundwater flow is shortly specified in 1.1. The singularity in the problem is a result of problem abstraction, convenient for handling the problem technically. The real case is that a borehole has a finite radius, but very small compared to the problem domain, which is inconvenient for meshing. On the other hand, a problem of a single borehole in homogeneous medium can be efficiently solved analytically. There are several either empirical or theoretical-based methods, introducing the analytical solution of the radial flow in the local scale to be coupled with a coarser mesh numerical solution – the analytical element

DOI: 10.21136/panm.2016.05

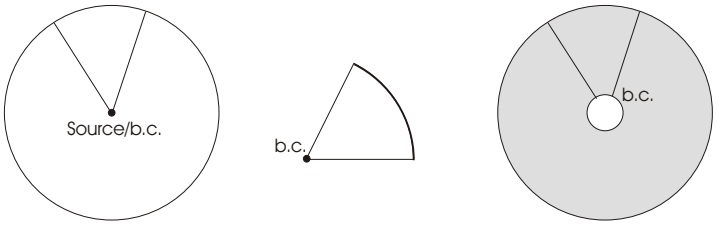


Figure 1: Left: radial flow around borehole in a singular form. Middle: configuration of boundary for numerical solution. Right: radial flow in a regular problem form.

method [2], Peaceman model in the field of reservoir engineering [6, 3], or the concept of extended finite element method (XFEM) with the enrichment functions based on the local analytical solution, e.g. [4].

The background is also different than studies aiming to approximate the singular problem solution and interpreting the numerical precision, e.g. [1] for the Dirac right-hand side with an a-priori set mesh refinement. Instead, we study the mesh dependence of the approximate solution in relation to a replacement problem, which gives a simpler understanding in the context of the finite borehole radius (alternatively to [3]). For such case, a sequence of fixed meshes is used, made by standard generators based on prescribed step at the boundary. This is subject of the first part of the paper and also a background for the second part, where we introduce a specific groundwater geometric configuration with analogous features but different interpretation of the mesh dependence.

1.1. Problem and singularity characterisation

The groundwater flow in its simplest form is a potential field, governed by linear Darcy's law and the mass balance equation,

$$v = K\nabla p\,, \qquad \nabla \cdot v = f\,, \tag{1}$$

where p is pressure head, v is flux density (velocity), K is hydraulic conductivity, and f are sources/sinks. Flux q meaning the integral of v is used in the solved problems.

In the borehole inflow configuration of Fig. 1, the singularity appears for the Dirac right-hand side, i.e. finite flux concentrated to a point as infinite spatial density, resulting to a generalised solution with infinite pressure at the point. Another formulation is with given finite pressure in the borehole, which can either be a boundary condition (formally, for the circular sector domain), or an additional constraint on the pressure solution together with related degree of freedom in the source/sink function f. The same is the asymptotic case of a finite borehole problem, solved analytically below.

For a real borehole, neither case is physically realistic, as the measured values of

flux and pressure are always finite, and any kind of solution needs to introduce the borehole diameter as a parameter.

2. Borehole-driven radial flow problem

The example problem is a case of radial-symmetric flow into a borehole, which can be expressed either by a circle with the borehole in its center or by any sector of such circle (Fig. 1), with the radius r as one variable. The analytical solution is simply derived from 1D radial form of (1) by applying the separation of variables method. With pressure boundary conditions $p(r_1) = p_1$ and $p(r_2) = p_2$, the solution is

$$q(r) = 2\pi K \frac{p_2 - p_1}{\ln \frac{r_2}{r_1}}\,, \qquad \text{i.e.} \quad q(r) = \text{const} = q\,, \tag{2}$$

$$p(r) = p_1 + (p_2 - p_1)\,\frac{\ln \frac{r}{r_1}}{\ln \frac{r_2}{r_1}}\,, \tag{3}$$

considering the flux q over the full circumference. We see the singularity as the asymptotic behavior of the formulas for $r_1 \to 0$, i.e. $q \to 0$ for finite p_1 while $p_1 \to -\infty$ for a non-zero q. We study the first case in the following work, i.e. the dependence $q_{[r_1]}$, which is our notation for the parametric dependence on the problem geometry, to distinguish from the solution as a function of its space variable.

The study is based on comparison between the numerical solution of the singular problem with dependence on a mesh step h, and the analytical solution of the regular problem (2) with dependence on r_1. We use the circular sector geometry for the numerical problem (2D meshing). The boundary condition $p = p_1$ representing the borehole (singularity) is introduced to one node value of the standard finite element discretisation with piecewice linear base functions. The parameters used are $r_2 = 10$, $K = 1$, $p_1 = 0$ and $p_2 = 1$.

The comparison is made in two ways: the numerical $q_{[h]}$ dependence against analytical $q_{[r_1]}$ dependence for the choice $r_1 = h$ and an "inverse" problem of finding an effective borehole diameter r_1 to fit the numerical q value by the analytical one. We use more variants of mesh topology and, for each, a set of meshes of varying step h at the borehole boundary point is generated. So we can check other possible influences than the h value.

In the mesh set of type A, each refinement is generated independently, only based on the prescribed step h at the borehole and a different step along the outer boundary, resulting in either more uniform or more graded meshes. Three such sets of different domain angles $\varphi = 45°$, $60°$, and $90°$ are generated, denoted as A-45, A-60, and A-90. The set of type B, for $\varphi = 60°$ only, is constructed from the coarsest uniform mesh by sequential splitting of each triangular element uniformly, resulting in the same topology of all meshes in the set.

The results for a range of mesh steps are presented in Fig. 2. The parametric dependence of the flux is visually very similar between the analytical solution and

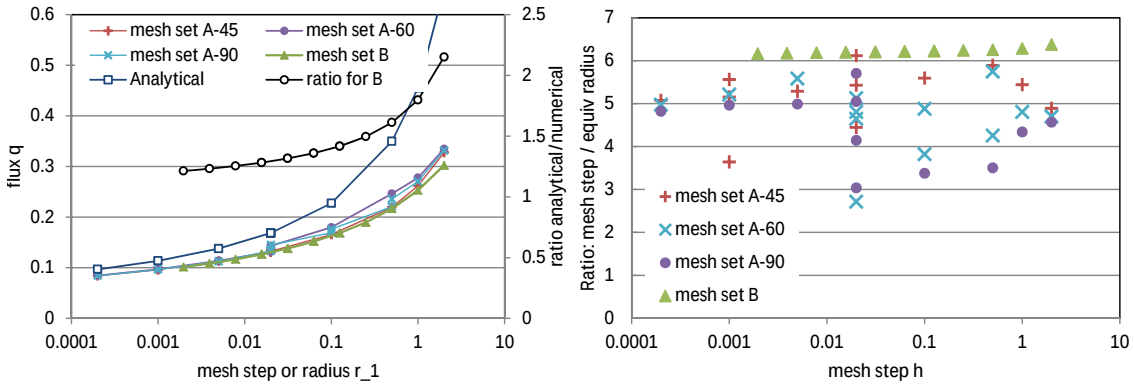


Figure 2: Relation of the numerical and analytical solution of radial flow for various mesh topologies: left is the resulting flux depending on either the mesh step or the borehole radius (scaled to equivalent of $\varphi = 60°$), right is the ratio of equivalent analytical solution radius (r_1) to a given numerical discretisation step.

the numerical solution (for $r_1 = h$), but the fluxes are not proportional, as seen from the evaluated ratio.

The effects of different meshes of the same h are often invisible for $q_{[h]}$, in general less then 10%, but they are detected in the inverse problem of effective r_1. The ratio of the given h and the fitted r_1, depending on h, is plot in Fig. 2 (right). The dependence can be evaluated as a constant, but with significant deviations related to different origin of the mesh. It leads to a hypothesis that the effective behaviour of the numerical solution with "point boundary" corresponds to the desired solution of the radial flow with particular radius, proportional to the mesh step. A precise relation could depend on the choice of numerical scheme and mesh topology.

3. Fracture-block coupling problem

3.1. Real-world motivation

The rock hydraulic conductivity K can change over many orders of magnitude and blocks of very different K are often parts of a single modelling problem. Within the low-permeable rocks, the water can be conducted along planes like fractures or tectonic faults; these are domains with orders of magnitude larger than K and small thickness. To get a measure of contribution to the total flux in a domain, the transmissivity is defined as a product of K and the thickness. If the fracture is represented as plane, the problem, in its vertical cross-section (2D), is a composition of a rectangle and a line (Fig. 3). It creates, at their contact, a singularity within the rectangle domain, similar to the borehole point source problem. Again, it is an effect of the model abstraction, while no sharp K changes or block edges would exist in the real rock.

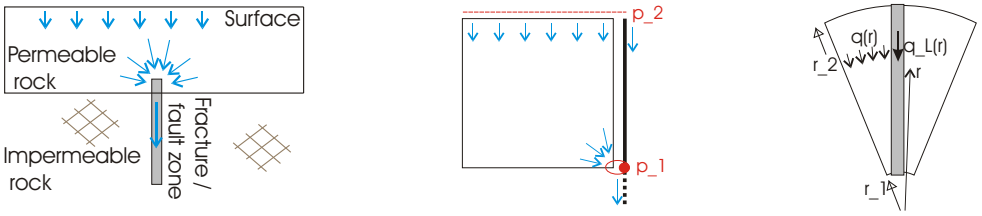


Figure 3: Left: real-world problem of fracture to permeable zone contact, which is in its part an analogue of the radial flow problem. Middle: model problem of fracture-porous block contact for the numerical study with the boundary conditions. Right: derived simpler analogue of coupled radial flow (circle) and uniform flow (line).

3.2. Problem configuration

The test problem to demonstrate numerical features is illustrated in Fig. 3 (middle). The configuration is an analogue of the problem in Decovalex-2015 benchmark [5], where the question of possible mesh dependence arised. Besides using reflection symmetry, there are two main differences against the real-world concept (Fig. 3 left):

One is in extending the fracture (line) domain along the whole rock (rectangle) domain in order to establish a communication between the domains which would not be possible for the mixed-hybrid finite elements through one node. Considering the fracture transmissivity is not significantly larger than the rock block transmissivity (realistic assumption because of the large rock volume), the problem should not be affected much quantitatively.

The second adaptation excludes the part of the line not in contact with the rectangle. The total flux is controlled by a "serial connection" of the line along the rectangle and the line below the rectangle. The latter is controlled by a linear relation of pressure gradient and flux, so the difficulty related to the singularity is present only in the rectangle part, which we concentrate on.

The domain dimensions are 50×50, as well as $r_2 = 50$ in the problem of Section 3.3 for comparison. The boundary pressure values are $p_1 = 0$ and $p_2 = 1$, and the coefficients are listed in Section 3.4.

3.3. Simplified analytical solution

To get an analytical solution for comparison, the problem needs to be significantly simplified (Fig. 3 right): a circular sector between two radii, r_1 inner and r_2 outer, and a line with coordinates between the same r_1 and r_2. We assume radial symmetry and ideal contact between the domains, i.e. a common value of pressure $p(r)$ for both the radial flow and the uniform flow. There are two variables for the flux, $q(r)$ in the circular domain and $q_L(r)$ in the line domain. For simplicity, we consider the unit thickness of the circular domain and the unit cross-section of the line domain, without loss of generality.

The system of equations for the choice of the circular sector angle φ is

$$q(r) + q_L(r) = \text{const} = Q, \tag{4}$$

$$q(r) = \varphi K\, r\, \frac{dp}{dr}, \tag{5}$$

$$q_L(r) = K_L \frac{dp}{dr}, \tag{6}$$

where K and K_L are hydraulic conductivities of the respective subdomains.

The analytical solution is a generalisation of the radial flow, with mostly technical differences. Substituting both q and q_L into the first (mass balance) equation, we get a form ready for separation of variables, p and r. Then two constants, the total flux Q and the integration constant, are evaluated from the two boundary conditions. The results are

$$Q = \varphi K \frac{p_2 - p_1}{\ln \frac{\varphi K r_2 + K_L}{\varphi K r_1 + K_L}}\,, \tag{7}$$

$$p(r) = p_1 + (p_2 - p_1) \frac{\ln \frac{\varphi K r + K_L}{\varphi K r_1 + K_L}}{\ln \frac{\varphi K r_2 + K_L}{\varphi K r_1 + K_L}}\,. \tag{8}$$

Additionally, we can derive $q_L(r)$ and $q(r)$. Then we evaluate the asymptotic problem behavior for $r_1 \to 0$. Obviously, the term K_L regularizes the solution, so that $Q_{[r_1]}$ converges to a finite value composed, at the boundary, of finite $q_{L[0]}(0)$ and zero $q_{[0]}(0)$.

3.4. Parameter sensitivity

The asymptotic behavior for $r_1 \to 0$ is strongly related to the magnitude of φK versus K_L, which is demonstrated in Fig. 4 for $K = 10^{-8}$ and two choices $K_L = 10^{-7}$ and $K_L = 10^{-10}$. For dominating conductivity of the circular domain (Fig. 4 left), the certain range of r_1 dependence is similar to the singularity case of the radial flow alone, requiring substantially small r_1 to exhibit the convergence through the contribution of the line domain. For small r_1, most of water comes through the line domain at the r_1 boundary and the flux $q_{[r_1]}(r_1)$ decreases much more quickly with r_1 than would for the pure radial flow problem. Contrary, for dominating conductivity of the line domain (Fig. 4 right), the q contribution quickly vanishes with decreasing r_1 and the changes of $Q_{[r_1]}$ are relatively smaller.

3.5. Numerical tests

We compare the dependence of the analytical solution on r_1 (total flux $Q_{[r_1]}$) with the dependence of the numerical solution (flux through the Dirichlet boundary bottom right corner) on mesh step h in Fig. 5. We note that, contrary to the previous case of radial flow, the solution used for the comparison is for a significantly simplified problem – in particular, not capturing that the 2D domain is not exactly radially symmetric and there is not necessarily an equilibrium between the domains. Two

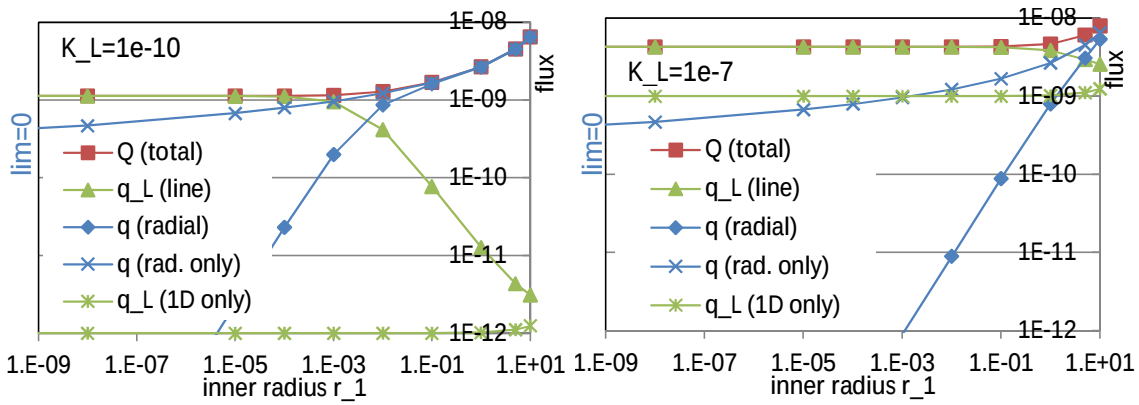


Figure 4: Analytical solution of the coupled circle-line flow problem showing convergence with decreasing inner radius, different for either dominant circle (left) or dominant line (right). The "radial only" and "line only" cases are solutions on each of the subdomains uncoupled, with the same boundary conditions.

different finite-element versions are used – the standard linear FEM with shared degrees of freedom of the rectangle side and the line and the mixed-hybrid (MH-FEM), with separate degrees of freedom.

There is a lot of common in the mesh-dependence with the radius-dependence, but the quantitative relation is not so clear as for the radial flow in Section 2. The mesh dependence disappears for sufficiently small h, which suggests the convergence of the solution like in the simplified analytical problem for $r_1 \to 0$.

The physical parameter sensitivity is also well reproduced: there is a little mesh dependence for large K_L (even negligible for $K_L = 10^{-6}$, not shown), contrary to the significant mesh dependence for small K_L, disappearing for very small h. The mesh step size, necessary for $Q_{[h]}$ to get steady, is typically one order of magnitude larger then the radius, for which the analytical solution dependence on r_1 disappear, and this position is roughly proportional to K_L (Fig. 5).

On the other hand, the trends of $Q_{[h]}$ differ between the numerical schemes. The standard FEM solution Q decreases with mesh refinement similarly to the analytical $Q_{[r_1]}$ (total flux). The MH-FEM solution rises with the mesh refinement, similarly to the curve of the analytical $q_{L[r_1]}$ (line-only flux). It can be explained by a structure of the discrete unknowns: in the used implementation of the standard FEM, the boundary condition is prescribed to a shared node while in the MH-FEM, the b.c. is introduced to the line (see the red b.c. circle in Fig. 3 middle), which is then coupled to the rectangle, making the line domain more significant for the overall hydraulic resistance.

The flux at the mesh refinement limit lies between the analytical solution for $\varphi = \pi/3$ and $\varphi = \pi/2$. It means that only a part of the numerical problem domain (corresponds to $\varphi = \pi/2$) is effectively covered by the flow, as it is deviated from the radial symmetry (more for larger fracture contribution).

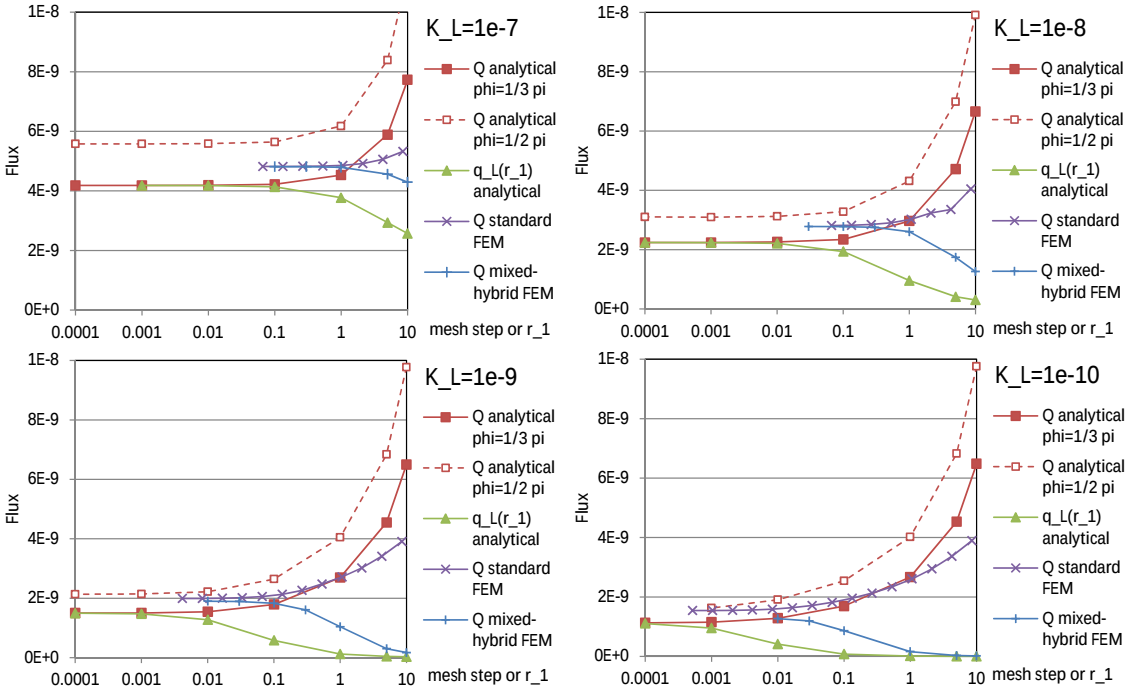


Figure 5: Mesh dependence of the block-fracture problem numerical solution compared to the radius dependence of the circle-line analytical solution (two choices of φ) for $K = 10^{-8}$ and a range of K_L, from fracture-dominant to block-dominant.

4. Conclusion

We have shown for the radial flow problem that the mesh dependence of the numerical solution, resulting from the singularity property of the point source/boundary, can have a physical meaning equivalent to the dependence of the real flow on the borehole radius. The mesh step providing the fit is roughly proportional to the borehole radius with a factor between 5 and 6 for meshes close to uniform while a larger deviation appears for more graded meshes. It could be a topic for further study to predict the relation theoretically from a numerical scheme. Such mesh choice can be useful as an alternative for adaptive mesh refinement based on error analysis. The error in flux is appropriate for groundwater data accuracy.

For the fracture-rock coupling problem, the extension of the fracture along the block regularizes the problem, although the mesh dependence is still present depending on the singularity component dominance. The mesh and the physical parameter dependence can be predicted by a relatively simple analytically solvable problem, in particular the position of visual convergence. Also, we have shown how the solution is sensitive on the position of the discrete unknowns.

Acknowledgements

This work was realised with the institutional support of the Faculty of Mechatronics, Informatics, and Interdisciplinary Studies, TU Liberec, and supported by the Radioactive Waste Repository Authority (SÚRAO), contract code SO2013/077.

References

[1] Apel, T., Benedix, O., Sirch, D., and Vexler, B.: A priori mesh grading for an elliptic problem with Dirac right-hand side. SIAM J. Numer. Anal. **49(3)** (2011), 992–1005.

[2] Bakker, M. and Strack, O. D. L.: Analytic elements for multiaquifer flow. J. Hydrology **271(1–4)** (2003), 119–129.

[3] Ding, Y. and Jeannin, L.: A new methodology for singularity modelling in flow simulations in reservoir engineering. Comp. Geosci. **5** (2001), 93–119.

[4] Exner, P. and Březina, J.: Partition of unity methods for approximation of point water sources in porous media. Applied Math. Comp. **273** (2016), 21–32.

[5] Hokr, M., Shao, H., Gardner, W.P., et al.: Real-case benchmark for flow and tracer transport in the fractured rock. Environ. Earth Sci. **75** (2016), 1273.

[6] Peaceman, D.: Interpretation of well-block pressures in numerical reservoir simulation. SPE J. **18(3)** (1978), 183–194.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

THE ROLE OF SOMMERVILLE TETRAHEDRA IN NUMERICAL MATHEMATICS

Radim Hošek

Department of Mathematics, University of West Bohemia
Univerzitní 22, 306 14 Plzeň, Czech Republic
radhost@kma.zcu.cz
and
Institute of Mathematics, Czech Academy of Sciences
Žitná 25, 115 67 Prague, Czech Republic
hosek@math.cas.cz

Abstract: In this paper we summarize three recent results in computational geometry, that were motivated by applications in mathematical modelling of fluids. The cornerstone of all three results is the genuine construction developed by D. Sommerville already in 1923. We show *Sommerville tetrahedra* can be effectively used as an underlying mesh with additional properties and also can help us prove a result on boundary-fitted meshes. Finally we demonstrate the universality of the Sommerville's construction by its direct generalization to any dimension.

Keywords: simplicial tessellations, simplicial mesh, Sommerville tetrahedron, well-centered mesh, boundary-fitted mesh, high dimension

MSC: 51M20, 51M04, 65N30, 65N50.

1. Introduction

Many computational methods require or prefer simplicial meshes as the underlying geometrical playground. In two dimensions the best triangle among all, measured by various regularity criteria, is the equilateral triangle, which is a space-filler. In higher dimensions the situation is different, as already for $d = 3$ the equilateral tetrahedron cannot tile the space, see [13].

If the equilateral tetrahedron cannot be taken as the standard, is there any other playing such role what concerns space-filling? The answer is affirmative, as we show in the sequel. Moreover, an answer to that question will be generalized to a general dimension.

DOI: 10.21136/panm.2016.06

The cornerstone of this paper is the construction proposed by Sommerville already in 1923, see [14]. It takes the unit equilateral triangle $A_0A_1A_2$ as a base and creates the points $B_0, B_1, B_2, \ldots$ above the three original points satisfying

$$B_z = [A_{i(z)}, zp], z \in \mathbb{Z}, \qquad \text{where} \quad i(z) \equiv z \mod 3, \tag{1}$$

and p is a positive parameter. Then the tetrahedra are defined as convex hulls of four consecutive points, which we denote $\mathrm{co}\{B_z, B_{z+1}, B_{z+2}, B_{z+3}\} =: K_3^z$. Three such tetrahedra are sketched in Figure 1. Obviously, this construction enables to fill the whole infinite triangular prism by copies of a single element. Repeating this construction appropriately above all triangles, one gets a face-to-face tessellation of the three-dimensional space, determined up to a positive constant p, that consists of congruent tetrahedra, whose representative is denoted by $K_3(p)$. For more details we refer to [7].

The paper is devoted to three recent author's results based on the above construction. These can be found in their full detail in [7], [9] and [8]; here we provide their brief summary with some additional comments. Each of these results is presented in a separate section.

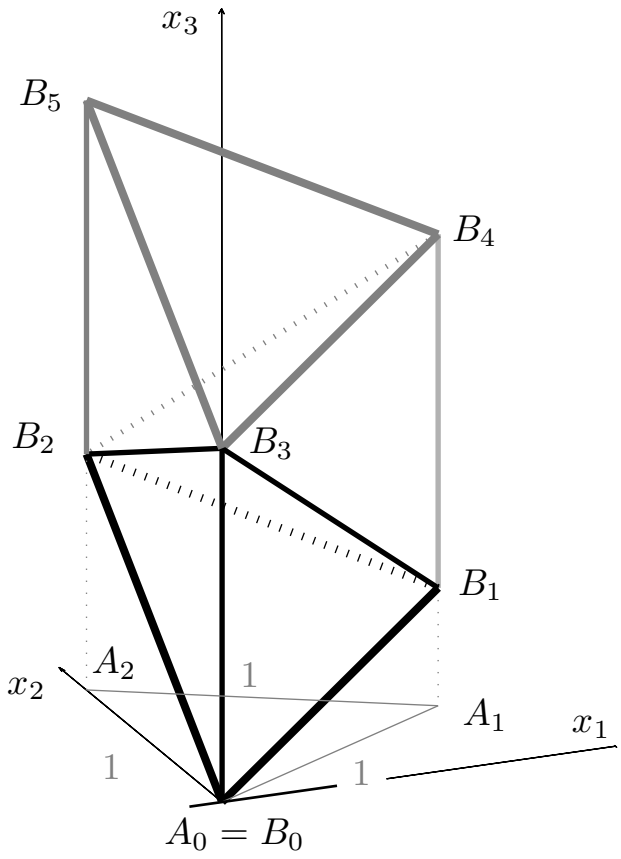


Figure 1: Illustration of the Sommerville's construction.

2. Well-centered Sommerville tetrahedra and their shape-optimization

This result was motivated by the work of Feireisl et al. [4], in which the convergence of a numerical scheme to the compressible Navier-Stokes-Fourier system in three spatial dimensions is proven. While the target system is confined to a smooth bounded domain Ω, the numerical scheme is supposed to be defined on a family of polyhedral domains $\{\Omega_h\}_{h\to 0}$, for which $\Omega \subset \Omega_h$ and $\max_{x\in\partial\Omega_h} \operatorname{dist}[x, \partial\Omega] \leq h$. This approach, known as *variational crime*, see e.g. [2], is known to decrease the rate of convergence, upon the condition that conforming elements are used. This is, however, not the case in [4], where non-conforming Crouzeix-Raviart elements are used for velocity.

Numerical domains Ω_h are supposed to admit face-to-face tetrahedral meshes $\mathcal{T}_h$ (where h denotes the characteristic diameter of the elements), satisfying the strong regularity property.

Definition 1 (Strong regularity)**.** Let $\{\mathcal{T}_h\}_{h\to 0}$ be a family of meshes. If there exists $\theta_0 > 0$ independent of h such that for any $\mathcal{T}_h$ and any $K \in \mathcal{T}_h$ it holds that

$$\theta(K) := \frac{\varrho(K)}{\operatorname{diam} K} \geq \theta_0, \tag{2}$$

where $\varrho(K)$ is the radius of the largest ball contained in K, then we say that $\{\mathcal{T}_h\}_{h\to 0}$ is a *strongly regular* family.

One can also define strong regularity with different regularity ratios. Equivalency of some of these definitions can be found in [1]. The terms *shape regular* or *regular* family of meshes can be found within the literature for the property (2).

Further, the tetrahedral elements of the mesh in [4] is assumed to satisfy so-called *well-centered property*, introduced by VanderZee, see e.g. [16]. A well-centered simplex contains its circumcenter in its interior; this ensures that the segment connecting the circumcenters of two neighbouring elements is perpendicular to their common facet and does not degenerate. This property is used in the numerical scheme for the balance of temperature. For the sake of brevity, we use the term *well-centered* mesh instead of the more proper *d-well-centered* mesh.

Definition 2 (Well-centered property)**.** Let $K_d := \operatorname{co}\{V_0, V_1, \ldots, V_d\}$ be an d-dimensional simplex. We say that K_d is well-centered if its circumcenter lies in the interior of K_d.

We would like to point out that in two dimensions the well-centeredness coincides with acuteness, while in higher dimensions it is no longer true, see some illustrations in [15].

With all the above said, the idea for finding the polyhedral domains Ω_h and meshes $\mathcal{T}_h$ was the following. First, to find a face-to-face well-centered tetrahedral tessellation of the whole three-dimensional space with the size of the elements not

exceeding h and then to pick those elements whose intersection with Ω is non-empty. Their union then builds Ω_h.

We use the Sommerville's result, which can be scaled to provide the desired tessellation. By virtue of a sufficient condition introduced by VanderZee, [16, Theorem 1] and elementary geometric calculations, we are able to determine the range of parameters, for which the elements are well-centered.

Theorem 1 ([7], Theorem 3.3)**.** *The tetrahedra constructed by the method described in Section 1 are well-centered if and only if* $p \in \left(0, \sqrt{2}/2\right)$.

Basically any parameter from given range would give a satisfactory mesh. However, it is obvious that parameters in the middle of the interval are *better* than those at its edges; for p small we obtain flat tetrahedra of the *wedge* type that are close to degenerate ones, while for $p \to \sqrt{2}/2$ the distances of neighbouring circumcenters degenerate. Therefore, we determine a *shape optimal parameter* within this range.

Theorem 2. *Let* $K_3(p)$ *be a tetrahedron constructed by the method in Section 1 and let* θ *be the regularity ratio defined by (2). Then* $\theta(p) := \theta(K_3(p))$ *is maximal for* $p = p^\star = \sqrt{2}/4$.

The value $p^\star$ is optimal also for regularity ratio (6) that is used later in Section 4 and also for the ratio of circumradius and inradius of an element, which is the original assertion [7, Theorem 4.3].

While the general $p \in (0, \sqrt{2}/2)$ gives a well-centered mesh that consists of a tetrahedra that are congruent to each other, for $p^\star$ we get a mesh build by copies of a single element, which also trivially implies the strong regularity property. The element is an *equifacial tetrahedron*, see [6]. As for Naylor [12], it is the most regular tetrahedron, whose copies tile the three dimensional space. Thus it can (and in the next section will) be used as a reference tetrahedron for measuring the shape regularity.

3. Strongly regular family of boundary-fitted meshes

The second result is also motivated by a numerical scheme for compressible flow on an unfitted mesh. For establishing error estimates to a numerical scheme for compressible Navier-Stokes equation in three dimensions in [3], the weak-strong uniqueness principle, see [5], is used. For this reason, the existence of a strong solution is assumed. But the system is known to possess strong solution only on sufficiently smooth domains. Therefore, the target system is confined to a bounded domain $\Omega \in C^3$, while the numerical scheme is designed on a tetrahedral mesh $\mathcal{T}_h$ that fills a polyhedral domain Ω_h. There is no inclusion of the domains Ω_h and Ω assumed, but both domains shall be close to each other. We require for all $x \in \partial\Omega_h$ that

$$\operatorname{dist}[x, \partial\Omega] \leq d_\Omega h^2, \tag{3}$$

with the constant d_Ω depending solely on the geometry of Ω. This is easily ensured by placing the vertices of the polyhedral domain Ω_h at the boundary of the smooth

domain Ω. We call such mesh *boundary-fitted*. To prove (3), one just uses the Taylor expansion.

Again, for the convergence, we need to assume that the family of polyhedral domains is strongly regular. The question of existence of a strongly regular family of boundary-fitted simplicial meshes to a C^2 domain is affirmative in two dimensions thanks to [10]. We attack the three-dimensional case and the result reads as follows.

Theorem 3 ([9], Theorem 1)**.** *Let Ω be a bounded domain in $\mathbb{R}^3$ of the class C^2, with the minimal radius of an osculation sphere equal to r_Ω. Let for some h_1* sufficiently small *there exists $(\Omega_{h_1}, \mathcal{T}_{h_1})$ an approximative domain with boundary-fitted mesh and let*

$$\theta(K) \geq \frac{\alpha}{r_\Omega} \text{diam } K,$$

for any $K \in \mathcal{T}_{h_1}$, where θ is defined in (2) and $\alpha > \alpha_0 = 32(2+\sqrt{5})\sqrt{2/3}$.

Then there exists a strongly regular family of boundary-fitted meshes $\{\mathcal{T}_h\}_{h\to 0}$.

The assumption is easy to be fulfilled, as the initial regularity requirement gets weaker with decreasing discretization parameter. The proof is based on the result of Křížek, see [11], that shows that a Sommerville tetrahedron, i.e. tetrahedron $K_3(p^\star)$ from Section 2, can be decomposed into eight identical tetrahedra that are similar to the original one. As a consequence, any tetrahedron of the size h can be decomposed into eight tetrahedra of the size not exceeding $h/2$ while the regularity is preserved. Hence we decompose the initial mesh, the newly established vertices that lie on $\partial\Omega_h$ get shifted to the smooth boundary $\partial\Omega$ and we show, that the regularity does not deteriorate *too much*. The shifts of these vertices are performed by affine mappings. This was our motivation to employ a new regularity criterion, based on the similarity of a tetrahedron with the reference one, Sommerville tetrahedron.

Definition 3. Let $K = \text{co}\{A, B, C, D\}$ be a tetrahedron,

$$\mathcal{A}_K := \{F_K; F_K \text{ an affine transformation}, F_K(\widetilde{K}) = K\}$$

be a set of all affine transformations mapping Sommerville tetrahedron $\widetilde{K}$ onto K. Then we define the *Sommerville regularity ratio* of tetrahedron K as

$$\kappa(K) = \max_{F_K \in \mathcal{A}_K} \frac{\sigma_{\min}(F_K)}{\sigma_{\max}(F_K)},$$

where $\sigma_{\min}(F_K), \sigma_{\max}(F_K)$ are the minimal and maximal singular values of F_K.

We are able to show, that this regularity criterion is equivalent to the other standard ones in the sense of strong regularity, hence the whole proof can be performed in the terms of κ. The details of the laborious and technical proof can be found in [9]. Here we just point out that the final argument is based on the following inequalities,

$$\prod_{j=0}^{n-1}(1-aq^j) > \lim_{n\to\infty}\prod_{j=0}^{n-1}(1-aq^j) = P(a,q) > 0,$$

for any $n \in \mathbb{N}$ and $a, q \in [0, 1]$.

4. Space-filling simplices in general dimension

The last of the triplet of results is motivated by the Sommerville's construction itself. One can view it as a method of creating the tessellation of d-dimensional space out of $(d-1)$-dimensional one.

The idea is to take a simplex of a tessellation of the $(d-1)$-dimensional space and create the infinite prism made of d-dimensional simplices above it. More specifically, for a simplex $K \in \mathcal{T}_{d-1}$, where $K = \mathrm{co}\{A_0, A_1, \ldots, A_{d-1}\}$ we construct points B_z satisfying

$$B_z = [A_{i(z)}, zp], z \in \mathbb{Z}, \qquad \text{where} \quad i(z) \equiv z \mod d, \tag{4}$$

compare (4) with (1). The simplices are defined as convex hulls of $d+1$ consecutive points B_z. Performing the same above all simplices of the original $(d-1)$-dimensional tessellation, one recovers a face-to-face simplicial tessellation of d-dimensional space, as it is summarized in the following lemma.

Lemma 1 ([8], Lemma 2.2)**.** *Let $d \geq 2$ and $\mathcal{T}_{d-1} = \{K_{d-1}^k\}_{k \in \mathbb{Z}^{d-1}}$ be a simplicial tessellation of $(d-1)$-dimensional space such that the graph constructed from vertices and edges of $\mathcal{T}_{d-1}$ is a d-vertex-colorable graph. Then*

- *there exists $\mathcal{T}_d = \{L_d^l\}_{l \in \mathbb{Z}^d}$ a simplicial tessellation of d-dimensional space with additional shape parameter p_d,*
- *any connected compact subset of $\mathcal{T}_d$ is a face-to-face mesh,*
- *$\mathcal{T}_d$ is a $(d+1)$-vertex-colorable graph.*

The vertex coloring is a tool which ensures the face-to-face property and guarantees that above a vertex (that is shared by several simplices) the new points are constructed consistently, in the same heights above each element. Lemma 1 provides us with the induction step, the initial step is given as a straight line discretized equidistantly using $p_1 > 0$ with points of alternating colors. Thus, we can state the following.

Theorem 4 ([8], Theorem 2.1)**.** *For any d-dimensional space there exists a d-parametric family of simplicial tessellations $\mathcal{T}_d(\mathbf{p}), \mathbf{p} = (p_1, p_2, \ldots, p_d), p_i > 0$. For $\mathbf{p}$ fixed, all elements $K \in \mathcal{T}_d(\mathbf{p})$ have the same d-dimensional measure equal to*

$$\mathrm{meas}_d K = \prod_{i=1}^{d} p_i. \tag{5}$$

Moreover, every connected compact subset of the tessellation builds a face-to-face mesh.

We obtained a tessellation that is determined up to a d-dimensional vector of positive parameters $\mathbf{p} = (p_1, \ldots, p_d)$. Therefore, we determine the shape-optimal

vector of parameters. To benefit from the equivolumetricity property (5), we decided to optimize the ratio

$$\vartheta(K) = \frac{\text{meas}_d K}{(\text{diam } K)^d}, \qquad d \geq 2. \tag{6}$$

In particular, we are looking for an element $K^\star$ and a vector $\mathbf{p}^\star$ satisfying

$$\vartheta(K^\star(\mathbf{p}^\star)) = \sup_{\mathbf{p}\in\mathbb{R}^d_+} \min_{K\in\mathcal{T}_d(\mathbf{p})} \frac{\text{meas}_d K}{(\text{diam } K)^d}, \tag{7}$$

as in general all the elements are not equal. We are optimizing the *worst* element, which is the one with largest diameter. Luckily, there is only a limited number of candidates for the diameter, therefore (7) can be viewed as an optimization problem with nonlinear constraints. Such optimum must satisfy so-called Karush-Kuhn-Tucker conditions. These are always necessary, but sufficient only when the optimized function is convex.

Since we are not able to show the convexity, we prove that the minimizer exists and that there is a unique vector $\mathbf{p}^\star$ that satisfies these conditions. Then $\mathbf{p}^\star$ must be the minimizer. To be precise, the above is true after fixing p_1, which obviously plays the role of a scaling parameter and as such does not affect the shape of the simplices. The statement reads as follows.

Theorem 5 ([8], Theorem 3.1)**.** *Let $d \geq 2$ and let $\mathcal{T}_d(\mathbf{p})$ be a tessellation constructed through the procedure introduced above. Then there exists a unique one-dimensional vector half-space*

$$P^\star = \left\{ \mathbf{p}^{\star,\kappa} \in \mathbb{R}^d_+ | \mathbf{p}^{\star,\kappa} = \kappa \mathbf{p}^\star, \kappa > 0, p_1^\star = 1, p_2^\star = \frac{1}{\sqrt{3}}, p_j^\star = \frac{1}{j-1}\sqrt{\frac{2}{3}}, j \in \{3,\dots,d\} \right\},$$

of optimal parameters that realize

$$\sup_{\mathbf{p}\in\mathbb{R}^d_+} \min_{K\in\mathcal{T}_d(\mathbf{p})} \frac{\text{meas}_d K}{(\text{diam } K)^d}. \tag{8}$$

The detailed proof can be found in [8]. Here we just point out two interesting remarks.

The result of the optimization would be the same, if one optimizes at every level of the construction, which is a one-dimensional optimization that is much easier. In other words, a shape optimal tessellation cannot be created from a sub-optimal tessellation of a hyperplane.

As it was already mentioned in Section 2, for $d = 3$ we obtain again the (equifacial) Sommerville tetrahedron. One can verify that for the choice $\kappa = \sqrt{3}/2$ we get unit equilateral triangle for $d = 2$, which was the base for construction in Section 2, indeed $\kappa p_3^\star = \sqrt{2}/4$.

Acknowledgements

This work was supported by grant SGS-2016-003 of the University of West Bohemia.

References

[1] Brandts, J., Korotov, S., and Křížek, M.: On the equivalence of ball conditions for simplicial finite elements in $\mathbf{R}^d$. Appl. Math. Lett. **22(8)** (2009), 1210–1212.

[2] Brenner, S. and Scott, R.: *The mathematical theory of finite element methods.* Texts in Applied Mathematics. Springer New York, 2008.

[3] Feireisl, E., Hošek, R., Maltese, D., and Novotný, A.: Error estimates for a numerical method for the compressible navierstokes system on sufficiently smooth domains. ESAIM: M2AN **51(1)** (2017), 279–319.

[4] Feireisl, E., Hošek, R., and Michálek, M.: A convergent numerical method for the full Navier–Stokes–Fourier system in smooth physical domains. SIAM J. Numer. Anal. **54(5)** (2016), 3062–3082.

[5] Feireisl, E., Jin, B. J., and Novotný, A.: Relative entropies, suitable weak solutions, and weak-strong uniqueness for the compressible navier–stokes system. J. Math. Fluid Mech. **14(4)** (2012), 717–730.

[6] Hajja, M. and Walker, P.: Equifacial tetrahedra. International Journal of Mathematical Education in Science and Technology **32(4)** (2001), 501–508.

[7] Hošek, R.: Face-to-face partition of 3D space with identical well-centered tetrahedra. Appl. Math. **60(6)** (2015), 637–651.

[8] Hošek, R.: Construction and shape optimization of simplical meshes in d-dimensional space. Submitted to Disc. Comp. Geom. Preprint available at ArXiv.org, June 2016.

[9] Hošek, R.: Strongly regular family of boundary-fitted tetrahedral meshes of bounded C^2 domains. Appl. Math. **61(3)** (2016), 233–251.

[10] Korotov, S., Křížek, M., and Neittaanmäki, P.: On the existence of strongly regular families of triangulations for domains with a piecewise smooth boundary. Appl. Math. **44(1)** (1999), 33–42.

[11] Křížek, M.: An equilibrium finite element method in three-dimensional elasticity. Appl. Mat. **27** (1982), 46–75.

[12] Naylor, D. J.: Filling space with tetrahedra. Int. J. Numer. Methods Eng. **44(10)** (1999), 1383–1395.

[13] Senechal, M.: Which tetrahedra fill space? Math. Mag. **54(5)** (1981), 227–243.

[14] Sommerville, D.: Space-filling tetrahedra in Euclidean space. Proc. Edinburgh Math. Soc. **41** (1923), 49–57.

[15] VanderZee, E., Hirani, A. N., and Guoy, D.: Triangulation of simple 3D shapes with well-centered tetrahedra. CoRR, abs/0806.2332, 2008.

[16] VanderZee, E., Hirani, A. N., Guoy, D., Zharnitsky, V., and Ramos, E. A.: Geometric and combinatorial properties of well-centered triangulations in three and higher dimensions. Comput. Geom. (2013).

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

NUMERICAL MODELLING OF STEADY AND UNSTEADY FLOWS OF GENERALIZED NEWTONIAN FLUIDS

Radka Keslerová, David Trdlička, Hynek Řezníček

Czech Technical University in Prague, Faculty of Mechanical Engineering,
Department of Technical Mathematics,
Karlovo nám. 13, 121 35 Prague, Czech Republic
Radka.Keslerova@fs.cvut.cz, David.Trdlicka@fs.cvut.cz, Hynek.Reznicek@fs.cvut.cz

Abstract: This work presents the numerical solution of laminar incompressible viscous flow in a three dimensional branching channel with circular cross section for generalized Newtonian fluids. This model can be generalized by cross model in shear thinning meaning. The governing system of equations is based on the system of balance laws for mass and momentum. Numerical tests are performed on a three dimensional geometry, the branching channel with one entrance and two outlet parts. Numerical solution of the described model is based on central finite volume method using explicit Runge–Kutta time integration. The steady state solution is achieved for $t \to \infty$. In this case the artificial compressibility method will be applied. In the case of unsteady computation artificial compressibility method is considered.

Keywords: viscous fluids flow, generalized Newtonian fluids, cross model, finite volume method, Runge-Kutta scheme, artificial compressibility method

MSC: 65L06, 65N08, 76A05, 76A10, 76D05

1. Introduction

Branching of pipes occurs in many technical or biological applications. In [13] the effects of viscoelasticity on the pitchfork bifurcation using a numerical finite volume method was investigated. Results from both the upper-convected Maxwell and Oldroyd-B models show that the instability occurs at lower Reynolds numbers for viscoelastic fluids in comparison to the Newtonian base case. In [12] computational fluid dynamics simulations of steady viscoelastic flows through a planar two dimensional T-junction is considered and the influence of constitutive model and fluid elasticity upon the main recirculating flow characteristics formed at the junction and the shear stress fields is studied. In [4] a comparative numerical study of non-Newtonian fluid models capturing shear-thinning and viscoelastic effects of blood flow in idealized and realistic stenosed vessels was presented. Reference [2] proposes to analyze the

DOI: 10.21136/panm.2016.07

pulsatile flow of blood through catheterized stenosed artery considering the Carreau--Yasuda model. The effects of tapering angle, body acceleration and magnetic field are also taken into account. In [1] the unsteady pulsatile magneto-hydrodynamic blood flows through porous arteries concerning the influence of externally imposed periodic body acceleration and a periodic pressure gradient are numerically simulated. Paper [7] shows the mathematical models that represent non-Newtonian flow of blood through a stenosed artery in the presence of a transverse magnetic field. Here, the rheology of the flowing blood is characterised by a generalised Power law model.

In previous works [9] and [10] we studied the numerical simulation of generalized Newtonian and Oldroyd-B fluids flow in 2D branching channel. In this article the problem of the unsteady numerical simulation for the generalized Newtonian fluids flow is presented. The modelled domain is the three dimensional branching channel with T-junction.

2. Mathematical model

The governing system of equations is the system of generalized Navier-Stokes equations, see [3]. This system consists of the continuity equation

$$\operatorname{div} \boldsymbol{u} = 0 \tag{1}$$

and the momentum equation

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla P + \operatorname{div} \mathbf{T}, \tag{2}$$

where P is the pressure, ρ is the constant density, $\boldsymbol{u}$ is the velocity vector. The symbol $\mathbf{T}$ represents the stress tensor.

For the viscous fluids Newtonian mathematical model for the definition of the stress tensor $\mathbf{T}$ is considered (see e.g. [4])

$$\mathbf{T} = 2\mu\mathbf{D}, \tag{3}$$

where μ is the dynamic viscosity and tensor $\mathbf{D}$ is the symmetric part of the velocity gradient, $\mathbf{D} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$.

For the numerical modelling of the generalized Newtonian fluids flow it is necessary to generalize the mathematical models. In this case the viscosity function $\mu\dot{\gamma}$ is defined by cross model (for more details see [8, 14, 15])

$$\mu(\dot{\gamma}) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (\lambda\dot{\gamma})^b)^a}, \quad \dot{\gamma} = 2\sqrt{\frac{1}{2}\operatorname{tr} \mathbf{D}^2}, \tag{4}$$

with dynamical viscosities $\mu_0 = 1.6 \cdot 10^{-1}\ Pa \cdot s$, $\mu_\infty = 3.6 \cdot 10^{-3}\ Pa \cdot s$, constants $a = 1.23, b = 0.64$ and time parameter $\lambda = 8.2\ s$. For Newtonian flow modelling, the viscosity is kept constant and equal to μ_∞. The relationships between dynamic viscosity μ and shear rate $\dot{\gamma}$ is shown in Fig. 1.

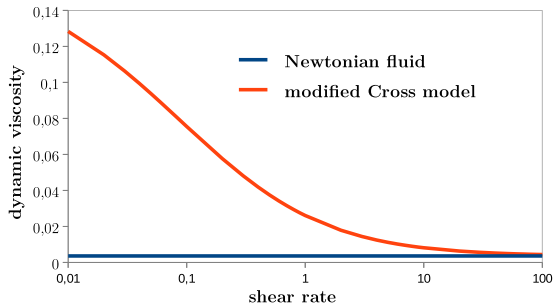


Figure 1: Viscosity μ - shear rate $\dot{\gamma}$ relationships for the two selected viscosity models.

3. Numerical solution

The mathematical models described above are solved numericaly by the artificial compressibility approach combined with the finite-volume discretization. The artificial compressibility method [5, 6, 8, 9] is used to obtain an equation for pressure. It means that the continuity equation is completed by a pressure time derivative term $\frac{\partial p}{\beta^2 \partial t}$, where β is a positive parameter, making the inviscid part of the system of equations hyperbolic

$$\frac{1}{\beta^2}\frac{\partial p}{\partial t} + \operatorname{div} \boldsymbol{u} = 0. \tag{5}$$

The parameter β for steady simulation is chosen equal to the maximum inlet velocity. This value ensures good convergence to steady state but is not large enough to make the transient solution accurate in time. Therefore it is suitable for steady flows only. The system including the modified continuity equation and the momentum equations can be written

$$\tilde{R}_\beta W_t + F_x^c + G_y^c + H_z^c = F_x^v + G_y^v + H_z^v, \quad \tilde{R}_\beta = \operatorname{diag}(\frac{1}{\beta^2}, 1, \cdots, 1), \tag{6}$$

where W is the vector of unknowns. By superscripts c and v we denote the inviscid and the viscous fluxes, respectively.

The space discretization is done by a cell-centered finite-volume method with hexahedral finite volumes [6, 8, 11]. The arising system of ODEs is integrated in time by the explicit multistage Runge–Kutta scheme [9].

The flow is modelled in a bounded computational domain where a boundary is divided into three mutually disjoint parts: a solid wall, an outlet and an inlet. At the inlet Dirichlet boundary condition for velocity vector and for the stress tensor is used. For the pressure, homogeneous Neumann boundary condition is used. At the outlet parts the pressure value is prescribed and for the velocity vector and the stress tensor homogeneous Neumann boundary condition is used. The no-slip boundary condition for the velocity vector is used on the wall. For the pressure and stress tensor homogeneous Neumann boundary condition is considered.

3.1. Unsteady computation

For numerical solution of unsteady flows the artificial compressibility method with unsteady boundary conditions is applied [6]. This method consists in modifying continuity equation in the same way as was described above. The question arises, how big the artificial compressibility parameter β should be in numerical simulation. Numerical calculations show that $\beta = 6\ m \cdot s^{-1}$ (100x higher than for steady simulation) is the proper choice.

Two different forms of unsteady boundary conditions are considered. First, prescribing the pressure value at the outlet (branch). Second, prescribing the normal velocity component u at the inlet as

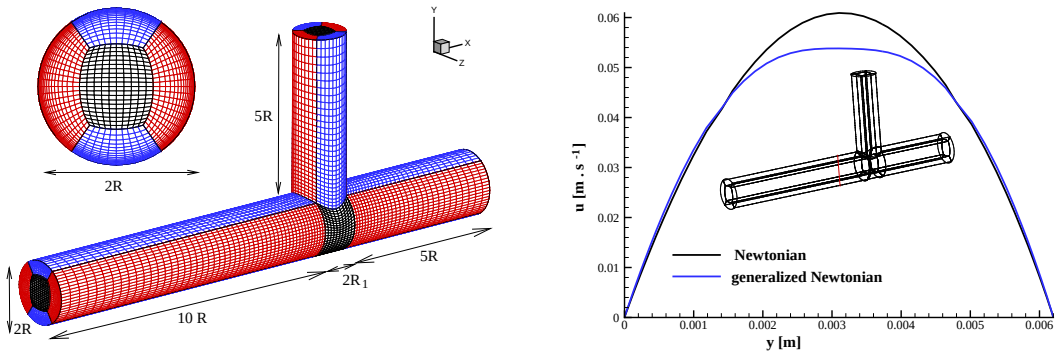
$$up = up_{in}\left(1 + K\sin(\omega t)\right). \tag{7}$$

Here ω is the angular velocity defined as $\omega = 2\pi f$, where f is a frequency. The symbol up denotes the pressure or the velocity according to used unsteady boundary condition. Symbol up_{in} is the constant value obtained from the steady simulation. Constant K is for the pressure 0.5 and for the velocity it is equal to reference velocity value. Other boundary condition are considered as in the steady case.

In this work three values of the frequency are tested: $f = 2, 5, 10\ Hz$.

4. Numerical results

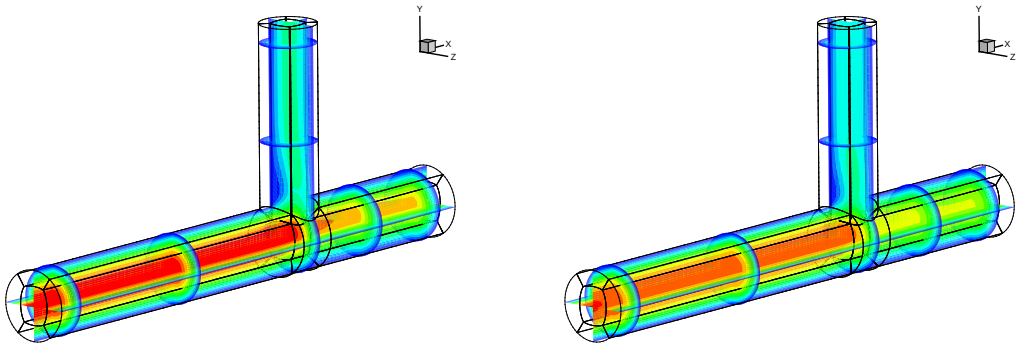
This section deals with the comparison of the numerical results of generalized Newtonian fluids flow for steady case. Numerical tests are performed in an idealized branching channel with the circular cross-section. Fig. 2 (left) shows the shape of the tested domain. The computational domain is discretized using a block-structured, wall fitted mesh with hexahedral cells. The domain is divided to 19 blocks with 125 000 cells.



(a) structure of the domain

(b) axial velocity profile

Figure 2: Structure of the tested domain (left) and axial velocity profile of tested fluids (right). Reference radius $R = 0.0031\ m$.



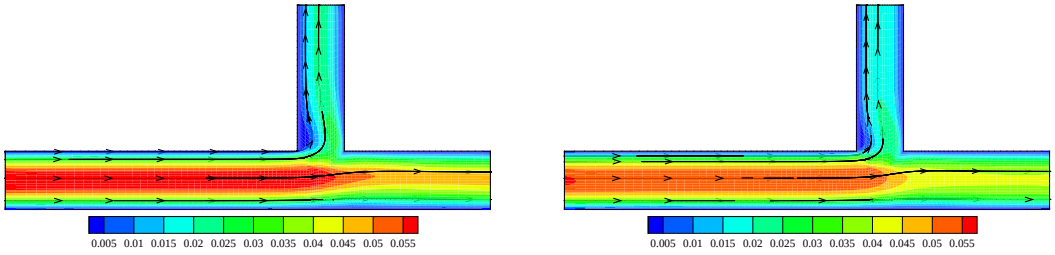
(a) Newtonian (b) generalized Newtonian

Figure 3: Velocity isolines of steady flows for generalized Newtonian fluids.

As initial condition the following model parameters are used: reference radius $R = 0.0031\ m$, branch radius $R_1 = 0.0025\ m$, dynamic viscosity $\mu = 0.0036\ Pa \cdot s$, reference velocity $U_0 = 0.0615\ m \cdot s^{-1}$, constant density $\rho = 1050\ kg \cdot m^{-3}$. At the inlet the Dirichlet boundary conditions for velocity are used, the parabolic profile with reference velocity value U_0. At the outlet the constant pressure values are prescribed. In Fig. 2 the axial velocity profile close to the branching is shown. The line for Newtonian fluids is similar to the parabolic line, as was assumed. It is clear that the shear-thinning fluids attain lower maximum velocity in the central part of the channel (close to the axis of symmetry) which is compensated by the increase of local velocity in the boundary layer close to the wall.

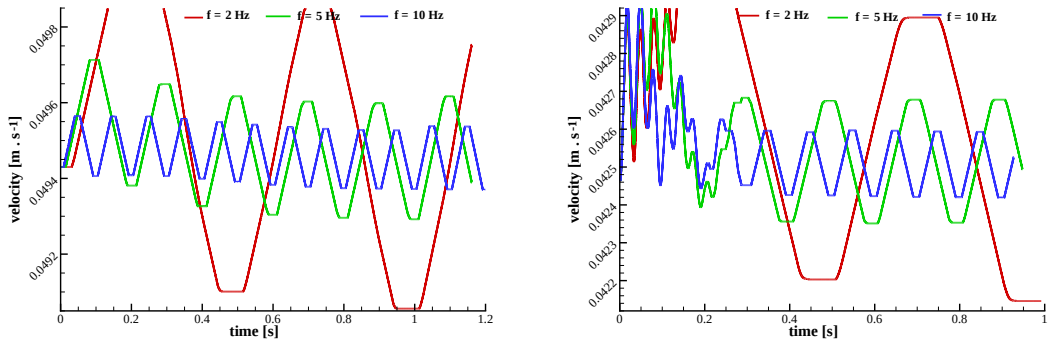
In Fig. 3 the velocity isolines and the cuts through the channel are shown. The axial velocity isolines in the center-plane area for tested fluids are shown in Fig. 4. It can be observed from these that the size of separation region for generalized Newtonian fluids is smaller than for Newtonian fluids.

These steady numerical results are used as initial condition for unsteady numerical computation. The artificial compressibility method with high β parameter is used for the unsteady numerical simulation.



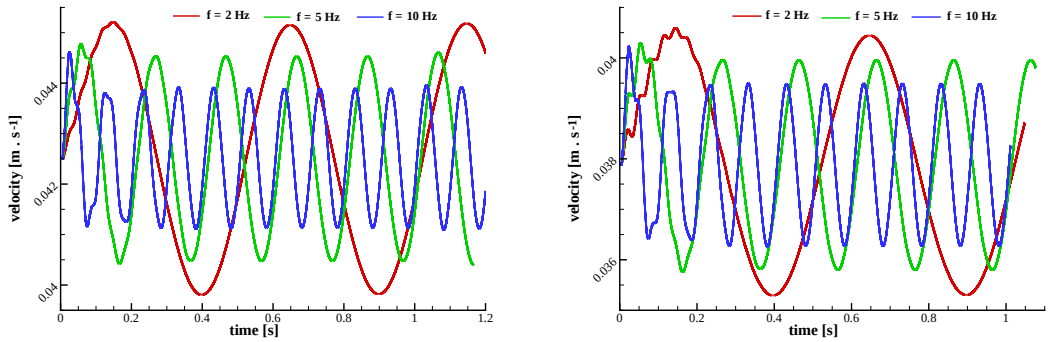
(a) Newtonian (b) generalized Newtonian

Figure 4: Axial velocity isolines in the center-plane area for generalized Newtonian fluids.



(a) Newtonian (b) generalized Newtonian

Figure 5: The graphs of the velocity as the function of time for three tested values of frequency - oscillation of the pressure at the outlet.



(a) Newtonian (b) generalized Newtonian

Figure 6: The graphs of the velocity as the function of time for three tested values of frequency - pulsation of the velocity at the inlet.

Figs. 5 and 6 show the magnitude of the velocity during the time period. The velocity is taken from the point inside the domain. In Fig. 5 the pressure at the outlet (branch) and in Fig. 6 the velocity at the inlet are prescribed by (7).

The numerical results (the magnitude of the velocity) are affected by the choice of the oscillation's variable (pressure or velocity) as well as by the velocity field. In the case where the boundary function (7) was used for pulsation of the velocity at the inlet, the numerical results (Fig. 6) for Newtonian and generalized Newtonian fluids are very similar independently of the position in the domain. On the other hand if the oscillation of the pressure at the outlet is considered than the resulting magnitudes of the velocity (taken in the same point as in the previous case) are different for Newtonian and generalised Newtonian fluids (6). It is affected by the type of fluid and by the behaviour of fluids in the branching (the size of separation region).

5. Conclusion

Classical Newtonian model as well as its generalized (shear-thinning) modification have been considered to model flow in the branching channel with T-junction. The goal of this study is to investigate shear-thinning effects in steady flow simulations. From the presented velocity profile, it is clear that the shear thinning fluids (generalized Newtonian fluids) attain lower maximum velocity in the central part of the channel (close to the axis of symmetry) which is compensated by the increase of local velocity in the boundary layer close to the wall.

The numerical method used to solve the governing equations seems to be sufficiently robust and efficient for the appropriate resolution of the given class of problems.

For unsteady simulation Newtonian and generalized Newtonian fluids were considered. An artificial compressibility approach was considered for numerical solution of unsteady governing equations. The artificial compressibility parameter β was set to be 6 $m \cdot s^{-1}$. Several values of frequency were tested. Pressure value in the branch outlet and the velocity value in the inlet were prescribed by a periodic function. The numerical results given by graphs of the velocity as the function of time in the point in the domain were presented.

For the future work, extending this unsteady simulation for generalized Oldroyd-B fluids flow will be considered. The dual-time stepping method will be used.

Acknowledgments

This work was supported by the grant SGS16/206/OHK2/3T/12.

References

[1] Akbarzadeh, P.: Pulsatile magneto-hydrodynamic blood flows through porous blood vessels using a third grade non-Newtonian fluids model. Computer Methods and Programs in Biomedicine **126** (2016), 3–19.

[2] Ali, N., Zaman, A., Sajid, M., Nietoc, J. J., and Torres, A.: Unsteady non-Newtonian blood flow through a tapered overlapping stenosed catheterized vessel. Math. Biosci. **269** (2015), 94–103.

[3] Beneš, L., Louda, P., Kozel, K., Keslerová, R., and Štigler, J.: Numerical simulations of flow through channels with T-junction. Appl. Math. Comput. **219** (2013), 7225–7235.

[4] Bodnar, T., Sequeira, A., and Prosi, M.: On the shear-thinning and viscoelastic effects of blood flow under various flow rates. Appl. Math. Comput. **217** (2010), 5055–5067.

[5] Chorin, A. J.: A numerical method for solving incompressible viscous flow problem. J. Comput. Phys. **135** (1967), 118–125.

[6] Honzátko, R.: *Numerical simulations of incompressible flows with dynamical and aeroelastic effects*. Ph.D. thesis, Czech Technical University in Prague, 2007.

[7] Ikbal, M. A., Chakravartya, S., Wong, K. K. L., Mazumdar, J., and Mandal, P. K.: Unsteady response of non-Newtonian blood flow through a stenosed artery in magnetic field. J. Comput. Appl. Math. **230** (2009), 243–259.

[8] Jonášová, A.: *Computational modelling of hemodynamics for non-invasive assessment of arterial bypass graft patency*. Ph.D. thesis, University of West Bohemia, Pilsen, 2014.

[9] Keslerová, R. and Kozel, K.: Numerical simulation of viscous and viscoelastic fluids flow by finite volume method. In: J. Fořt, J. Fürst, J. Halama, R. Herbin and F. Hubert (Eds.), *Finite volumes for complex applications VI — problems and perspectives*, pp. 589–596. Springer-Verlag Berlin Heidelberg, 2011.

[10] Keslerová, R. and Kozel, K.: Numerical simulation of generalized Newtonian and Oldroyd-B fluids flow. In: J. Chleboun, K. Segeth, J. Šístek and T. Vejchodský (Eds.), *Proceedings of Programs and Algorithms of Numerical Mathematics 16*, pp. 112–117. IM ASCR, Prague, 2013.

[11] LeVeque, R.: *Finite-volume methods for hyperbolic problems.* Cambridge University Press, 2004.

[12] Matos, H. M. and Oliveira, P. J.: Steady flows of constant-viscosity viscoelastic fluids in a planar T-junction. Journal of Non-Newtonian Fluid Mechanics **213** (2014), 15–26.

[13] Poole, R. J., Haward, S. J., and Alves, M. A.: Symmetry-breaking bifurcations in T-channel flows: effects of fluid viscoelasticity. Procedia Engineering **79** (2014), 28–34.

[14] Rabby, M. G., Razzak, A., and Molla, M. M.: Pulsatile non-Newtonian blood flow through a model of arterial stenosis. Procedia Engineering **56** (2013), 225–231.

[15] Vimmr, J., Jonášová, A., and Bublík, O.: Modelling of blood's non-Newtonian behaviour in patient-specific aorto-coronary bypass grafts. In: *Proc. 10th World Congress on Computational Mechanics*, pp. 1634–1653. São Paulo, Blucher, 2014.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

DISCONTINUOUS GALERKIN METHOD FOR A 2D NONLOCAL FLOCKING MODEL

Václav Kučera, Andrea Živčáková

Charles University in Prague, Faculty of Mathematics and Physics
Sokolovská 83, 186 75 Praha, Czech Republic
kucera@karlin.mff.cuni.cz, zivcakova@karlin.mff.cuni.cz

Abstract: We present our work on the numerical solution of a continuum model of flocking dynamics in two spatial dimensions. The model consists of the compressible Euler equations with a nonlinear nonlocal term which requires special treatment. We use a semi-implicit discontinuous Galerkin scheme, which proves to be efficient enough to produce results in 2D in reasonable time. This work is a direct extension of the authors' previous work in 1D.

Keywords: discontinuous Galerkin method, semi-implicit time discretization, nonlocal problems, flocking dynamics

MSC: 65M60, 35Q92, 35Q35

1. Introduction

The study of emergent collective behavior and phenomena in natural and artificial systems is a very popular and diverse field, cf. [1], [9], [10] for an overview. One of the topics of interest is that of global coordination of behavior seen in flocks of birds or other similar self-propelled entities. The study of such problems leads to descriptions on various levels (particle, kinetic and hydrodynamic) and various models of the underlying behavior of the individuals, cf. [10]. The paper [7] deals with the derivation of a hydrodynamic limit of a certain modification of the famous Cucker-Smale model [2], [3]. The resulting partial differential equation consists of the compressible Euler equation of gas dynamics, with an additional nonlinear nonlocal term. The presence of this term leads to difficulties in constructing an efficient numerical scheme, which would produce results in a reasonable time (e.g. hours) even on very coarse grids in 1D, cf. [7].

This short note presents results obtained using a two-dimensional version of the 1D numerical scheme presented in [8]. The scheme is based on a semi-implicit time discretization of the discontinuous Galerkin (DG) scheme from [6] originally applied to the compressible Euler equations. In [8], the semi-implicit scheme was extended to include the nonlinear and nonlocal interaction terms of the considered flocking

DOI: 10.21136/panm.2016.08

model in an efficient way. Here we show how to perform the discretization in the 2D case and present numerical experiments obtained with the resulting scheme.

2. Mathematical model

We consider the hydrodynamic model of flocking derived in [7] as a macroscopic limit of a modification of the Cucker-Smale model [2], [3].

Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, be a bounded domain and $0 < L < +\infty$ is the length of a time interval. We set $Q_L := \Omega \times (0, L)$. We treat the following problem: Find $\rho, E : Q_L \to \mathbb{R}$, $\boldsymbol{u} = (u_1, \dots, u_d) : Q_L \to \mathbb{R}^d$ such that

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \boldsymbol{u}) &= 0, \\ \frac{\partial (\rho \boldsymbol{u})}{\partial t} + \operatorname{div}(\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p &= \lambda \boldsymbol{\mathcal{A}}(\rho, \boldsymbol{u}), \\ \frac{\partial E}{\partial t} + \operatorname{div}\Big(\boldsymbol{u}\,(E + p)\Big) &= \lambda \mathcal{B}(\rho, \boldsymbol{u}, E), \end{aligned} \tag{1}$$

where ρ denotes the density, E energy, p pressure and $\boldsymbol{u}$ velocity. The relations between E and p are

$$E = \rho\left(\frac{T}{\gamma - 1} + \frac{|\boldsymbol{u}|^2}{2}\right), \; p = \rho T, \tag{2}$$

where $\gamma = \frac{d+2}{d}$ is the adiabatic constant and T temperature. These quantities describe the macroscopic behavior of agents behaving according to the microscopic model considered in [7]. In this context, the basic variables must be interpreted in the Boltzmannian framework - e.g. momentum $\rho\boldsymbol{u}$ and temperature T are the first and second moments of the density distribution function $f(x, \boldsymbol{v}, t)$ in the corresponding kinetic (mesoscopic) model from which the hydrodynamic model (1) is derived, cf. [7].

The right-hand side functions $\boldsymbol{\mathcal{A}}$ and $\mathcal{B}$ are given by

$$\begin{aligned} \boldsymbol{\mathcal{A}}(\rho, \boldsymbol{u})(\boldsymbol{x}, t) &= \int_{\mathbb{R}^d} \widetilde{\boldsymbol{n}}(\boldsymbol{x}, \boldsymbol{y}) b(\boldsymbol{x}, \boldsymbol{y}) \Big(\boldsymbol{u}(\boldsymbol{y}, t) - \boldsymbol{u}(\boldsymbol{x}, t)\Big) \cdot \widetilde{\boldsymbol{n}}(\boldsymbol{x}, \boldsymbol{y}) \rho(\boldsymbol{x}, t) \rho(\boldsymbol{y}, t) \mathrm{d}\boldsymbol{y}, \\ \mathcal{B}(\rho, \boldsymbol{u}, E)(\boldsymbol{x}, t) &= \int_{\mathbb{R}^d} b(\boldsymbol{x}, \boldsymbol{y}) \rho(\boldsymbol{x}, t) \Big(\rho(\boldsymbol{y}, t) \widetilde{\boldsymbol{n}}(\boldsymbol{x}, \boldsymbol{y}) \cdot \boldsymbol{u}(\boldsymbol{x}, t) \widetilde{\boldsymbol{n}}(\boldsymbol{x}, \boldsymbol{y}) \cdot \boldsymbol{u}(\boldsymbol{y}, t) \\ &\quad - \frac{2}{d} E(\boldsymbol{y}, t)\Big) \mathrm{d}\boldsymbol{y}, \end{aligned} \tag{3}$$

where

$$b(\boldsymbol{x}, \boldsymbol{y}) = \frac{\lambda K}{(\lambda + |\boldsymbol{x} - \boldsymbol{y}|^2)^{\beta+1}}, \; \widetilde{\boldsymbol{n}}(\boldsymbol{x}, \boldsymbol{y}) = (\widetilde{n}_1, \dots, \widetilde{n}_d) = \frac{\boldsymbol{x} - \boldsymbol{y}}{|\boldsymbol{x} - \boldsymbol{y}|}, \tag{4}$$

and $K, \lambda > 0$ and $\beta \geq 0$ are given constants.

By omitting the right-hand side terms $\boldsymbol{\mathcal{A}}, \mathcal{B}$ from (1), we obtain the compressible Euler equations. In this light, we rewrite system (1) as a system of conservation laws with a right-hand side source terms:

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = \boldsymbol{g}(\boldsymbol{w}) \quad \text{in } Q_L, \tag{5}$$

where

$$\begin{aligned} \boldsymbol{w} &= (w_1, \ldots, w_{d+2}) = (\rho, \rho u_1, \ldots, \rho u_d, E)^\top \in \mathbb{R}^{d+2}, \\ \boldsymbol{f}_s(\boldsymbol{w}) &= \big(\rho u_s, \rho u_s u_1 + \delta_{s1}, \rho u_s u_2 + \delta_{s2} p, (E+p)u_s\big)^\top, \\ \boldsymbol{g}(\boldsymbol{w}) &= \big(0, \boldsymbol{\mathcal{A}}(\boldsymbol{w}), \mathcal{B}(\boldsymbol{w})\big)^\top = \big(0, \mathcal{A}_1(\boldsymbol{w}), \ldots, \mathcal{A}_d(\boldsymbol{w}), \mathcal{B}(\boldsymbol{w})\big)^\top. \end{aligned} \tag{6}$$

Here δ is the Kronecker delta. The vector-valued function $\boldsymbol{w}$ is called the *state vector* and the functions $\boldsymbol{f}_s, s = 1, \ldots, d$, are the *Euler fluxes*. In (5), we write the right-hand side terms $\boldsymbol{\mathcal{A}}, \mathcal{B}$ as functions of the state vector $\boldsymbol{w}$, although in (1), they are written terms of the nonconservative variables. Expressing $\boldsymbol{\mathcal{A}}, \mathcal{B}$ in $\boldsymbol{w}$ in a suitable way is a key ingredient in our scheme and will be described in detail in Section 3.4.

The resulting system is equipped with the initial condition $\boldsymbol{w}(x, 0) = \boldsymbol{w}^0(x)$ for $x \in \Omega$. In 1D case we use periodic boundary conditions and in 2D we chose conditions corresponding to solid impermeable walls, i.e. $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on $\partial\Omega$.

Euler fluxes are homogeneous functions, which implies the useful relations

$$\boldsymbol{f}_s(\boldsymbol{w}) = \mathbb{A}_s(\boldsymbol{w})\boldsymbol{w}, \quad \mathbb{A}_s = \frac{D\boldsymbol{f}_s}{D\boldsymbol{w}}, \quad s = 1, \ldots, d. \tag{7}$$

Furthermore, the Euler flux is *diagonally hyperbolic*: The matrix

$$\mathbb{P}(\boldsymbol{w}, \bar{\boldsymbol{n}}) := \sum_{s=1}^{d} \mathbb{A}_s(\boldsymbol{w})\bar{n}_s \tag{8}$$

is diagonalizable with real eigenvalues, where $\bar{\boldsymbol{n}} = (\bar{n}_1, \ldots, \bar{n}_d)$ denotes a unit vector. This means there exists $\mathbb{T}(\boldsymbol{w}, \bar{\boldsymbol{n}}) \in \mathbb{R}^{d+2,d+2}$ and a diagonal matrix $\mathbb{D}(\boldsymbol{w}, \bar{\boldsymbol{n}}) \in \mathbb{R}^{d+2,d+2}$ such that

$$\mathbb{P}(\boldsymbol{w}, \bar{\boldsymbol{n}}) = \mathbb{T}\mathbb{D}\,\mathbb{T}^{-1}, \quad \text{where } \mathbb{D}(\boldsymbol{w}, \bar{\boldsymbol{n}}) = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{d+2}). \tag{9}$$

3. Discretization

Let $\mathcal{T}_h$ be a triangulation of Ω, i.e. a partition of $\overline{\Omega}$ into a finite number of closed simplices with mutually disjoint interiors. By $\mathcal{F}_h$ we denote the system of all faces (i.e. nodes in 1D) of $\mathcal{T}_h$. For each $\Gamma \in \mathcal{F}_h$ we choose and fix a unit normal $\boldsymbol{n}_\Gamma$, which, for $\Gamma \subset \partial\Omega$ has the same orientation as the outer normal to Ω.

For each *interior* face $\Gamma \in \mathcal{F}_h$ there exist two neighbours $K_\Gamma^{(L)}$, $K_\Gamma^{(R)} \in \mathcal{T}_h$ such that $\Gamma = K_\Gamma^{(L)} \cap K_\Gamma^{(R)}$. We use the convention that $\boldsymbol{n}_\Gamma$ is the outer normal to the element $K_\Gamma^{(L)}$. For a function v piecewise defined on $\mathcal{T}_h$ and $\Gamma \in \mathcal{F}_h$ we introduce:

$$v|_\Gamma^{(L)} = \text{ the trace of } v|_{K_\Gamma^{(L)}} \text{ on } \Gamma, \qquad v|_\Gamma^{(R)} = \text{ the trace of } v|_{K_\Gamma^{(R)}} \text{ on } \Gamma,$$
$$\langle v \rangle_\Gamma = \tfrac{1}{2}\big(v|_\Gamma^{(L)} + v|_\Gamma^{(R)}\big), \qquad [v]_\Gamma = v|_\Gamma^{(L)} - v|_\Gamma^{(R)},$$

where on $\partial\Omega$, we define $v|_\Gamma^{(R)}$ using the boundary conditions as in [6].

If $[\,\cdot\,]_\Gamma$ and $\langle\,\cdot\,\rangle_\Gamma$ appear in an integral of the form $\int_\Gamma \dots \,\mathrm{d}S$, we omit the subscript Γ and write simply $[\,\cdot\,]$ and $\langle\,\cdot\,\rangle$. We shall use the following notation:

$$\int_{\mathcal{F}_h} F(x)\,\mathrm{d}S = \sum_{\Gamma\in\mathcal{F}_h} \int_\Gamma F(x)\,\mathrm{d}S$$

and similarly for $\int_{\partial K} F(x)\,\mathrm{d}S$ etc.

Let $m \geq 0$ be an integer. The approximate solution will be sought in the space of discontinuous piecewise polynomial functions

$$\boldsymbol{S}_h := [S_h]^{d+2}, \text{ where } S_h = \{v;\, v|_K \in P^m(K), \forall K \in \mathcal{T}_h\}.$$

Here $P^m(K)$ denotes the space of all polynomials on K of degree $\leq m$.

3.1. Discontinuous Galerkin space semidiscretization

Similarly as in [6] in the case of the Euler equations and in [8] in the case of the 1D flocking model, we multiply (5) by a test function $\boldsymbol{\varphi} \in \boldsymbol{S}_h$, integrate over $K \in \mathcal{T}_h$ and apply integration by parts in the convective terms and rearrange:

$$\int_\Omega \frac{\partial \boldsymbol{w}}{\partial t} \cdot \boldsymbol{\varphi}\,\mathrm{d}\boldsymbol{x} + \sum_{s=1}^d \left(\int_{\mathcal{F}_h} \boldsymbol{f}_s(\boldsymbol{w})\, n_s \cdot [\boldsymbol{\varphi}]\,\mathrm{d}S - \sum_{K\in\mathcal{T}_h} \int_K \boldsymbol{f}_s(\boldsymbol{w}) \cdot \frac{\partial \boldsymbol{\varphi}}{\partial x_s}\mathrm{d}\boldsymbol{x} \right) = \int_\Omega \boldsymbol{g}(\boldsymbol{w}) \cdot \boldsymbol{\varphi}\mathrm{d}\boldsymbol{x}. \tag{10}$$

Since $\boldsymbol{w}$ is approximated by a discontinuous function, we use a *numerical flux* $\mathbf{H}_s(\boldsymbol{w}^{(L)}, \boldsymbol{w}^{(R)}, \boldsymbol{n})$ in the boundary integral term:

$$\int_{\mathcal{F}_h} \sum_{s=1}^d \boldsymbol{f}_s(\boldsymbol{w}) n_s \cdot [\boldsymbol{\varphi}]\,\mathrm{d}S \approx \int_{\mathcal{F}_h} \mathbf{H}(\boldsymbol{w}^{(L)}, \boldsymbol{w}^{(R)}, \boldsymbol{n}) \cdot [\boldsymbol{\varphi}]\,\mathrm{d}S. \tag{11}$$

Specifically, in our implementation we used the Vijayasundaram numerical flux, cf. Section 3.2.

Now we can define the following forms defined for $\boldsymbol{w}, \boldsymbol{\varphi} \in H^1(\Omega, \mathcal{T}_h)$.
Convective form:

$$b_h(\boldsymbol{w}, \boldsymbol{\varphi}) = \int_{\mathcal{F}_h} \mathbf{H}(\boldsymbol{w}^{(L)}, \boldsymbol{w}^{(R)}, \boldsymbol{n}) \cdot [\boldsymbol{\varphi}]\,\mathrm{d}S - \sum_{K\in\mathcal{T}_h} \int_K \sum_{s=1}^d \boldsymbol{f}_s(\boldsymbol{w}) \cdot \frac{\partial \boldsymbol{\varphi}}{\partial x_s}\,\mathrm{d}\boldsymbol{x}.$$

Right-hand side source form:

$$l_h(\boldsymbol{w}, \boldsymbol{\varphi}) = -\int_\Omega \boldsymbol{g}(\boldsymbol{w}) \cdot \boldsymbol{\varphi} \, \mathrm{d}\boldsymbol{x}.$$

Definition 1. *We say that* $\boldsymbol{w}_h \in C^1([0,T]; S_h)$ *is a DG solution of problem (5) if* $\boldsymbol{w}_h(0) = \boldsymbol{w}_h^0$*, an* $\boldsymbol{S}_h$*-approximation of the initial condition* $\boldsymbol{w}^0$ *and*

$$\frac{d}{dt}(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) + b_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) + l_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = 0 \quad \forall \boldsymbol{\varphi}_h \in \boldsymbol{S}_h, \ \forall t \in (0,T). \tag{12}$$

3.2. Numerical flux

The choice of the numerical flux is a very important topic in the finite volume and DG schemes. We use the *Vijayasundaram* numerical flux, cf. [11], which is suitable for our semi-implicit time discretization. This numerical flux is based on the flux vector splitting concept, and can be viewed as an extension of the upwind numerical flux to systems of equations. We use the diagonal hyperbolicity (9) and define the *positive* and *negative* parts of matrix $\mathbb{P}$:

$$\mathbb{P}^{\pm}(\boldsymbol{w}, \bar{\boldsymbol{n}}) = \mathbb{T}(\boldsymbol{w}, \bar{\boldsymbol{n}}) \mathbb{D}^{\pm}(\boldsymbol{w}, \bar{\boldsymbol{n}}) \mathbb{T}^{-1}(\boldsymbol{w}, \bar{\boldsymbol{n}}), \ \mathbb{D}^{\pm}(\boldsymbol{w}, \bar{\boldsymbol{n}}) = \mathrm{diag}(\lambda_1^{\pm}, \lambda_2^{\pm}, \ldots, \lambda_{d+2}^{\pm}), \tag{13}$$

where $\lambda^+ = \max\{0, \lambda\}$, $\lambda^- = \min\{0, \lambda\}$. Then $\mathbb{P}(\boldsymbol{w}, \bar{\boldsymbol{n}}) = \mathbb{P}^+(\boldsymbol{w}, \bar{\boldsymbol{n}}) + \mathbb{P}^-(\boldsymbol{w}, \bar{\boldsymbol{n}})$ and we can define the Vijayasundaram numerical flux as

$$\mathbf{H}_{VS}(\boldsymbol{w}^{(L)}, \boldsymbol{w}^{(R)}, \bar{\boldsymbol{n}}) = \mathbb{P}^+\left(\tfrac{\boldsymbol{w}^{(L)} + \boldsymbol{w}^{(R)}}{2}, \bar{\boldsymbol{n}}\right) \boldsymbol{w}^{(L)} + \mathbb{P}^-\left(\tfrac{\boldsymbol{w}^{(L)} + \boldsymbol{w}^{(R)}}{2}, \bar{\boldsymbol{n}}\right) \boldsymbol{w}^{(R)}. \tag{14}$$

Explicit formulas for $\mathbb{P}, \mathbb{T}, \mathbb{T}^{-1}$ and $\mathbb{D}$ can be found e.g. in [5].

3.3. Time discretization

After choosing some basis of the space $\boldsymbol{S}_h$, equation (12) represents a system of nonlinear ordinary differential equations, which must be discretized with respect to time. Due to severe time step restrictions caused by the nonlocality and nonlinearity of system (1), we want to avoid using an explicit scheme. However an implicit time discretization is also very expensive due to its nonlinearity. Therefore we choose the semi-implicit scheme of [6] as a basis and apply it to our problem.

Let $0 = t_0 < t_1 < t_2 < \ldots$ be a partition of time interval $[0, L]$ and define $\tau_k = t_{k+1} - t_k$. We use a first order backward difference approximation for the time derivative, i.e.

$$\frac{\partial \boldsymbol{w}_h(t_{k+1})}{\partial t} \approx \frac{\boldsymbol{w}_h^{k+1} - \boldsymbol{w}_h^k}{\tau_k},$$

where $\boldsymbol{w}_h^k \approx \boldsymbol{w}_h(t_k)$ and $\boldsymbol{w}_h^k \in \boldsymbol{S}_h$. The backward Euler scheme reads

$$\left(\frac{\boldsymbol{w}_h^{k+1} - \boldsymbol{w}_h^k}{\tau_k}, \boldsymbol{\varphi}_h\right) + b_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + l_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = 0 \quad \forall \boldsymbol{\varphi}_h \in \boldsymbol{S}_h, \tag{15}$$

for all $k = 0, 1, \ldots$. Equation (15) is nonlinear with respect to the unknown $\boldsymbol{w}_h^{k+1}$, therefore we linearize the scheme.

In the convective form, we linearize the interior terms using the homogeneity (7) as $\boldsymbol{f}_s(\boldsymbol{w}_h^{k+1}) \approx \mathbb{A}_s(\boldsymbol{w}_h^k)\boldsymbol{w}_h^{k+1}$. In the boundary terms, we use the Vijayasundaram numerical flux (14) and linearize by taking the matrices $\mathbb{P}_s^+$ and $\mathbb{P}_s^-$ from the previous time level. Thus we get the linearize convective form

$$\begin{aligned}\tilde{b}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = &- \sum_{K\in\mathcal{T}_h} \int_K \sum_{s=1}^{d} \mathbb{A}_s(\boldsymbol{w}_h^k)\boldsymbol{w}_h^{k+1} \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s} \,\mathrm{d}\boldsymbol{x} \\ &+ \int_{\mathcal{F}_h} \Big(\mathbb{P}^+\big(\langle \boldsymbol{w}_h^k\rangle, \boldsymbol{n}\big)\boldsymbol{w}_h^{k+1,(L)} + \mathbb{P}^-\big(\langle \boldsymbol{w}_h^k\rangle, \boldsymbol{n}\big)\boldsymbol{w}_h^{k+1,(R)}\Big) \cdot [\boldsymbol{\varphi}_h] \,\mathrm{d}S.\end{aligned} \tag{16}$$

In the source terms we again linearize them to obtain the approximation $l_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) \approx \tilde{l}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h)$, cf. Section 3.4 for details.

Collecting all the considerations, we obtain the following semi-implicit DG scheme:

Definition 2. *We say that the sequence $\boldsymbol{w}_h^k \in \boldsymbol{S}_h, k = 0, 1, \ldots,$ is a semi-implicit DG solution of problem (5) if for all $\boldsymbol{\varphi}_h \in S_h$*

$$\left(\frac{\boldsymbol{w}_h^{k+1} - \boldsymbol{w}_h^k}{\tau_k}, \boldsymbol{\varphi}_h\right) + \tilde{b}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) + \tilde{l}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = 0. \tag{17}$$

3.4. Linearization of the source terms l_h

As the 1D case is treated in [8], we demonstrate the linearization of the nonlocal terms only in 2D. First, we rewrite the right-hand side integrals $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ in terms of the conservative variables.

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{w}(\boldsymbol{x},t)) &= \int_\Omega \begin{pmatrix} 0 \\ \mathcal{A}_1(\boldsymbol{w}(\boldsymbol{x},t)) \\ \mathcal{A}_2(\boldsymbol{w}(\boldsymbol{x},t)) \\ \mathcal{B}(\boldsymbol{w}(\boldsymbol{x},t)) \end{pmatrix} \mathrm{d}\boldsymbol{y} \\ &= \int_\Omega b(\boldsymbol{x},\boldsymbol{y})\rho(\boldsymbol{x},t) \begin{pmatrix} 0 \\ \widetilde{n}_1\rho(\boldsymbol{y},t)\big(\boldsymbol{u}(\boldsymbol{y},t) - \boldsymbol{u}(\boldsymbol{x},t)\big)\cdot\widetilde{\boldsymbol{n}}(\boldsymbol{x},\boldsymbol{y}) \\ \widetilde{n}_2\rho(\boldsymbol{y},t)\big(\boldsymbol{u}(\boldsymbol{y},t) - \boldsymbol{u}(\boldsymbol{x},t)\big)\cdot\widetilde{\boldsymbol{n}}(\boldsymbol{x},\boldsymbol{y}) \\ \rho(\boldsymbol{y},t)\boldsymbol{u}(\boldsymbol{x},t)\cdot\widetilde{\boldsymbol{n}}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{u}(\boldsymbol{y},t)\cdot\widetilde{\boldsymbol{n}}(\boldsymbol{x},\boldsymbol{y}) - E(\boldsymbol{y},t) \end{pmatrix} \mathrm{d}\boldsymbol{y} \\ &= \int_\Omega b(\boldsymbol{x},\boldsymbol{y})\mathbb{U}\big(\boldsymbol{w}(\boldsymbol{y},t)\big)\boldsymbol{w}(\boldsymbol{x},t)\mathrm{d}\boldsymbol{y},\end{aligned} \tag{18}$$

where $\mathbb{U}\big(\boldsymbol{w}(\boldsymbol{y},t)\big)$ is

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \widetilde{n}_1^2 w_2(\boldsymbol{y},t) + \widetilde{n}_1\widetilde{n}_2 w_3(\boldsymbol{y},t) & -\widetilde{n}_1^2 w_1(\boldsymbol{y},t) & -\widetilde{n}_1\widetilde{n}_2 w_1(\boldsymbol{y},t) & 0 \\ \widetilde{n}_1\widetilde{n}_2 w_2(\boldsymbol{y},t) + n_2^2 w_3(\boldsymbol{y},t) & -\widetilde{n}_1\widetilde{n}_2 w_1(\boldsymbol{y},t) & -\widetilde{n}_2^2 w_1(\boldsymbol{y},t) & 0 \\ -w_4(\boldsymbol{y},t) & \widetilde{n}_1\widetilde{n}_2 w_3(\boldsymbol{y},t) + \widetilde{n}_1^2 w_2(\boldsymbol{y},t) & \widetilde{n}_1\widetilde{n}_2 w_2(\boldsymbol{y},t) + \widetilde{n}_2^2 w_3(\boldsymbol{y},t) & 0 \end{pmatrix}.$$

Approximating $\boldsymbol{w}(\boldsymbol{x},t) \approx \boldsymbol{w}_h^{k+1}(\boldsymbol{x})$ and $\boldsymbol{w}(\boldsymbol{y},t) \approx \boldsymbol{w}_h^{k}(\boldsymbol{y})$ leads to the linearized form

$$\tilde{l}_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = -\int_\Omega \left(\int_\Omega b(\boldsymbol{x},\boldsymbol{y}) \mathbb{U}\big(\boldsymbol{w}_h^k(\boldsymbol{y})\big)\, \mathrm{d}\boldsymbol{y} \right) \boldsymbol{w}_h^{k+1}(\boldsymbol{x}) \cdot \boldsymbol{\varphi}_h(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{19}$$

If we use a basis for $\boldsymbol{S}_h$ consisting of functions whose support is exactly one element, the form (19) does not change the structure of the system matrix, since it contributes only to the block-diagonal. This is important as other expressions than (18) are possible, however they lead to a full system matrix which is undesirable.

We note, that the computation of the source terms (19) is very time consuming due to their nonlocal nature. Even if the basis functions of $\boldsymbol{S}_h$ are local, in order to evaluate $\tilde{l}_h$, we must compute the inner integral $\int_\Omega b(\boldsymbol{x},\boldsymbol{y}) \mathbb{U}_2\big(\boldsymbol{w}_h^k(\boldsymbol{y})\big)\, \mathrm{d}\boldsymbol{y}$, which is time consuming due to the slow decay of the function $b(\boldsymbol{x},\boldsymbol{y})$.

3.5. Shock capturing and treatment of vacuum

Numerical experiments from the 1D case show that the solution of (1) typically contains quickly moving shocks and near vacuum states. Often, one observes shocks neighboring a vacuum, even in the stationary case, which is impossible for the Euler equations themselves. The situation is quite similar in 2D. To treat these problems, we include the shock capturing terms of [6]. Furthermore, special attention must be given to the treatment of the occurrence of vacuum - when ρ, p or T are near zero, or even numerically negative due to spurious oscillations in the solution, the matrices $\mathbb{A}, \mathbb{P}^+, \mathbb{P}^-$ are no longer defined and the computation collapses. We therefore use the "postprocessing" approach from [8]. The newly computed state $\boldsymbol{w}_h^k$ is modified thus: if $\rho < \varepsilon$ or $T < \varepsilon$, then set $\rho := \varepsilon$ or $T := \varepsilon$ and recompute the energy E using relation (2). This defines a new state $\widetilde{\boldsymbol{w}}_h^k$ which is used in (17) instead of $\boldsymbol{w}_h^k$ to compute $\boldsymbol{w}_h^{k+1}$. In our case, we use $\varepsilon := 10^{-5}$. In combination with the shock capturing procedure of [6], this yields a sufficiently robust scheme.

4. Numerical tests

We considered the 2D problem on a unit square. In the first numerical experiment was prescribed the initial density distribution as a Gaussian bump given by $\rho(x,y) = \exp(-10|(x,y) - (0.5,0.5)|^2)$ with constant temperature $T = 10$ and the velocity distribution $\boldsymbol{u}(x,y) = (0,0)$. This is a two-dimensional analogue of the problem solved in [8]. Similarly as in the 1D case, after an initial phase of rapid oscillations, was observed the concentration of the hump. Density plots at chosen time instances are given in Figure 1 (ordered left to right, top to bottom). Due to the concentration of density, the solution converges to a tall "spike", so the graphs are cut of at the same value of 1.1 for clarity.

The second numerical experiment consisted of two neighboring Gaussian bumps, one smaller than the other in magnitude. In this case, the two groups merged into one single "flock", as seen in Figure 2.

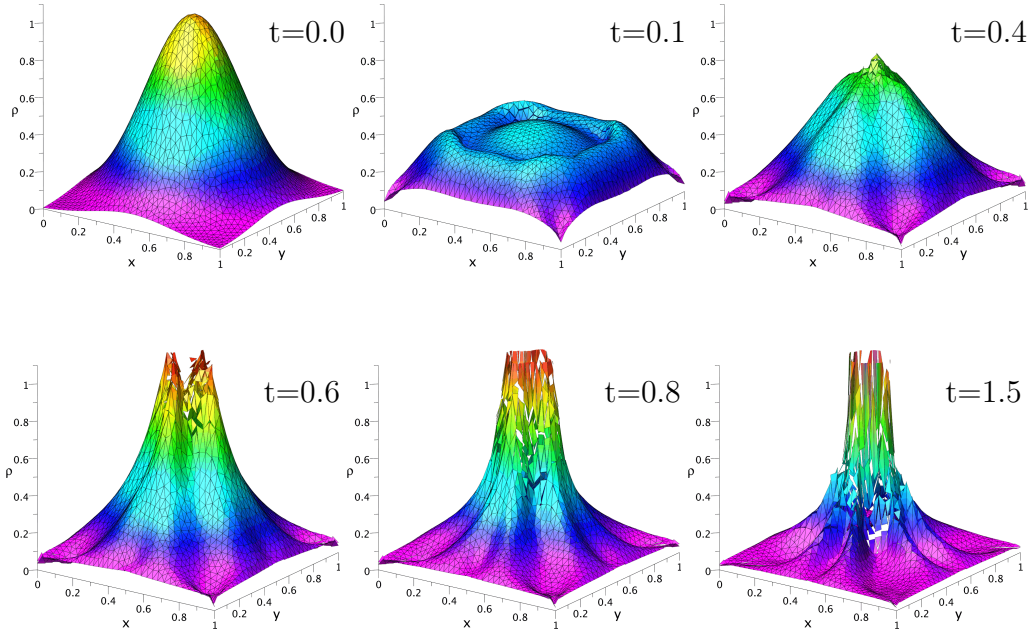


Figure 1: Numerical experiment 1 – density distribution.

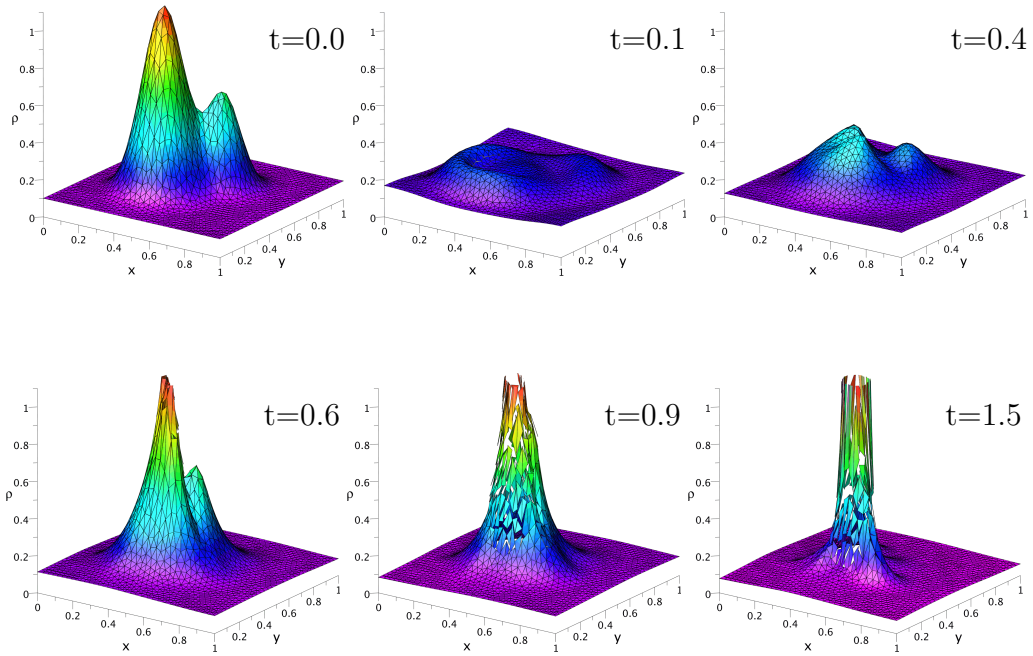


Figure 2: Numerical experiment 2 – density distribution.

We performed both numerical experiments on an unstructured mesh consisting of 3057 element with piecewise linear approximation. The constants needed in (4) were chosen as $K = 10$, $\beta = 0.1$, $\lambda = 1$, following [7].

5. Conclusions

In this paper we have presented a practical application of the discontinuous Galerkin method to nonlocal problems, namely for a complicated nonlinear and nonlocal version of the compressible Euler equations describing the dynamics of flocks of birds. Straightforward discretizations of the problem are extremely inefficient due to its nonlocal nature and the need to evaluate the nonlocal terms too many times. We shown how to obtain numerical solutions in reasonable time using a very efficient time discretization of the discontinuous Galerkin method.

Acknowledgements

The research of V. Kučera is supported by the Grant No. P201/13/00522S of the Czech Science Foundation. The research of A. Živčáková is supported by the Charles University in Prague, project GA UK No. 758214.

References

[1] Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E.: *Self-organization in biological systems*. Princeton University Press, 2003.

[2] Cucker, F. and Smale, S.: Emergent behavior in flocks. IEEE Trans. Automat. Control **52** (2007), 852–862.

[3] Cucker, F. and Smale, S.: On the mathematics of emergence. Japan J. Math. **2** (2007), 197–227.

[4] Davis, T. A. and Duff, I. S.: A combined unifrontal/multifrontal method for unsymmetric sparse matrices. ACM Transact. on Math. Soft. **25** (1999), 1–19.

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

[6] Feistauer, M. and Kučera, V.: On a robust discontinuous Galerkin technique for the solution of compressible flow. J. Comput. Phys., **224** (2007), 208–221.

[7] Fornasier, M., Haškovec J., and Toscani, G.: Fluid dynamic description of flocking via Povzner-Boltzmann equation. Physica D **240 (1)** (2011), 21–31.

[8] Kučera, V. and Živčáková, A.: Numerical solution of a new hydrodynamic model of flocking dynamics. In: *Programs and Algorithms of Numerical Mathematics 17*, pp. 124–129. Institute of Mathematics, CAS, Prague, 2015.

[9] Muntean, A. and Toschi, F. (Eds.): *Collective dynamics from bacteria to crowds*, Springer, 2014.

[10] Naldi, G., Pareschi, L., and Toscani, G. (Eds.): *Mathematical modeling of collective behavior in socio-economic and life sciences*, Birkhauser, Boston, 2010.

[11] Vijayasundaram, G.: Transonic flow simulation using upstream centered scheme of Godunov type in finite elements. J. Comput. Phys. **63** (1986), 416–433.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

ON THE OPTIMIZATION OF INITIAL CONDITIONS FOR A MODEL PARAMETER ESTIMATION

Ctirad Matonoha[1], Štěpán Papáček[2], Stefan Kindermann[3]

[1] Institute of Computer Science, The Czech Academy of Sciences
Pod Vodárenskou věží 271/2, 182 07 Prague 8, Czech Republic
matonoha@cs.cas.cz

[2] Institute of Complex Systems, University of South Bohemia in České Budějovice,
FFPW USB, CENAKVA
Zámek 136, 373 33 Nové Hrady, Czech Republic
spapacek@frov.jcu.cz

[3] Industrial Mathematics Institute, University of Linz
Altenbergerstr. 69, 4040 Linz, Austria
kindermann@indmath.uni-linz.ac.at

Abstract: The design of an experiment, e.g., the setting of initial conditions, strongly influences the accuracy of the process of determining model parameters from data. The key concept relies on the analysis of the sensitivity of the measured output with respect to the model parameters. Based on this approach we optimize an experimental design factor, the initial condition for an inverse problem of a model parameter estimation. Our approach, although case independent, is illustrated at the FRAP (Fluorescence Recovery After Photobleaching) experimental technique. The core idea resides in the maximization of a sensitivity measure, which depends on the initial condition. Numerical experiments show that the discretized optimal initial condition attains only two values. The number of jumps between these values is inversely proportional to the value of a diffusion coefficient D (characterizing the biophysical and numerical process). The smaller value of D is, the larger number of jumps occurs.

Keywords: FRAP, sensitivity analysis, optimal experimental design, parameter estimation, finite differences

MSC: 65M32, 65N21, 49Q10, 49Q12, 65D25

1. Introduction

Image processing is one of the fastest growing areas in informatics and applied mathematics. However, it is not a rare case that a large amount of data, e.g., spatio-temporal FRAP (Fluorescence Recovery After Photobleaching) images, is routinely

DOI: 10.21136/panm.2016.09

generated without a clear idea about further data processing. The FRAP technique is based on measuring the fluorescence intensity (proportional to non-bleached particles concentration) in a region of interest (being usually an Euclidian 2D domain) in response to a high-intensity laser pulse. The laser pulse (the so-called *bleach*) causes an irreversible loss in fluorescence of some particles residing originally in the bleached area, without any damage to intracellular structures. After the bleach, we observe the change in fluorescence intensity in a monitored region reflecting the diffusive transport of fluorescent particles from the area outside the bleach [9].

The aim of this paper is to establish the link between experimental conditions (protocol) and the accuracy of the results. The idea is presented in a simplified case study of FRAP data processing [8], [3]. It serves as a paradigmatic example of the inverse problem of the diffusion parameter estimation from spatio-temporal measurements of fluorescent particle concentration. A natural question is how the experimental settings influence the accuracy of resulting parameter estimates. There are many rather empirical recommendations related to the design of a photobleaching experiment, e.g., the bleach spot shape and size, the region of interest (its location and size), or the total time of measurement, see [9]. However, we should have a more rigorous tool for the choice of experimental design factors. This goal can be achieved through a reliable process model, i.e., the Fickian diffusion equation, and through performing the subsequent sensitivity analysis with respect to the model parameters. Thus, we can define an optimization problem as the maximization of the sensitivity measure described in Section 2. The special focus of this paper concerns the search for the optimal initial condition that in its discretized form represents the bleaching pattern [2], [5].

The paper is organized as follows. In Section 2, we define the sensitivity measure and formulate the optimization problem. Section 3 describes a numerical approach to reach the optimal initial condition. In Section 4, we provide a numerical example to show that the features of the optimal initial condition strongly depend on the diffusion coefficient. Finally, some conclusions are presented in Section 5.

2. Problem formulation

We consider the Fickian diffusion problem with a *constant* diffusion coefficient $D > 0$ and assume a spatially radially symmetric observation domain, i.e., the data are observed on a cylinder with the radius R and height T. In FRAP, the simplest governing equation for the spatio-temporal distribution of fluorescent particle concentration $u(r,t)$ is the diffusion equation as follows[1]

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r}\right), \tag{1}$$

[1]We consider the diffusion equation in polar coordinates since both the whole boundary value problem and the bleaching pattern used in the FRAP experiment have the rotational (axial) symmetry. In our preceding papers [8], [4], we employed the Cartesian coordinate system.

where $r \in (0, R],\ t \in [0, T]$, with the initial and Neumann boundary conditions

$$u(r, 0) = u_0(r), \quad \frac{\partial u}{\partial r}(R, t) = 0. \tag{2}$$

The main issue in FRAP and related estimation problems is to find the value of the diffusion coefficient D from spatio-temporal measurements of the concentration $u(r, t)$, see [7], [8].

Obviously, the measured data are discrete and each data entry quantifies the variable u at a particular spatio-temporal point (r, t) in a finite domain, i.e.,

$$u(r_i, t_j), \quad i = 0 \dots n, \quad j = 0 \dots m,$$

where i is the spatial index uniquely identifying the pixel position where the value of fluorescence intensity u is measured and j is the time index (the initial condition corresponds to $j = 0$). Usually, the data points are uniformly distributed both in time (the time interval Δt between two consecutive measurements is constant) and space, i.e., on an equidistant mesh with the step-size Δr, see [4].

Given the data as above, the diffusion coefficient D can be computed numerically by solving the inverse problem to (1)–(2). Because of unavoidable noise in the data, one obtains an estimated value $\overline{D}$ which reasonably well approximates the true D. It can be shown [1], [4], that for our case of single scalar parameter estimation and white noise as data error assumed, the expected relative error in D depends on the data noise and a factor, which we call the global semi-relative squared sensitivity S_{GRS}, as follows

$$\mathbb{E}\left(\left|\frac{\overline{D} - D}{D}\right|^2\right) \sim \frac{\sigma^2}{S_{GRS}}, \tag{3}$$

where $\mathbb{E}$ is the expected value and σ^2 denotes the variance of the additive Gaussian noise. The sensitivity measure S_{GRS}, that depends on the initial condition, is defined on a spatio-temporal mesh by

$$S_{GRS} = D^2 \sum_{i=0}^{n} \sum_{j=1}^{m} \left[\frac{\partial}{\partial D} u(r_i, t_j)\right]^2, \tag{4}$$

where $\frac{\partial}{\partial D} u(r_i, t_j)$ is the usual sensitivity of the model output at the spatio-temporal point (r_i, t_j) with respect to the parameter D. It is obvious from this estimate that if the noise level is fixed, the estimation of D can only be improved by switching to an experimental design with a higher sensitivity.

The sensitivity measure (4) involves several design parameters. If all the above parameters $R, T, \Delta r, \Delta t$ are fixed, there is only one way to maximize the sensitivity measure S_{GRS}: to consider the *initial condition* u_0 in (2) as the experimental design parameter. In the discretized problem, the aim is to find the initial condition $(u_0(r_0), \dots, u_0(r_n))^T \in \mathcal{R}^{n+1}$ such that S_{GRS} is maximized and hence the expected

error in D is minimized. In order to do so, we establish the bounds where the initial condition is considered: $\underline{u_0} \le u_0 \le \overline{u_0}$, where $\underline{u_0}, \overline{u_0} \in \mathcal{R},\ \underline{u_0} < \overline{u_0}$. The optimization problem can be formulated as follows

$$u_0^{opt} = \arg \max_{u_0 \in \mathcal{R}^{n+1}} S_{GRS}(u_0) \quad \text{subject to} \quad \underline{u_0} \le u_0 \le \overline{u_0}, \tag{5}$$

where, for brevity, u_0 is interpreted as a vector.

3. Optimization of the initial condition

Based on the parameters R, T, it is convenient to introduce the following scaling of the space and time coordinates and to define a scaled diffusion coefficient δ

$$\tilde{r} := \frac{r}{R}, \quad \tilde{t} := \frac{t}{T}, \quad \delta := \frac{DT}{R^2}. \tag{6}$$

The concentration u in the scaled coordinates $\tilde{r}, \tilde{t}$ then satisfies the equation

$$\frac{\partial u}{\partial \tilde{t}} = \delta \left(\frac{\partial^2 u}{\partial \tilde{r}^2} + \frac{1}{\tilde{r}} \frac{\partial u}{\partial \tilde{r}} \right), \tag{7}$$

where $\tilde{r} \in [0,1],\ \tilde{t} \in [0,1]$, with initial and Neumann boundary conditions

$$u(\tilde{r}, 0) = u_0(\tilde{r}), \quad \frac{\partial u}{\partial \tilde{r}}(1, \tilde{t}) = 0. \tag{8}$$

Let us fix $n+1$ as a number of spatial points and m as a number of time measurements. Consider a spatio-temporal grid $\{\tilde{r}_i, \tilde{t}_j\},\ i = 0 \dots n,\ j = 0 \dots m$, where $\tilde{r}_0 = 0,\ \tilde{r}_n = 1,\ \tilde{t}_0 = 0,\ \tilde{t}_m = 1$, with corresponding spatial and time steps $\Delta\tilde{r} = \frac{1}{n}$ and $\Delta\tilde{t} = \frac{1}{m}$, respectively. Consequently, $u(\tilde{r}_i, 0) = u_0(\tilde{r}_i),\ i = 0 \dots n$, represent the initial condition (evaluated at discrete points $\tilde{r}_i$) and $\frac{\partial u}{\partial \tilde{r}}(1, \tilde{t}_j) = 0,\ j = 1 \dots m$, represent the Neumann boundary condition.

We will use a finite difference Crank-Nicolson scheme to compute a numerical solution $u_{i,j} := u(\tilde{r}_i, \tilde{t}_j),\ i = 0 \dots n-1,\ j = 1 \dots m$, of the initial boundary value problem (7)–(8). After some algebraic manipulation [10] we arrive at a linear system

$$A u_{\cdot,j} = g \tag{9}$$

for $(u_{0,j}, \dots, u_{n-1,j})^T$ with a three-diagonal symmetric positive definite matrix

$$A = \begin{bmatrix} \frac{1}{4}\gamma^+ & -hs_0 & & & & \\ -hs_0 & \gamma^+ & -hs_1 & & & \\ & -hs_1 & 2\gamma^+ & -hs_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -hs_{n-3} & (n-2)\gamma^+ & -hs_{n-2} \\ & & & & -hs_{n-2} & (n-1)\gamma^+ - hs_{n-1} \end{bmatrix}$$

and a right-hand side

$$\begin{aligned} g_0 &= \tfrac{1}{4}\gamma^- u_{0,j-1} + hs_0 u_{1,j-1}, \\ g_i &= hs_{i-1}u_{i-1,j-1} + i\,\gamma^- u_{i,j-1} + hs_i u_{i+1,j-1}, \quad i = 1,\dots,n-2, \\ g_{n-1} &= hs_{n-2}u_{n-2,j-1} + ((n-1)\gamma^- + hs_{n-1})u_{n-1,j-1}. \end{aligned}$$

The Neumann boundary condition implies that $u_{n,j} = u_{n-1,j}$. Here

$$h = \frac{\Delta\tilde{t}}{\Delta\tilde{r}}, \quad \gamma^+ = \frac{\Delta\tilde{r}}{\delta} + h, \quad \gamma^- = \frac{\Delta\tilde{r}}{\delta} - h, \quad s_k = \frac{2k+1}{4},\ k = 0,\dots,n-1.$$

The formula (4) for S_{GRS} involves the derivative of the solution $u(r,t)$ of (1)–(2) with respect to the diffusion parameter D. Taking the scaled variables (6) and using the derivative of a composite function, we find that

$$D\frac{\partial u}{\partial D} = D\frac{\partial u}{\partial \delta}\frac{\partial \delta}{\partial D} = \frac{DT}{R^2}\frac{\partial u}{\partial \delta} = \delta\frac{\partial u}{\partial \delta} = \delta\frac{\partial u}{\partial \tilde{t}}\frac{\partial \tilde{t}}{\partial \delta} = -\frac{Dt}{\delta R^2}\frac{\partial u}{\partial \tilde{t}} = -\tilde{t}\frac{\partial u}{\partial \tilde{t}}. \tag{10}$$

Thus the scaled sensitivity measure (4) has the form

$$S_{GRS} = \delta^2 \sum_{i=0}^{n}\sum_{j=1}^{m}\left[\frac{\partial}{\partial\delta}u(\tilde{r}_i,\tilde{t}_j)\right]^2 = \sum_{i=0}^{n}\sum_{j=1}^{m}\left[\tilde{t}_j\,\frac{\partial}{\partial\tilde{t}}u(\tilde{r}_i,\tilde{t}_j)\right]^2. \tag{11}$$

Replacing the derivative with a finite difference, and using the fact that $\tilde{t}_j = j\Delta\tilde{t}$, the sensitivity measure S_{GRS} can be approximated as follows

$$\begin{aligned} S_{GRS} &\approx \sum_{i=0}^{n}\sum_{j=1}^{m}\left[j\Delta\tilde{t}\,\frac{u(\tilde{r}_i,\tilde{t}_j) - u(\tilde{r}_i,\tilde{t}_{j-1})}{\Delta\tilde{t}}\right]^2 \\ &= \sum_{j=1}^{m} j^2 \sum_{i=0}^{n}\left[u_{i,j} - u_{i,j-1}\right]^2 =: S_{app}(u_0(\tilde{r})). \end{aligned} \tag{12}$$

The values $u_{i,j}$ are computed from $u_{i,j-1}$ using (9), thus no extra work is necessary. The problem (5) of finding the optimal initial condition u_0 maximizing the sensitivity measure S_{app} (the approximation of S_{GRS}) can be formulated algorithmically as follows.

Algorithm 1. *Data: a cylinder radius R, height T, number of spatial points $n+1$, time measurements m, and a diffusion coefficient D.*

1. *Perform the scaling of variables (6) to obtain the value δ.*
2. *Let an initial condition $u_0 \in \mathcal{R}^{n+1}$, $u_0 \in [\underline{u_0}, \overline{u_0}]$ be given.*
3. *Compute $u_{i,j}$, $i = 0\dots n$, by solving the linear system (9) for $j = 1\dots m$.*
4. *Compute the value S_{app} using (12).*
5. *Repeat steps 2-4 to find u_0 such that S_{app} is maximal.*

4. Numerical example

As an example to demonstrate the optimal configurations of the initial condition let us choose parameters

$$R = 1, \quad T = 1, \quad n = 30, \quad m = 200, \quad \underline{u_0} = 0, \quad \overline{u_0} = 1$$

and use Algorithm 1 to find such an initial condition $(u_0(r_0), \ldots, u_0(r_n))^T \in \mathcal{R}^{n+1}$ that maximizes S_{app} (12) for $1/\delta = 5, 10, \ldots, 225$ (notice the inverse values of δ).[2]

To solve the optimization problem (5), we used a global optimization method from the UFO system [6]. This method uses local optimization methods for finding local minima. Briefly speaking, we choose an initial $u_0^{(0)} = (1/2, \ldots, 1/2)^T$ and for $k = 0, 1, \ldots$, until the optimality conditions are satisfied, we update the next iterate $u_0^{(k+1)}$ from u_0^k based on the function value $S_{app}(u_0^{(k)})$ and its gradient.

Figure 1 shows the results. For each $1/\delta$ we obtained a solution on the boundary of the feasible region. Thus, $u_0^{opt}(r_i) \in \{1, 0\}$ is a binary-valued vector (there exist non-zero components of u_0^{opt}). As the components of u_0^{opt} attain only two values 1 and 0, the vertical lines indicate the non-zero components of u_0^{opt}. A small number of jumps between 1 and 0 in u_0^{opt} occurs for large values of δ. When δ decreases ($1/\delta$ increases), the number of jumps increases.

Figure 2 shows the time evolution of the solution $u_{i,j}$ computed using (9) with the optimal initial condition u_0^{opt} in case of $\delta = 1/20$, i.e., computed vectors $u(r_i, t_j)$ for $j = 10, 20, \ldots, 200$ with $u(r_i, t_0) = u_0^{opt}(r_i),\ i = 0, \ldots, n$. For increasing time index $j \to \infty$ the solution tends to a steady-state solution.

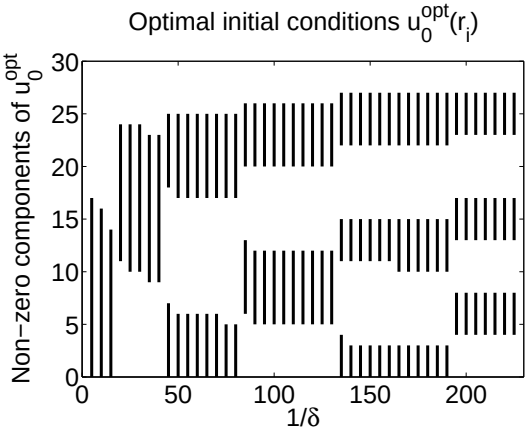


Figure 1: The result of optimization problem (5): vertical lines indicate the non-zero components of u_0^{opt}.

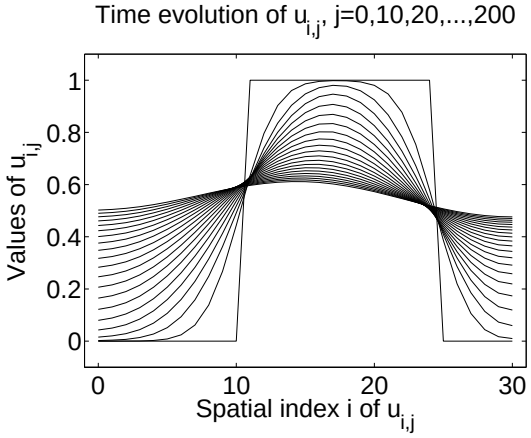


Figure 2: Optimal u_0^{opt} for $\delta = 0.05$ and the time evolution of the solution $u_{i,j}$ computed using (9).

[2]The corresponding original diffusion coefficient is $D = \delta$, see (6).

The following table shows the output from the UFO system for $\delta = 1/20$, where `F` denotes $-S_{app}$ and `X` denotes $(u_0^{opt}(r_0), \ldots, u_0^{opt}(r_n))^T$. The non-zero components of the solution have indices $i = 11, \ldots, 24$, i.e., two jumps between $\underline{u_0} = 0$ and $\overline{u_0} = 1$ occur (see Figure 1).

```
EXTREM   1 :
F = -0.2537705605D-01
X =  0.0000000000D+00  0.0000000000D+00  0.0000000000D+00  0.0000000000D+00
     0.0000000000D+00  0.0000000000D+00 -0.1734723476D-17  0.0000000000D+00
     0.0000000000D+00  0.0000000000D+00  0.0000000000D+00  0.1000000000D+01
     0.1000000000D+01  0.1000000000D+01  0.1000000000D+01  0.1000000000D+01
     0.1000000000D+01  0.1000000000D+01  0.1000000000D+01  0.1000000000D+01
     0.1000000000D+01  0.1000000000D+01  0.1000000000D+01  0.1000000000D+01
     0.1000000000D+01  0.0000000000D+00  0.0000000000D+00  0.0000000000D+00
     0.0000000000D+00  0.0000000000D+00  0.0000000000D+00
```

5. Conclusion

In this study, the problem of the optimal initial condition for further identification of a constant diffusion coefficient was formulated. We set a sensitivity measure S_{GRS} as the optimality criterion to be maximized in order to have the expected error minimal, see (4). Afterwards, we used the finite difference scheme to discretize both the scaled initial boundary value problem (7)–(8) and the sensitivity measure S_{app} eqrefsgrs. Our numerical results indicate that there exists specific optimal initial condition u_0^{opt} that maximizes the sensitivity measure S_{app} and therefore minimizes the error in the model parameter estimate (diffusion coefficient D), see (3). In discrete points $r_0, \ldots, r_n$, the components of the vector u_0^{opt} attain only two values ($\underline{u_0}$ and $\overline{u_0}$) and the number of jumps between these values depends on the diffusion coefficient D. The smaller value of D is (i.e., the slower the particle mobility due to the diffusion process is), the larger number of jumps occurs. These jumps in fact represent the discontinuities leading to more complex bleaching patterns, see [5] for more details.

Acknowledgements

This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807), by the OeAD within the programme "Aktion Oesterreich-Tschechien (AOeCZ-Universitaetslehrerstipendien)", and by the Ministry of Education, Youth and Sport of the Czech Republic projects CENAKVA (No. CZ.1.05/2.1.00/01.0024) and CENAKVA II (No. LO1205 under the NPU I program).

References

[1] Bates, D. M. and Watts, D.G.: *Nonlinear regression analysis: Its applications.* John Wiley & Sons, New York, 1988.

[2] Blumenthal, D., Goldstien, L., Edidin, M., and Gheber, L. A.: Universal approach to FRAP analysis of arbitrary bleaching patterns. Scientific Reports **5**, 11655 (2015); doi: 10.1038/srep 11655.

[3] Kaňa, R., Kotabová, E., Lukeš, M., Papáček, Š., Matonoha, C., Liu, L. N., Prášil, O., and Mullineaux, C. W.: Phycobilisome mobility and its role in the regulation of light harvesting in red algae. Plant Physiology **165(4)** (2014), 1618–1631.

[4] Kindermann, S. and Papáček, Š.: On data space selection and data processing for parameter identification in a reaction-diffusion model based on FRAP experiments. Abstract and Applied Analysis, **2015** (2015), Article ID 859849, 17 p., doi:10.1155/2015/859849.

[5] Kindermann, S. and Papáček, Š.: Optimization of the shape (and topology) of the initial conditions for diffusion parameter identification. Submitted.

[6] Lukšan, L., Tůma, M., Matonoha, C., Vlček, J., Ramešová, N., Šiška, M., and Hartman, J.: UFO 2014 — Interactive System for Universal Functional Optimization. TR V-1218, ICS AS CR, Prague, 2014.

[7] Mullineaux, C. W., Tobin, M. J., and Jones, G. R.: Mobility of photosynthetic complexes in thylakoid membranes. Nature **390** (1997), 421–424.

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

[9] Sbalzarini I. F.: *Analysis, Modeling and Simulation of Diffusion Processes in Cell Biology.* VDM Verlag Dr. Müller, Saarbrücken, 2009.

[10] Vitásek, E.: *Základy teorie numerických metod pro řešení diferenciálních rovnic.* Academia Praha, 1994.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

CALCULATION OF INDUSTRIAL ROBOT TRAJECTORY IN FRAME COMPOSITE PRODUCTION

Jaroslav Mlýnek[1], Tomáš Martinec[2], Michal Petrů[2]

[1] Department of Mathematics and Didactics of Mathematics, FP
jaroslav.mlynek@tul.cz
[2] Institute for Nanomaterials, Advanced Technologies and Innovations
tomas.martinec@tul.cz, michal.petru@tul.cz
Technical University of Liberec
Studentská 2, 461 17 Liberec, Czech Republic

Abstract: This article is focused on calculating the trajectory of an industrial robot in the production of composites for the automotive industry. The production technology is based on the winding of carbon fibres on a polyurethane frame. The frame is fastened to the end-effector of the robot arm (i.e. robot-end-effector, REE). The passage of the frame through the fibre processing head is determined by the REE trajectory. The position of the fibre processing head is fixed and is composed of three fibre guide wheels with coils of carbon fibres. The fibre processing head winds three layers of filaments onto the frame. The polyurethane frame is determined by the local Euclidean coordinate system E_3, which has its origin in the REE. We use a mathematical model and matrix calculus to compute the trajectory of the REE to guarantee the desired passage of the frame through the fibre processing head. The translation and rotation matrices of the local coordinate system (of the REE) are calculated with respect to the base coordinate system of the robot.

Keywords: robot trajectory, transformation matrix, Euler angle of rotation, orthogonal group

MSC: 14P10, 15A04, 15A24

1. Introduction

Composite materials are extensively used in many branches of industry. These materials successfully replace traditional materials. The technology based on the winding of carbon fibres by an industrial robot on a polyurethane frame is now widespread in the manufacturing of composites. After the winding process, the composite is thermally hardened.

The fibre processing head is fixedly placed in the workspace of the industrial robot and its coordinates are specified in the base coordinate system of the robot.

DOI: 10.21136/panm.2016.10

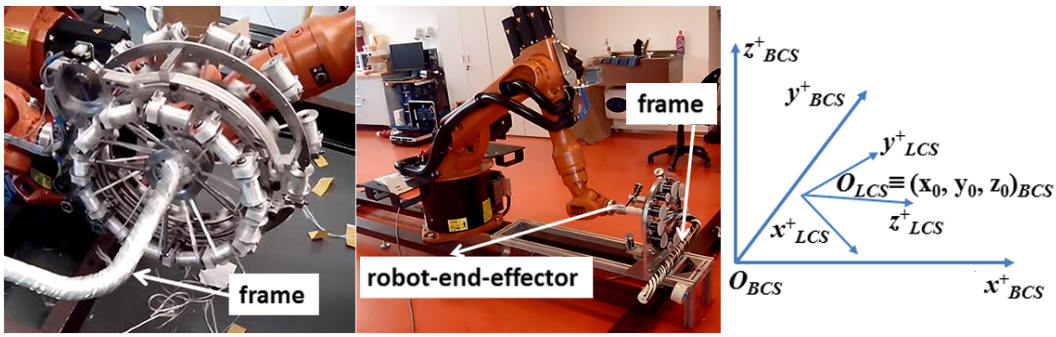


Figure 1: a) The fibre processing head with the three guide wheels for the filament winding of three layers on the frame. b) The robot with the frame attached to the REE and fibre processing head only with one fibre guide wheel. c) The base coordinate robot system and the local coordinate system of the REE.

The fibre processing head contains three fibre guide wheels, each of which includes twelve fixed coils along the periphery (see Fig. 1a). The outer fibre guide wheels rotate around their common axis, the central fibre guide wheel is static. The fibre processing head winds gradually three layers of filaments at the angles of 45°, 0° and −45° on the frame during its passage through the fibre processing head (see Fig. 1a). The frame is attached to the REE (see Fig. 1b). The passage of the frame through the fibre processing head is controlled by the movement of the REE.

Industrial robots suppliers often offer commercial software modules to control the robots. These modules are used in areas such as welding, pressing, cutting and packaging. However, the available software tools are not suitable for our purposes.

2. Mathematical model

A mathematical model of the passage of the frame through the fibre processing head developed in order to calculate the REE trajectory is described in this chapter. Using the robot base right-hand Euclidean coordinate system E_3 (BCS), we will describe the REE movements and rotations during the passage of the frame through the fibre processing head. The local right-hand Euclidean coordinate system E_3 (LCS) of the REE (see Fig. 1c) is also taken into account. To avoid any confusion, the points and vectors with the coordinates in the BCS are labelled with the subscript $_{BCS}$, while the points and vectors with the coordinates in the LCS are labelled with the subscript $_{LCS}$.

2.1. Industrial robot

The origin of the LCS is positioned in the REE. The actual position of the LCS with respect to the BCS is determined by the tool-centre-point (TCP). The robot central unit controls the movement of the REE while using the current TCP. The TCP contains six values $TCP = (x, y, z, a, b, c)$. The first three parameters

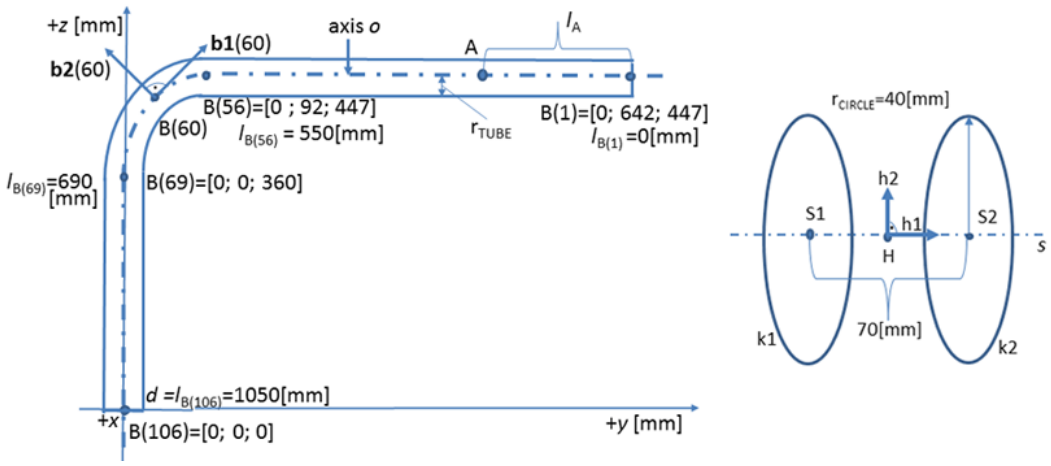


Figure 2: a) Example of vertical section through a polyurethane frame (see Fig. 1b) in LCS. The frame is connected to the REE in the point $B(106)$, $N = 106$. b) Model of the fibre processing head in BCS.

specify the coordinates of the origin of the LCS with respect to the BCS. The parameters a, b and c indicate the angles of the rotations of the LCS around the z, y and x axes with respect to the BCS.

2.2. Composite frame with a circular cross section

The polyurethane frame is described by its central axis o and its radius r_{TUBE} (see Fig. 2a). The central axis o is defined in the LCS of the REE through a discrete set of points $B(i)_{LCS}$ and the unit tangent vectors $\mathbf{b1}(i)_{LCS}$ at that points, $1 \leq i \leq N$. The initial point $B(1)_{LCS}$ and the end point $B(N)_{LCS}$ can coincide in the case of a closed frame. In addition, the unit vector $\mathbf{b2}(i)_{LCS}$ $(1 \leq i \leq N)$ lies in the plane orthogonal to the vector $\mathbf{b1}(i)_{LCS}$ and defines the upward direction at the moment when the point $B(i)_{BCS}$ passes through the fibre processing head.

The points $B(i)_{BCS}$ and vectors $\mathbf{b1}(i)_{LCS}$, $\mathbf{b2}(i)_{LCS}$ are prescribed by a composite designer to ensure an optimal passage of the frame through the centre of the fibre processing head. We assume that the discrete set of points $B(i)_{LCS}$ defines the shape of the frame with a sufficient accuracy. If one or more parts of the axis o are line segments, then it is sufficient to define only the end points of these line segments. The variable l represents the distance between the point $B(1)_{LCS}$ and a point on the axis o. The distance is measured as the o-arc length (see Fig. 2a, a general point A and distance l_A).

2.3. Fibre processing head

The position of the fibre processing head is supposed to be fixed. It consists of three fibre guide wheels which wind three layers of fibres on the frame under angles of 45°, 0° and −45°. The outside rotating wheels are indicated in Fig. 2b as circles $k1$

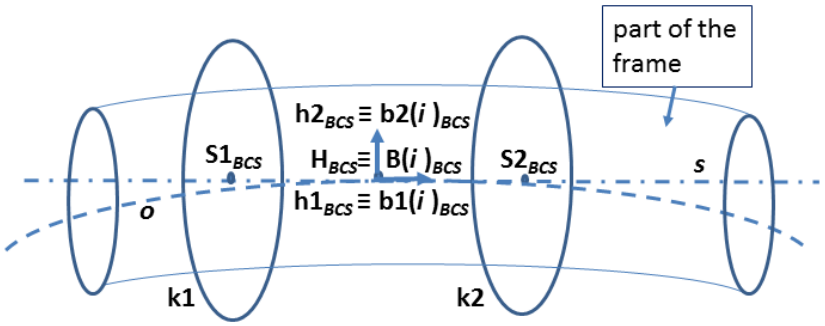


Figure 3: The scheme of the passing frame through the fibre processing head in the i-th step.

and $k2$ with centres $S1_{BCS}$ and $S2_{BCS}$, respectively. Both $k1$ and $k2$ have the same radius $r_{CIRCLE} > r_{TUBE}$ and their centres lie on the axis s of the head. The static middle fibre guide wheel is not important for our model. The best results can be achieved if the frame central axis o passes the head through its central point H_{BCS} and the tangent vector $\mathbf{b1}(i)_{BCS}$ $(1 \leq i \leq N)$ of the axis o is aligned with the axis s of the head, represented by a unit vector $\mathbf{h1}_{BCS}$. The longitudial rotation of the frame is governed by the angle of the vectors $\mathbf{h2}_{BCS} = (0, 0, 1, 0)^T$ and $\mathbf{b2}(i)_{BCS}$ $(1 \leq i \leq N)$. Thus, each point of the frame central axis o should pass through H_{BCS}. The desired orientation of the frame is then given by the vectors $\mathbf{h1}_{BCS}$ and $\mathbf{h2}_{BCS}$.

3. Calculation of the trajectory

The main idea of calculating the REE trajectory is described in this chapter. We remind that frame is fixed to the REE. The goal is to calculate the REE trajectory that ensures a gradual passage of the axis o through the centre H_{BCS} of the head in the desired direction $\mathbf{h1}_{BCS}$. The frame's initial point of passage is $B(1)_{LCS}$ and the end point is $B(N)_{LCS}$. The REE trajectory is determined by the sequence of the TCP_i values, where $1 \leq i \leq N$. The initial position of the REE corresponds to the value TCP_0.

In the admissible REE position, the two orthogonal vectors and their common initial point originally defined in the LCS are in the same position in the BCS as the two fixed orthogonal vectors and their common initial point specified in the BCS (see Fig. 3); therefore

$$H_{BCS} \equiv B(i)_{BCS},\ \mathbf{h1}_{BCS} \equiv \mathbf{b1}(i)_{BCS},\ \mathbf{h2}_{BCS} \equiv \mathbf{b2}(i)_{BCS}. \tag{1}$$

The BCS position and orientation of the REE in the i-th step of the passing of the frame through the fibre processing head are uniquely determined by the relation (1).

The central unit of the robot changes the position of the REE on the basis of the actual values of the TCP. The movement of the REE occurs in the transition from the TCP_{i-1} to the TCP_i, where the linear or cubic interpolation of the control parameters is applied.

3.1. Procedure of the TCP_i calculation

In this part we focus on the calculation of TCP_i, where $1 \leq i \leq N$. Points, vectors and matrices are represented in a homogeneous form (i.e. general point $A = (x_A, y_A, z_A, 1)^T$, vector $\mathbf{a} = (x_a, y_a, z_a, 0)^T$, this form is suitable for differentiation of operations with points and with vectors). The Euclidean norm of vector is further also used.

We calculate transformation matrix $\mathbf{T}_i$ from LCS to BCS for the i-th step of passing the frame through the fibre processing head. The transformation matrix $\mathbf{T}_i$ is generally the product of the translation matrix $\mathbf{L}_i$ and the rotation matrix $\mathbf{Q}_i$, i.e.

$$\mathbf{T}_i = \mathbf{L}_i \cdot \mathbf{Q}_i \ . \tag{2}$$

After calculating the rotation matrix $\mathbf{Q}_i$ we can decompose $\mathbf{Q}_i$ and determine the Euler angles of the LCS rotations with respect to BCS. The knowledge of the translation matrix $\mathbf{L}_i$ and the Euler angles leads to the determining of the TCP_i.

3.1.1. Determination of the rotation matrix Q_i in the relation (2)

We determine the rotation matrix Q_i ensuring the validity of the last two identifications in the relation (1).

To determine $\mathbf{Q}_i$ we suppose temporarily that the origins of the BCS and LCS are identical and that $TCP_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1}, a_{i-1}, b_{i-1}, c_{i-1})$ is specified. The matrix $\mathbf{Q}_{i-1}$ is defined by the relation (see [4], p. 31)

$$\mathbf{Q}_{i-1} = \mathbf{Rot}(z, a_{i-1}) \cdot \mathbf{Rot}(y, b_{i-1}) \cdot \mathbf{Rot}(x, c_{i-1}),$$

where $\mathbf{Rot}(z, a_{i-1})$ is the orthogonal rotation matrix of the LCS around the axis z by angle a_{i-1} and similarly for the orthogonal matrices $\mathbf{Rot}(y, b_{i-1})$ and $\mathbf{Rot}(x, c_{i-1})$. These three rotation matrices are

$$\begin{pmatrix} ca & -sa & 0 & 0 \\ sa & ca & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} cb & 0 & sb & 0 \\ 0 & 1 & 0 & 0 \\ -sb & 0 & cb & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cc & -sc & 0 \\ 0 & sc & cc & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

where sa and ca indicate $\sin a_{i-1}$, $\cos a_{i-1}$ and similarly for cb, sb, cc, sc.

Subsequently, we perform the following steps.

1) The vector $\mathbf{b1}(i)_{LCS}$ is expressed in the BCS as $\mathbf{b1}(i)_{BCS} = \mathbf{Q}_{i-1} \cdot \mathbf{b1}(i)_{LCS}$. The deviation α of the vectors $\mathbf{h1}_{BCS}$ and $\mathbf{b1}(i)_{BCS}$ is determined by using their scalar product.

2) We then calculate the cross product $\mathbf{n}_{BCS} = \mathbf{h1}_{BCS} \times \mathbf{b1}(i)_{BCS}$. The vector $\mathbf{n}_{BCS}$ is orthogonal to both the vectors $\mathbf{h1}_{BCS}$ and $\mathbf{b1}(i)_{BCS}$. The vector $\mathbf{n}_{BCS}$ is normalized, i.e. $\mathbf{n}_{BCS} = \mathbf{n}_{BCS} / \|\mathbf{n}_{BCS}\|$.

3) Now, the vector $\mathbf{b1}(i)_{BCS}$ is rotated by the angle α around the vector $\mathbf{n}_{BCS}$ to vector $\mathbf{h1}_{BCS}$ (after rotation, the vector $\mathbf{b1}(i)_{BCS}$ coincides with the vector $\mathbf{h1}_{BCS}$). If we denote the components of the unit vector $\mathbf{n}_{BCS} = (n_1, n_2, n_3, 0)^T$ then the matrix $\mathbf{Rot}(\mathbf{n}_{BCS}, \alpha)$ is of the form (see [4], p. 34)

$$\mathbf{Rot}(\mathbf{n}_{BCS}, \alpha) = \left(\begin{array}{c|c|c|c} c + n_1^2(1-c) & n_1 n_2(1-c) - n_3 s & n_1 n_3(1-c) + n_2 s & 0 \\ n_1 n_2(1-c) + n_3 s & c + n_2^2(1-c) & n_2 n_3(1-c) - n_1 s) & 0 \\ n_1 n_3(1-c) - n_2 s & n_2 n_3(1-c) + n_1 s & c + n_3^2(1-c) & 0 \\ 0 & 0 & 0 & 1 \end{array}\right), \tag{4}$$

where s and c denote $s = \sin\alpha$, $c = \cos\alpha$. Then it is true that

$$\mathbf{h1}_{BCS} \equiv \mathbf{b1}(i)_{BCS} := \mathbf{Rot}(\mathbf{n}_{BCS}, \alpha) \cdot \mathbf{b1}(i)_{BCS} = \mathbf{Rot}(\mathbf{n}_{BCS}, \alpha) \cdot \mathbf{Q}_{i-1} \cdot \mathbf{b1}(i)_{LCS}.$$

At the same time the vector $\mathbf{l}_{BCS} = \mathbf{Rot}(\mathbf{n}_{BCS}, \alpha) \cdot \mathbf{Q}_{i-1} \cdot \mathbf{b2}(i)_{LCS}$ is calculated.

4) The deviation β of the vectors $\mathbf{h2}_{BCS}$ and $\mathbf{l}_{BCS}$ is determined by using their scalar product. We define the rotation matrix $\mathbf{Rot}(\mathbf{h1}_{BCS}, \beta)$ describing the rotation of the vector $\mathbf{l}_{BCS}$ around $\mathbf{h1}_{BCS}$ to $\mathbf{h2}_{BCS}$. Then it is true

$$\mathbf{h2}_{BCS} \equiv \mathbf{b2}(i)_{BCS} := \mathbf{Rot}(\mathbf{h1}_{BCS}, \beta) \cdot \mathbf{Rot}(\mathbf{n}_{BCS}, \alpha) \cdot \mathbf{Q}_{i-1} \cdot \mathbf{b2}(i)_{LCS}.$$

The resulting rotation matrix

$$\mathbf{Q}_i = \mathbf{Rot}(\mathbf{h1}_{BCS}, \beta) \cdot \mathbf{Rot}(\mathbf{n}_{BCS}\alpha) \cdot \mathbf{Rot}(z, a_{i-1}) \cdot \mathbf{Rot}(y, b_{i-1}) \cdot \mathbf{Rot}(x, c_{i-1}), \tag{5}$$

where the elements of the matrix $\mathbf{Rot}(\mathbf{h1}_{BCS}, \beta)$ are defined analogously as the elements of the matrix $\mathbf{Rot}(\mathbf{n}_{BCS}, \alpha)$ in (4). Now, it is true that $\mathbf{h1}_{BCS} \equiv \mathbf{b1}(i)_{BCS} = \mathbf{Q}_i \cdot \mathbf{b1}(i)_{LCS}$ and $\mathbf{h2}_{BCS} \equiv \mathbf{b2}(i)_{BCS} = \mathbf{Q}_i \cdot \mathbf{b2}(i)_{LCS}$ in the relation (1).

3.1.2. Calculation of the Euler angles

Any right-hand rotation of the Euclidean space E_3 around a given unit vector $\mathbf{v}$ is determined by the orthogonal matrix $\mathbf{Q} = \mathbf{Rot}(\mathbf{v}, \vartheta)$, where ϑ is the angle of rotation. It is true that $\det(\mathbf{Q}) = 1$ and the elements of the matrix $\mathbf{Q}$ are of the form (4). These rotation matrices create the orthogonal group SO(3) (see [1]). Each rotation matrix $\mathbf{Q}$ can be written in the form (see [4], p. 32)

$$\mathbf{Q} = \mathbf{Rot}(z, a) \cdot \mathbf{Rot}(y, b) \cdot \mathbf{Rot}(x, c), \tag{6}$$

where the matrices $\mathbf{Rot}(z, a)$, $\mathbf{Rot}(y, b)$, and $\mathbf{Rot}(x, c)$ are the orthogonal matrices of rotations around the axes z, y, x and are of the form (3). Values a, b, and c are the corresponding Euler angles. We note that the Euler angles are not uniquely determined by relation (6) (see [3]).

Now, we describe the procedure to determine the Euler angles entering the matrix $\mathbf{Q}_i$ in (5). The rotation matrix $\mathbf{Q}_i$ can be decomposed in accordance with (6) in the form $\mathbf{Q}_i = \mathbf{Rot}(z, a_i) \cdot \mathbf{Rot}(y, b_i) \cdot \mathbf{Rot}(x, c_i)$. Both sides of the equation (6) written for the matrix $\mathbf{Q}_i$ are multiplied by the matrix $\mathbf{Rot}(z, a_i)^T$. The matrix $\mathbf{Rot}(z, a_i)$ is orthogonal, therefore $\mathbf{Rot}(z, a_i)^{-1} = \mathbf{Rot}(z, a_i)^T$. The equation (6) then reads

$$\mathbf{Rot}(z, a_i)^T \cdot \mathbf{Q}_i = \mathbf{Rot}(y, b_i) \cdot \mathbf{Rot}(x, c_i). \tag{7}$$

The rotation angles a_i, b_i and c_i are calculated by comparing suitably selected corresponding elements from the matrix product on the left and right side of equation (7). By writing the rotation matrix $\mathbf{Q_i}$ in the form

$$\mathbf{Q}_i = \begin{pmatrix} q_{11}(i) & q_{12}(i) & q_{13}(i) & 0 \\ q_{21}(i) & q_{22}(i) & q_{23}(i) & 0 \\ q_{31}(i) & q_{32}(i) & q_{33}(i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{8}$$

we obtain Euler angles a_i, b_i, and c_i by the following expressions

$$\begin{aligned} a_i &= \text{ATAN2}(q_{21}(i),\ q_{11}(i)), \\ b_i &= \text{ATAN2}(-q_{31}(i),\ q_{11}(i) \cdot \cos a_i + q_{21}(i) \cdot \sin a_i), \\ c_i &= \text{ATAN2}(q_{13}(i) \cdot \sin a_i - q_{23}(i) \cdot \cos a_i,\ q_{22}(i) \cdot \cos a_i - q_{12}(i) \cdot \sin a_i). \end{aligned} \tag{9}$$

The ATAN2$(arg1, arg2)$ function (common in many programming languages) calculates the value of the arctangent function for the argument $arg1/arg2$. The signs of both input parameters are involved in the determining of the output angle of the ATAN2 function ($-\pi <$ ATAN2$(arg1, arg2) \leq \pi$).

3.1.3. Determination of the translation matrix $\mathbf{L}_i$ in the relation (2)

In general, the origin of the BCS and the origin of the LCS have different positions. We have to translate the LCS relative to the BCS so that $B(i)_{BCS} \equiv H_{BCS}$. We determine the translation vector $\mathbf{u}(i)_{BCS}$ as follows

$$\mathbf{u}(i)_{BCS} := H_{BCS} - \mathbf{Q}(i) \cdot B(i)_{LCS} - (x_{i-1}, y_{i-1}, z_{i-1}, 0)^T, \tag{10}$$

where $\mathbf{Q}(i)$ is given by (5) and x_{i-1}, y_{i-1}, and z_{i-1} are the first three parameters of TCP_{i-1}. Then, see (2),

$$\mathbf{L}_i = \begin{pmatrix} 1 & 0 & 0 & x_{\mathbf{u}(i)} \\ 0 & 1 & 0 & y_{\mathbf{u}(i)} \\ 0 & 0 & 1 & z_{\mathbf{u}(i)} \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where $x_{\mathbf{u}(i)}$, $y_{\mathbf{u}(i)}$, and $z_{\mathbf{u}(i)}$ are the components of the vector $\mathbf{u}(i)_{BCS}$, see (10).

The REE is in the admissible position (1) after the transformation (2) of the LCS of the REE.

The described procedure allows to determine $TCP_i = (x_{\mathbf{u}(i)}, y_{\mathbf{u}(i)}, z_{\mathbf{u}(i)}, a(i), b(i), c(i))$ for $1 \leq i \leq N$ and thereby the whole REE trajectory when the frame passes through the fibre processing head.

4. Conclusion

The algorithm described in Chapter 3 calculates the 3D trajectory of the REE of an industrial robot during the production of composites using fibre winding technology. The described algorithm allows to determine the exact trajectory of the REE, which provides a significant advantage over the users of the extended teach-in principle (technician searches for a suitable trajectory using the robot control panel — teach pendant). Also, the possibility to accurately determine the desired trajectory of the REE by the presented algorithm can be beneficial for optimizing the REE trajectory. The application of the algorithm is completely independent of the type of robot and software tools.

The use of the procedure for determining the trajectory of the REE does not increase production costs and can significantly speed up the determination of the robot trajectory.

The practical results of the robot trajectory calculation during the winding process of composite production are described in [2].

Acknowledgements

This work was supported by grant No. TF02000051 of the Institute for Nanomaterials, Advanced Technologies and Innovations, Technical University of Liberec.

References

[1] Baker, A.: *Matrix Groups: An introduction to lie group theory.* Springer-Verlag, London, 2002.

[2] Martinec, T., Mlýnek, J., and Petrů, M.: Calculation of the robot trajectory for the optimum directional orientation of fibre placement in the manufacture of composite profile frames. In: Robotics and Computer-Integrated Manufacturing, vol. 35 (2015), 42–54.

[3] Salabough, G. G.: Computing Euler angles from a rotation matrix. `www.soi.city.ac.uk/ sbbh635/publications/euler.pdf` (1999).

[4] Sciavicco, L. and Siciliano, B.: *Modelling and control of robot manipulators.* Springer-Verlag, London, 2004.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

COMPARING NUMERICAL INTEGRATION SCHEMES FOR A CAR-FOLLOWING MODEL WITH REAL-WORLD DATA

Jan Přikryl, Miroslav Vaniš

Czech Technical University in Prague, Faculty of Transportation Sciences
Konviktská 16, CZ-110 00 Praha, Czech Republic
prikryl@fd.cvut.cz, vanismir@fd.cvut.cz

Abstract: A key element of microscopic traffic flow simulation is the so-called car-following model, describing the way in which a typical driver interacts with other vehicles on the road. This model is typically continuous and traffic micro-simulator updates its vehicle positions by a numerical integration scheme. While increasing the order of the scheme should lead to more accurate results, most micro-simulators employ the simplest Euler rule. In our contribution, inspired by [1], we will provide some additional details that have to be addressed when implementing higher-order numerical integration schemes for CFMs and we will show that the theoretical gain of higher-order methods is unfortunately masked out by the stochastic nature of real-world traffic flow.

Keywords: numerical integration, Runge-Kutta, Euler, trapezoid, ballistic update, car-following model, intelligent driver model, traffic flow

MSC: 65L06, 65L07, 68Q17

1. Introduction

In time-continuous car-following models (CFMs) employed by some microscopic traffic flow simulators, the acceleration of individual vehicles is described by a function of the driver's characteristic behavior and the surrounding traffic. This formulation leads to a coupled set of ODEs which is identical to that of physical particles following Newtonian dynamics with the physical forces replaced by fictious "social forces". While many CFMs have been formulated directly in discrete time in the form of difference equations, or fully discretely as cellular automata, and can be therefore evaluated directly, time-continuous CFMs must be evaluated using a numerical integration scheme in all but the most trivial analytically solvable cases [3].

The interaction between different vehicles on the road is best described by a process where vehicles react primarily on the driving behaviour of their leaders, i.e. vehicles that drive in front of the modelled vehicles [5]. If formulated in continuous

DOI: 10.21136/panm.2016.11

form, this class of CFMs is often represented by a coupled system of ODEs modelling movement of a single vehicle,

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = f(\mathbf{y}, t), \tag{1}$$

where $\mathbf{y} = [x, v]^{\mathrm{T}}$ is the state vector of a vehicle, composed of position x and speed v, and the vector function f represents the specific CFM. A wide class of follow-the-leader CFMs that are used in microscopic simulators can be defined using a vehicle acceleration function a^{mic} as

$$\begin{aligned} \frac{\mathrm{d}x_i}{\mathrm{d}t} &= v_i, \\ \frac{\mathrm{d}v_i}{\mathrm{d}t} &= a^{\mathrm{mic}}(s_i, v_i, v_{i-1}), \end{aligned} \tag{2}$$

where $i = 1, \ldots, n$ is the index of actual vehicle from the fixed n number of vehicles, x_i denotes the position of the front bumper of vehicle i, v_i its speed and s_i bumper to bumper gap between the current vehicle and its leader (for the first vehicle we have $s_1 = \infty$, otherwise $s_i = x_{i-1} - x_i - \ell_{i-1}$, where ℓ_i is the length of the i-th vehicle). By convention, we assume that for the i-th vehicle, the vehicle $i - 1$ is the leader.

1.1. Intelligent driver model

In their recent article [1], Treiber and Kanagaraj compare different numerical integration schemes for *intelligent driver model* (IDM) [2]. The IDM is a time-continuous car-following model for the simulation of freeway and urban traffic, developed in part by the first author of [1]. The model assumes that drivers maintain certain minimal time and space gap from their leading vehicle, and at the same time try to keep their preferred speed and acceleration and deceleration profile. The model does not work with an explicitly given reaction time of the driver, and typically it is assumed that the driver reaction time is constant and equal to the step size of the underlying numerical integration scheme.

The IDM acceleration function is given by

$$a^{\mathrm{mic}}(s_i, v_i, v_{i-1}) = a \left(1 - \left(\frac{v_i}{v_0} \right)^{\delta} - \left(\frac{s^*(v_i, v_{i-1})}{s_i} \right)^2 \right) \tag{3}$$

with

$$s^*(v_i, v_{i-1}) = s_0 + v_i\, T + \frac{v_i(v_i - v_{i-1})}{2\sqrt{ab}}, \tag{4}$$

where the model parameters are a, b – maximum safe acceleration and deceleration, v_0 – preferred speed, δ – acceleration exponent (typically equal to 4), and s_0, T – minimum space and time gap to the leader vehicle.

2. Numerical methods

Similarly to the authors of [1] we will investigate four different integration schemes. The first three are well-known numerical methods for integrating ODEs, namely

- explicit Euler method, defined as

$$\begin{aligned} \mathbf{k}_1 &= f(\mathbf{y}, t), \\ \mathbf{y}(t+h) &= \mathbf{y} + h\mathbf{k}_1, \end{aligned} \tag{5}$$

- explicit trapezoidal rule (Heun's method)

$$\begin{aligned} \mathbf{k}_1 &= f(\mathbf{y}, t), \qquad \mathbf{k}_2 = f(\mathbf{y} + h\mathbf{k}_1, t + h), \\ \mathbf{y}(t+h) &= \mathbf{y} + \frac{h}{2}(\mathbf{k}_1 + \mathbf{k}_2), \end{aligned} \tag{6}$$

- and the standard fourth-order Runge-Kutta method (RK4)

$$\begin{aligned} \mathbf{k}_1 &= f(\mathbf{y}, t), \qquad \mathbf{k}_2 = f\left(\mathbf{y} + \frac{h}{2}\mathbf{k}_1, t + \frac{h}{2}\right), \\ \mathbf{k}_3 &= f\left(\mathbf{y} + \frac{h}{2}\mathbf{k}_2, t + \frac{h}{2}\right), \qquad \mathbf{k}_4 = f(\mathbf{y} + h\mathbf{k}_3, t + h), \\ \mathbf{y}(t+h) &= \mathbf{y} + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k_4}). \end{aligned} \tag{7}$$

Besides the three standard approaches outlined above, the authors of [1] use an alternative first-order integration scheme called *ballistic update*, which can be used only in special cases where Eq. (1) represents Newtonian dynamic acceleration equations. The rule can be interpreted as a mixed first-order, second-order update consisting of an Euler update for the speeds, and a trapezoidal update for the positions,

$$\mathbf{y}(t+h) = \begin{pmatrix} \mathbf{x}(t+h) \\ \mathbf{v}(t+h) \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} + h \begin{pmatrix} \mathbf{v} \\ a(\mathbf{x}, \mathbf{v}) \end{pmatrix} + \frac{1}{2}h^2 \begin{pmatrix} a(\mathbf{x}, \mathbf{v}) \\ \mathbf{0} \end{pmatrix}, \tag{8}$$

where the last term makes the difference between ballistic and Euler method. The acceleration is computed only once per the time step, so the order of the method stays the same as in the Euler method. The trapezoidal rule needs to calculate the acceleration two times and the fourth order Runge-Kutta four times. For CFMs, calculating the acceleration function is an essential part of their numerical complexity.

3. Comparing the numerical integration methods

In order to compare the performance of integration schemes introduced in Section 2 we need to have a reference solution of the studied problem. For all the schemes, both decreasing the time step and increasing the method order should lead to more accurate results at the price of higher computational demands. We will therefore evaluate the difference to the reference solution as a function of the computational effort that was spent at obtaining the tested solution.

3.1. Reference solution

The reference solution of a simulation involving IDM cannot be obtained analytically, except for trivial cases without vehicle interaction. We will follow the argumentation of the original paper and generate a reference solution for simulation scenarios using RK4 and time step $h_{\text{ref}} = 10^{-4}\,\text{s}$. The global discretisation error will be evaluated for speeds of a single vehicle as an average of local absolute errors,

$$\epsilon = \left\| v_i^{\text{num}} - v_i^{\text{ref}} \right\| = \frac{1}{m} \sum_{j=1}^{m} |v_i^{\text{num}}(jh) - v_i^{\text{ref}}(jh)|, \tag{9}$$

where $v_i^{\text{num}}(jh)$ is the speed of the i-th vehicle at time $t = jh$ and $v_i^{\text{ref}}(jh)$ is the reference solution for the same vehicle at the same time step.

3.2. Numerical complexity

The computational demands of a given integration scheme can be expressed in terms of *numerical complexity* as

$$C = \frac{p}{h}, \tag{10}$$

where p denotes the number of evaluations of the acceleration function for one step of the integration scheme (for Euler method and ballistic update $p = 1$, for trapezoidal rule $p = 2$, for RK4 we have $p = 4$) and h is the time step.

4. Implementation details

Reference [1] contains almost complete information needed to re-implement the original experiments of the authors. We will now briefly overview the important parts, adding one detail that has been omitted from the original paper.

4.1. Initial and boundary conditions

For the original (synthetic) simulation scenarios the following conditions hold: At time 0 s all vehicles are stopped, $v_i(0) = 0$, and their positions are $x_1(0) = 0$, $\forall i > 1 : x_i(0) = x_{i-1}(0) - \ell_{i-1} - s_i$. Furthermore, a boundary condition for the first vehicle acceleration is given – Treiber and Kanagaraj assume free-flow conditions for the first vehicle,

$$a_1(v_1, t) = a_{\text{free}}(v_1) = a^{\text{mic}}(\infty, v_1, v_1). \tag{11}$$

These conditions lead to an autonomous ODE.

For the real world scenario we have vehicles that are entering the simulation at externally prescribed time instants τ_i, and hence we have $v_i(\tau_i) = v_{0,i}$ and $x_i(\tau_i) = 0$. Special care has to be taken to keep the gap between the entered vehicle and its leader large enough.

5. Heuristics for stopping vehicles

The discussed integration schemes assume smooth f. Due to finite update times, all equations will lead to negative speeds in cases where the vehicle stops between the updates. The authors of [1] suggested the following heuristics to estimate the stopping position directly: when the computed speed after the final step of integration scheme is negative, the position of the stopped vehicle is determined by a variant of the ballistic rule instead of the originally calculated position:

$$x_i(t + \widetilde{h}) = x_i(t) - \frac{v_i^2(t)}{2a_i^{\mathrm{mic}}(t)}. \tag{12}$$

Here, $\widetilde{h}$ could be h or $h/2$ (for the RK4 method). In addition, the speed of the vehicle is reset to zero.

5.1. Position of the leader

Except for the Euler method and the ballistic update, the integration of IDM for the current vehicle requires the knowledge of the leader vehicle state $\mathbf{y}_{\mathrm{lead}}(t+h)$ (for the computation of $\mathbf{k}_2$ of trapezoidal rule and $\mathbf{k}_4$ of RK4) and, for RK4, also the leader vehicle state at the intermediate point $\mathbf{y}_{\mathrm{lead}}(t+h/2)$, which has to be used to correctly compute values of $\mathbf{k}_2$ and $\mathbf{k}_3$. While the former value can be easily obtained by performing the integration step on an ordered sequence of vehicles, thus updating the state of the leader vehicle before updating the state of its follower, the original paper does not mention how the state $\mathbf{y}_{\mathrm{lead}}(t + h/2)$ is computed.

We have tested simple linear approximation, trapezoidal rule, and different combinations of intermediate RK4 states. Our results, presented in Section 2, indicate that in [1] the state at the intermediate point is computed as

$$\mathbf{y}_{\mathrm{lead}}\left(t + \frac{h}{2}\right) = \mathbf{y}_{\mathrm{lead}}(t) + \frac{h}{2}\left(\mathbf{k}_2 + \mathbf{k}_3\right). \tag{13}$$

5.2. Parallelisation

Note that while the Euler and the ballistic updates may be executed on arbitrary vehicle regardless of the updated state of its leader, the ordering condition for trapezoidal and RK4 update effectively prevents vehicle-level parallelisation. However, vehicles within independent lanes may be still updated in parallel.

6. Results

To verify the results presented in [1] and to compare these results with real-world measured data, a simple custom micro-simulator has been created in Python programming language, using standard extension libraries NumPy, SciPy and Matplotlib. The simulator is able to simulate a single lane road equipped with vehicle counting detectors and provide floating car data (speed, position, acceleration, and gap-to--leader) for all simulated vehicles.

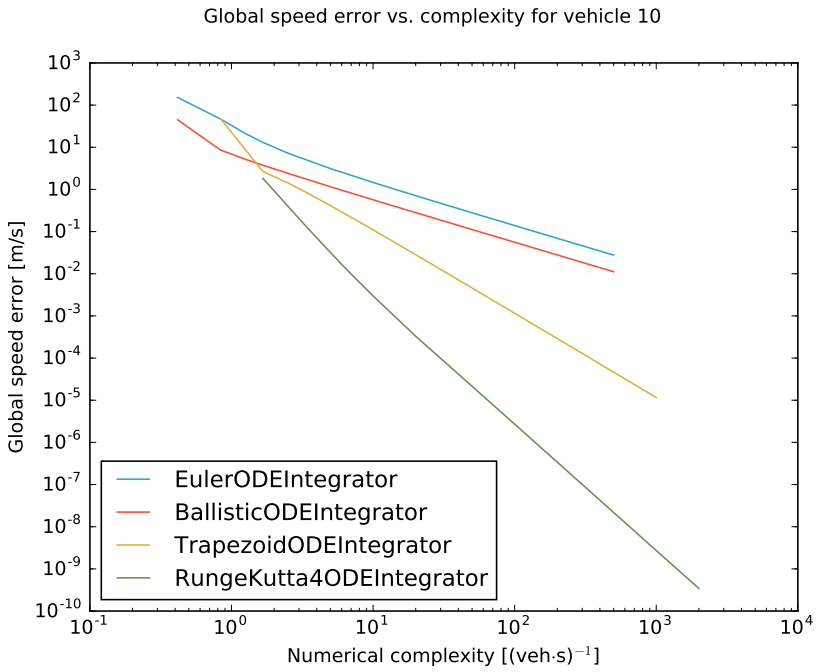


Figure 1: Global discretisation error of the 10-th vehicle speed as a function of the numerical complexity for the four update schemes. As with the original paper [1], the simulation interval has been limited to $[0, 60]$ seconds.

6.1. Original scenario

In order to verify the implemented integration schemes, we first replicated the original synthetic start-stop scenario of Treiber and Kanagaraj. We started with 20 identical vehicles queuing at the red light. At time $t = 0$, traffic light turned green and the queued cars started moving for 670 metres, where the next signalized intersection was located and the cars needed to stop again. We ran the simulation of this scenario for all four numerical integration schemes mentioned in Section 2 for 16 different integration steps ranging from 2.4 s to 0.002 s. If implemented correctly, we would expect to observe results similar to that of [1], with RK4 being the most precise of the tested methods in terms of the L_1 global error metric (9). This is indeed true in case that the intermediate point in RK4 scheme is computed using Eq. (13), see Fig. 1.

6.2. Real world data

In order to test the results on real world data, the same data set from the southern leg of the Prague Ring (SOKP) as in Ref. [4] has been used. From the database of weekday measurements, two working days have been selected. One day of detector measurements for passenger vehicles from SOKP gantry at km 20.1 has been used to calibrate IDM parameters. Then, another working day has been used as an input

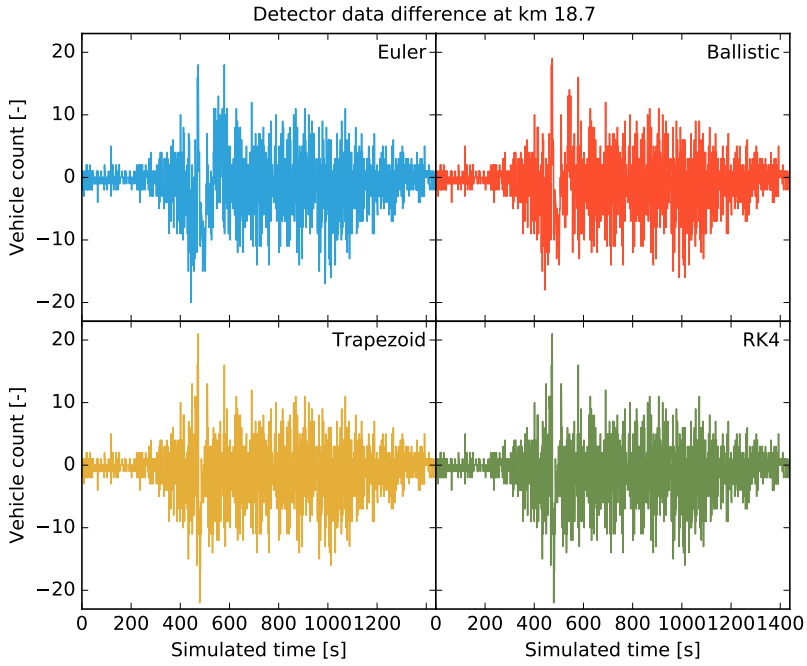


Figure 2: Global error of the vehicle count for the four update schemes with respect to real measurements.

to vehicle generator and the simulation result has been compared to data measured at km 18.7. The fixed step $h = 0.5\,\mathrm{s}$ (that roughly corresponds to average driver reaction time) has been used for all four integration schemes. As we can see in Fig. 2, results for the different integration schemes are almost identical. When we compute different error metrics, the similarity of the results becomes even more obvious – see Tab. 1.

We believe these results confirm the fact the even a well-accepted model as the IDM is only an approximation of the reality (which is to greater or lesser extent true for all mathematical models). The traffic flow is inherently a phenomenon with a very strong stochastic component, and as such it is very difficult to model – even if we would be able to replicate the properties of every vehicle with a minimal error, it is impossible to predict the behaviour of a human driving the vehicle.

7. Conclusions

Generally, when integrating ODEs, the fourth-order Runge–Kutta (RK4) method is the de-facto standard and other methods are rarely used. However, this is not the case for integrating CFMs in traffic simulations – here, Euler's method is still the most widespread one [1]. One of the reasons is given by the authors of [1]: for typical traffic-related situations, RK4 cannot reach its theoretical consistency order $p = 4$ as the smoothness conditions for the integrated function are rarely satisfied.

Method	MSE	$\max \lvert\Delta\rvert$	# hits
Euler	20.88	20	259
Ballistic	19.73	19	258
Trapezoid	20.19	22	262
RK4	20.67	21	261

Table 1: Comparison on real world data. MSE stands for mean squared error, $\max \lvert\Delta\rvert$ is the maximum difference from the reference value, and # hits column contains the count of occurrences where the simulation results were within ± 1 vehicle from the reference.

Inspired by this observation, which was made using synthetic experiments, we have used the same group of integration schemes to simulate real traffic between two measurement points on a highway. Our result show that due to the stochastic nature of traffic, the performance of all integration schemes is almost identical, suggesting that using Euler's method (or ballistic update) can be justified by its low computational demands.

The reader could correctly object that selecting only a single step size, namely $h = 0.5\,\mathrm{s}$, for comparison, may be unfair to higher order methods as their benefits would become more pronounced for larger h. Unfortunately, in CFM context, h also often reflects the average reaction time of a driver and its choice is therefore limited to values between circa 0.5 and 1 second.

The Python source code of our experiments is available from the GitHub of the fist author at http://github.com/jprk/panm18.

References

[1] Treiber, M. and Kanagaraj V.: Comparing numerical integration schemes for time-continuous car-following models. Phys. A **419** (2015), 183–195.

[2] Treiber, M., Hennecke, A., and Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E **62** (2000), 1805-–1824.

[3] Treiber, M. and Kesting, A.: *Traffic flow dynamics: data, models and simulation.* Springer, Berlin, 2013.

[4] Horňák, I. and Přikryl, J.: Experimental comparison of traffic flow models on traffic data. Programs and Algorithms of Numerical Mathematics **17** (2015), 86–91.

[5] Gazis, D. C., Herman, R., and Rothery, R. W.: Nonlinear follow-the-leader models of traffic flow. Oper. Res. **9** (1961), 545–567.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

NUMERICAL INTEGRATION IN THE TREFFTZ FINITE ELEMENT METHOD

Petra Rozehnalová

Brno University of Technology
Veveří 95, Brno, Czech Republic
rozehnalova.p@fce.vutbr.cz

Abstract: Using the high order Trefftz finite element method for solving partial differential equation requires numerical integration of oscillating functions. This integration could be performed, instead of classic techniques, also by the Levin method with some modifications. This paper shortly describes both the Trefftz method and the Levin method with its modification.

Keywords: Trefftz finite element method, Levin method

MSC: 65N30, 65D30

1. Introduction

The Trefftz finite element method is a method for solving boundary value problems applied already in the 1970. The integrals of oscillating functions often appear during the solving procedure. These integrals can be calculated using the modified Levin method. In this paper we briefly introduce the Trefftz and Levin methods with its necessary modification to be applicable to the Trefftz finite element.

2. Trefftz finite element method

Trefftz finite elements are finite elements based on the usage of the auxiliary unknown defined on the edges (faces) of elements that links together the primary unknown defined on each element. This method is described in [1] or [5]. Description of implementation aspect is in [6].

Here, let us briefly present the method on the following model problem. Consider that we are seeking to find the solution of the Laplace equation in a domain $\Omega \subset \mathbb{R}^2$ endowed with the boundary conditions

$$\begin{aligned} \Delta u &= 0 \quad \text{in} \quad \Omega, \\ u &= \bar{u} \quad \text{on} \quad \Gamma_u, \\ \frac{\partial u}{\partial n} &= \bar{q} \quad \text{on} \quad \Gamma_q, \end{aligned} \tag{1}$$

where $\bar{u}, \bar{q}$ are known functions, and n is the normal to the boundary $\Gamma = \Gamma_u \cap \Gamma_q$.

DOI: 10.21136/panm.2016.12

Let the domain Ω be divided into elements; and over each element Ω_e, we assume the function u in the form

$$u = \sum_{j=1}^{M} c_j \, N_j \quad \text{on element } \Omega_e,$$

where c_j are unknown constants and N_j are known functions to be chosen such that

$$\Delta N_j = 0 \quad \text{in element } \Omega_e, \text{ for } j = 1, 2, \ldots, M.$$

It can be shown that this equation is satisfied by any of the following functions:

$$1,\ r\cos\theta,\ r\sin\theta, \ldots,\ r^m \cos m\theta,\ r^m \sin m\theta, \ldots, \tag{2}$$

where r and θ are a pair of polar coordinates.

Let us introduce an auxiliary function $\tilde{u}$ defined on element boundary only:

$$\tilde{u} = \sum_{i=1}^{N} d_i \, \tilde{N}_i,$$

where d_i stands for nodal displacement and $\tilde{N}_i$ are standard shape functions.

Let us denote $q = \frac{\partial u}{\partial n}$ and $q_1 = \frac{\partial u}{\partial x_1}, q_2 = \frac{\partial u}{\partial x_2}$. Following the approach in [5], let us introduce the functional Ψ_e,

$$\Psi_e = \frac{1}{2}\int_{\Omega_e} q_1^2 + q_2^2 \,\mathrm{d}\Omega - \int_{\Gamma_e} q\,\tilde{u}\,\mathrm{d}\Gamma + \int_{\Gamma_{eq}} \bar{q}\,\tilde{u}\,\mathrm{d}\Gamma = \frac{1}{2}\int_{\Gamma_e} q\,u\,\mathrm{d}\Gamma - \int_{\Gamma_e} q\,\tilde{u}\,\mathrm{d}\Gamma + \int_{\Gamma_{eq}} \bar{q}\,\tilde{u}\,\mathrm{d}\Gamma, \tag{3}$$

where Γ_e is the boundary of the element Ω_e and $\Gamma_{eq} = \Gamma_e \cap \Gamma_q$. The minimalization of the variational functional Ψ_e for the all elements provides the solution of (1).

3. Levin method

The Levin method is an effective way for the numerical integration of rapidly oscillating functions. It is described in [2] for one- and two-dimensional integrals; more details, numerical examples, and error analysis are provided in [3], [4].

Let us briefly introduce this method. The integration problem is transformed into an ordinary differential equation problem to be numerically solved by for example the collocation method.

We consider integrals of the form

$$I = \int_a^b f^t \cdot w \,\mathrm{d}x, \tag{4}$$

where f is a vector of smooth and non-oscillating functions, $f(x)=(f_1(x),\ldots,f_m(x))^t$, w is a vector of oscillating functions, $w(x)=(w_1(x),w_2(x),\ldots w_m(x))^t$, and a, b are real and finite. We also assume that w satisfies

$$w'(x) = A(x)\,w(x),$$

where A is an $m\times m$ matrix of non-oscillating functions.

We would like to find the vector $p(x)=(p_1(x),p_2(x),\ldots p_m(x))^t$ such that

$$(p^t\cdot w)' = f^t\cdot w.$$

Subsequently,

$$I=\int_a^b \left(p^t\cdot w\right)'\mathrm{d}x = \int_a^b (p')^t\cdot w + p^t\cdot w'\,\mathrm{d}x = \int_a^b (p')^t\cdot w + p^t\cdot A\,w\,\mathrm{d}x = \int_a^b \left(p'+A^t\,p\right)^t\cdot w\,\mathrm{d}x,$$

Hence, the vector p should satisfy

$$p' + A^t\,p = f.$$

Then, the integral is computed as

$$\int_a^b f^t\cdot w\,\mathrm{d}x = p^t(b)\cdot w(b) - p^t(a)\cdot w(a). \tag{5}$$

Example As an example, let us compute the integral

$$\int_0^{2\pi} x^2\cos(rx)\,\mathrm{d}x = \int_0^{2\pi}\left(x^2,0\right)\cdot(\cos(rx),\sin(rx))\,\mathrm{d}x,$$

where $r\in\bigodot\mathbb{N}$. In the notation used above, $f(x)=(x^2,0)^t$ and $w(x)=(\cos(rx),\sin(rx))^t$.

Then,

$$w'(x)=\begin{bmatrix}\cos(rx)\\ \sin(rx)\end{bmatrix}' = \begin{bmatrix}0 & -r\\ r & 0\end{bmatrix}\begin{bmatrix}\cos(rx)\\ \sin(rx)\end{bmatrix} = A(x)\,w(x).$$

We are looking for the vector $p=(p_1,p_2)$ which satisfies

$$\begin{aligned} p_1' + rp_2 &= x^2,\\ p_2' - rp_1 &= 0.\end{aligned}$$

The general solutions are

$$p_1(x) = C_1\cos(rx) + C_2\sin(rx) + \frac{2x}{r^2}, \tag{6}$$

$$p_2(x) = C_1\sin(rx) - C_2\cos(rx) + \frac{x^2}{r} - \frac{2}{r^3}. \tag{7}$$

The formula (5) is valid for any solution of (6). We choose the solution for which $C_1 = C_2 = 0$. Hence,

$$\int_0^{2\pi} x^2 \cos(rx)\,\mathrm{d}x = \left[\left(\frac{2x}{r^2}, \frac{x^2}{r} - \frac{2}{r^3}\right) \cdot (\cos(rx), \sin(rx))^t\right]_0^{2\pi} = \frac{4\pi}{r^2}.$$

4. Integration in the Trefftz method

Using functions (2) in functional (3) leads to line integrals of oscillating functions.

It depends on the implementation, but it is usual that only values of the integrated function are accessible. In this content, it is obvious that success of Levin method relies on rewriting w' as $A\,w$. In this situation, namely, when only values of w are known, finding matrix A could be a problem.

The oscillating function w can be approximated by the trigonometric interpolation polynomial

$$w(x) \doteq a_0 + \sum_{i=1}^{n} \left(a_i \cos(\alpha_i x) + b_i \sin(\alpha_i x)\right),$$

where the coefficients a_0, a_i, b_i, $i = 1 \ldots N$, can be effectively computed by the discrete fast Fourier transform.

Then, the integral of the form (4) can be approximated by

$$I \doteq \int_a^b f\,a_0 + \sum_{i=1}^{n} f\,\left(a_i \cos(\alpha_i x) + b_i \sin(\alpha_i x)\right)\mathrm{d}x = a_0 \int_a^b f\,\mathrm{d}x + \int_a^b \sum_{i=1}^{n} f\,\left(a_i \cos(\alpha_i x) + b_i \sin(\alpha_i x)\right)\mathrm{d}x.$$

In this form, the integral is suitable for the Levin method.

Acknowledgements

This research has been supported by the specific university research project of Brno University of Technology, No. FAST-S-16-3385, in cooperation with the project LO1408 AdMaS UP (Advanced Materials, Structures and Technologies, Ministry of Education, Youth and Sports of the Czech Republic, National Sustainability Programme I).

References

[1] Jirousek, J. and Wróblewski, A.: T-elements: State of the art and future trends. Arch. Comput. Methods Engrg. **3** (1996), 323–434.

[2] Levin, D.: Procedures for computing one- and two-dimensional integrals of functions with rapid irrebular oscillations. Math. Comp. **38** (1982).

[3] Levin, D.: Fast integration of rapidly oscillatory functions. J. Comput. Appl. Math. **67** (1996).

[4] Levin, D.: Analysis of a collocation method for integrationg rapidly oscillatory functions. J. Comput. Appl. Math. **78** (1997).

[5] Qin, Q.: Trefftz finite element method and its applications. Applied Mechanics Reviews **58** (2005).

[6] Qin, Q. and Wang, H.: *MATLAB and C programming for Trefftz finite element methods*. CRC Press, 2008.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

MATHEMATICAL MODELLING OF ROCK BOLT REINFORCEMENT

David Runt, Jaroslav Novotný, Jan Pruška

Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
david.runt@fsv.cvut.cz, jaroslav.novotny@fsv.cvut.cz, Pruska@fsv.cvut.cz

Abstract: Rock bolts as construction elements are often used in underground civil engineering projects. This work deals with their numerical modelling. Aydan special finite elements for the description of rock bolts and hexahedral quadratic finite elements for the description of rock massif were used. A code for the computation of stiffness matrices and right hand sides of these elements was developed. The code was tested on several simple test examples and their results were compared with the analytical solution. Stresses in a rock massif in the surrounding of an excavation reinforced by rock bolts were computed. The results show that the use of rock bolts can reduce the areas of maximal mechanical stress in the vicinity of excavations.

Keywords: rock bolt reinforcement, mathematical modelling, the finite element method

MSC: 65N30, 74B05, 74G15, 74L10

1. Introduction

Rock bolts as reinforcing construction elements are often used in underground civil engineering projects (Fig. 1).

Several special finite elements for rock bolt modelling were developed. The most widely used element was presented by Aydan [1]. The so-called Aydan element consists of two groups of nodes. The first group represents a rod sub-element, which is a simple model of a steel bar. The remaining nodes are located on the interface of cement grout and rock massif. The connection of the bar with the surrounding rock by cement grout is represented by the joint action of both groups of nodes. This paper is focused on the six-node type of the Aydan element with quadratic shape functions, which is used in 3D models.

The computation of the stiffness matrix of this element and its application in the 3D model of two tunnels reinforced by rock bolts is described. The rock bolts

DOI: 10.21136/panm.2016.13

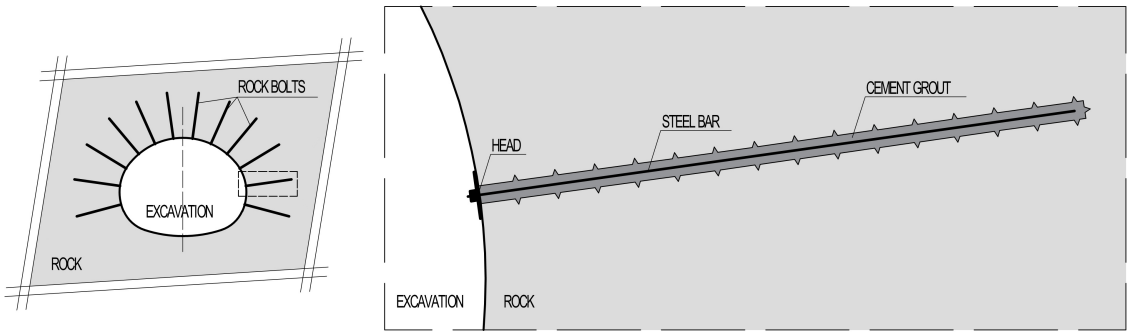


Figure 1: Rock bolts.

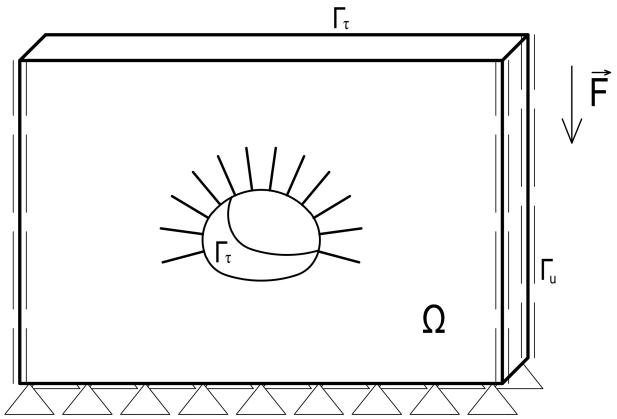


Figure 2: Classical formulation of the linear elasticity problem.

are fastened by cement grout along their full length. The geometry of the model corresponds to the characteristic cross section of the Brusnice tunnel that is a part of the Blanka tunnel complex, an underground part of the Prague City Ring Road.

2. Classical formulation of a linear elasticity problem

Differential equations describe real physical processes inside the material. The classical formulation of the linear elasticity problem is described in [3], for instance.

We consider a linear elastic body that occupies a domain Ω (Fig. 2). We look for the vector of displacements $\mathbf{u} = (u_1, u_2, u_3)$ satisfying Lamé equations in the domain Ω, see Equations (1),

$$(\lambda + \mu) \sum_{j=1}^{3} \frac{\partial^2 u_j}{\partial x_i \partial x_j} + \sum_{j=1}^{3} \frac{\partial^2 u_i}{\partial x_j^2} + F_i = 0, \quad i = 1, 2, 3 \tag{1}$$

where λ, μ are the Lamé coefficients, x_i and F_i stand for the i-th coordinate com-

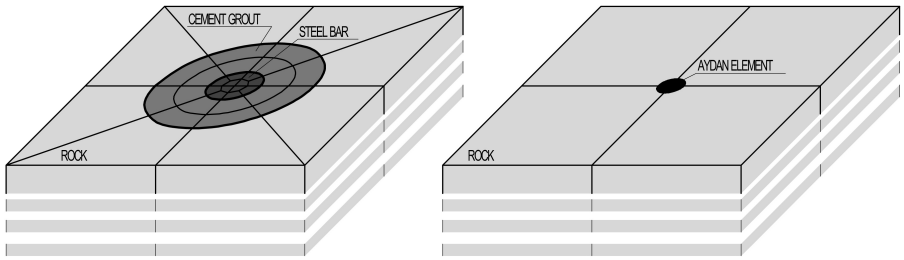


Figure 3: Finite element discretization of the rock bolt reinforcement without and with a special rock bolt element.

ponent and volumetric load component, respectively. Displacements are prescribed on Γ_u and the stress vector is given on Γ_τ (see Fig. 2), that is,

$$u_i = 0, \quad i = 1, 2, 3 \text{ on } \Gamma_u \tag{2}$$

$$\sum_{j=1}^{3} \tau_{ij}\nu_j = 0, \quad i = 1, 2, 3 \text{ on } \Gamma_\tau \tag{3}$$

where $\tau = (\tau_{ij})$ is the stress tensor and $\nu = (\nu_1, \nu_2, \nu_3)$ is the outward unit normal vector to the boundary of Ω.

We prescribed zero displacements on the bottom, on the sides and on the front and back faces of the domain Ω, see Equation (2). The zero stress vector is prescribed on the excavation surface and on the top surface of the body Ω, see Equation (3).

3. The finite element method

3.1. Rock bolt element of the Aydan type

Because a detailed discretization of the rock bolt reinforcement including the steel bar and fastening material needs a generation of a complicated finite element mesh, special finite elements were derived, see Fig. 3.

The Aydan rock bolt element with quadratic shape functions has six nodes (Fig. 4). Three of them represent the steel rod (nodes 1, 2 and 3). The others are located on the interface between the fastening material and the surrounding rock. The rock bolt element is connected to the elements, which represent rock massif, by nodes 4, 5 and 6 (Fig. 4). The connection of the bar with surrounding rock massif by cement grout is represented by the joint action of both groups of the nodes.

Several simplifications were considered during the derivation of the stiffness matrix of the Aydan element. The steel rod and body formed by the fastening material are assumed axially symmetric and coaxial bodies. Both mentioned materials are considered homogeneous, isotropic and linear elastic. Therefore the dependence between stresses and deformations of these materials is described by linear Hooke's law. The radius of the rock bolt is negligible with respect to its length. Therefore, nodes 1 and 4 have identical coordinates. The same is valid for nodes 2 and 5 or 3

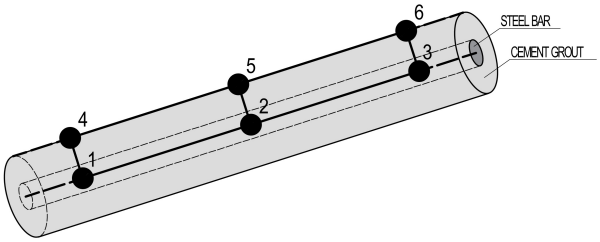


Figure 4: Aydan six-node rock bolt element.

and 6. However, the assumption of negligible radius cannot be applied to the process of derivation of the stiffness matrix. Only three types of deformations of the Aydan element are included in the computation:

- relative longitudinal deformation of the steel bar caused by different axial displacements of nodes 1, 2 and 3,
- relative cross shear deformation of the steel bar caused by different radial displacements of nodes 1, 2 and 3,
- relative longitudinal shear deformation of fastening material caused by different axial displacements of nodes 1, 2 and 3 with respect to nodes 4, 5 and 6,
- relative cross deformation of fastening material caused by different radial displacements of nodes 1, 2 and 3 with respect to nodes 4, 5 and 6,

To define the element stiffness matrix, we introduce matrices $\mathbf{D}$ and $\mathbf{B}$. In detail,

$$\mathbf{D} = \begin{bmatrix} E^t & 0 & 0 & 0 & 0 & 0 \\ 0 & G^t & 0 & 0 & 0 & 0 \\ 0 & 0 & G^t & 0 & 0 & 0 \\ 0 & 0 & 0 & G^z & 0 & 0 \\ 0 & 0 & 0 & 0 & D^z & 0 \\ 0 & 0 & 0 & 0 & 0 & D^z \end{bmatrix}.$$

Here, E^t is Young's modulus of steel, G^t and G^Z is the shear modulus of steel and fastening material, D^Z is Young's modulus of the fastening material multiplied by two. The multiplicative factor of two reflects the effects of enlacement of the cement grout by the rock massif.

Next,

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_3 & -\mathbf{B}_3 \end{bmatrix},$$

where

$$\mathbf{B}_1 = \begin{bmatrix} N_1' & 0 & 0 & N_2' & 0 & 0 & N_3' & 0 & 0 \\ 0 & N_1' & 0 & 0 & N_2' & 0 & 0 & N_3' & 0 \\ 0 & 0 & N_1' & 0 & 0 & N_2' & 0 & 0 & N_3' \end{bmatrix},$$

$$\mathbf{B}_3 = \begin{bmatrix} cN_1 & 0 & 0 & cN_2 & 0 & 0 & cN_3 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 \end{bmatrix},$$

B_2 is a zero matrix, the prime denotes the derivative with respect to ξ (see (4)), and $c = 2/((r^h + r^t)l$, where $l = \ln(r^h/r^t)$ and $r^t,\ r^h$ are the respective diameters of the steel bar and the rock bolt borehole, the latter is equivalent to the diameter of the body formed by the fastening material.

The constant c was derived in [1]. Shape functions are quadratic:

$$\begin{aligned} N_1 &= 0.5 \cdot \xi \cdot (\xi - 1), \\ N_2 &= 1 - \xi^2, \\ N_3 &= 0.5 \cdot \xi \cdot (\xi + 1), \end{aligned} \tag{4}$$

where $\xi \in [-1; 1]$ is the local coordinate. The stiffness matrix of the rock bolt referential element is

$$\mathbf{K} = \int_{-1}^{1} \mathbf{B}^T \mathbf{D} \mathbf{B} \, \mathrm{d}\xi,$$

where $\mathbf{B}^T =$ is the transpose of $\mathbf{B}$.

Constant values of the displacements across the cross section of the rock bolt are considered. Therefore, the volume integral is reduced to one-dimensional integral with the integration area of the length of the rock bolt element. Three-point Gaussian numerical integration was used for the calculation of the one-dimensional integral.

Finally, it is necessary to transform the stiffness matrix from local to global system of coordinates.

3.2. Hexahedron - rock element for 3D model

Rock massif is represented by hexahedral elements with 20 nodes [2] in 3D model. Eight nodes are located in vertices, remaining twelve are located in the centres of edges. Gaussian numerical integration of third order was used for calculating stiffness matrices and right hand sides.

4. Test example and comparison with analytical solution

An analytical expression for stresses in excavation is known only for several simple cases. For a circular excavation without rock bolts, such solution is described in [6]. For a circular excavation with rock bolts, an analytical solution is known only if an

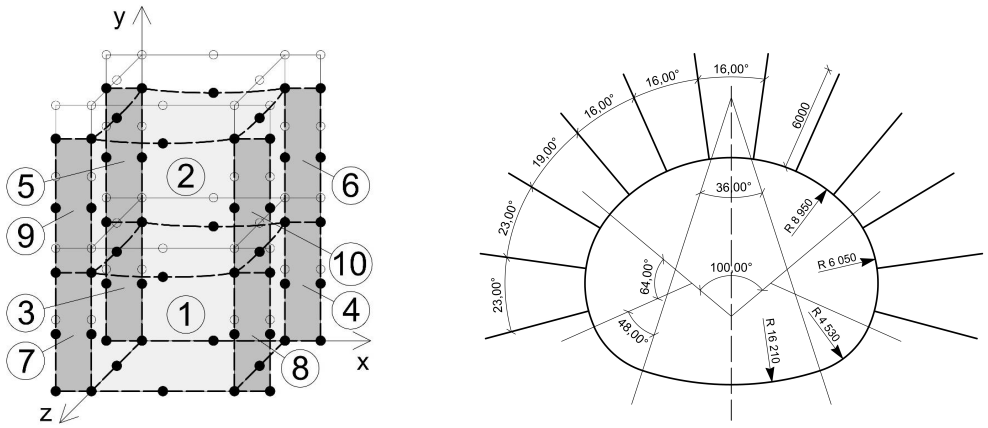


Figure 5: Test example consisting of two bricks and eight Aydan elements (left), and shape of excavation and spacing of rock bolts (right).

averaging of the rock bolt and rock material properties is introduced, see [4]. Therefore we studied several simple problems where the analytical solution can easily be derived. One of the simple problems used for the verification of our implementation of the Aydan rock bolt element is presented below, see Fig. 5.

This example consists of two bricks (elements 1 and 2) of $1 \times 2 \times 1$ m reinforced by eight Aydan elements (elements 3 - 10). Zero displacements are prescribed at the base of the body, the other displacements are restricted to the $y-$direction. The following material properties and loads were considered:

$$\begin{aligned} E_{rock} &= 500 \text{ MPa}, \ \nu_{rock} = 0.2, \\ E_{steel} &= 210 \text{ GPa}, \ \nu_{steel} = 0.3, \\ E_{grout} &= 30 \text{ GPa}, \ \nu_{grout} = 0.2, \ f_y = -25 \text{ kN} \cdot \text{m}^{-3}. \end{aligned}$$

Analytical solution was easy to compute from the equation

$$\frac{d}{dy}\left(EA\frac{du}{dy}\right) + f_y = 0,$$

where A denotes the area of the cross-section. The stiffness of the cross-section was defined by the sum of stiffnesses of the rock bolts and the rock. Results for both numerical and analytical solution are summarized in Table 1. The shape of the deformed body is also depicted in Figure 5.

5. Model of rock bolt reinforcement

The geometry of the model corresponds to the characteristic cross section of the two-tube Brusnice tunnel, which is a part of the Blanka tunnel complex. Two three lane motorways are situated inside of these two tunnels. All data necessary for the model creation were taken from [7].

Solution	Displacement of rock at y = 1 m	Displacement of rock at y = 2 m
Numerical	from -0.341×10^{-4} to -0.402×10^{-4} m	from -0.468×10^{-4} to -0.532×10^{-4} m
Analytical	-0.383×10^{-4} m	-0.511×10^{-4} m

Table 1: Comparison of numerical and analytical solution at $y = 1$ and $y = 2$ m.

5.1. Input data

Each of the two excavations is composed of four types of circular arcs, has a height of 12.8 metres, a width of 16.6 metres, and an area of almost 180.0 square metres. Twelve rock bolts six metres long are placed in the tunnel arch. The angle between two adjacent rock bolts varies from 18° to 23° (Fig. 5 right).

The distance between parallel planes containing rock bolt bundles in the direction of the axis of the tunnel is 1.25 m. All 24 rock bolts are located in the central plane of the model. Therefore, thickness of the model periodic segment is considered also 1.25 m. Each rock bolt is, in fact, a steel bar with a radius of 2.0 cm that is fastened by cement grout along its full length in the borehole with a diameter of 6.0 cm.

The width of the whole model is 150.0 metres and the height varies from 51.4 metres up to 62.6 metres. The thickness of a rock cover is 13.3 metres for the left excavation and 12.8 metres for the right one (Fig. 6 top). For this model we consider boundary conditions that were already described in the classical formulation of the problem. The rock massif is formed by mildly eroded slates, which are very common in the surrounding of the tunnel. Steel and cement grout are another materials contained in the model. It is necessary to prescribe Young's modulus, Poisson's ratio and specific density for all the materials,

$$\begin{aligned} E_{rock} &= 400 \text{ MPa}, \ \nu_{rock} = 0.28, \ \rho_{rock} = 2450 \text{ kg} \cdot \text{m}^{-3}, \\ E_{steel} &= 210 \text{ GPa}, \ \nu_{steel} = 0.3, \\ E_{grout} &= 30 \text{ GPa}, \ \nu_{grout} = 0.2. \end{aligned}$$

In the development of the finite element mesh, a circular zone around the excavation was created in order to properly couple the Aydan elements with the brick elements (Fig. 6 bottom). The mesh consists of 61,368 hexahedrons and 360 rock bolt elements and it is represented by 297,931 nodes. The mesh is composed of four layers of elements with the same thickness in the direction of the axis of the tunnel.

5.2. Results

We considered two studies. In the first case, we assumed the excavations without any rock bolts. Then the model with rock bolts was used. An influence of rock bolts is recognizable only in the close vicinity of the excavations, therefore we analyse the results only in this detailed area, especially in the surrounding of the left excavation. Stress σ_{yy} in the vertical direction and stress σ_{xx} in the horizontal direction is depicted (Fig. 7). The influence of rock bolts is most evident, if the

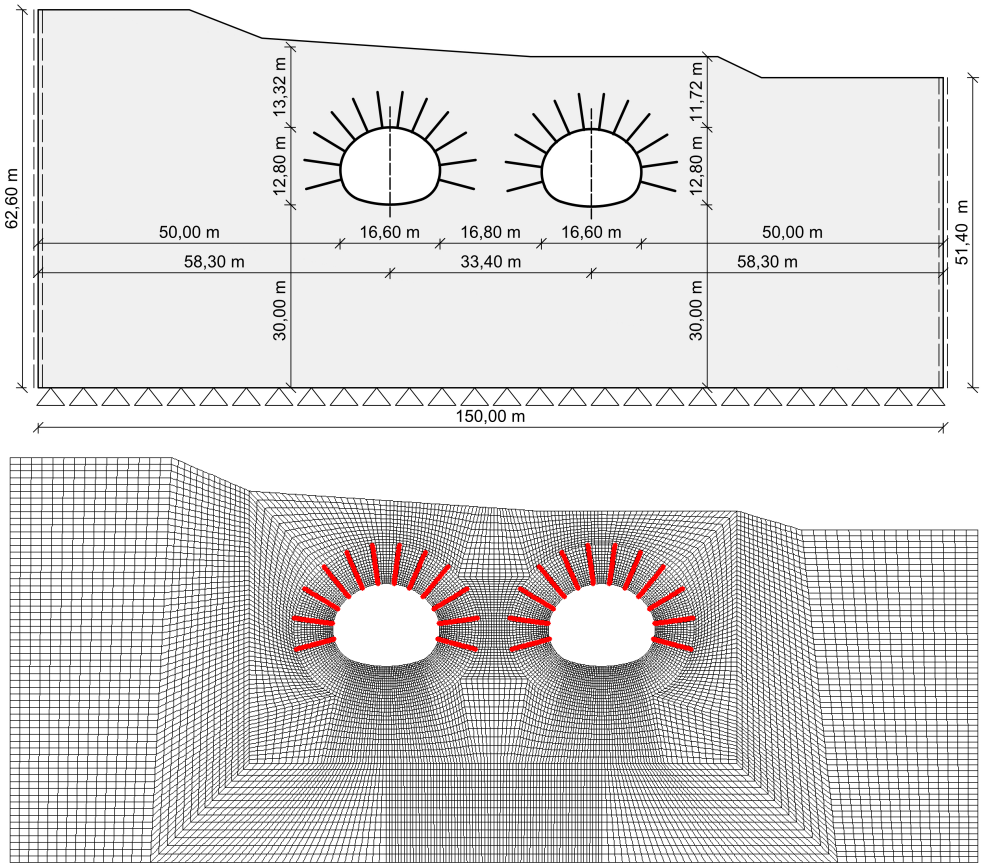


Figure 6: Geometry of the whole model (top), and finite element mesh including the position of rock bolts (bottom).

direction of their axis is similar to the direction of the considered stress. Areas with large stress are redistributed into several smaller areas, although the stress need not decrease (Fig. 8).

Rock bolts placed above the excavation have greater impact on the vertical stress and rock bolts placed on the sides have greater impact on the horizontal stress. Stresses in the area above the excavation are relatively low, so rock bolts cannot achieve full activation here. Presence of rock bolts is most evident in the horizontal stress and mainly on the sides of the excavations (Fig. 9). The location of the rock bolts is clearly apparent from the local stress anomalies.

6. Conclusion

The influence of rock bolts is recognizable from our results. Rock bolts reduce local extreme values of stresses. In general, they mildly raise stresses in the radial direction and they help to create rock arch and improve the stability of the excavation. When using rock bolts, areas with large stress are redistributed into several

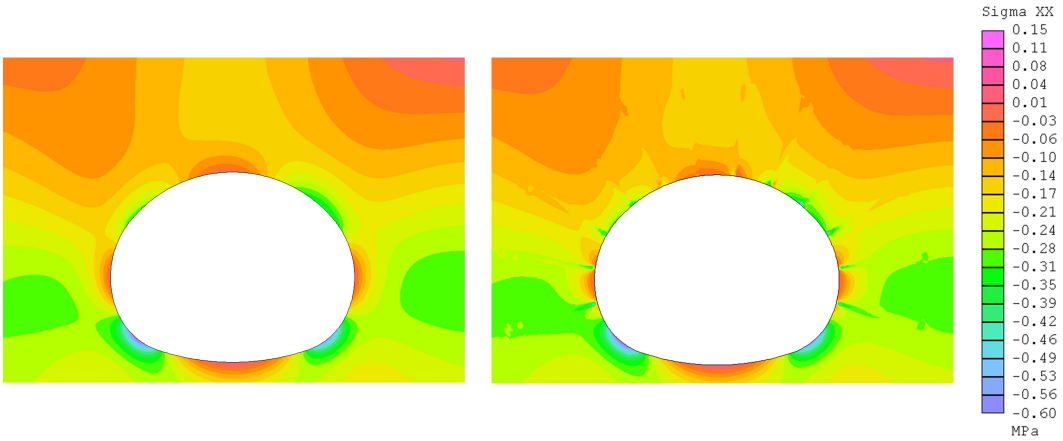


Figure 7: Stress σ_{xx} in the surrounding of the left excavation without rock bolts (left) and with rock bolts (right).

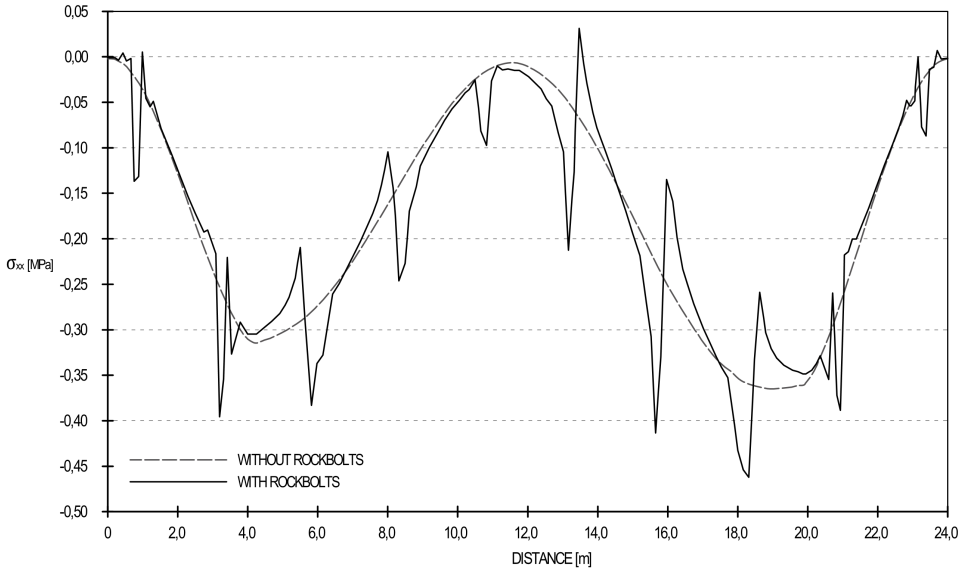


Figure 8: Stress σ_{xx} along the upper part of the excavation arc – without rock bolts (dashed line) and with rock bolts (solid line).

smaller areas, although the stress need not decrease. Described influence corresponds to theoretical knowledge of the functioning of rock bolt reinforcement. It is possible to combine special rock bolt elements with other types of elements with appropriate shape functions and this is the way how to create complex numerical models of reinforced excavations. Due to the simplicity of the rock bolt element it is quite easy to create its different modifications with various shape functions. These modifications were described by Aydan [1], Chao [5] or Runt [8]. The resulting elements can be used in both 2D and 3D models.

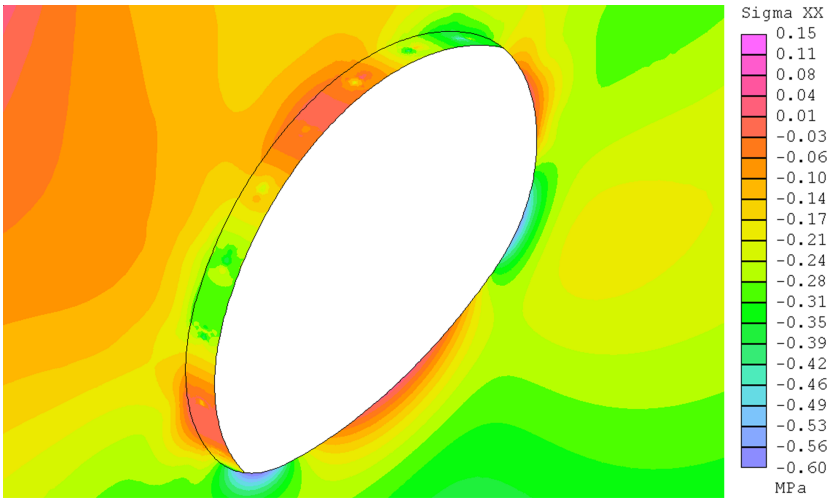


Figure 9: Stress σ_{xx} inside the excavation.

Acknowledgements

This work was supported by the grant project SGS16/004/OHK1/1T/11 provided by Czech Technical University in Prague, Faculty of Civil Engineering.

References

[1] Aydan, O.: *The stabilisation of rock engineering structures by rockbolts*. Nagoya University, Nagoya, 1989.

[2] Bathe, K. J.: *Finite element procedures*. Prentice-Hall, New Jersey, 1996.

[3] Brdička, M.: *Continuum mechanics*. Czechoslovak Academy of Sciences, Prague, 1959 (in Czech).

[4] Carranza-Torres, C.: Analytical and numerical study of the mechanics of rock-bolt reinforcement around tunnels in rock masses. Rock. Mech. Rock. Engng. **42** (2009), 175–228.

[5] Chao, T.: *The numerical modelling of rockbolts in geomechanics by finite element methods*. Ph.D. thesis, Brunel University, London, 1998.

[6] Jaeger, J. C., Cook N. G. W.: *Fundamentals of rock mechanics*. Chapman and Hall, London, 1979.

[7] Nosek, J.: *Evaluation of reliability of structural analyses of tunnels in the Czech Republic on the bases of analyses of the monitoring results*. Ph.D. thesis, Czech Technical University in Prague, Prague, 2015 (in Czech).

[8] Runt, D.: *Rock bolt element in the finite element method*. M.Sc. thesis, Czech Technical University in Prague, Prague, 2016 (in Czech).

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

A PARTICULAR SMOOTH INTERPOLATION THAT GENERATES SPLINES

Karel Segeth

Institute of Mathematics, Czech Academy of Sciences
Žitná 25, CZ-115 67 Prague 1, Czech Republic
segeth@math.cas.cz

Abstract: There are two grounds the spline theory stems from – the algebraic one (where splines are understood as piecewise smooth functions satisfying some continuity conditions) and the variational one (where splines are obtained via minimization of some quadratic functionals with constraints). We use the general variational approach called *smooth interpolation* introduced by Talmi and Gilat and show that it covers not only the cubic spline and its 2D and 3D analogues but also the well known tension spline (called also spline with tension). We present the results of a 1D numerical example that characterize some properties of the tension spline.

Keywords: data interpolation, smooth interpolation, spline interpolation, tension spline, Fourier series, Fourier transform

MSC: 65D05, 65D07, 41A05, 41A63, 42A38

1. Introduction

The minimum curvature approach to interpolation, which produces e.g. cubic splines, usually provides a visually nice smooth curve or surface. However, in some cases the method can create artificial oscillations. A remedy proposed by Schweikert [4] is known as *tension spline*. The functional minimized includes the first derivative term in addition to the second derivative term.

The *smooth approximation* [8] is an approach to data interpolating or data fitting that employs the variational formulation of the problem in a normed space with constraints representing the approximation conditions. The cubic spline interpolation in 1D is also known to be the approximation of this kind.

For the cubic spline, the objective is to minimize the L^2 norm of the second derivative of the approximating function. A more complex criterion then is to minimize, with some weights chosen, the L^2 norms of several (or possibly all) derivatives of a sufficiently smooth approximating function. In the paper, we are concerned with the tension spline constructed by means of the smooth approximation theory

DOI: 10.21136/panm.2016.14

(cf. also [3]), i.e. with the exact interpolation of the data at nodes and, at the same time, with the smoothness of the interpolating curve and its first derivative.

For the sake of simplicity, we are mostly concerned with the 1D case in the paper. Assuming the approach of [8] and [5], we introduce the problem to be solved and the tools necessary to this aim in Sec. 3. We also present the general existence theorem for smooth interpolation proven in [5]. We use the basis system $\exp(\mathrm{i}kx)$ of exponential functions of pure imaginary argument for smooth interpolation problems in Sec. 4. In Sec. 5 we investigate some properties of this basis suitable for preserving the smoothness of the interpolation and for generating the tension spline in 1D, 2D, and 3D. We also present a 1D numerical example and discuss it to illustrate some properties of smooth interpolation in Sec. 6.

2. Problem of data interpolation

Basic notation and fundamental statements are presented, e.g., in [6]. Let us have a finite number N of (complex, in general) measured (sampled) values $f_1, f_2, \ldots, f_N \in C$ obtained at N nodes $X_1, X_2, \ldots, X_N \in R^n$. The nodes are assumed to be mutually distinct. We are usually interested also in the intermediate values corresponding to other points in some domain. Assume that $f_j = f(X_j)$ are measured values of some continuous function f while z is an approximating function to be constructed. The dimension n of the independent variable can be arbitrary.

Definition 1 (Interpolation)**.** *The* interpolating function *(interpolant) z is constructed to fulfil the interpolation conditions*

$$z(X_j) = f_j, \quad j = 1, \ldots, N. \tag{1}$$

Various additional conditions can be considered, e.g. minimization of some functionals applied to z.

The problem of data interpolation does not have a unique solution. The property (1) of the interpolating function is uniquely formulated by mathematical means but there are also requirements on the *subjective perception* of the behavior of the approximating curve or surface between nodes that can hardly be formalized.

The general *problem of smooth approximation (smooth curve fitting, data smoothing)*, where the interpolation condition (1) is not applied, is treated in more detail e.g. in [5], [8].

For the sake of simplicity we now put $n = 1$ and assume that $X_1, X_2, \ldots, X_N \in \Omega$, where either $\Omega = [a, b]$ is a finite interval or $\Omega = (-\infty, \infty)$. We will turn back to general $n \geq 1$ in Sec. 5.

3. Smooth interpolation

We introduce an inner product space to formulate the additional constraints in the problem of smooth approximation [6], [8]. Let $\widetilde{\mathcal{W}}$ be a linear vector space of complex valued functions g continuous together with their derivatives of all orders

on the interval Ω. Let $\{B_l\}_{l=0}^{\infty}$ be a sequence of nonnegative numbers and L the smallest nonnegative integer such that $B_L > 0$ while $B_l = 0$ for $l < L$. For $g, h \in \widetilde{\mathcal{W}}$, put

$$(g,h)_L = \sum_{l=0}^{\infty} B_l \int_{\Omega} g^{(l)}(x)[h^{(l)}(x)]^* \,\mathrm{d}x, \tag{2}$$

$$|g|_L^2 = \sum_{l=0}^{\infty} B_l \int_{\Omega} |g^{(l)}(x)|^2 \,\mathrm{d}x, \tag{3}$$

where * denotes the complex conjugate.

If $L = 0$ (i.e. $B_0 > 0$), consider functions $g \in \widetilde{\mathcal{W}}$ such that the value of $|g|_0$ exists and is finite. Then $(g,h)_0 = (g,h)$ has the properties of *inner product* and the expression $|g|_0 = \|g\|$ is *norm* in a normed space $W_0 = \widetilde{\mathcal{W}}$.

Let $L > 0$. Consider again functions $g \in \widetilde{\mathcal{W}}$ such that the value of $|g|_L$ exists and is finite. Let $P_{L-1} \subset \widetilde{\mathcal{W}}$ be the subspace whose basis $\{\varphi_p\}$ consists of monomials

$$\varphi_p(x) = x^{p-1}, \quad p = 1, \ldots, L.$$

Then $(\varphi_p, \varphi_q)_L = 0$ and $|\varphi_p|_L = 0$ for $p, q = 1, \ldots, L$. Using (2) and (3), we construct the *quotient space* $\widetilde{\mathcal{W}}/P_{L-1}$ whose zero class is the subspace P_{L-1}. Finally, considering $(\cdot,\cdot)_L$ and $|\cdot|_L$ in every equivalence class, we see that they represent the inner product and norm in a normed space $W_L = \widetilde{\mathcal{W}}/P_{L-1}$ [6].

W_L is the normed space where we minimize functionals and measure the smoothness of the interpolation. For an arbitrary $L \geq 0$, choose a *basis system* of functions $\{g_k\} \subset W_L$, $k = 1, 2, \ldots$, that is complete and orthogonal (in the inner product in W_L), i.e., $(g_k, g_m)_L = 0$ for $k \neq m$, $(g_k, g_k)_L = |g_k|_L^2 > 0$. If $L > 0$ then it is, moreover, $(\varphi_p, g_k)_L = 0$ for $p = 1, \ldots, L$, $k = 1, 2, \ldots$. The set $\{\varphi_p\}$ is empty for $L = 0$.

Definition 2 (Smooth data interpolation)**.** *The* problem of smooth data interpolation [8] *consists in finding the coefficients A_k and a_p of the interpolant*

$$z(x) = \sum_{k=1}^{\infty} A_k g_k(x) + \sum_{p=1}^{L} a_p \varphi_p(x) \tag{4}$$

such that

$$z(X_j) = f_j, \quad j = 1, \ldots, N, \tag{5}$$

and

$$\text{the quantity } |z|_L^2 \text{ attains its minimum.} \tag{6}$$

Apparently, the infinite sum in (4) is inconvenient for practical computation. Therefore, we introduce the *generating function*

$$R_L(x,y) = \sum_{k=1}^{\infty} \frac{g_k(x) g_k^*(y)}{|g_k|_L^2}. \tag{7}$$

We state in Theorem 1, which follows, that a finite linear combination of the values of the generating function R_L at particular nodes is used for the practical interpolation instead of the infinite linear combination in (4). Further put

$$R = [R_L(X_i, X_j)], \quad i, j = 1, \ldots, N,$$

where R is an $N \times N$ square Hermitian matrix, and if $L > 0$ then introduce an $N \times L$ matrix

$$\Phi = [\varphi_p(X_j)], \quad j = 1, \ldots, N, \ p = 1, \ldots, L.$$

Theorem 1. *Let $X_i \neq X_j$ for all $i \neq j$. Assume that the generating function (7) converges for all $x, y \in \Omega$. If $L > 0$ let* rank $\Phi = L$. *Then the problem of smooth interpolation (4) to (6) has the unique solution*

$$z(x) = \sum_{j=1}^{N} \lambda_j R_L(x, X_j) + \sum_{p=1}^{L} a_p \varphi_p(x), \tag{8}$$

where the coefficients λ_j, $j = 1, \ldots, N$, and a_p, $p = 1, \ldots, L$, are the unique solution of a nonsingular system of $N + L$ linear algebraic equations.

Proof. The proof is given in [5]. □

4. A particular basis function system

Recall that we have put $n = 1$. Let the function f to be approximated be 2π-periodic in $[0, 2\pi]$. We choose exponential functions of pure imaginary argument for the periodic basis system $\{g_k\}$ in W_L. The following theorem shows important properties of the system.

Theorem 2. *Let there be an integer s, $s \geq L$, such that $B_l = 0$ for all $l > s$ in W_L. The system of periodic exponential functions of pure imaginary argument*

$$g_k(x) = \exp(-\mathrm{i}kx), \quad x \in [0, 2\pi], \ k = 0, \pm 1, \pm 2, \ldots, \tag{9}$$

is complete and orthogonal in W_L.

Proof. The proof is given in [6]. □

The range of k implies a minor change in the notation introduced above. For the basis system (9), notice that the generating function

$$R_L(x, y) = \sum_{k=-\infty}^{\infty} \frac{g_k(x) g_k^*(y)}{|g_k|_L^2} = \sum_{k=-\infty}^{\infty} \frac{\exp(-\mathrm{i}k(x-y))}{|g_k|_L^2} \tag{10}$$

is the Fourier series in $L^2(0, 2\pi)$ with the coefficients $|g_k|_L^{-2}$, where

$$|g_k|_L^2 = 2\pi \sum_{l=L}^{\infty} B_l k^{2l} \tag{11}$$

according to (3).

Let now the function f to be approximated be nonperiodic on $(-\infty, \infty)$ and $f^{(l)}(\pm\infty) = 0$ for all $l \geq 0$. Let us define the generating function $R_L(x, y)$ as the Fourier transform of the function $|g_k|_L^{-2}$ of continuous variable k,

$$R_L(x, y) = \int_{-\infty}^{\infty} \frac{\exp(-\mathrm{i}k(x-y))}{|g_k|_L^2} \, \mathrm{d}k, \tag{12}$$

if the integral exists. Using the effect of transition from the Fourier series (10) with the coefficients $|g_k|_L^{-2}$ to the Fourier transform (12) of the function $|g_k|_L^{-2}$ of continuous variable k (cf., e.g., [6]), we have transformed the basis functions, enriched their spectrum, and released the requirement of periodicity of f. Moreover, if the integral (12) does not exist in the usual sense, in many instances we can calculate $R_L(x, y)$ as the Fourier transform $\mathcal{F}$ of the generalized function $|g_k|_L^{-2}$ of k.

5. Tension spline

To finish the definition of the inner product and norm (2), (3) in a particular space W_L we now choose a particular sequence $\{B_l\}$ and set, therefore, the minimization properties of the smooth interpolant. Let us thus put (cf. [3])

$$B_l = 0 \text{ for all } l \text{ with the exception of } B_1 = \alpha^2, \ \alpha > 0, \text{ and } B_2 = 1. \tag{13}$$

It means that we have $L = 1$ and minimize the L^2 norm of the first derivative (characterizing oscillations) multiplied by α^2 plus the L^2 norm of the second derivative (characterizing the curvature) of the interpolant (4) in the form (8), i.e.

$$z(x) = \sum_{j=1}^{N} \lambda_j R_1(x, X_j) + a_1. \tag{14}$$

We get

$$|g_k|_1^2 = 2\pi(\alpha^2 k^2 + k^4)$$

from (11). Let $r = |x - y|$. We arrive at

$$\begin{aligned} R_1(x, y) &= \mathcal{F}\left(\frac{1}{2\pi k^2(\alpha^2 + k^2)}\right) = \frac{1}{2\pi}\mathcal{F}\left(\frac{1}{\alpha^2 k^2} - \frac{1}{\alpha^2(k^2 + \alpha^2)}\right) \\ &= -\frac{1}{2\alpha^3}(\alpha r + \exp(-\alpha r)), \end{aligned} \tag{15}$$

where $\mathcal{F}$ denotes the Fourier transform of a generalized function (see [2], p. 375, formula 14 and p. 377, formula 29; and [1], formula 8.469.3), cf. [7]. We see that this version of smooth approximation is equivalent to the tension spline interpolation [4] but introduced in a way different from [3].

For dimension $n > 1$, l is a multiindex, k, x, y are vectors, and the formula (14) remains the same.

If $n = 2$ then $|g_k|_1^2 = 4\pi^2(\alpha^2(k_1^2 + k_2^2) + (k_1^2 + k_2^2)^2)$ and we arrive at

$$\begin{aligned} R_1(x, y) &= \mathcal{F}\left(\frac{1}{4\pi^2(\alpha^2(k_1^2 + k_2^2) + (k_1^2 + k_2^2)^2)}\right) \\ &= -\frac{1}{2\pi\alpha^2}(\ln(\tfrac{1}{2}\alpha r) - \mathrm{K}_0(\alpha r) + C), \end{aligned}$$

where C is a constant that can be included into a_1 in (14) and K_0 is the modified Bessel function of the second kind (see [2], p. 382, formula 13 and p. 380, formula 5).

Moreover, if $n = 3$ then $|g_k|_1^2 = 8\pi^3(\alpha^2(k_1^2 + k_2^2 + k_3^2) + (k_1^2 + k_2^2 + k_3^2)^2)$. We finally have

$$\begin{aligned} R_1(x, y) &= \mathcal{F}\left(\frac{1}{8\pi^3(\alpha^2(k_1^2 + k_2^2 + k_3^2) + (k_1^2 + k_2^2 + k_3^2)^2)}\right) \\ &= \frac{1}{4\pi\alpha^2}\,\frac{1 - \exp(-\alpha r)}{r} \end{aligned}$$

(see [2], p. 382, formula 13 and p. 380, formula 1).

At $r = 0$, the above functions $R_1(x, y)$ are defined as a limit for $r \to 0$.

There are further practical examples of smooth interpolation where the integral (12) that defines the generating function can be calculated with the help of the Fourier transform. It is easy to show that one of them is the 1D *cubic spline interpolation* and its analogues in 2D and 3D [6].

6. Computational comparison

We present results of a simple numerical experiment with the tension spline for $n = 1$. We employ the complete and orthogonal system (9) and the sequence (13) to introduce the space W_1. We use the interpolant (14), where R_1 is given by (15). The function to be interpolated is

$$f(x) = 8 - \frac{2}{1 + 16x^2}. \tag{16}$$

Apparently, it has "almost a negative pole" at $x = 0$. The tension spline interpolation of the function (16) has been constructed in several equidistant grids of N nodes on $[-1, 1]$ and for several values of α^2 including also $\alpha^2 = 0$, i.e. the cubic spline.

Some of the results of interpolation are in Fig. 1. We put $N = 9$ and compare tension splines with $\alpha^2 = 0$, $\alpha^2 = 1\,000$, and $\alpha^2 = 10\,000$. The interpolants are in the upper part of the figure, their first derivatives in the lower part along the x axis.

We see that the tension splines do not differ substantially from each other but their derivatives are very unlike. The derivative of the cubic spline is a smooth function while the derivative of the tension spline with $\alpha^2 = 10\,000$ is similar to a piecewise constant function with smooth changes between the constant levels. This corresponds to the behavior of the tension spline if examined in a different scale: it

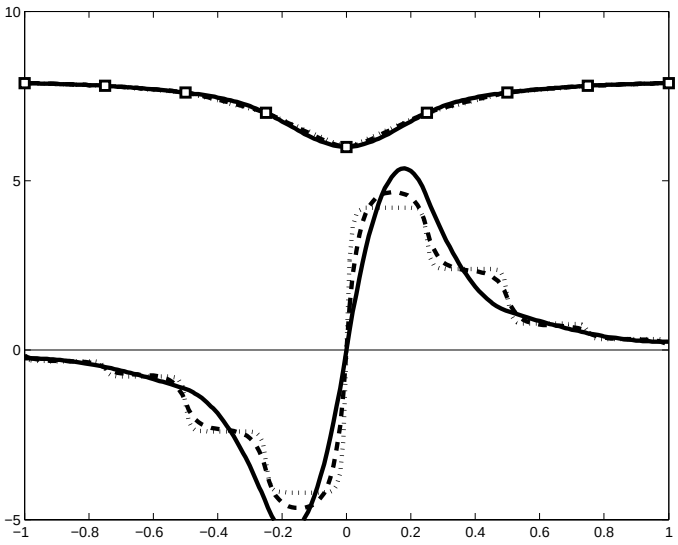


Figure 1: $N = 9$. The horizontal axis: independent variable, the vertical axis: interpolant (in the upper part of the figure) and its derivative (in the lower part). Cubic spline (tension $\alpha^2 = 0$): solid line, tension spline ($\alpha^2 = 1\ 000$): dashed line, tension spline ($\alpha^2 = 10\ 000$): dotted line.

resembles a piecewise linear curve but it is smooth, not sharp-cornered also at nodes, i.e. its derivative is continuous.

A proper choice of the parameter α^2 can provide a compromise interpolation solution with both tension spline and its derivative so smooth that they give a good, pleasing subjective impression.

7. Conclusion

We have shown that the generating function for the tension spline interpolation can be obtained by means of the Fourier transform of generalized functions. To this end we have employed the integral definition (12) of the generating function and some known formulae for the Fourier transform. The Fourier transform can be successfully used to determine the generating function also in several other cases including $n = 2$ and $n = 3$. Moreover, the same approach can be applied to smooth approximation where no interpolation conditions (1) are prescribed. The example in Fig. 1 is a very simple illustration of the 1D smooth interpolation case.

Acknowledgements

This work has been supported by the Czech Science Foundation through Grant 14-02067S and by project RVO 67985840.

References

[1] Gradshteyn, I. S. and Ryzhik, I. M.: *Table of integrals, series, and products.* Academic Press, Boston, 1994.

[2] Kreĭn, S. G. (Ed.): *Functional analysis* (Russian). Nauka, Moskva, 1964.

[3] Mitáš, L. and Mitášová, H.: General variational approach to the interpolation problem. Comput. Math. Appl. **16** (1988), 983–992.

[4] Schweikert, D. G.: An interpolation curve using a spline in tension. J. Math. Phys. **45** (1966), 312–317.

[5] Segeth, K.: Some computational aspects of smooth approximation. Computing **95** (2013), S695–S708.

[6] Segeth, K.: A periodic basis system of the smooth approximation space. Appl. Math. Comput. **267** (2015), 436–444.

[7] Segeth, K.: A note on tension spline. In: *Applications of Mathematics 2015*, pp. 217–224. Institute of Mathematics CAS, Prague, 2015.

[8] Talmi, A. and Gilat, G.: Method for smooth approximation of data. J. Comput. Phys. **23** (1977), 93–123.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

IMPLICIT CONSTITUTIVE SOLUTION SCHEME FOR MOHR-COULOMB PLASTICITY

Stanislav Sysala[1], Martin Čermák[2]

[1] Institute of Geonics, Czech Academy of Sciences
Studentská 1768, 708 00 Ostrava, Czech Republic
stanislav.sysala@ugn.cas.cz
[2] IT4Innovations National Supercomputing Centre, VŠB-TU Ostrava
17. listopadu 15, 708 00 Ostrava, Czech Republic
martin.cermak@vsb.cz

Abstract: This contribution summarizes an implicit constitutive solution scheme of the elastoplastic problem containing the Mohr-Coulomb yield criterion, a nonassociative flow rule, and a nonlinear isotropic hardening. The presented scheme builds upon the subdifferential formulation of the flow rule leading to several improvements. Mainly, it is possible to detect a position of the unknown stress tensor on the Mohr-Coulomb pyramid without blind guesswork. Further, a simplified construction of the consistent tangent operator is introduced. The presented results are important for an efficient solution of incremental boundary value elastoplastic problems.

Keywords: Mohr-Coulomb plasticity, implicit constitutive solution scheme, consistent tangent operator

MSC: 34L10, 74C05, 74D10, 74L10, 90C25

1. Introduction

We focus on a solution of an elastoplastic constitutive problem containing the *Mohr-Coulomb yield criterion* and a consequent construction of the *consistent tangent operator* which is important for Newton-like methods in elastoplasticity. This constitutive problem is broadly exploited in soil and rock mechanics and many various solution schemes were suggested. For their detailed overview and historical development, we refer the recent papers [1] and [3], respectively. Nevertheless, it is still a challenging problem due to its technical complexity. It follows from the fact that the Mohr-Coulomb yield surface is a *hexagonal pyramid* aligned with the hydrostatic axis in terms of *principal stresses*.

We consider the Mohr-Coulomb constitutive initial-value problem introduced in [2, Sections 6.3–6.6] which can optionally contain the *nonassociative flow rule* and

DOI: 10.21136/panm.2016.15

the *nonlinear isotropic hardening*. The solution scheme mainly depends on a formulation of the problem and its discretization. For example, the scheme based on the multisurface representation of the flow rule and the implicit Euler discretization is derived in detail in [2, Section 8.2]. To improve this conventional scheme, we use the *subdifferential formulation* of the flow rule instead of the multisurface one. The subdifferential-based implicit solution concept was proposed in [5] for yield criteria containing 1 or 2 singular points on the yield surface. Then it was extended to the Mohr-Coulomb problem in [4]. Here, we summarize the main results from [4] and write the solution scheme in more readable form.

The rest of the contribution is organized as follows. In Section 2, the Mohr-Coulomb constitutive problem discretized by the implicit Euler method is introduced. Section 3 contains selected theoretical results characterizing the problem. Based on these results, the improved solution scheme is introduced, see Section 4. Finally, some concluding remarks are mentioned in Section 5.

Besides scalar variables, we work mainly with second and fourth order tensors. For easier orientation in the text, we denote the second order tensors by bold letters and the fourth order tensors by capital blackboard letters, e.g., $\mathbb{D}_e$ or $\mathbb{I}$. The symbols $\otimes$ and : mean the tensor product and the biscalar product, respectively (see, e.g., [2]). We also use the following notation: $\mathbb{R}_+ := \{z \in \mathbb{R};\ z \geq 0\}$ and $\mathbb{R}^{3\times3}_{sym}$ for the space of symmetric, second order tensors.

2. Formulation of the discretized problem

Let $\boldsymbol{\sigma}, \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^p \in \mathbb{R}^{3\times3}_{sym}$, $\bar{\varepsilon}^p, \kappa, \lambda \in \mathbb{R}_+$ denote the stress tensor, the strain tensor, the plastic strain tensor, the hardening variable, the thermodynamical hardening force, and the plastic multiplier, respectively. The spectral decomposition of the stress tensor reads as:

$$\boldsymbol{\sigma} = \sum_{i=1}^{3} \sigma_i \mathbf{e}_i \otimes \mathbf{e}_i, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3, \tag{1}$$

where $\sigma_1, \sigma_2, \sigma_3$ are the ordered eigenvalues (the principal stresses) of $\boldsymbol{\sigma}$ and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are the corresponding eigenvectors. Recall that $\sigma_1, \sigma_2, \sigma_3$ are uniquely defined with respect to the prescribed ordering. The Mohr-Coulomb yield function and the related plastic potential are defined as follows:

$$f(\boldsymbol{\sigma}, \kappa) = (1 + \sin\phi)\sigma_1 - (1 - \sin\phi)\sigma_3 - 2(c_0 + \kappa)\cos\phi, \tag{2}$$

$$g(\boldsymbol{\sigma}) = (1 + \sin\psi)\sigma_1 - (1 - \sin\psi)\sigma_3, \tag{3}$$

respectively. Here, the material parameters $c_0 > 0$, $\phi, \psi \in (0, \pi/2)$ represent the initial cohesion, the friction angle, and the dilatancy angle, respectively. It is worth mentioning that g is a convex function and thus one can use its subdifferential $\partial g(\boldsymbol{\sigma})$. Further, define the fourth order tensor

$$\mathbb{D}_e = \frac{1}{3}(3K - 2G)\boldsymbol{I} \otimes \boldsymbol{I} + 2G\mathbb{I}, \tag{4}$$

representing linear and isotropic elastic response, where $K, G > 0$ denote the bulk and shear moduli, respectively, $\boldsymbol{I}$ is the second order identity tensor ($[\boldsymbol{I}]_{ij} = \delta_{ij}$, $i, j = 1, 2, 3$), and $\mathbb{I}$ is the fourth order identity tensor ($[\mathbb{I}]_{ijkl} = \delta_{ik}\delta_{jl}$, $i, j, k, l = 1, 2, 3$). Finally, it holds that $\kappa = H(\bar{\varepsilon}^p)$, where H is a nondecreasing, continuous, and piecewise smooth function satisfying $H(0) = 0$. As in [2], we let this function in an abstract form.

The elastoplastic constitutive initial value problem is defined on a pseudo-time interval $[0, T]$. With respect to the implicit Euler discretization, we consider a partition $0 = t_0 < t_1 < \ldots < t_k < \ldots < t_N = T$, fix a step k and denote $\boldsymbol{\sigma} := \boldsymbol{\sigma}(t_k)$, $\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon}(t_k)$, $\boldsymbol{\varepsilon}^p := \boldsymbol{\varepsilon}^p(t_k)$, $\bar{\varepsilon}^p := \bar{\varepsilon}^p(t_k)$, $\triangle\lambda = \lambda(t_k) - \lambda(t_{k-1})$, $\bar{\varepsilon}^{p,tr} := \bar{\varepsilon}^p(t_{k-1})$, $\boldsymbol{\varepsilon}^{tr} := \boldsymbol{\varepsilon}(t_k) - \boldsymbol{\varepsilon}^p(t_{k-1})$, and $\boldsymbol{\sigma}^{tr} := \mathbb{D}_e : \boldsymbol{\varepsilon}^{tr}$. Here, the superscript tr is the standard notation for the so-called trial variables which are known (see, e.g., [2]). The k-th step problem reads as:

Given $\boldsymbol{\sigma}^{tr}$ and $\bar{\varepsilon}^{p,tr}$. Find $\boldsymbol{\sigma}$, $\bar{\varepsilon}^p$, and $\triangle\lambda$ satisfying:

$$\left.\begin{aligned} \boldsymbol{\sigma} = \boldsymbol{\sigma}^{tr} - \triangle\lambda\mathbb{D}_e : \boldsymbol{\nu}, \quad \boldsymbol{\nu} \in \partial g(\boldsymbol{\sigma}), \\ \bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi), \\ \triangle\lambda \geq 0, \quad f(\boldsymbol{\sigma}, H(\bar{\varepsilon}^p)) \leq 0, \quad \triangle\lambda f(\boldsymbol{\sigma}, H(\bar{\varepsilon}^p)) = 0. \end{aligned}\right\} \tag{5}$$

Notice that the remaining unknown variables can be computed from the solution components $\boldsymbol{\sigma}$, $\bar{\varepsilon}^p$, and $\triangle\lambda$. For example, it holds that $\boldsymbol{\varepsilon}^p(t_k) = \boldsymbol{\varepsilon}(t_k) - \mathbb{D}_e^{-1} : \boldsymbol{\sigma}(t_k)$.

3. Useful theoretical results

In this section, we summarize some theoretical results concerning problem (5). This framework is important for understanding of the solution scheme introduced in Section 4.

The first result enables to write problem (5) only in terms of principal stresses. For its derivation, it was necessary to find the subdifferential $\partial g(\boldsymbol{\sigma})$ in closed form with respect to (3), see [4, Lemma 4.1].

Lemma 1. *Let $(\boldsymbol{\sigma}, \bar{\varepsilon}^p, \triangle\lambda)$ be a solution to (5) for given $\boldsymbol{\sigma}^{tr}$ and $\bar{\varepsilon}^{p,tr}$. Let σ_i, σ_i^{tr}, $i = 1, 2, 3$, be the ordered eigenvalues of $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}^{tr}$, respectively. Then $(\sigma_1, \sigma_2, \sigma_3, \bar{\varepsilon}^p, \triangle\lambda)$ is a solution to:*

$$\left.\begin{aligned} \sigma_i = \sigma_i^{tr} - \triangle\lambda\left[\tfrac{2}{3}(3K - 2G)\sin\psi + 2G\nu_i\right], \quad i = 1, 2, 3, \\ \bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi), \\ \triangle\lambda \geq 0, \;\; (1 + \sin\phi)\sigma_1 - (1 - \sin\phi)\sigma_3 - 2(c_0 + H(\bar{\varepsilon}^p))\cos\phi \leq 0, \\ \triangle\lambda\left[(1 + \sin\phi)\sigma_1 - (1 - \sin\phi)\sigma_3 - 2(c_0 + H(\bar{\varepsilon}^p))\cos\phi\right] = 0, \end{aligned}\right\} \tag{6}$$

where ν_1, ν_2, ν_3 are the eigenvalues of $\boldsymbol{\nu} \in \partial g(\boldsymbol{\sigma})$ satisfying

$$\left.\begin{aligned} 1 + \sin\psi \geq \nu_1 \geq \nu_2 \geq \nu_3 \geq -1 + \sin\psi, \quad \nu_1 + \nu_2 + \nu_3 = 2\sin\psi, \\ (\nu_1 - 1 - \sin\psi)(\sigma_1 - \sigma_2) = 0, \quad (\nu_3 + 1 - \sin\psi)(\sigma_2 - \sigma_3) = 0. \end{aligned}\right\} \tag{7}$$

Conversely, if $(\sigma_1, \sigma_2, \sigma_3, \bar{\varepsilon}^p, \triangle\lambda)$ *is a solution to (6) then* $(\boldsymbol{\sigma}, \bar{\varepsilon}^p, \triangle\lambda)$ *solves (5), where* $\boldsymbol{\sigma} = \sum_{i=1}^{3} \sigma_i \mathbf{e}_i^{tr} \otimes \mathbf{e}_i^{tr}$ *and* $\mathbf{e}_1^{tr}, \mathbf{e}_2^{tr}, \mathbf{e}_3^{tr}$ *are the eigenvectors of* $\boldsymbol{\sigma}^{tr}$ *with respect to the ordering* $\sigma_1^{tr} \geq \sigma_2^{tr} \geq \sigma_3^{tr}$.

A further simplification of the problem is possible under additional assumptions on the solution to problem (6). First, assume $\triangle\lambda = 0$. Then the *elastic response* appears and it holds: $\sigma_i = \sigma_i^{tr}$, $i = 1, 2, 3$, $\bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr}$, and

$$f(\boldsymbol{\sigma}^{tr}, H(\bar{\varepsilon}^{p,tr})) = (1 + \sin\phi)\sigma_1^{tr} - (1 - \sin\phi)\sigma_3^{tr} - 2(c_0 + H(\bar{\varepsilon}^{p,tr}))\cos\phi \leq 0. \quad (8)$$

In fact, (8) is a necessary and sufficient condition for $\triangle\lambda = 0$. If $\triangle\lambda > 0$ then the unknown principal stresses lie on the yield surface of the Mohr-Coulomb pyramid as follows from $(6)_4$. We distinguish four possible positions on the yield surface: the smooth portion ($\sigma_1 > \sigma_2 > \sigma_3$), the left edge ($\sigma_1 = \sigma_2 > \sigma_3$), the right edge ($\sigma_1 > \sigma_2 = \sigma_3$), and the apex ($\sigma_1 = \sigma_2 = \sigma_3$). This terminology follows from [2]. For each position, one can introduce a special solution scheme, the so-called *return-mapping scheme*. These schemes are introduced in Sections 4.4-4.7. Briefly speaking, nonlinear equations $q_s^{tr}(\triangle\lambda) = 0$, $q_l^{tr}(\triangle\lambda) = 0$, $q_r^{tr}(\triangle\lambda) = 0$, and $q_a^{tr}(\triangle\lambda) = 0$ are derived within these schemes, respectively. After finding their solutions, one can easily compute the remaining unknowns. However, only one type of the return-mapping usually leads to the solution of problem (6) and the remaining schemes produce incorrect solutions. To find the correct scheme, we define the intervals C_s^{tr}, C_l^{tr}, C_r^{tr}, C_a^{tr} introduced in Section 4.1. These intervals are mutually disjoint, their union is equal to $\mathbb{R}_+$, and either $C_l^{tr} = \emptyset$ or $C_r^{tr} = \emptyset$. For example, the return to the smooth portion appears if the solution of $q_s^{tr}(\triangle\lambda) = 0$ belongs to C_s^{tr}. Analogous criteria hold for the remaining return types.

It seems that one must successively solve the nonlinear equations with q_s^{tr}, q_l^{tr}, q_r^{tr}, q_a^{tr} to find the correct scheme. Similar blind guesswork is also introduced, e.g., in [2, Section 8.2]. Nevertheless, the presented approach enables to derive a priori decision criteria to detect the stress position on the yield surface, without any blind guesswork. To this end, we introduce the following useful result [4, Lemma 4.2].

Lemma 2. *There exists a unique function* $q^{tr} : \mathbb{R}_+ \to \mathbb{R}$ *satisfying:*

(*i*) $q^{tr}|_{C_s^{tr}} = q_s^{tr}$, $q^{tr}|_{C_l^{tr}} = q_l^{tr}$, $q^{tr}|_{C_r^{tr}} = q_r^{tr}$, $q^{tr}|_{C_a^{tr}} = q_a^{tr}$.

(*ii*) q^{tr} *is continuous, piecewise smooth, and decreasing in* $\mathbb{R}_+$.

(*iii*) $q^{tr}(0) = f(\boldsymbol{\sigma}^{tr}, H(\bar{\varepsilon}^{p,tr}))$.

(*iv*) $q^{tr}(\gamma) \to -\infty$ *as* $\gamma \to +\infty$.

The properties of the function q^{tr} have many important consequences. First, they imply the main solvability result [4, Theorems 4.4–4.6].

Theorem 3. *Problems (5) and (6) have unique solutions and the solution component* $\triangle\lambda$ *satisfies* $q^{tr}(\triangle\lambda) = 0$.

Second, one can easily detect one of the intervals C_s^{tr}, C_l^{tr}, C_r^{tr}, C_a^{tr} where values of q^{tr} change the sign. It leads to the a priori decision criteria introduced in Section 4.2. Finally, by Lemma 2, one can easily investigate properties of the stress-strain constitutive operator: $\boldsymbol{\sigma} = \boldsymbol{T}(\boldsymbol{\varepsilon}^{tr}; \bar{\varepsilon}^{p,tr})$. It is expected that this mapping is Lipschitz continuous and semismooth as follows from the discussion in [4].

The generalized derivative (in Clark's sense) of $\boldsymbol{T}$ represents the *consistent tangent operator*. This derivative defines the fourth order tensor $\mathbb{T}$, i.e., if $\boldsymbol{T}$ is differentiable at $(\boldsymbol{\varepsilon}^{tr}; \bar{\varepsilon}^{p,tr})$ then $\mathbb{T} = \partial \boldsymbol{T}/\partial \boldsymbol{\varepsilon}^{tr}$. The formulas defining $\mathbb{T}$ for each stress position are introduced in Sections 4.3-4.7. In case of the associative plasticity, i.e., if $\psi = \phi$, the tangent stiffness matrix is symmetric, otherwise it is nonsymmetric.

Let us note that the stress-strain operator is substituted into the balance equation leading to the incremental boundary value elastoplastic problem. The consistent tangent operator is used for assembling of the tangent stiffness matrix which is important for solving this problem by Newton-like methods [2, 4, 5].

4. Solution scheme

This section is organized as follows. Section 4.1 contains an auxilliary notation. In Section 4.2, a priori decision criteria for the elastic response and the returns to the smooth portion, the left edge, the right edge, and the apex of the yield surface are summarized. The solution schemes for these cases are introduced in parallel Sections 4.3–4.7, respectively.

4.1. Auxilliary notation

Recall that $\boldsymbol{\varepsilon}^{tr}$ and $\bar{\varepsilon}^{p,tr}$ are known in (5). The ordered eigenvalues $\varepsilon_1^{tr} \geq \varepsilon_2^{tr} \geq \varepsilon_3^{tr}$ of $\boldsymbol{\varepsilon}^{tr}$ can be determined using the Haigh-Westergaard coordinates (see, e.g., [2]). Other auxilliary notation is summarized below:

- $\sigma_i^{tr} = \frac{1}{3}(3K - 2G)(\varepsilon_1^{tr} + \varepsilon_2^{tr} + \varepsilon_3^{tr}) + 2G\varepsilon_i^{tr}$, $i = 1, 2, 3$ — trial principal stresses
- $\mathbb{E}^{tr,2}$, $[\mathbb{E}^{tr,2}]_{ijkl} = \delta_{ik}[\boldsymbol{\varepsilon}^{tr}]_{lj} + \delta_{jl}[\boldsymbol{\varepsilon}^{tr}]_{ik}$ — Fréchet derivative of $(\boldsymbol{\varepsilon}^{tr})^2$
- $\gamma_{s,l}^{tr} = \dfrac{\sigma_1^{tr} - \sigma_2^{tr}}{2G(1 + \sin\psi)}$, $\gamma_{s,r}^{tr} = \dfrac{\sigma_2^{tr} - \sigma_3^{tr}}{2G(1 - \sin\psi)}$,
- $\gamma_{l,a}^{tr} = \dfrac{\sigma_1^{tr} + \sigma_2^{tr} - 2\sigma_3^{tr}}{2G(3 - \sin\psi)}$, $\gamma_{r,a}^{tr} = \dfrac{2\sigma_1^{tr} - \sigma_2^{tr} - \sigma_3^{tr}}{2G(3 + \sin\psi)}$
- $C_s^{tr} = \big(0, \min\{\gamma_{s,l}^{tr}, \gamma_{s,r}^{tr}\}\big)$, $C_l^{tr} = \big[\gamma_{s,l}^{tr}, \gamma_{l,a}^{tr}\big)$,
- $C_r^{tr} = \big[\gamma_{s,r}^{tr}, \gamma_{r,a}^{tr}\big)$, $C_a^{tr} = \big[\max\{\gamma_{l,a}^{tr}, \gamma_{r,a}^{tr}\}, +\infty\big)$
- $S = \frac{4}{3}(3K - 2G)\sin\psi\sin\phi + 4G(1 + \sin\psi\sin\phi)$

- $L = \frac{4}{3}(3K - 2G)\sin\psi\sin\phi + G(1+\sin\psi)(1+\sin\phi) + 2G(1-\sin\psi)(1-\sin\phi)$
- $R = \frac{4}{3}(3K - 2G)\sin\psi\sin\phi + 2G(1+\sin\psi)(1+\sin\phi) + G(1-\sin\psi)(1-\sin\phi)$
- $A = 4K\sin\psi\sin\phi$
- $h(\gamma) = 2\left[c_0 + H\left(\bar{\varepsilon}^{p,tr} + \gamma(2\cos\phi)\right)\right]\cos\phi$
- $q_s^{tr}(\gamma) = (1+\sin\phi)\sigma_1^{tr} - (1-\sin\phi)\sigma_3^{tr} - h(\gamma) - S\gamma$
- $q_l^{tr}(\gamma) = \frac{1}{2}(1+\sin\phi)(\sigma_1^{tr} + \sigma_2^{tr}) - (1-\sin\phi)\sigma_3^{tr} - h(\gamma) - L\gamma$
- $q_r^{tr}(\gamma) = (1+\sin\phi)\sigma_1^{tr} - \frac{1}{2}(1-\sin\phi)(\sigma_2^{tr} + \sigma_3^{tr}) - h(\gamma) - R\gamma$
- $q_a^{tr}(\gamma) = \frac{2}{3}(\sigma_1^{tr} + \sigma_2^{tr} + \sigma_3^{tr})\sin\phi - h(\gamma) - A\gamma$
- $H_1 = h'(\triangle\lambda) = 4H'(\bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi))\cos^2\phi$ — possibly, we take the derivative from the left if $h'(\triangle\lambda)$ does not exist

4.2. A priori decision criteria

The criteria introduced below are mutually disjoint, i.e., for a given pair $(\boldsymbol{\varepsilon}^{tr}, \bar{\varepsilon}^{p,tr})$, only one possibility is realized.

The elastic response:

- $(1+\sin\phi)\sigma_1^{tr} - (1-\sin\phi)\sigma_3^{tr} - 2(c_0 + H(\bar{\varepsilon}^{p,tr}))\cos\phi \leq 0$

The return to the smooth portion of the yield surface:

- $(1+\sin\phi)\sigma_1^{tr} - (1-\sin\phi)\sigma_3^{tr} - 2(c_0 + H(\bar{\varepsilon}^{p,tr}))\cos\phi > 0$
- $q_s^{tr}(\min\{\gamma_{s,l}^{tr}, \gamma_{s,r}^{tr}\}) < 0$

The return to the left edge of the yield surface:

- $(1+\sin\phi)\sigma_1^{tr} - (1-\sin\phi)\sigma_3^{tr} - 2(c_0 + H(\bar{\varepsilon}^{p,tr}))\cos\phi > 0$
- $\gamma_{s,l}^{tr} < \gamma_{l,a}^{tr}$, $q_l^{tr}(\gamma_{s,l}^{tr}) \geq 0$, $q_l^{tr}(\gamma_{l,a}^{tr}) < 0$

The return to the right edge of the yield surface:

- $(1+\sin\phi)\sigma_1^{tr} - (1-\sin\phi)\sigma_3^{tr} - 2(c_0 + H(\bar{\varepsilon}^{p,tr}))\cos\phi > 0$
- $\gamma_{s,r}^{tr} < \gamma_{r,a}^{tr}$, $q_r^{tr}(\gamma_{s,r}^{tr}) \geq 0$, $q_r^{tr}(\gamma_{r,a}^{tr}) < 0$

The return to the apex of the yield surface:

- $(1+\sin\phi)\sigma_1^{tr} - (1-\sin\phi)\sigma_3^{tr} - 2(c_0 + H(\bar{\varepsilon}^{p,tr}))\cos\phi > 0$
- $q_a^{tr}(\max\{\gamma_{l,a}^{tr}, \gamma_{r,a}^{tr}\}) \geq 0$

Notice that other very useful necessary conditions for the returns to the smooth portion, the left and right edges are introduced in Sections 4.4–4.6, respectively. These conditions were derived in [4] and simplify the construction of the consistent tangent operator.

4.3. Solution scheme for the elastic response

- $\triangle\lambda = 0$
- $\sigma_i = \sigma_i^{tr},\ i = 1, 2, 3$
- $\bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr}$
- $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{tr}$
- $\mathbb{T} = \mathbb{D}_e$

4.4. Solution scheme for the return to the smooth portion

It is worth mentioning that $\varepsilon_1^{tr} > \varepsilon_2^{tr} > \varepsilon_3^{tr}$ is a necessary condition for this return. Therefore, the following auxilliary formulas are well-defined:

$$
\begin{aligned}
\boldsymbol{E}_i^{tr} &= \frac{(\boldsymbol{\varepsilon}^{tr} - \varepsilon_j^{tr}\boldsymbol{I})(\boldsymbol{\varepsilon}^{tr} - \varepsilon_k^{tr}\boldsymbol{I})}{(\varepsilon_i^{tr} - \varepsilon_j^{tr})(\varepsilon_i^{tr} - \varepsilon_k^{tr})}, \quad i \neq j \neq k \neq i, \ \ i = 1, 2, 3, \\
\boldsymbol{F}_{s,\phi}^{tr} &= 2G(1 + \sin\phi)\boldsymbol{E}_1^{tr} - 2G(1 - \sin\phi)\boldsymbol{E}_3^{tr} + \frac{2}{3}(3K - 2G)\sin\phi\boldsymbol{I}, \\
\boldsymbol{F}_{s,\psi}^{tr} &= 2G(1 + \sin\psi)\boldsymbol{E}_1^{tr} - 2G(1 - \sin\psi)\boldsymbol{E}_3^{tr} + \frac{2}{3}(3K - 2G)\sin\psi\boldsymbol{I}, \\
\mathbb{E}_i^{tr} &= \frac{\mathbb{E}^{tr,2} - (\varepsilon_j^{tr} + \varepsilon_k^{tr})\mathbb{I} - (2\varepsilon_i^{tr} - \varepsilon_j^{tr} - \varepsilon_k^{tr})\boldsymbol{E}_i^{tr} \otimes \boldsymbol{E}_i^{tr}}{(\varepsilon_i^{tr} - \varepsilon_j^{tr})(\varepsilon_i^{tr} - \varepsilon_k^{tr})} \\
&\quad - \frac{(\varepsilon_j^{tr} - \varepsilon_k^{tr})[\boldsymbol{E}_j^{tr} \otimes \boldsymbol{E}_j^{tr} - \boldsymbol{E}_k^{tr} \otimes \boldsymbol{E}_k^{tr}]}{(\varepsilon_i^{tr} - \varepsilon_j^{tr})(\varepsilon_i^{tr} - \varepsilon_k^{tr})}, \quad i \neq j \neq k \neq i, \ i = 1, 2, 3.
\end{aligned}
$$

It is well-known that $\boldsymbol{E}_1^{tr}, \boldsymbol{E}_2^{tr}, \boldsymbol{E}_3^{tr}$ define the eigenprojections of $\boldsymbol{\varepsilon}^{tr}$ [2]. Further, it holds: $\boldsymbol{E}_i^{tr} = \partial\varepsilon_i^{tr}/\partial\boldsymbol{\varepsilon}^{tr}$ and $\mathbb{E}_i^{tr} = \partial\boldsymbol{E}_i^{tr}/\partial\boldsymbol{\varepsilon}^{tr}$, $i = 1, 2, 3$. The solution scheme for the return to the smooth portion reads as:

- $\triangle\lambda \in C_s^{tr}$ and solves $q_s^{tr}(\triangle\lambda) = 0$
- $\sigma_1 = \sigma_1^{tr} - \triangle\lambda\left[\frac{2}{3}(3K - 2G)\sin\psi + 2G(1 + \sin\psi)\right]$
- $\sigma_2 = \sigma_2^{tr} - \triangle\lambda\left[\frac{2}{3}(3K - 2G)\sin\psi\right]$
- $\sigma_3 = \sigma_3^{tr} - \triangle\lambda\left[\frac{2}{3}(3K - 2G)\sin\psi - 2G(1 - \sin\psi)\right]$
- $\bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi)$
- $\boldsymbol{\sigma} = \sigma_1\boldsymbol{E}_1^{tr} + \sigma_2\boldsymbol{E}_2^{tr} + \sigma_3\boldsymbol{E}_3^{tr}$
- $\mathbb{T} = \sum_{i=1}^{3}\left[\sigma_i\mathbb{E}_i^{tr} + 2G\boldsymbol{E}_i^{tr} \otimes \boldsymbol{E}_i^{tr}\right] + \frac{1}{3}(3K - 2G)\boldsymbol{I} \otimes \boldsymbol{I} - \dfrac{1}{S + H_1}\boldsymbol{F}_{s,\psi}^{tr} \otimes \boldsymbol{F}_{s,\phi}^{tr}$

4.5. Solution scheme for the return to the left edge

For this return, the only one sharp inequality is guaranteed: $\varepsilon_2^{tr} > \varepsilon_3^{tr}$. We use the following auxilliary and well-defined formulas:

$$\begin{aligned}
\boldsymbol{E}_3^{tr} &= \frac{(\boldsymbol{\varepsilon}^{tr} - \varepsilon_1^{tr}\boldsymbol{I})(\boldsymbol{\varepsilon}^{tr} - \varepsilon_2^{tr}\boldsymbol{I})}{(\varepsilon_3^{tr} - \varepsilon_1^{tr})(\varepsilon_3^{tr} - \varepsilon_2^{tr})}, \quad \boldsymbol{E}_{12}^{tr} = \boldsymbol{I} - \boldsymbol{E}_3^{tr}, \\
\boldsymbol{F}_{l,\phi}^{tr} &= G(1+\sin\phi)\boldsymbol{E}_{12}^{tr} - 2G(1-\sin\phi)\boldsymbol{E}_3^{tr} + \frac{2}{3}(3K-2G)\sin\phi\boldsymbol{I}, \\
\boldsymbol{F}_{l,\psi}^{tr} &= G(1+\sin\psi)\boldsymbol{E}_{12}^{tr} - 2G(1-\sin\psi)\boldsymbol{E}_3^{tr} + \frac{2}{3}(3K-2G)\sin\psi\boldsymbol{I}, \\
\mathbb{E}_3^{tr} &= \frac{\mathbb{E}^{tr,2} - (\varepsilon_1^{tr}+\varepsilon_2^{tr})\mathbb{I} - [\boldsymbol{\varepsilon}^{tr}\otimes\boldsymbol{E}_{12}^{tr} + \boldsymbol{E}_{12}^{tr}\otimes\boldsymbol{\varepsilon}^{tr}] + (\varepsilon_1^{tr}+\varepsilon_2^{tr})\boldsymbol{E}_{12}^{tr}\otimes\boldsymbol{E}_{12}^{tr}}{(\varepsilon_3^{tr}-\varepsilon_1^{tr})(\varepsilon_3^{tr}-\varepsilon_2^{tr})} \\
&\quad + \frac{(\varepsilon_1^{tr}+\varepsilon_2^{tr}-2\varepsilon_3^{tr})\boldsymbol{E}_3^{tr}\otimes\boldsymbol{E}_3^{tr} + \varepsilon_3^{tr}[\boldsymbol{E}_{12}^{tr}\otimes\boldsymbol{E}_3^{tr} + \boldsymbol{E}_3^{tr}\otimes\boldsymbol{E}_{12}^{tr}]}{(\varepsilon_3^{tr}-\varepsilon_1^{tr})(\varepsilon_3^{tr}-\varepsilon_2^{tr})}.
\end{aligned}$$

It is possible to prove that the definitions of $\mathbb{E}_3^{tr}$ introduced here and in Section 4.4 are equivalent under the assumption $\varepsilon_1^{tr} > \varepsilon_2^{tr} > \varepsilon_3^{tr}$. The solution scheme for the return to the left edge reads as:

- $\triangle\lambda \in C_l^{tr}$ and solves $q_l^{tr}(\triangle\lambda) = 0$
- $\sigma_1 = \sigma_2 = \frac{1}{2}(\sigma_1^{tr} + \sigma_2^{tr}) - \triangle\lambda\left[\frac{2}{3}(3K-2G)\sin\psi + G(1+\sin\psi)\right]$
- $\sigma_3 = \sigma_3^{tr} - \triangle\lambda\left[\frac{2}{3}(3K-2G)\sin\psi - 2G(1-\sin\psi)\right]$
- $\bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi)$
- $\boldsymbol{\sigma} = \sigma_1\boldsymbol{E}_{12}^{tr} + \sigma_3\boldsymbol{E}_3^{tr}$
- $$\begin{cases} \mathbb{T} = (\sigma_3-\sigma_1)\mathbb{E}_3^{tr} + G\boldsymbol{E}_{12}^{tr}\otimes\boldsymbol{E}_{12}^{tr} + 2G\boldsymbol{E}_3^{tr}\otimes\boldsymbol{E}_3^{tr} + \frac{1}{3}(3K-2G)\boldsymbol{I}\otimes\boldsymbol{I} \\ \quad -\dfrac{1}{L+H_1}\boldsymbol{F}_{l,\psi}^{tr}\otimes\boldsymbol{F}_{l,\phi}^{tr} \end{cases}$$

4.6. Solution scheme for the return to the right edge

For this return, the inequality $\varepsilon_1^{tr} > \varepsilon_2^{tr}$ is guaranteed. We use the following auxilliary and well-defined formulas:

$$\begin{aligned}
\boldsymbol{E}_1^{tr} &= \frac{(\boldsymbol{\varepsilon}^{tr} - \varepsilon_2^{tr}\boldsymbol{I})(\boldsymbol{\varepsilon}^{tr} - \varepsilon_3^{tr}\boldsymbol{I})}{(\varepsilon_1^{tr} - \varepsilon_2^{tr})(\varepsilon_1^{tr} - \varepsilon_3^{tr})}, \quad \boldsymbol{E}_{23}^{tr} = \boldsymbol{I} - \boldsymbol{E}_1^{tr}, \\
\boldsymbol{F}_{r,\phi}^{tr} &= 2G(1+\sin\phi)\boldsymbol{E}_1^{tr} - G(1-\sin\phi)\boldsymbol{E}_{23}^{tr} + \frac{2}{3}(3K-2G)\sin\phi\boldsymbol{I}, \\
\boldsymbol{F}_{r,\psi}^{tr} &= 2G(1+\sin\psi)\boldsymbol{E}_1^{tr} - G(1-\sin\psi)\boldsymbol{E}_{23}^{tr} + \frac{2}{3}(3K-2G)\sin\psi\boldsymbol{I}, \\
\mathbb{E}_1^{tr} &= \frac{\mathbb{E}^{tr,2} - (\varepsilon_2^{tr}+\varepsilon_3^{tr})\mathbb{I} - [\boldsymbol{\varepsilon}^{tr}\otimes\boldsymbol{E}_{23}^{tr} + \boldsymbol{E}_{23}^{tr}\otimes\boldsymbol{\varepsilon}^{tr}] + (\varepsilon_2^{tr}+\varepsilon_3^{tr})\boldsymbol{E}_{23}^{tr}\otimes\boldsymbol{E}_{23}^{tr}}{(\varepsilon_1^{tr}-\varepsilon_2^{tr})(\varepsilon_1^{tr}-\varepsilon_2^{tr})} \\
&\quad + \frac{(\varepsilon_2^{tr}+\varepsilon_3^{tr}-2\varepsilon_1^{tr})\boldsymbol{E}_1^{tr}\otimes\boldsymbol{E}_1 + \varepsilon_1^{tr}[\boldsymbol{E}_{23}^{tr}\otimes\boldsymbol{E}_1^{tr} + \boldsymbol{E}_1^{tr}\otimes\boldsymbol{E}_{23}^{tr}]}{(\varepsilon_1^{tr}-\varepsilon_2^{tr})(\varepsilon_1^{tr}-\varepsilon_3^{tr})}.
\end{aligned}$$

It is possible to prove that the definitions of $\mathbb{E}_1^{tr}$ introduced here and in Section 4.4 are equivalent under the assumption $\varepsilon_1^{tr} > \varepsilon_2^{tr} > \varepsilon_3^{tr}$. The solution scheme for the return to the right edge reads as:

- $\triangle\lambda \in C_r^{tr}$ and solves $q_r^{tr}(\triangle\lambda) = 0$
- $\sigma_1 = \sigma_1^{tr} - \triangle\lambda \left[\frac{2}{3}(3K - 2G)\sin\psi + 2G(1 + \sin\psi)\right]$
- $\sigma_2 = \sigma_3 = \frac{1}{2}(\sigma_2^{tr} + \sigma_3^{tr}) - \triangle\lambda \left[\frac{2}{3}(3K - 2G)\sin\psi - G(1 - \sin\psi)\right]$
- $\bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi)$
- $\boldsymbol{\sigma} = \sigma_1 \boldsymbol{E}_1^{tr} + \sigma_3 \boldsymbol{E}_{23}^{tr}$
- $$\begin{cases} \mathbb{T} = (\sigma_1 - \sigma_3)\mathbb{E}_1^{tr} + 2G\boldsymbol{E}_1^{tr} \otimes \boldsymbol{E}_1^{tr} + G\boldsymbol{E}_{23}^{tr} \otimes \boldsymbol{E}_{23}^{tr} + \frac{1}{3}(3K - 2G)\boldsymbol{I} \otimes \boldsymbol{I} \\ \qquad -\dfrac{1}{R + H_1}\boldsymbol{F}_{r,\psi}^{tr} \otimes \boldsymbol{F}_{r,\phi}^{tr} \end{cases}$$

4.7. Solution scheme for the return to the apex

- $\triangle\lambda \in C_a^{tr}$ and solves $q_a^{tr}(\triangle\lambda) = 0$
- $\sigma_1 = \sigma_2 = \sigma_3 = \frac{1}{3}(\sigma_1^{tr} + \sigma_2^{tr} + \sigma_3^{tr}) - \triangle\lambda[2K\sin\psi]$
- $\bar{\varepsilon}^p = \bar{\varepsilon}^{p,tr} + \triangle\lambda(2\cos\phi)$
- $\boldsymbol{\sigma} = \sigma_1 \boldsymbol{I}$
- $\mathbb{T} = K\left(1 - \dfrac{A}{A + H_1}\right)\boldsymbol{I} \otimes \boldsymbol{I}$

5. Conclusion

The subdifferential-based constitutive solution scheme for the Mohr-Coulomb model was introduced. This technique has several advantages in comparison to the current ones. First, it enabled a deeper analysis of the constitutive problem. Second, a priori decision criteria were derived for each position of the unknown stress tensor on the yield surface. Finally, for each return type, we specified the necessary conditions on multiplicity of $\varepsilon_1^{tr}, \varepsilon_2^{tr}, \varepsilon_3^{tr}$. Such conditions are crucial for the correct definition of the consistent tangent operator $\mathbb{T}$. Without this knowledge, an additional branching in the definition of $\mathbb{T}$ must be introduced as in [2, Appendix A].

The presented solution schemes were implemented in Matlab codes for the analysis of slope stability in 2D and 3D. The codes are publicly available in [6] and the used numerical techniques are described in [4].

Acknowledgements

This work was supported by The Ministry of Education, Youth and Sports (of the Czech Republic) from the National Programme of Sustainability (NPU II), project "IT4Innovations excellence in science - LQ1602".

References

[1] Clausen, J., Damkilde, L., Andersen, L. V.: Robust and efficient handling of yield surface discontinuities in elastoplastic finite element calculations. Engineering Computations **32 (6)** (2015), 1722–1752.

[2] de Souza Neto, E. A., Perić, D., Owen, D. R. J: *Computational methods for plasticity: theory and application.* Wiley, 2008.

[3] Karaoulanis, F. E.: Implicit numerical integration of nonsmooth multisurface yield criteria in the principal stress space. Arch. Computat. Methods Eng. **20 (3)** (2013), 263–308.

[4] Sysala, S., Cermak, M.: Subdifferential-based implicit return-mapping operators in Mohr-Coulomb plasticity. Submitted, 2016.
`http://arxiv.org/abs/1508.07435`

[5] Sysala, S., Cermak, M., Koudelka, T., Kruis, J., Zeman, J., Blaheta, R.: Subdifferential-based implicit return-mapping operators in computational plasticity. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **96 (11)** (2016), 1318–1338.

[6] Experimental Matlab codes concerning slope stability and Mohr-Coulomb models. `www.ugn.cas.cz/?p=publish/output.php` (or `www.ugn.cas.cz` – Publications – Other outputs – SS-MC-NP-3D, SS-MC-NH, SS-MC-NP-Acontrol), 2016.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

COMPUTATIONAL MODELLING OF THERMAL CONSUMPTION OF BUILDINGS WITH CONTROLLED INTERIOR TEMPERATURE

Jiří Vala

Brno University of Technology, Faculty of Civil Engineering,
Institute of Mathematics and Descriptive Geometry, 602 00 Brno, Veveří 95
vala.j@fce.vutbr.cz

Abstract: New materials, structures and technologies used in civil engineering impeach traditional evaluations of the annual thermal consumption of buildings, based on the quasi-stationary estimate of the thermal resistance of the building envelope, or some operational parts of such building with the guaranteed temperature. The complete proper physical analysis, applying the principles of thermodynamics and appropriate constitutive relations for particular material layers and air in rooms, is not realistic because of the stochastic character of physical processes, of the need of identification of a lot of material characteristics, of the barely predictable behaviour of users, as well as of the limited financial budgets of investors. From the soliciting compromise solutions, this paper pays attention to the simplified formulation of dynamic behaviour of a building as a nonlinear thermal system, referring to certain analogy with the analysis of electrical circuits. Numerical solution applies the finite element technique, the method of lines and the spectral analysis of differential operators. The practical implementation is performed in MATLAB.

Keywords: building heat transfer, computational optimization, Fourier analysis

MSC: 80A20, 35K20, 65K10

1. Introduction

Application of advanced materials, structures and technologies in civil engineering forces new approaches to the physical, mathematical and computational analysis of both new and reconstructed buildings. Moreover, a lot of European and national directives and technical standards requires the evaluation of various thermal certificates, as the European directive [25], forcing the "passive house" properties, taken from [9], for all building actions starting 2020 and later. Unfortunately, the complete proper physical analysis of thermal behaviour of buildings, based on the application of the principles of thermodynamics and appropriate constitutive relations for particular material layers and air in rooms, is not realistic because of the stochastic character

DOI: 10.21136/panm.2016.16

of physical processes, of the need of identification of a lot of material characteristics, of the barely predictable behaviour of users, as well as of the limited financial budgets of investors. The periodically updated list `www.buildingenergysoftwaretools.com` (formerly hosted by the US Department of Energy) contains about 500 relevant software tools; however, a physically transparent, intuitive, simple, robust and reliable numerical solver for the evaluation of energy consumption of a building, supporting the optimization of its design, too, is still not available.

The progress in this research area in the last 2 decades can be documented on [19], [15] and [22] and on numerous further references therein. Strong physical and geometrical simplifications are often overcome using some elements of soft computing by [17]: fuzzy sets, rough classifications, genetic algorithms, etc. However, some their assumptions, connecting the thermal analysis with the simplified one-dimensional study of electrical circuits, relying on the finite difference and lumped mass approaches, can be removed or modified, following the weak formulation and using the properties of the Fourier decomposition. In this paper we shall come from the consideration of a building as a thermal system, suggested in [20]; for the prediction and control of energy consumption for heating of rooms (including more references) cf. [12].

2. A model problem

Let us consider an open set (e. g. a domain) Ω in the 3-dimensional Euclidean space R^3, supplied by the Cartesian coordinate system $x = (x_1, x_2, x_3)$, with its boundary $\partial\Omega$ where the local vector of (formally outward) unit normal $n(x) = (n_1(x), n_2(x), n_3(x))$ can be introduced. The usual notation for the Hamilton operator $\nabla = (\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3)$ and for the Laplace one $\Delta = \nabla \cdot \nabla$, with the central dot used for the scalar product in R^3, will be used. Moreover, let us consider a time interval $I = [0, \tau]$ with some real positive τ; the limit passage $\tau \to \infty$ is not prohibited, the dot symbol is reserved for partial derivatives with respect to the time $t \in I$, whereas the prime symbol will mean the derivative with respect to the following variable everywhere. For the brevity of notation, in this short paper we shall assume an (at least macroscopically) isotropic material (not homogeneous in general) located in Ω; the straightforward generalization is left to the reader.

The conservation principle for a scalar quantity $w(x,t)$ on Ω and its flux $\eta(w(x,t))$, following [3], p. 5, and [10], p. 9, reads

$$\dot{\varepsilon}(w) + \nabla \cdot \eta(w) = \mathcal{F}(w) \tag{1}$$

where $\varepsilon(w(x,t))$ evaluates the non-stationary redistribution of $w(x,t)$ and $\mathcal{F}(w(x,t), x, t)$ the sources occurring on $\Omega \times I$. Some constitutive relations for η and ε are needed: here we shall introduce them in the (rather simple) forms

$$\dot{\varepsilon}(w) = \varepsilon'(w)\dot{w} = \kappa(w)\dot{w}\,, \tag{2}$$

$$\eta(w) = -\nabla\beta(w) = -\beta'(w)\nabla w = -\lambda(w)\nabla w \tag{3}$$

where β are certain a priori known functions of w (for their relation to the so-called enthalpy transform see [18], p. 252) and κ and λ are the material characteristics, in general functions of w again (but not of x and t explicitly), used in most engineering applications. We shall work with the unified boundary condition

$$\nabla\eta(w)\cdot n = \mathcal{G}(w)\,, \tag{4}$$

on $\partial\Omega$, containing the sources $\mathcal{G}(w(x,t),x,t)$; some special forms of both $\mathcal{F}$ and $\mathcal{G}$ will be introduced later. Let us remark that such definitions of $\mathcal{F}$ and $\mathcal{G}$ are open to handle relations between different quantities w, as temperature, velocity components or density, considered as reference variables in conservation principles for energy, momentum and mass.

Inserting only the 1st equation from (3) into (1), we obtain

$$\dot{\varepsilon}(w) - \Delta\beta(w) = \mathcal{F}(w)\,. \tag{5}$$

Applying the Kirchhoff transform, $\varepsilon(w)$ in (5) can be simplified to w, using certain modified variable w; for more details see [23], with several important references to [18]. In particular, for the constant values of κ and λ from (2) and (3) such transform is very easy. Unfortunately, in most other cases this makes namely (4) rather complicated and is not friendly to numerical approximations, thus we shall work with the original forms of (1), (2), (3) and (4) here. To solve w from such system of equation of evolution, the knowledge of initial values $w_0(x)$ of $w(x,t)$ with $t=0$ for any $x\in\Omega$ is also needed, i.e.

$$w(.,0) = w_0\,. \tag{6}$$

Using the Green - Ostrogradskiĭ theorem formally, we are able to convert (at least in the sense of distributions) (1), (2), (3) and (4) to

$$(v,\kappa(.)\dot{w}) + (\nabla v,\lambda(.)\nabla w) = (v,\mathcal{F}(.)) + \langle v,\mathcal{G}(.)\rangle \tag{7}$$

with any test function $v(x)$ from an appropriate function space introduced on Ω, scalar products $(.,.)$ on $L^2(\Omega)$ and $L^2(\Omega)^3$, or their generalizations in the sense of dualities in non-unitary spaces, similarly $\langle .,.\rangle$ on $L^2(\partial\Omega)$, etc. Then (7) can be understood as a weak (but physically transparent) condition for the conservation of a scalar quantity w.

A generalization of this approach to more complicated configurations, including a finite number of open sets Ω and a set of their mutual boundary parts, is straightforward, although not quite easy. We shall discuss this approach for the special case of the simplified evaluation of thermal transfer in buildings. In this case we are allowed to formulate only one principle of conservation of (thermal) energy (or enthalpy), with the (absolute) temperature w as the reference variable. A building is considered as a thermal system where all particular elements and subsystems (represented by Ω here) are coupled using the interface thermal fluxes from all admissible

adjacent elements and subsystems, as well as from the external environment. All terms (.) here, for the simplicity, refer to the possible dependence of related quantities on w; the significance of such dependence may be very different in practical engineering applications. In such reasonable model, $\mathcal{F}$ and $\mathcal{G}$, needed in (7), come:

i) $\mathcal{F}$ from the heat sources in rooms, controlled to satisfy requirements of technical standards (which can be quantified as an optimization problem or using some results from the control theory), together with certain well-being of inhabitants (whose quantification relies to expert decisions, soft-computing or advanced statistical approaches – cf. [14]) and ([7]),

ii) $\mathcal{F}$ from the obligatory air exchange in rooms, with similar limitations as i),

iii) $\mathcal{G}$ from the heat convection, driven by external temperature development in day and year quasi-cycles, expressible as

$$\mathcal{G}(w) = \alpha(.)(w_\times - w) \tag{8}$$

on $\partial\Omega$ where α (a function of w and $w_\times$ in general) is the transfer factor and $_\times$ refers to the external environment, which can be repeated literally for the transfer between rooms and separating structures, too,

iv) $\mathcal{G}$ from the heat radiation by the Stefan - Boltzmann law,

$$\mathcal{G}(w) = \sigma(.)(w_\times^4 - w^4) = \sigma(.)(w_\times - w)(w_\times + w)(w_\times^2 + w^2) \tag{9}$$

on $\partial\Omega$ where σ replaces α from iii) (its value is related usually to the theoretical one, derived for the unshaded perfectly black surface, utilizing some multiplicative correction factors),

v) $\mathcal{G}$ from the direct and diffusive solar heat radiation, due to the geographical location, slope and orientation of building surfaces, their (natural or artificial) shading, etc. – for more details see [12].

3. Fourier approximations

In general, using the Einstein convention for all sum indices $i, j, k \in \{1, 2, \ldots, n\}$, with $n \to \infty$ theoretically, we can seek for the evolution of w, utilizing the multiplicative Fourier decompositions, by [1], p. 215, and [6], p. 346, in the form

$$w(x, t) = \varphi_i(x) u_i(t) \tag{10}$$

where φ_i form a sufficiently rich system of functions defined on Ω (or on a union of a finite number of such open sets, as sketched above) approximating the needed space of all φ (e. g. the Sobolev space $W^{1,2}(\Omega)$ or some weighted space derived from it – see [18], p. 14) and u_i represent some (a priori unknown) functions of time. The choice of φ_i refers frequently to various finite element techniques – cf. [5], pp. 299, 323.

Assuming that some system of (fundamental) solutions of (7), i. e. those independent of (6)) U_{ik}, is available, we can rewrite (10) as

$$w(x,t) = \varphi_i(x) U_{ik}(t) c_k \tag{11}$$

with some unknown real constants c_k, satisfying (6) (at least) in the least squares (minimum variation) sense – cf. [4], p. 133. All test function in (7) can be then selected as $v = \varphi_j$. Consequently we have a system of linear algebraic equations

$$\left[(\varphi_j, \kappa(.)\varphi_i)\dot{U}_{ik} + (\nabla\varphi_j, \lambda(.)\nabla\varphi_i) U_{ik} \right] c_k = (\varphi_j, \mathcal{F}(.)) + \langle \varphi_j, \mathcal{G}(.) \rangle \,. \tag{12}$$

Let us notice that for practical computations the following heuristic linearization is useful: all values referenced by (.) should be estimated from the preceding time, as the initial choice for the algorithm of their successive improvement inside any time step. The proper mathematical and numerical analysis of relevant algorithms is needed; however, in this paper we shall sketch it only in one special case.

4. Thermal performance of buildings

For the computational analysis of thermal performance of buildings, applying the system approach, as introduced above, taking (11) and (12) to account, we can convert (7) to the matrix form

$$M\dot{u} + Ku = f + \dot{g} \tag{13}$$

where the real square matrices M and K come for two left-side additive term of (7), together with all appropriate parts of (8) by iii) and (9) by iv), the vector f is then generated by i), v) and the remaining terms from iii) and iv) and the vector g by ii); another supplement to iii) is allowed to come for the properly controlled artificial heating, to save the temperature in rooms at the level required by technical standards. Roughly speaking: M represents the thermal accumulation properties, K the thermal insulation ones, whereas $f(t)$ and $g(t)$ (namely their additive components) correspond to various types of time-variable heat sources. The best design of a vector of unknowns $u(t)$ seems to be that it refers to temperature values on a sufficiently rich set of points characterizing the building (typically for the application of Lagrange interpolation in the finite element method); its n components can be obtained from (10) in the form

$$(\varphi_i, \varphi_k) u_i(t) = (w(.,t), \varphi_k) \,. \tag{14}$$

Let us notice the close relation of (13) to the generalized eigenvalue problem

$$MV = V\Lambda \tag{15}$$

where Λ is a diagonal matrix of all eigenvalues and V contain all eigenvectors in its corresponding columns; both V and Λ can be positive and real, which depends

on the suitable (classical Fourier, finite element, wavelet, etc.) choice of the basis φ_i ($i \in \{1, 2, \dots n\}$). Consequently the method of variation of constants, coming from (15), in account with (6), gives the result

$$u(t) = V \exp(-\Lambda t) V^T M u_0 + V \int_0^t \exp(\Lambda(\tau - t)) V^T (f(\tau) + \dot{g}(\tau)) \, \mathrm{d}\tau \,. \tag{16}$$

In particular, for a finite positive time step h, to evaluate $u^1 \approx u(h)$ from u^0 derived from (14) and (6) (and similarly $u^2 \approx u(2h)$ from u^1, etc.), assuming that f and g are (for simplicity) linear splines defined on the set $\{0, h, 2h \dots\}$ with the accurate values $f^s = f(sh)$ and $g^s = g(sh)$ for $s \in \{0, 1, 2, \dots\}$, we come to the result, useful for the design of practical algorithms,

$$u^1 - V \exp(-\Lambda h) V^T M u^0 = V(I - \exp(-\Lambda h)) \cdot \tag{17}$$

$$\cdot \left(\Lambda^{-1} V^T \frac{g^1 - g^0}{h} - \Lambda^{-2} V^T \frac{f^1 - f^0}{h} \right) + V \Lambda^{-1} V^T f^1 - V \Lambda^{-1} \exp(-\Lambda h) V^T f^0 \,.$$

The Taylor expansion $I - \exp(-\Lambda h) \approx \Lambda h$, with the error $\mathcal{O}(h^2)$, applied to (17), leads to the Euler explicit scheme (unstable for real buildings and climatic records frequently), whereas the implicit one can be forced by the backward integration with the exchanged role of u^0 and u^1; for the comparison to the Crank - Nicholson, etc. schemes and the detailed stability and accuracy analysis see [2], pp. 639, 646. Here we intend to demonstrate namely the implementation of the influence of solar radiation to the computational algorithm based on (17) and the possibility of optimal control of artificial heating.

5. Solar radiation

Solar radiation can be handled using the surface thermal fluxes $\mathcal{G}$, contributing (in the relevant finite element or similar discretization – for more details see [23]) to f in (13), whose evaluation as the sum of beam (direct) and diffusive radiation intensities q_b and q_d comes from the analysis of mutual Sun – Earth positions and from the location and orientation of a building. The preliminaries for such calculations are explained in [21], [16] and [11]; their assessment to a real building object (due to its location, orientation, surface properties, etc.) relies on [15], the detailed analysis of this type is contained in [13], p. 47.

Following [15], let us introduce the surface reflexivity ρ, together with certain shading factors ρ_b and ρ_d. Let us also consider the geographical latitude ϕ, the azimuth angle γ (depending on the building orientation, related to south), and the angle β between the building surface and the horizontal plane. The evaluation day and year cycles of Sun – Earth positions requires the knowledge of declination $\delta(N) = 0.13027\dot{7}\pi \sin(2\pi(284 + N)/365)$ (presented here only for the standard year length of 365 days for simplicity) and of the day time angle $\omega(\xi) = \pi\xi$ for $\xi = (1 - (t - t_\times))/t_*$; $N \in \{1, \dots, 365\}$ refers to the day in the year, $t_\times$ to the

initial day time and t_* (formally) to the standard day length. Further astronomical corrections and generalizations of $\delta(N)$, $\omega(\xi)$, etc., are available, but not presented and implemented here, because of i) the substantial extension of computational formulae, ii) the lack of methods and input data for reasonable setting of corresponding physical characteristics. Neglecting such corrections and generalizations, the zenith angle ς comes from the relation

$$\cos\varsigma = \sin\phi\sin\delta(N) + \cos\phi\cos\delta(N)\cos\omega(\xi)\,, \tag{18}$$

whereas the real angle of incidence of sunrays ϑ needs to calculate

$$\begin{aligned}\cos\vartheta = \cos\varsigma &+ \sin\phi(\sin\delta(N)(\cos\beta-1) + \cos\delta(N)\sin\beta\cos\gamma\cos\omega(\xi)) \\ &+ \cos\phi(\cos\delta(N)(\cos\beta-1)\cos\omega(\xi) - \sin\delta(N)\sin\beta\cos\gamma) \\ &+ \cos\delta(N)\sin\beta\sin\gamma\sin\omega(\xi)\,.\end{aligned} \tag{19}$$

Making use of (18) and (19), together with the measurement results for q_b and q_d, we are finally able to evaluate

$$\begin{aligned}FLU\mathcal{G} =& \rho_b q_b \min(\cos\vartheta, 0)/\min(\cos\varsigma, 0.087) + \rho_d q_d(1+\cos\vartheta)/2 \\ &+\ \rho(q_b+q_d)(1-\cos\vartheta)/2\,.\end{aligned} \tag{20}$$

The above sketched model of solar radiation does not violate the formal linearity of (13). Up to now, we have relied on the small changes of $\kappa(.)$, $\lambda(.)$, $\mathcal{F}(.)$ and $\mathcal{G}(.)$ in (7), with the acceptable values taken from the previous time step in practical calculations, following (13), with some potential improvements a posteriori. Nevertheless, the strongly nonlinear heat radiation by (9) should be discussed and handled separately. In more details: f^0 and f^1 in (13) can be written as $\widetilde{f}^0 + S(u^0)^4$ and $\widetilde{f}^1 + S(u^1)^4$ where $(.)^4$ means the 4^{th} power of all components of $(.)$ (later also $(.)^3$ the 3^{rd} power), S is a sparse diagonal matrix and $\widetilde{f}^0$ and $\widetilde{f}^1$ are independent of u^0 and u^1. The principal difficulty is the poor knowledge of u^1 a priori (unlike u^0); only some initial estimate u^1_* is available, e. g. $u^1_* \approx u^0$. For the brevity of notation, let $\widetilde{u}^1$ be the result (instead of u^1) obtained from (13) with the correct f^0, but with f^1 replaced by $\widetilde{f}^1$. Consequently we have

$$u^1 = \widetilde{u}^1 + PS(u^1)^4\,. \tag{21}$$

where $P = V\Lambda^-1V^T - V(I-\exp(-\Lambda h))\Lambda^{-2}V^T/h$.

The exact Newton method, as derived in [8], pp. 101, 144, applied to (21), gives

$$(I - 4PS(u^1_*)^3)(u^1 - u^1_*) = \widetilde{u}^1 - u^1_* + PS(u^1_*)^4\,, \tag{22}$$

which can be repeated (with u^1_* replaced by u^1) in the iteration procedure (if needed, to reach the required accuracy). However, in most application cases the additional term PSu^1 is not dominating on the right side of (21), consequently the first estimate u^1_* on the left side of (22) can be taken as fixed, which slightly disturbs the Newton iteration process, but improves the algorithmic efficiency.

6. Heating control

Assuming (for simplicity) the constant temperature in rooms in each time, the conservation of (at least) prescribed temperature level $\bar{u}^s$ ($s \in \{1, 2, \ldots\}$), e. g. of selected elements of u from (13) corresponding (using the transform (14)) just to rooms with some heating equipments, is required. However, the heating power of these equipments $\mathcal{Q}^s$, valid for $(s-1)h < t \leq sh$, is limited, i. e. $0 \leq \mathcal{Q}^s \leq \mathcal{Q}^{\max}$ for some upper bounds $\mathcal{Q}^{\max}$ for particular rooms (which is a vector of much lesser length than n in practice). Such upper bounds are design parameters of a building, due to the volume, location and assumed exploitation of rooms.

Consequently we have to distinguish between 4 following artificial heating regimes: 1) no heating during the summer season, 2) inactive heating for sufficient temperature outside the summer season, i. e. $u^s \geq \overline{u}^s$, 3) normal heating: setting of $\mathcal{Q}^s$ is needed, 4) insufficient heating for $\mathcal{Q}^s \geq \mathcal{Q}^{s\,\max}$ by 3) when the reduction of $\mathcal{Q}^s$ to $\mathcal{Q}^{\max}$ is necessary. Clearly, our principal aim is to evaluate $\mathcal{Q}^s$ for the regime 3); the modifications for all other regimes are obvious.

Let us compose W from the lines of the matrix $V(I - \exp(-\Lambda h))\Lambda^{-1}V^T$ corresponding to the temperature in rooms with artificial heating, and take $\mathcal{Q}$ as certain additional $\dot{g}$ from artificial heating in (13). Using the least squares technique, we are able to evaluate $\mathcal{Q}^s$ from the rather simple formula

$$W^T W \mathcal{Q}^s = W^T(\widehat{u}^s - \overline{u}) \tag{23}$$

where $\widehat{u}^s$ corresponds to the hypothetical regime 1), active for $(s-1)h < t \leq sh$ (thus $\widehat{u}^s$ can be evaluated by (17) directly), and $\overline{u}$ is related (using (14 again) to some prescribed temperature level $\overline{w}$, time-independent here for simplicity (but allowed to have different values in particular rooms, due to the "thermal stability" requirements in modern technical standards, unlike the classical "passive house" evaluation by [9]). We can see that the system of linear algebraic equations (23) is small, thus its solution should be inexpensive; moreover no further inequalities occur in the computational algorithm.

In general, the Newton iterations by (21), including its above discussed modification, disturb the direct evaluation of $\mathcal{Q}^1, \mathcal{Q}^2, \ldots$ from (23). Fortunately, no additional iterative tricks are needed in most application cases; the corresponding modifications of $\mathcal{Q}^1, \mathcal{Q}^2, \ldots$ due to $u^1, u^2, \ldots$ including the heat radiation can be incorporated into the inexact Newton process, as sketched in the preceding section.

7. Illustrative example

Our illustrative example presents the thermal analysis of a simple model building, considered as one great room, supplied by building enclosure with particular constructive and insulation layers, located near Brno (Czech Republic); the more complex example of a real building structure will be referenced in *Conclusion*. All calculations are based on the one-dimensional model simplification, coming from the

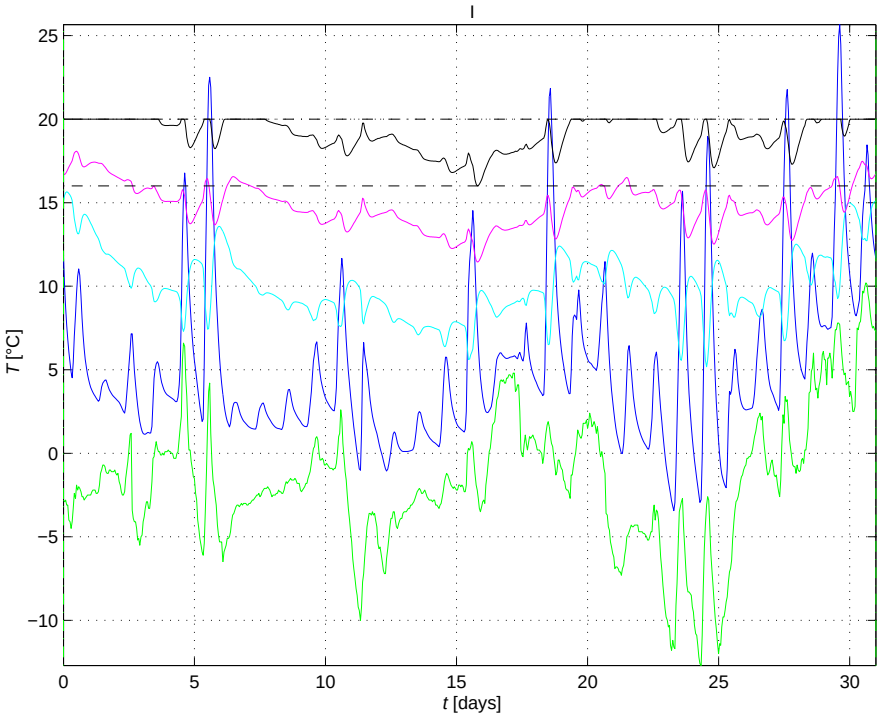


Figure 1: Temperature in a room and structures in January of the 1st year.

linearized heat conduction (no air flow is allowed) and on the application of the finite element method with linear splines as basis functions, with the original software implementation inside the MATLAB environment. Thus it is useful to compare our model with the approach of [22], based on the analogy with the analysis of electrical RC-circuits; a relevant model circuit is described in [15], p. 447, and [22], p. 156, in all details. The "lumped masses" by [22] generate diagonal matrices M (in our notation); here we come to (slightly more general) sparse matrices M, with technical details explained in [13], p. 62.

The following meteorological data in the time steps $h = 1$ hour, averaged to the periodical form of the so-called reference year, from the international airport Brno - Tuřany are available: i) the environmental temperature, ii) the total solar radiation intensity, iii) the diffusive component of ii). Such data are sufficient to perform all above sketched evaluations, including the annual energy consumption, crucial for the thermal design of a building. The heating is insufficient, with the aim of proper testing of switching between regimes 1), 2), 3), 4). No summer shading (or air conditioning) is installed, to force the unwanted summer increase of temperature in rooms. Three figures document the above sketched direct calculation procedure, starting from the hypothetical inside temperature $u^0 = 20\,°\mathrm{C}$ everywhere. The same minimum temperature level is assumed to be satisfied during the whole year. Visible assertion of the thermal radiation due to the Stefan - Boltzmann law is limited to the slight temperature decrease in some nights.

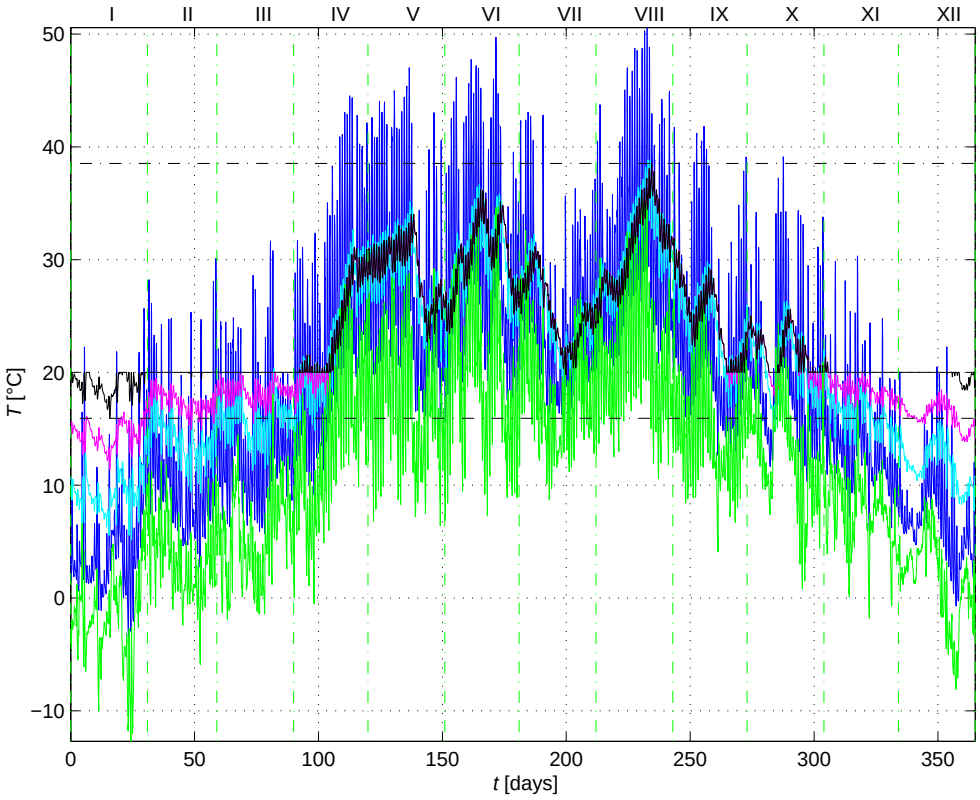


Figure 2: Temperature in a room and structures during the 3rd year.

Figure 1 shows the development of temperature in selected locations of constructive and insulation layers of walls, floors and roof (here just in selected layer centers and interfaces) in the first month (January) of the first calculation year. Namely the lower oscillating curve represents the environmental temperature, the curve rather close to the required $u^0 = 20\,°\mathrm{C}$. The non-negligible decrease of indoor temperature, under $u^0 = 20\,°\mathrm{C}$ corresponds to the strange design of a building for software testing purposes, as introduced above.

The climatic input data are available just for one reference year in practice, thus for simulations considering more years they have to be applied as periodic ones, with the period of 1 year. Apart from the artificial initial condition, one can expect the nearly periodic development of temperature after a small number of years. Figure 2 presents the same computational results as Figure 1, but for the (less synoptic) development of temperature in the third year (the results for the fourth year are nearly the same). The summer overheating of rooms, in some cases even over the prescribed environmental temperature (the maximum inside temperature exceeding 38 °C occurs in August, which corresponds to the temperature on the outer building surface, oscillating up to 1 °C) is caused by the effect of solar radiation, whereas the same radiation in winter, in the direction close to the normal one to most walls and windows, helps to reduce the effect of the insufficient heating. Such idea comes

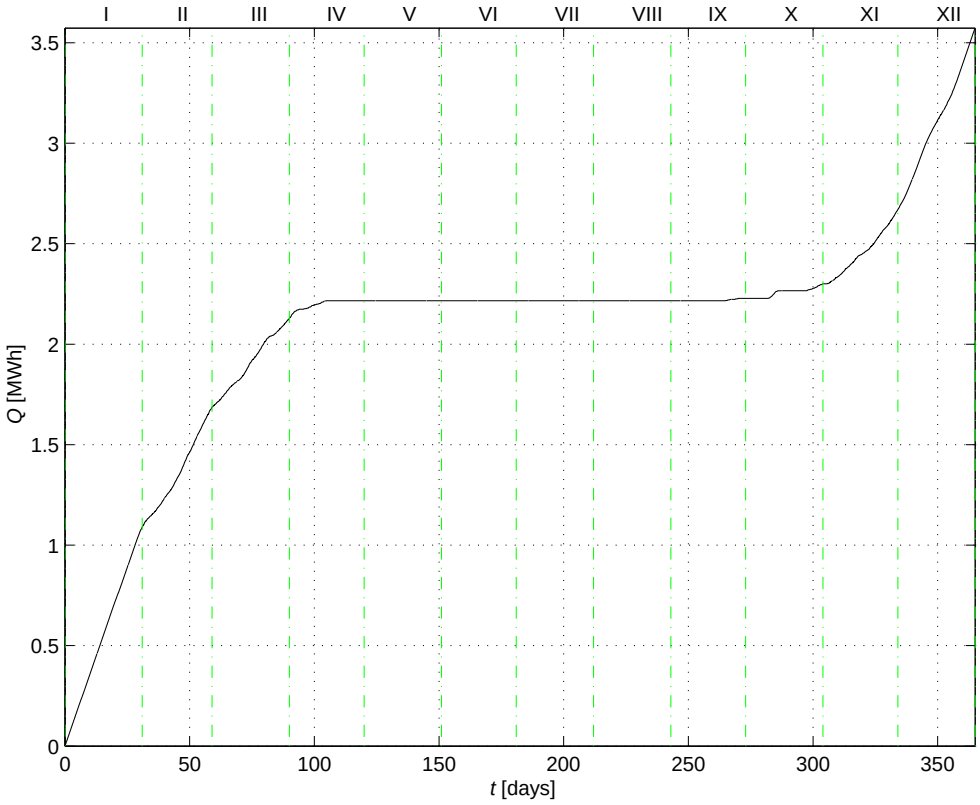


Figure 3: Total consumption of energy for artificial heating during the 3rd year.

even from the ancient literature (Aischylos, Socrates); however, the implementation of some active cooling system (air-conditioning) or at least a passive one (shading) should be expected in modern buildings. More extensive comparative calculations with 34 relevant figures and with the proper discussion of above sketched physical processes are presented in [13], p. 68, including certain identification and optimization considerations.

Finally Figure 3 shows the total consumption of energy, corresponding to Figure 2. Its seemingly regularized increase, with nearly invisible day quasi-cycles, reflects the accumulation ability of applied building materials and structures, as well as the reasonable control of artificial heating. The similar figure corresponding to Figure 1 degenerates just to a linearly increasing function (because of winter climatic conditions together with the low heating power).

8. Conclusion

We have demonstrated how the Fourier analysis can contribute to the development of computational algorithms of thermal transfer and related phenomena, including the simplified system analysis of thermal performance of buildings, namely the simulation of energy consumption of building structures, respecting the significant effect of solar radiation and the proper control of heating equipments. The

corresponding software in MATLAB is still in development; some more references to preliminary results can be found in [12]. The application of this software to various classes of real buildings and the development of further optimization tools belong to the research priorities for the near future.

The robust and inexpensive numerical solver in MATLAB, working with linear algebra, vector operations and spectral decompositions, as sketched under (13), seems to be promising solution for the support of optimization of buildings, both residential and industrial ones, as for the cooling and freezing plants, for the advanced buildings with green roofs and walls (for their practical realization in Vienna see [7]), etc. For example, the evaluation of temperature development and the heating power requirements for the low-energy house in Ostrov u Macochy (35 km northern from Brno, some heuristic corrections of climatic data from Brno are needed), with 4 rooms and 26 planar interfaces with the carefully described inner structure, shows the good coincidence between the simulation results and the recorded energy consumption (time series compound from 5 years); for much more details see [13], p. 94. The time step can be still $h = 1$ hour, whereas the requirement of stability of the explicit Euler method (in the linearized case) would force $h < 161,63$ s.

The formal derivation of all convergence properties of implemented algorithms, regardless from their validation from engineering practice, is not trivial, just in the case of iterations caused by strong nonlinearities and optimization steps; the proper analysis are assumed to be prepared for (at least one) another, more extensive paper. The same is true for optimization problems, namely for the application of the Nelder - Mead method, using selected functions of the *optimization* toolbox from MATLAB, to the support of design of building objects by [13], p. 89, working with the above mentioned reference object again.

Acknowledgements

This research has been supported by the project of specific university research of Brno University of Technology, No. FAST-S-16-3385, in cooperation with the project LO1408 AdMaS UP (Advanced Materials, Structures and Technologies, Ministry of Education, Youth and Sports of the Czech Republic, National Sustainability Programme I).

References

[1] Barták, J., Herrmann, L., Lovicar, V., and Vejvoda, O.: *Partial differential equations of evolution*. Ellis Horwood, 1991.

[2] Baker, A. and Graves-Morris, D.: *Padé approximants*. Cambridge University Press, 2010.

[3] Bermúdez de Castro, A.: *Continuum termodynamics*. Birkhäuser, 2005.

[4] Bochev, P. B. and Gunzburger, M. D.: *Least-square finite element methods*. Springer, 2009.

[5] Brenner, S. C. and Ridgway Scott, L.: *The mathematical theory of finite element methods*. Springer, 2002.

[6] Brigola, R.: *Fourier-analysis und distributionen*. Co-Verlag, 2012.

[7] Burdajewicz, F., Korjenic, A., and Bednar, T.: Bewertung und Optimierung von dynamischen Dämmsystemen unter Berücksichtigung des Wiener Klimas. *Bauphysik* **33** (2011), 49–58.

[8] Drábek, P. and Milota, J.: *Methods of nonlinear analysis – application to differential equations*. Springer, 2013.

[9] Feist, W.: *Gestaltungsgrundlagen Passivhäuser*. Das Beispiel, Darmstadt, 1999.

[10] Ferziger, J. H. and Periç, M.: *Computational methods for fluid dynamics*. Springer, 2002.

[11] Gliah, O., Kruczek T., and Thibault, J.: The effective sky temperature: An enigmatic concept. Heat and Mass Transfer **47** (2011), 1171–1180.

[12] Jarošová, P. and Vala, J.: New approaches to the thermal design of energy saving buildings. Advanced Materials Research **1126** (2015), 174–180.

[13] Jarošová, P.: *Optimization of selected parameters for the design of objects with low energy demand.* (In Czech.) Dissertation thesis, Brno University of Technology, submitted 2017.

[14] Kala, Z.: Sensitivity and reliability analysis of lateral-torsion buckling resistance of steel beams. Archives of Civil and Mechanical Engineering **15** (2015), 1098–1107.

[15] Kämpf J. H. and Robinson, D.: A simplified thermal model to support analysis of urban resource flows. Energy and Buildings **39** (2007), 445–453.

[16] Nayak, J. K. and Sukhatme, S. P.: *Principles of thermal collection and storage*. Tata McGraw Hill, 2008.

[17] Pacheco-Vega, A.: *Soft computing applications in thermal energy systems*. Springer, 2011.

[18] Roubíček, T.: *Nonlinear partial differential equations with applications*. Birkhäuser, 2005.

[19] Škrjanc, I., Župančič, B., Furlan, B., and Kainer, A.: Theoretical and experimental fuzzy modelling of building thermal dynamic response. Building and Environment **36** (2001), 1023–1038.

[20] Šťastník, S., and Vala, J.: On the thermal stability in dwelling structures. Building Research Journal **52** (2004), 31–55.

[21] Sukhatme, S. P.: *Solar energy*. Tata McGraw Hill, 1996.

[22] Underwood J. C.: An improved lumped parameter method for building thermal modelling. Energy and Buildings **79** (2014), 191–201.

[23] Vala, J.: Computational approaches to some inverse problems from engineering practice. *Programs and Algorithms of Numerical Mathematics* 17 in Dolní Maxov (Czech Republic, 2014), Institute of Mathematics CAS, Prague 2015, 215–230.

[24] White, R. E.: *Elements of matrix modeling and computing with MATLAB*. Chapmann & Hall / CRC, 2005.

[25] Directive 2010/31/EU of the European parliament and of the council on the energy performance of buildings. *Official Journal of the European Union* **L 153/13**, 2010.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

ON FINITE ELEMENT APPROXIMATION OF FLOW INDUCED VIBRATION OF ELASTIC STRUCTURE

Jan Valášek[1], Petr Sváček[1], Jaromír Horáček[2]

[1] Faculty of Mechanical Engineering, CTU in Prague
Karlovo nám. 13, Praha 2, 121 35, Czech Republic
valasek.jan@volny.cz, petr.svacek@fs.cvut.cz
[2] Institute of Thermomechanics, The Czech Academy of Sciences
Dolejškova 5, 182 00 Praha 8, Czech Republic
jaromirh@it.cas.cz

Abstract: In this paper the fluid-structure interaction problem is studied on a simplified model of the human vocal fold. The problem is mathematically described and the arbitrary Lagrangian-Eulerian method is applied in order to treat the time dependent computational domain. The viscous incompressible fluid flow and linear elasticity models are considered. The fluid flow and the motion of elastic body is approximated with the aid of finite element method. An attention is paid to the applied stabilization technique. The whole algorithm is implemented in an in-house developed solver. Numerical results are presented and the influence of different inlet boundary conditions is discussed.

Keywords: stabilized finite element method, 2D Navier-Stokes equations, vocal folds, aeroelasticity

MSC: 65M60, 35Q30, 76D05

1. Introduction

The flow induced vibration of elastic structure or more generally fluid-structure interaction problems (FSI) are important in many technical applications, see e.g. [3]. This contribution focuses on the simulation of human vocal folds vibration, see e.g. [5]. There are many numerical methods concerned with the solution of the fluid-structure interaction and also many approaches how to deal with the coupled problem. Basically these can be characterized as either the monolithic or partitioned scheme, see for example [7]. Monolithic solvers are usually more robust, but more computer time consuming. The partitioned scheme decouples the solution of both subproblems, but on the other hand it often requires subiterations.

This paper focuses on the application of the partitioned scheme for finite element method (FEM). To avoid possible instabilities of FEM due to high Reynolds

DOI: 10.21136/panm.2016.17

number flows the streamline upwind/Petrov-Galerkin (SUPG) method, pressure-stabilization/Petrov-Galerkin (PSPG) method together with 'div-div' stabilization are applied.

The structure of the paper is as follows. First the mathematical model consisting of the Navier-Stokes and linear elasticity equations is presented and the arbitrary Lagrangian-Eulerian method (ALE) is used. Further the flow problem is discretized in space by the stabilized finite element method. The numerical results of several test cases are shown.

2. Mathematical model

For the sake of simplicity the FSI problem is solved in 2D. The geometry of the problem topology is shown in Figure 1. The elastic structure (the vocal fold) is represented by the domain Ω^s. It is not necessary to distinguish between the shape at an arbitrary time t and the reference shape of the domain because of the adopted Lagrange approach for the deformation description.

The domain Ω^f_{ref} denotes the reference fluid domain, e.g. the domain at the time instant $t = 0$ with the common interface $\Gamma_{\mathrm{W_{ref}}} = \Gamma_{\mathrm{W_0}}$ between the fluid and structure domains. The fluid motion is solved with the ALE method, which enables the change of the reference domain Ω^f_{ref} to the domain Ω^f_t at any time instant t.

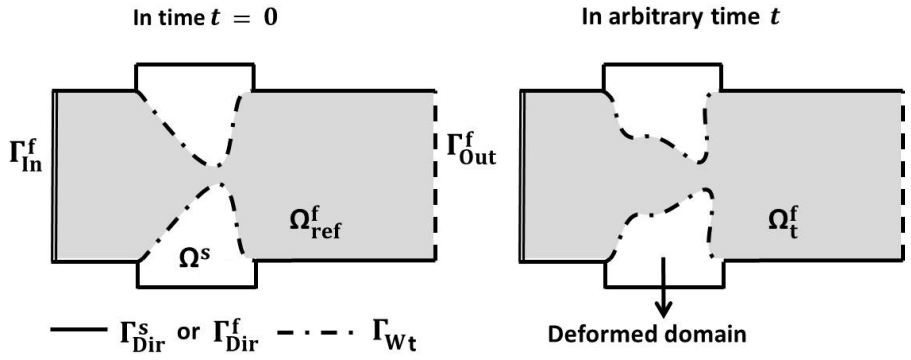


Figure 1: Geometry of vocal folds model with boundaries marked before (left) and after (right) deformation.

Elastic body. The deformation of the elastic body Ω^s_{ref} depends on establishing of dynamic equilibrium between the inertia forces and the applied surface and volume forces. This equilibrium is described by the partial differential equation, see e.g. [2]

$$\rho^s \frac{\partial^2 u_i}{\partial t^2} - \frac{\partial \tau^s_{ij}}{\partial x_j} = f^s_i \quad \text{in } \Omega^s \times (0, \mathrm{T}), \tag{1}$$

where $\mathbf{u}(X, t) = (u_1, u_2)$ denotes the displacement vector, f^s_i is the component of volume force, ρ^s is the structure density and τ_{ij} are the components of the Cauchy

stress tensor. These components are expressed by the generalized Hooke's law. Assuming the isotropic material and small displacements the components of the stress tensor $\boldsymbol{\tau}^s = (\tau_{ij}^s)$ are given by

$$\tau_{ij}^s = \lambda^s (\text{div } \mathbf{u})\, \delta_{ij} + 2\mu^s e_{ij}^s(\mathbf{u}), \tag{2}$$

where δ_{ij} is Kronecker's delta, $e_{jk}^s(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_k} + \frac{\partial u_k}{\partial x_j}\right)$ is the strain tensor, and λ^s, μ^s are Lame's constants related to the Young modulus of elasticity and Poisson's ratio. The partial differential equation (1) is enclosed with the following initial and boundary conditions

$$\begin{aligned}
&\text{a)} && \mathbf{u}(X,0) = \mathbf{u}_0(X), && \text{for } X \in \ \Omega^s, \\
&\text{b)} && \frac{\partial \mathbf{u}}{\partial t}(X,0) = \mathbf{u}_1(X) && \text{for } X \in \ \Omega^s, \\
&\text{c)} && \mathbf{u}(X,t) = \mathbf{u}_{\text{Dir}}(X,t) && \text{for } X \in \Gamma^s_{\text{Dir}},\ t \in (0,\text{T}), \\
&\text{d)} && \tau_{ij}^s(X,t)\, n_j^s(X) = q_i^s(X,t), && \text{for } X \in \Gamma^s_{\text{W}_{\text{ref}}},\ t \in (0,\text{T}),
\end{aligned} \tag{3}$$

where the $\Gamma_{\text{W}_{\text{ref}}}, \Gamma^s_{\text{Dir}}$ are mutually disjoint parts of the boundary $\partial\Omega = \Gamma_{\text{W}_{\text{ref}}} \cup \Gamma^s_{\text{Dir}}$ (see Figure 1) and $n_j^s(X)$ are components of the unit outer normal to $\Gamma_{\text{W}_{\text{ref}}}$.

ALE method. The ALE method uses a diffeomorphism A_t of the reference (undeformed) domain Ω^f_{ref} onto the time-dependent domain Ω^f_t at any time instant $t \in (0,\text{T})$. This mapping A_t is also required to satisfy

$$\frac{\partial A_t}{\partial t} \in C(\Omega^f_{ref}), \quad A_t(\partial\Omega^f_{ref}) = \partial\Omega^f_t, \quad t \in (0,\text{T}). \tag{4}$$

The ALE domain velocity is then defined by

$$\mathbf{w}_D(x,t) = \frac{\partial}{\partial t} A_t(X), \quad t \in (0,\text{T}),\ x = A_t(X) \in \Omega^f_t. \tag{5}$$

Afterwards ALE derivative is introduced as the time derivative with respect to a fixed point $X \in \Omega^f_{ref}$. The ALE derivative satisfies

$$\frac{D^A}{Dt} f(x,t) = \frac{df(A_t(X),t)}{dt} = \frac{\partial f}{\partial t}(x,t) + \mathbf{w}_D(x,t) \cdot \nabla f(x,t). \tag{6}$$

For more details see the article [9]. The practical construction of ALE mapping is described in [5] or [10].

Fluid motion. The fluid is assumed to be viscous and incompressible in the time dependent domain Ω^f_t. Its motion is modelled by the Navier-Stokes equations in the ALE form

$$\frac{D^A \mathbf{v}}{Dt} + ((\mathbf{v} - \mathbf{w}_D) \cdot \nabla)\mathbf{v} - \nu^f \Delta \mathbf{v} + \nabla p = \mathbf{0}, \quad \text{div } \mathbf{v} = 0 \quad \text{in } \Omega^f_t, \tag{7}$$

where $\mathbf{v}(x,t)$ denotes the fluid velocity, p is the kinematic pressure and ν^f is the kinematic fluid viscosity, see [5].

The problem (7) is equipped with an initial and the following boundary conditions

$$\begin{aligned}
&\text{a)} && \mathbf{v}(x,t) = \mathbf{w}_D(x,t) && \text{for } x \in \Gamma^f_{\mathrm{Dir}} \cup \Gamma_{\mathrm{W_t}},\ t \in (0,\mathrm{T}),\\
&\text{b)} && \mathbf{v}(x,t) = \mathbf{v}_{\mathrm{Dir}}(x,t) && \text{for } x \in \Gamma^f_{\mathrm{In}},\ t \in (0,\mathrm{T}),\\
&\text{c)} && p(x,t)\mathbf{n}^f - \nu^f \frac{\partial \mathbf{v}}{\partial \mathbf{n}^f}(x,t) = -\frac{1}{2}\mathbf{v}(\mathbf{v}\cdot\mathbf{n}^f)^- && \text{for } x \in \Gamma^f_{\mathrm{Out}},\ t \in (0,\mathrm{T}),
\end{aligned} \tag{8}$$

where $\mathbf{n}^f$ is unit outer normal to boundary $\partial\Omega^f_t$. The last condition (8 c) is the modified do-nothing boundary condition according to [1], which suppresses possible backward inlet through the outlet boundary.

Coupling. The solutions of problems (1) and (7) depend on each other via the boundary conditions on the common interface. Moreover, the location of the interface $\Gamma_{\mathrm{W_t}}$ at time t is not a priori known and is dependent on the establishing force equilibrium between the aerodynamic and the elastic forces. It is implicitly given by the deformation $\mathbf{u}$ as

$$\Gamma_{\mathrm{W_t}} = \left\{ x \in \mathbb{R}^2 \,|\, x = X + \mathbf{u}(X,t),\ X \in \Gamma_{\mathrm{W_{ref}}} \right\}. \tag{9}$$

First the so called dynamic boundary condition expressing effect of aerodynamic force $\mathbf{q}^s = (q^s_1, q^s_2)$ is prescribed at the interface for the structure, where

$$q^s_i(X,t) = -\sum_{j=1}^{2} \rho^f \left(-p\delta_{ij} + \nu^f \left(\frac{\partial \mathbf{v}_i}{\partial x_j} + \frac{\partial \mathbf{v}_j}{\partial x_i}\right)\right) n^f_j \Big|_{x = X + \mathbf{u}(X,t)}. \tag{10}$$

Further, the coupling of the problem is characterized by the so called kinematic boundary condition (8a), where the domain velocity $\mathbf{w}_D$ is equal to the structure velocity at the interface $\Gamma_{\mathrm{W_t}}$. In order to solve the problem, the strong coupling algorithm is implemented, see e.g. [5].

3. Discretization

Both parts of the FSI problem were discretized in space by the finite element method. For the time discretization the equidistant time step $\Delta t = \frac{T}{N}$, $N >> 1$ was used. The partition of the time interval is then given by $t_n = n\Delta t$. The functions $\mathbf{u}, \mathbf{v}, p$ are approximated at time instant t_n by $\mathbf{u}^n, \mathbf{v}^n, p^n$.

Elastic body. First, equation (1) is reformulated in a weak sense, the generalized Hooke's law (2) and the Green theorem is applied, which leads to

$$\left(\rho^s \frac{\partial^2 u_j}{\partial t^2}, \varphi_j\right)_{\Omega^s} + \left(\lambda^s (\mathrm{div}\ \mathbf{u})\,\delta_{ij} + 2\mu^s e^s_{ij}(\mathbf{u}), e^s_{ij}(\boldsymbol{\varphi})\right)_{\Omega^s} = \left(f^s_j, \varphi_j\right)_{\Omega^s} + \left(q^s_j, \varphi_j\right)_{\Gamma^s_{\mathrm{Neu}}}. \tag{11}$$

This equation needs to be satisfied for all $\boldsymbol{\varphi} = (\varphi_1, \varphi_2) \in V \times V$, where $V = \{\phi \in H^1(\Omega^s) | \phi = 0 \text{ on } \Gamma^s_{\text{Dir}}\}$ and $H^1(\Omega)$ is the Sobolev's space. The notation $(\cdot,\cdot)_M$ denotes scalar product in the space $L^2(M)$. The numerical solution $\mathbf{u}_h$ is now sought in the finite dimensional FE space, i.e. it can be expressed as the linear combination of basis functions $\mathbf{u}_h(x,t) = \sum_{i=1}^{N_h} \alpha_i(t)\boldsymbol{\varphi}_i(x)$, where the coefficients $\boldsymbol{\alpha}(t) = (\alpha_i(t))$ are unknowns. Then fulfilment of equation (11) leads to the system of ordinary differential equations of the second order

$$\mathbb{M}\ddot{\boldsymbol{\alpha}}(t) + \mathbb{C}\dot{\boldsymbol{\alpha}}(t) + \mathbb{K}\boldsymbol{\alpha}(t) = \mathbf{b}(t), \tag{12}$$

where the matrix $\mathbb{C}$ was added as a model of the proportional structural damping, see e.g. [5]. The vector $\mathbf{b}(t)$ has components $b_j(t) = \left(\mathbf{f}^s, \boldsymbol{\varphi}_j\right)_{\Omega^s} + \left(\mathbf{q}^s, \boldsymbol{\varphi}_j\right)_{\Gamma^s_{\text{Neu}}}$ and the elements of matrices $\mathbb{M} = (m_{ij}), \mathbb{K} = (k_{ij})$ are given by

$$m_{ij} = \left(\rho^s \frac{\partial^2 \boldsymbol{\varphi}_i}{\partial t^2}, \boldsymbol{\varphi}_j\right)_{\Omega^s}, \qquad k_{ij} = \left(\lambda^s (\text{div } \boldsymbol{\varphi}_i)\, \delta_{rl} + 2\mu^s e^s_{rl}(\boldsymbol{\varphi}_i), e^s_{rl}(\boldsymbol{\varphi}_j)\right)_{\Omega^s}. \tag{13}$$

The proportional damping matrix is chosen as $\mathbb{C} = \epsilon_1\mathbb{M} + \epsilon_2\mathbb{K}$ with appropriate choice of parameters ϵ_1, ϵ_2. This system is numerically approximated by the Newmark method, see e.g. [5].

Fluid motion. Equation (7) is first discretized in time by the backward differentiation formula of the second order (BDF2), see [10]. Furthermore, the nonlinear convective term is linearized using the value from the previous time step, $(\mathbf{v}\cdot\nabla)\mathbf{v}|_{t_{n+1}} \approx (\mathbf{v}^n \cdot \nabla)\mathbf{v}^{n+1}$. Afterwards the weak formulation is derived in the standard way, where on the outflow part of the boundary Γ^f_{Out} one extra application of Green theorem to the convective term according to [1] is performed. The functional spaces $\mathbf{X} = X \times X$, $X = H^1(\Omega^f)$ and $M = L^2(\Omega^f)$ are introduced. Then the solution of problem $V = (\mathbf{v}, p) = (\mathbf{v}^{n+1}, p^{n+1})$ is sought in the space $\mathbf{X} \times M$ such that $\mathbf{v}$ fulfills conditions (8 a) b)) and moreover

$$a(V, \Phi) = f(\Phi) \qquad \text{for all } \Phi = (\boldsymbol{\varphi}, q) \in \mathbf{W} \times L^2(\Omega^f), \tag{14}$$

where $\mathbf{W} = W \times W$, $W = \{\phi \in X | \phi = 0 \text{ on } \Gamma^f_{\text{Dir}} \cup \Gamma^f_{\text{In}} \cup \Gamma^f_{\text{W}_\text{t}}\}$. The bilinear form $a(\cdot,\cdot)$ and functional $f(\cdot)$ are defined as

$$\begin{aligned} a(V,\Phi) = &\left(\frac{3\mathbf{v}}{2\Delta t}, \boldsymbol{\varphi}\right)_{\Omega^f} + \frac{1}{2}((\overline{\mathbf{v}}^n - 2\mathbf{w}_D)\cdot\nabla)\mathbf{v}, \boldsymbol{\varphi})_{\Omega^f} - \frac{1}{2}((\overline{\mathbf{v}}^n\cdot\nabla)\boldsymbol{\varphi}, \mathbf{v})_{\Omega^f} + \\ &+\frac{1}{2}\left((\overline{\mathbf{v}}^n \cdot \mathbf{n}^f)^+ \mathbf{v}, \boldsymbol{\varphi}\right)_{\Gamma^f_{\text{Out}}} + \nu^f(\nabla\mathbf{v}, \nabla\boldsymbol{\varphi})_{\Omega^f} - (p, \text{div } \boldsymbol{\varphi})_{\Omega^f} + (q, \text{div}\,\mathbf{v})_{\Omega^f}, \\ f(\Phi) = &\left(\frac{4\overline{\mathbf{v}}^n - \overline{\mathbf{v}}^{n-1}}{2\Delta t}, \boldsymbol{\varphi}\right)_{\Omega_f}, \end{aligned} \tag{15}$$

where for a given time step t_{n+1} we set $\overline{\mathbf{v}}^i(x) = \mathbf{v}^i(A_{t_i}(A^{-1}_{t_{n+1}}(x)))$.

Stabilization of FEM. The instability arises primarily from unresolved high velocity gradients regions, which can be characterized by high values of local Reynold number Re_K. In order to overcome this phenomenon the streamline-upwind/Petrov-Galerkin method (SUPG) and pressure-stabilization method (PSPG) together with 'div-div stabilization' were applied, see [6]. The fully stabilized scheme is introduced with the additional terms added to equation (14) with shortened notation $\boldsymbol{\zeta} := ((\overline{\mathbf{v}}^n - \mathbf{w}_D) \cdot \nabla)\boldsymbol{\varphi} + \nabla q$

$$\mathcal{L}(V, \Phi) = \sum_{K \in \mathcal{T}_h} \delta_K \left(\frac{3\mathbf{v}}{2\Delta t} + ((\overline{\mathbf{v}}^n - \mathbf{w}_D) \cdot \nabla)\mathbf{v} + \nabla p - \nu \Delta \mathbf{v}, \boldsymbol{\zeta} \right)_K , \tag{16}$$

$$\mathcal{F}(\Phi) = \sum_{K \in \mathcal{T}_h} \delta_K \left(\frac{4\overline{\mathbf{v}}^n - \overline{\mathbf{v}}^{n-1}}{2\Delta t}, \boldsymbol{\zeta} \right)_K , \quad \mathcal{P}(V, \Phi) = \sum_{K \in \mathcal{T}_h} \tau_K \left(\nabla \cdot \mathbf{v}, \nabla \cdot \boldsymbol{\varphi} \right)_K ,$$

where parameters δ_K and τ_K are locally defined using local element length h_K as

$$\delta_K = \frac{h_K^2}{\tau_K}, \quad \tau_K = \nu \left(1 + Re^K + \frac{h_K^2}{\nu \Delta t} \right), \quad Re^K = \frac{h_K ||\overline{\mathbf{v}}^n - \mathbf{w}_D||_K}{2\nu}. \tag{17}$$

The stabilized problem now reads: find $V = (\mathbf{v}, p) \in \mathbf{X} \times M$ such that $\mathbf{v}$ fulfills conditions (8a)b)) and

$$a(V, \Phi) + \mathcal{L}(V, \Phi) + \mathcal{P}(V, \Phi) = f(\Phi) + \mathcal{F}(\Phi), \tag{18}$$

for all $\Phi = (\boldsymbol{\varphi}, q) \in \mathbf{W} \times L^2(\Omega^f)$. The numerical simulations were done using LBB stable P1-bubble/P1 elements. The solution of system (18) was performed by the mathematical library UMFPACK, see [4].

4. Numerical simulations

Numerical tests were performed on computational domain with vocal fold model M5 described in [8]. The height of the vocal fold was set to 6 mm.

Flow solver. First, the fluid flow through the fixed computational domain without interaction was computed. The inlet velocity was set to fully developed flow with maximum 1 m/s. The viscosity was set to $\nu^f = 1.47 \cdot 10^{-5}\,\mathrm{m/s^2}$ and the time step $\Delta t = 10^{-4}\,\mathrm{s}$ was chosen. Figure 2 shows a distribution of the flow velocity magnitude and pressure and illustrates the typical jet and vortex structures. Figure 3 shows the pressure difference between the inlet and the outlet computed both with (p_stab) and without stabilization (p_unstab). For this computation it was possible to use also the unstabilized FEM, the results are very similar.

FSI test. Subsequently, the coupled fluid-structure interaction problem in the same domain was solved. The vocal fold consists of two layers: The thin layer around the interface represents epithelium with Young modulus of elasticity and Poisson's

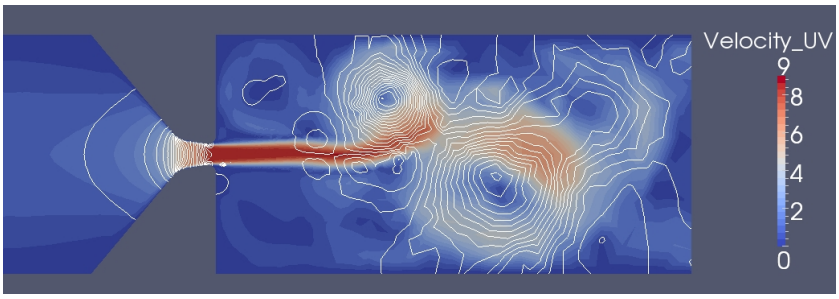


Figure 2: Flow field velocity together with pressure isolines are shown at the time instant $t = 0.3105\,\mathrm{s}$ for the case of the unstabilized FEM computation.

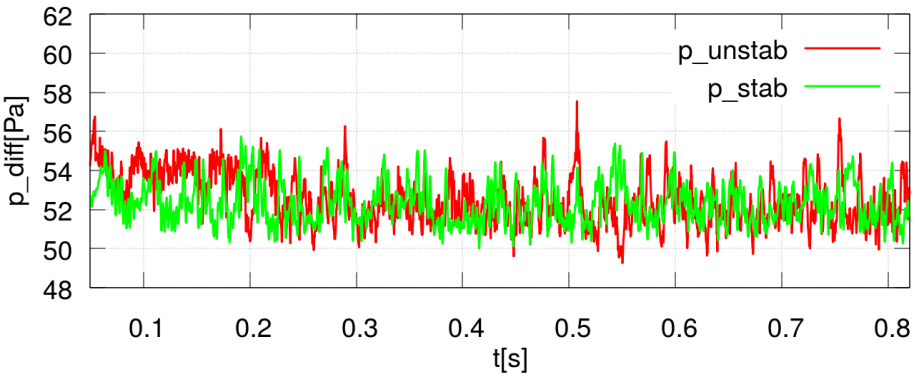


Figure 3: Pressure difference between the mean value on the inlet and the outlet.

ratio $E^s = 100\,\mathrm{kPa}$ and $\sigma = 0.4$, respectively. The rest of the vocal fold (muscle) is modelled with $E^s = 12\,\mathrm{kPa}$, $\sigma = 0.4$. The densities were set as $\rho^s = 1000\,\mathrm{kg/m^3}$, $\rho^f = 1.185\,\mathrm{kg/m^3}$ and damping parameters as $\epsilon_1 = 5.0\,\mathrm{s^{-1}}, \epsilon_2 = 2.0 \cdot 10^{-5}\,\mathrm{s}$. The eigenfrequencies and the eigenmodes were determined by the modal analysis. The first two are shown in Figure 4. They are the most important because the first mode represents motion dominantly in x-direction, whereas the second in the y-direction. The higher eigenfrequencies have a more complex shape of eigenmodes.

The same problem was solved either with the prescribed inlet velocity (parabolic profile with maximum $3\,\mathrm{m/s}$ - BC_velocity) or with the prescribed correspondent pressure difference ($272.55\,\mathrm{Pa}$ - BC_pressure). The interaction between elastic body and fluid flow was enabled after $0.1\,\mathrm{s}$ of computation, when the flow field was already fully developed. The numerically simulated displacements of one chosen node at the top of the bottom vocal fold are plotted in Figures 5 and 6 on the left and the Fourier transforms of the time signal are shown on the right.

After a transient part of the simulation, that corresponds to a sudden loading of the vocal folds by aerodynamic forces at the start of the interaction, the vocal folds vibrate with similar amplitudes in both cases around a new equilibrium point. The Fourier transform indicates the excitation of the first two eigenmodes. The first

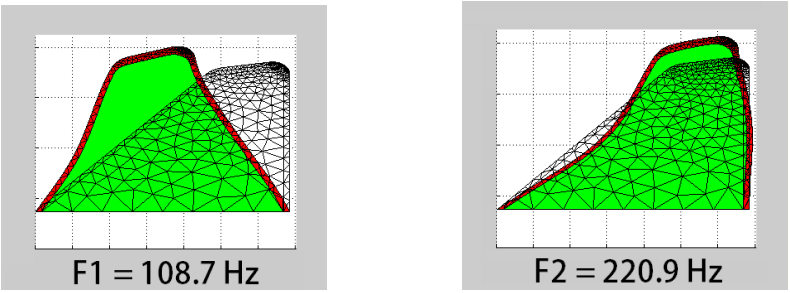


Figure 4: First and second eigenmode of vocal fold vibration with the correspondent eigenfrequencies.

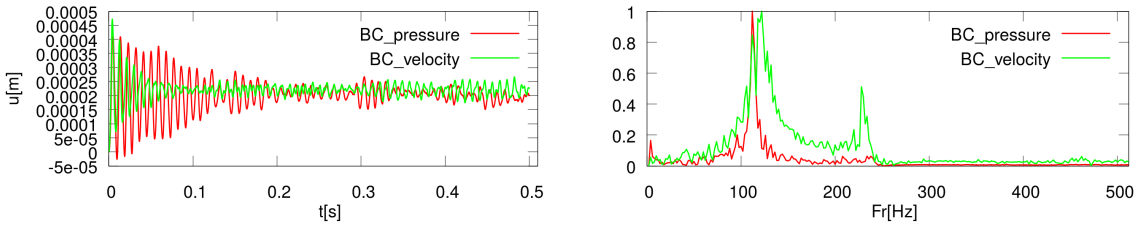


Figure 5: Simulated displacements of the point from the top of vocal fold in x-direction in time domain (left) and the normalized Fourier transformation of the signals (right). Time signal is plotted from the start of computation with the interaction.

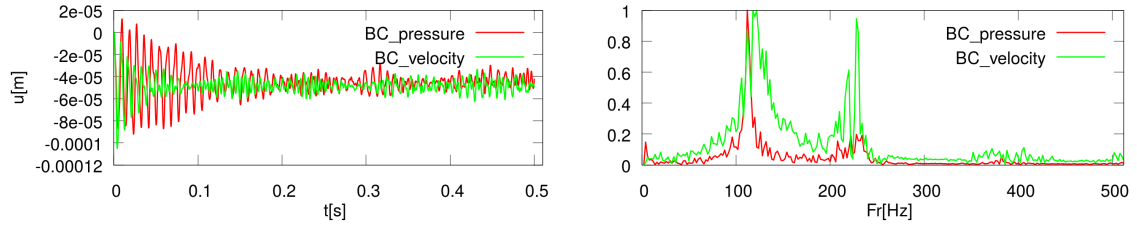


Figure 6: Simulated displacements of the point from the top of vocal fold in y-direction in time domain (left) and the normalized Fourier transformation of the signals (right). Time signal is plotted from the start of computation with the interaction.

eigenmode with the eigenfrequency of 108.7 Hz dominates in the spectrum for the x-component of the displacement (see Fig. 5). The frequency of the first eigenmode is also dominant for the y-component of the displacement but the second eigenmode with eigenfrequency 220.9 Hz is also excited noticeably (see Fig. 6). In the case of BC_velocity the excitation of the second mode is higher in comparison with the case of BC_pressure. This behaviour can be caused by different pressure distribution

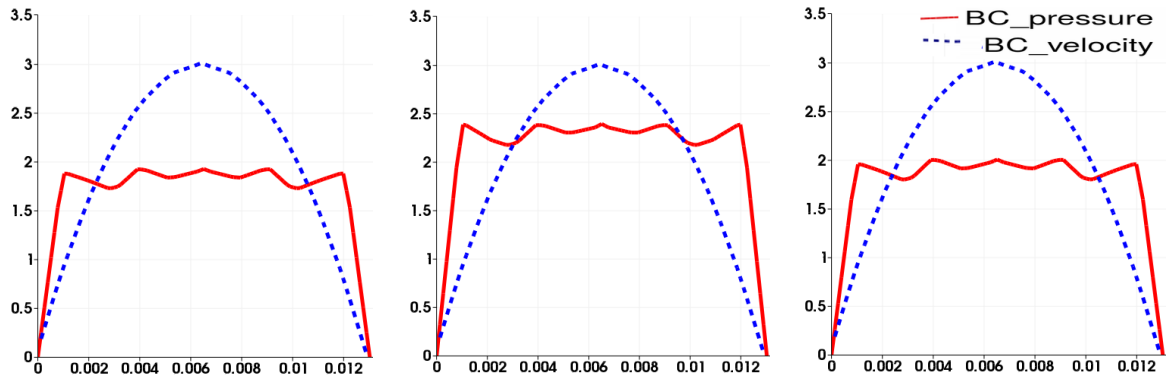


Figure 7: Velocity profile along the inlet at time instants $t = 0.1 + j * \Delta t$ [s], $j \in \{1, 4, 9\}$. Units are m/s and x-axis denotes distance from bottom of channel.

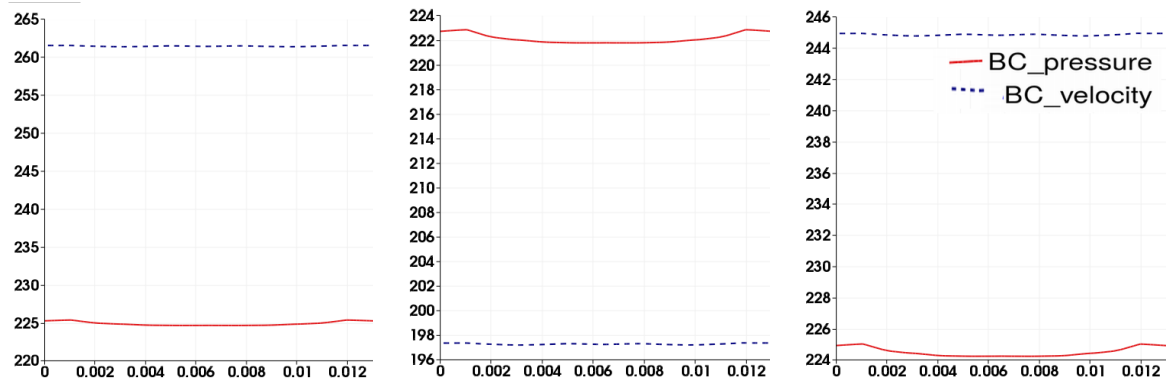


Figure 8: Kinematic pressure p along the inlet at time instants $t = 0.1 + j * \Delta t$ [s], $j \in \{1, 4, 9\}$. Units on the y-axis are in $\mathrm{Pa} \cdot \mathrm{m}^3/\mathrm{kg}$ and the x-axis denotes the distance from the bottom of the channel.

inside the fluid domain which is the dominant part of the aerodynamic force. If the pressure difference is prescribed, then the real pressure drop between the inlet and the outlet remains almost constant while inlet velocity slightly varies. On the other hand if velocity profile is prescribed, then the inlet velocity is fixed and pressure along the channel considerably varies. This is presented in Figures 7 and 8, where time instants were chosen as: 1.0005 s – start of the vocal folds opening, 1.0020 s – point of return from maximal displacement, 1.0045 s – channel closure at the end of the vibration cycle.

5. Conclusion

The article presents the mathematical description and derivation of numerical scheme for solution of FSI by FEM. Special attention is paid to the stabilization of FEM by additional terms introduced in weak formulation of the problem. These

SUPG, PSPG and 'div-div' stabilization methods enable to overcome numerical instabilities and to obtain more accurate results. The simulation of flow-induced vibration of vocal folds computed by an in-house developed program is shown and the influence of the boundary conditions is studied.

Acknowledgements

This work was supported by grant No. GA16-01246S of the Czech Science Foundation and by grant No. SGS16/206/OHK2/3T/12 of CTU in Prague.

References

[1] Braack, M. and Mucha, P. B.: Directional do-nothing condition for the Navier-Stokes equations. J. Comput. Math. **32** (2014), 507–521.

[2] Brdička, M., Samek, L., and Sopko, B.: *Continuum mechanics*. Academia, 2000.

[3] Clark, R. and Dowell, E. H.: *A modern course in aeroelasticity*. Springer, 2004.

[4] Davis, T. A.: *Direct methods for sparse linear systems*. SIAM, 2006.

[5] Feistauer, M., Sváček, P., and Horáček, J.: Numerical simulation of fluid-structure interaction problems with applications to flow in vocal folds. In: T. Bodnár, G. P. Galdi, and S. Nečasová (Eds.), *Fluid-structure Interaction and Biomedical Applications*, pp. 312–393. Birkhauser, 2014.

[6] Gelhard, T., Lube, G., Olshanskii, M. A., and Starcke, J. H.: Stabilized finite element schemes with LBB-stable elements for incompressible flows. J. Comput. Appl. Math. **177** (2005), 243–267.

[7] Richter, T.: *Numerical methods for fluid-structure interaction problems*. Ph.D. thesis, University of Heidelberg, Germany, 2010.

[8] Scherer, R. C. et al.: Intraglottal pressure profiles for a symmetric and oblique glottis with a divergence angle of 10 degrees. Journal of the Acoustical Society of America **109** (2001), 1616–1630.

[9] Takashi, N. 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.

[10] Valášek, J., Sváček, P., and Horáček, J.: On numerical approximation of fluid-structure interactions of air flow with a model of vocal folds. In: D. Šimurda and T. Bodnár (Eds.), *Topical problems of fluid mechanics 2016*, pp. 245–254. Institute of Thermomechanics, AS CR, v.v.i., 2015.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

PARALLEL STRATEGIES FOR SOLVING THE FETI COARSE PROBLEM IN THE PERMON TOOLBOX

Alena Vašatová[1], Jiří Tomčala[1], Radim Sojka[1], Marek Pecha[1,2], Jakub Kružík[1], David Horák[1,2], Václav Hapla[1], Martin Čermák[1]

[1] VSB-Technical University of Ostrava, IT4Innovations National Supercomputing Center
17. listopadu 15, Ostrava, 70833, Czech Republic
vaclav.hapla@vsb.cz, david.horak@vsb.cz
[2] VSB-Technical University of Ostrava, Dep. of Applied Mathematics
17. listopadu 15, Ostrava, 70833, Czech Republic

Abstract: PERMON (Parallel, Efficient, Robust, Modular, Object-oriented, Numerical) is a newly emerging collection of software libraries, uniquely combining Quadratic Programming (QP) algorithms and Domain Decomposition Methods (DDM). Among the main applications are contact problems of mechanics. This paper gives an overview of PERMON and selected ingredients improving scalability, demonstrated by numerical experiments.

Keywords: TFETI, quadratic programming, PERMON, PermonQP, PermonFLLOP, coarse problem

MSC: 65F05, 65M60, 65N55

1. Introduction

PERMON (Parallel, Efficient, Robust, Modular, Object-oriented, Numerical) [10], [12] is a newly emerging collection of software libraries, uniquely combining quadratic programming (QP) and Domain Decomposition Methods (DDM). There are two core modules in PERMON: PermonQP and PermonFLLOP. They are built on top of PETSc [3], [2], mainly its linear algebra part. They extend PETSc with new specific functionality, algorithms for large scale sparse QP problems and DDM of the Finite Element Tearing and Interconnecting (FETI) type. The same coding style is used so that users familiar with PETSc can utilize them with minimal effort. Among the main applications are contact problems of mechanics.

PermonQP provides a base for solution of quadratic programming (QP) problems. It includes data structures, transforms, algorithms, and supporting functions for QP. PermonQP is available for free under the FreeBSD open source license.

PermonFLLOP (FETI Light Layer on Top of PETSc) is an extension of PermonQP that adds support for DDM of the FETI type. PermonFLLOP is currently under preparation for publishing.

DOI: 10.21136/panm.2016.18

2. PermonQP

PermonQP, a general purpose QP solver, allows solving QPs with a symmetric positive semidefinite Hessian and any combination of linear equality and inequality constraints including unconstrained QP. It provides a basic framework for QP solution (data structures, transformations, and supporting functions), a wrapper of PETSc KSP linear solvers for unconstrained and equality-constrained QP, a wrapper of PETSc TAO optimization solvers [13] offering several additional algorithms for unconstrained and box-constrained QP, a variant of the augmented Lagrangian method called Semi-Monotonic Augmented Lagrangian with Bound and Equality (SMALBE), and several specific solvers for bound constrained minimization. General linear inequality constraints can be converted to bound constraints using dualization.

3. PermonFLLOP

PermonFLLOP (FETI Light Layer on Top of PETSc) is an extension of the PermonQP package, implementing the algebraic part of DDMs of the FETI type [7], [6], [5], [4]. Let us show how PermonFLLOP is implemented from the user's perspective. The domain has to be volume-meshed and decomposed using a partitioning software such as METIS [1]. Then virtually arbitrary Finite Element Method (FEM) implementation can be used to generate the subdomain stiffness matrices $\mathbf{K}^s$ and the subdomain load vectors $\mathbf{f}^s$ as sequential data for each subdomain Ω^s, $s = 1, \ldots, N_S$ independently. However, the local-to-global mapping $l2g$, mapping each subdomain's degrees of freedom to the global degrees of freedom, has to be produced in this phase.

Let us denote the number of processor cores used for the compuation by N_c. We assume here each processor core owns only one subdomain, $N_S = N_c$. PermonFLLOP has nevertheless a new experimental feature of allowing more than one subdomain per core, $N_S > N_c$, i.e. an array of $\mathbf{K}^s$ and $\mathbf{f}^s$ is passed per subdomain.

The "gluing" signed Boolean matrix $\mathbf{B}_g$ is constructed based on $l2g$ as described in [14]. The FEM software can skip the processing of the Dirichlet conditions and rather hand it over to PermonFLLOP, resulting in greater flexibility. PermonFLLOP allows to enforce Dirichlet boundary conditions either by the constraint matrix $\mathbf{B}_d$ (Total Finite Element Tearing and Interconnecting (TFETI) approach), or by a classical technique of embedding them directly into $\mathbf{K}^s$ and $\mathbf{f}^s$ (FETI-1 approach). It is also possible to mix these two approaches.

The inequality constraint matrix $\mathbf{B}_I$ describes linearized non-penetration conditions [5] on the contact zones. It is empty for linear (permanent contact only) problems. The global constraint right-hand side vector $\mathbf{c}$ possesses an analogous structure. Currently, PermonFLLOP requires $\mathbf{B}_I$ and $\mathbf{c}_I$ from the caller.

The subdomain nullspace matrix $\mathbf{R}^s$ is assembled using one of the following options. The first option is to use a numerical approach [8], and the second one is to generate $\mathbf{R}^s$ as rigid body modes from the mesh nodal coordinates [4]. The latter is typical for TFETI and is considered here.

Within PermonFLLOP, the local objects $\mathbf{K}^s$, $\mathbf{R}^s$ and $\mathbf{f}^s$ constitute the global distributed objects

$$\begin{aligned}\mathbf{K} &= \mathrm{diag}(\mathbf{K}^1, \ldots, \mathbf{K}^{N_S}),\\ \mathbf{R} &= \mathrm{diag}(\mathbf{R}^1, \ldots, \mathbf{R}^{N_S}),\\ \mathbf{f} &= [(\mathbf{f}^1)^T, \ldots, (\mathbf{f}^{N_S})^T]^T,\end{aligned}$$

where diag means a block-diagonal matrix consisting of the diagonal blocks in the parentheses.

In the PermonFLLOP's function `FllopSolve`, PermonFLLOP passes the global primal data $\mathbf{K}$, $\mathbf{f}$, $\mathbf{B}_E = \left[\begin{smallmatrix}\mathbf{B}_g^T & \mathbf{B}_d^T\end{smallmatrix}\right]^T$, $\mathbf{B}_I$ and $\mathbf{R}$ to PermonQP (Section 2), calls a specific series of QP transforms provided by PermonQP, resulting in the bound and equality constrained QP which is then solved with the `QPSSolve` function.

From the mathematical point of view, the called QP transforms (`QPT`) implement the following modifications. The original primal problem

$$\min\ \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{f}^T\mathbf{u}\ \text{ s.t. }\ \mathbf{B}_I\mathbf{u} \le 0\ \text{ and }\ \mathbf{B}_E\mathbf{u} = 0, \tag{1}$$

is transformed into the dual one by `QPTDualize`

$$\min\ \frac{1}{2}\boldsymbol{\lambda}^T\mathbf{F}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathbf{d}\ \text{ s.t. }\ \boldsymbol{\lambda}_I \ge 0\ \text{ and }\ \mathbf{G}\boldsymbol{\lambda} = \mathbf{e}, \tag{2}$$

We use the standard notation

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

with matrix $\mathbf{R}$, whose columns span the null space of $\mathbf{K}$ and represent rigid body or zero energy modes of subdomains, and $\mathbf{K}^\dagger$ denoting a generalized inverse of $\mathbf{K}$, i.e. a matrix satisfying $\mathbf{K}\mathbf{K}^\dagger\mathbf{K} = \mathbf{K}$. The constraint matrix $\mathbf{B} = \left[\begin{smallmatrix}\mathbf{B}_I^T & \mathbf{B}_E^T\end{smallmatrix}\right]^T$ can be constructed so that it has full rank, and then the Hessian $\mathbf{F}$ is positive definite with a relatively favourably distributed spectrum for application of the conjugate gradient method (CG).

The solution $\mathbf{u}$ can be evaluated by formula

$$\mathbf{u} = \mathbf{K}^\dagger(\mathbf{f} - \mathbf{B}^T\boldsymbol{\lambda}) + \mathbf{R}\boldsymbol{\alpha}. \tag{3}$$

Here,

$$\boldsymbol{\alpha} = -(\mathbf{R}^T\widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}}\mathbf{R})^{-1}\mathbf{R}^T\widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}}\mathbf{K}^\dagger(\mathbf{f} - \mathbf{B}^T\boldsymbol{\lambda})$$

denotes the vector of amplitudes, determining the contribution $\mathbf{R}\boldsymbol{\alpha}$ of the null space $\mathbf{R}$ to the solution $\mathbf{u}$. The matrix $\widetilde{\mathbf{B}}$ is defined as $\widetilde{\mathbf{B}} = \left[\begin{smallmatrix}\widetilde{\mathbf{B}}_I^T & \mathbf{B}_E^T\end{smallmatrix}\right]^T$ with $\widetilde{\mathbf{B}}_I$ formed by rows of $\mathbf{B}_I$ that correspond to the active constraints.

The problem of minimization on the subset of the affine space is transformed into the problem on subset of vector space by means of arbitrary $\widetilde{\boldsymbol{\lambda}}$ which satisfies $\mathbf{G}\widetilde{\boldsymbol{\lambda}} = \mathbf{e}$

while the solution is looked for in the form $\boldsymbol{\lambda} = \widehat{\boldsymbol{\lambda}} + \widetilde{\boldsymbol{\lambda}}$. The problem obtained by `QPTHomogenizeEq` then reads

$$\min \frac{1}{2}\widehat{\boldsymbol{\lambda}}^T\mathbf{F}\widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T(\mathbf{d} - \mathbf{F}\widetilde{\boldsymbol{\lambda}}) \text{ s.t. } \widehat{\boldsymbol{\lambda}}_I \geq -\widetilde{\boldsymbol{\lambda}_I} \text{ and } \mathbf{G}\widehat{\boldsymbol{\lambda}} = 0. \tag{4}$$

Further improvement is based on the observation, that the augmented Lagrangian for problem (4) can be decomposed by orthogonal projectors

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

on the kernel of $\mathbf{G}$ and on the image space of $\mathbf{G}^T$: $\mathrm{Im}\mathbf{P} = \mathrm{Ker}\mathbf{G}$, $\mathrm{Im}\mathbf{Q} = \mathrm{Im}\mathbf{G}^T$. Evaluating $(\mathbf{G}\mathbf{G}^T)^{-1}$, i.e. solving the linear system

$$\mathbf{G}\mathbf{G}^T\mathbf{x} = \mathbf{y}, \tag{5}$$

is called the coarse problem (CP). The modified formulation of the problem (4), obtained by `QPTEnforceEqByProjector`, then takes the form

$$\min \frac{1}{2}\widehat{\boldsymbol{\lambda}}^T\mathbf{P}\mathbf{F}\mathbf{P}\widehat{\boldsymbol{\lambda}} - \widehat{\boldsymbol{\lambda}}^T\mathbf{P}\mathbf{d} \text{ s.t. } \widehat{\boldsymbol{\lambda}}_I \geq -\widetilde{\boldsymbol{\lambda}_I} \text{ and } \mathbf{G}\widehat{\boldsymbol{\lambda}} = 0. \tag{6}$$

More details can be found in [12].

4. Coarse problem

FETI methods blend iterative and direct solvers. The main loop solving the dual problem is implemented by an iterative solver, e.g. CG. In each iteration, auxiliary problems related to the application of an unassembled system operator are solved: (1) $\mathbf{K}^\dagger$ application and (2) CP solution.

Parallelization is achieved mainly by distributing diagonal blocks of $\mathbf{K}$ over processors, each block reflecting a subdomain. We strive to maximize the number of subdomains to reduce the sizes of the subdomain stiffness matrices, accelerating their factorization and $\mathbf{K}^\dagger$ actions. Furthermore, thanks to the FETI operator condition number estimate [6], decomposition into more subdomains maintaining a fixed discretization parameter h leads to reduction of the condition number of $\mathbf{K}$ and thus the number of iterations.

A drawback is the increasing null space dimension which decelerates the CP solution – it is a kind of a communicating vessels effect. The natural coarse space matrix $\mathbf{G}$ is computed so that each core owns the sparse sequential matrices $\mathbf{R}^s$ and $\mathbf{B}^s$, and computes the local horizontal block $\mathbf{G}^s = (\mathbf{R}^s)^T(\mathbf{B}^s)^T$ without any communication, $\mathbf{G} = [(\mathbf{G}^1)^T, \ldots, (\mathbf{G}^{N_S})^T]^T$. The multiplication $\mathbf{G}\mathbf{G}^T = \mathbf{G} * \mathbf{G}^T$, factorization of $\mathbf{G}\mathbf{G}^T$, and the CP solutions (5) should be done in parallel, otherwise they form a computational and memory bottleneck. The sparsity pattern of $\mathbf{G}$ and $\mathbf{G}\mathbf{G}^T$ for the cube decomposed into 27 subdomains is illustrated in Fig. 1 and Fig. 2, respectively.

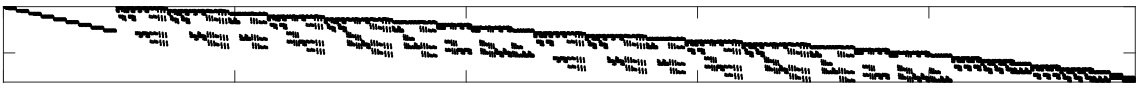


Figure 1: The sparsity pattern of $\mathbf{G}$ for a problem of the elastic cube with 27 subdomains an 2^3 elements per subdomain.

Figure 2: The sparsity pattern of $\mathbf{G}\mathbf{G}^T$ for the same problem as in Fig. 1.

We have suggested and compared several strategies for parallel CP solution [11], [9]. The explicit orthonormalization approach starts to fail when the nullspace is large (thousands). Hence, we have abandoned this approach. Let us describe in more detail the strategies tested in this work.

Strategy 1 (S1) Obtain a solution of CP by solving the system (5) iteratively (by CG or pipelined CG (PipeCG)) or by a direct method (by parallel direct solver (SuperLU_DIST)). For a direct solution, $\mathbf{G}\mathbf{G}^T$ is factorized in the preprocessing phase: $\mathbf{G}\mathbf{G}^T = \mathbf{L}^{\mathbf{G}\mathbf{G}^T}(\mathbf{L}^{\mathbf{G}\mathbf{G}^T})^T$. During the solution phase, each application of $(\mathbf{G}\mathbf{G}^T)^{-1}$ consists of the forward and backward substitution using a parallel direct solver: $\mathbf{x} = (\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{y}$ is solved by a two-step procedure as (1) $\mathbf{L}^{\mathbf{G}\mathbf{G}^\mathbf{T}}\mathbf{w} = \mathbf{y}$, (2) $(\mathbf{L}^{\mathbf{G}\mathbf{G}^\mathbf{T}})^T\mathbf{x} = \mathbf{w}$.

Strategy 2 (S2) An iterative or a parallel direct solver is employed for the computation of the *explicit inverse* of $\mathbf{G}\mathbf{G}^T$. During the preprocessing phase, $(\mathbf{G}\mathbf{G}^T)^{-1}$ is computed iteratively or by a direct method. In the solution phase, its application consists in the parallel dense matrix-vector product $(\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{y}$ in both cases.

The CP dimension is not large enough to justify the use of the whole global communicator. Instead, we propose a proper *partial* parallelization of this CP solution. We divide all processes of the global `PETSC_COMM_WORLD` communicator into the subcommunicators using PETSc built-in "pseudopreconditioner" `PCREDUNDANT`; the number of these subcommunicators is N_r (number of cores doing redundant work); this means the number of cores in each subcommunicator is $\approx N_c/N_r$.

In Strategy 2, the explicit inverse is assembled in the following way. Each of N_r subcommunicators is assigned a contiguous portion of N_n/N_r columns of the identity matrix taken as the right-hand side, where N_n is the dimension of the nullspace of $\mathbf{K}$, i.e. the number of columns of the matrix $\mathbf{R}$. The result of the forward/backward substitutions is the corresponding portion of N_n/N_r columns of the resulting explicit inverse $(\mathbf{G}\mathbf{G}^T)^{-1}$, stored as a $N_n \times (N_n/N_r)$ dense matrix distributed vertically across the subcommunicator. Taking advantage of the symmetry of $(\mathbf{G}\mathbf{G}^T)^{-1}$, each subcommunicator's block is transposed in parallel and the blocks are then merged one

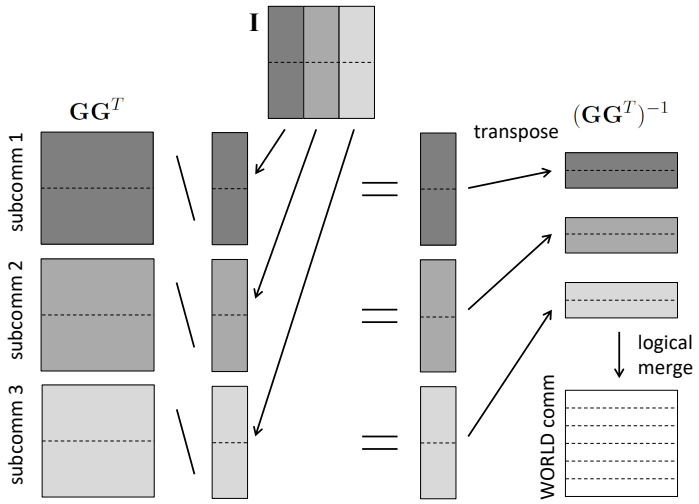


Figure 3: Scheme of $(\mathbf{GG}^T)^{-1}$ implementation using Strategy 2. Different colours represent different communicators.

below the other in the proper order forming the complete $(\mathbf{GG}^T)^{-1}$ matrix, divided into horizontal blocks distributed across the global communicator. Note that this merge means only logical reassignment from the subcommunicator to the global communicator with no actual data movements. A scheme of this strategy is depicted in Fig. 3.

5. Numerical experiments

The numerical experiments were performed at ARCHER, the latest UK National Supercomputing Service. It is based on a Cray XC30 supercomputer with 4920 nodes, 118,080 cores and 1.56 Petaflops of theoretical peak performance. All compute nodes are connected together in the Dragonfly topology by the Aries interconnect. Each compute node contains two 2.7 GHz, 12-core Ivy Bridge processors.

Firstly, we have performed a comparison of the CP strategies. For the CP solution the SuperLU_DIST solver performs better than the MUMPS solver. The GGtinv phase of the S1 is much cheaper in comparison with S2. On the other hand, S1 has much more expensive CP actions compared with S2. For a high number of expected CP actions, the second strategy starts to payoff because the high cost of preprocessing phase is offset by the cheapness of the CP action. The choice of an appropriate strategy therefore depends on the number of expected CP actions. This can be interesting for ill conditioned elasto-static problems but even more interesting for contact problems where the number of iterations is always higher. Finally, the greatest effect will be seen for all problems that are solved using outer iteration on top of FETI such as shape optimization, transient problems, or elasto-plasticity. Graphs in Fig. 4 show performance of both strategies on 8,000 subdomains (the CP size 48,000).

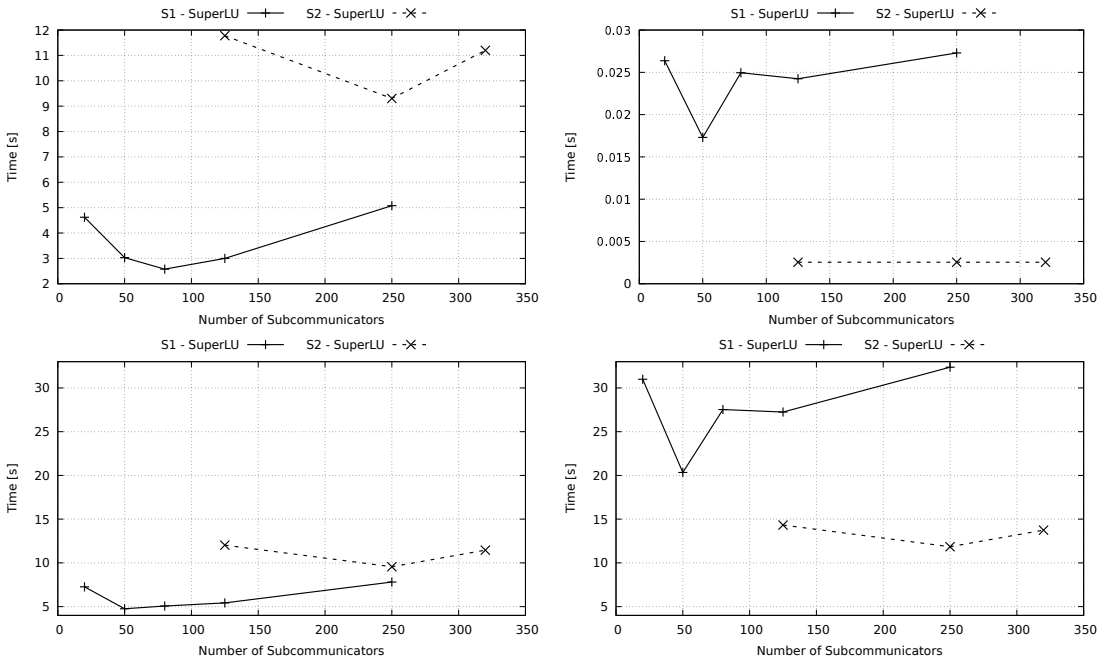


Figure 4: CP performance for the cube benchmark with 8,000 subdomains. $\mathbf{G}\mathbf{G}^T$ size is 48,000. Top left: CP setup time. Top right: CP solution time. Bottom left: Time of CP setup + 100 CP actions. Bottom right: Time of CP setup + 1000 CP actions actions.

Secondly, we have demonstrated PERMON capabilities using weak scalability tests with the S1 strategy. As a model 3D linear elasticity problem, we consider an elastic cube with the bottom face fixed generated by our PermonCube benchmark generation package. For the linear case (see Fig. 5), the top face is loaded with a vertical surface force $f_z = 465$ [N/mm^2] directed upwards. For the nonlinear case (see Fig. 7), the top face is loaded with a vertical surface force $f_z = -465$ [N/mm^2] directed downwards, and the right one is partially in contact with a rigid obstacle. In both cases, Young modulus is $E = 2 \cdot 10^5$ [MPa], and Poisson ratio is $\mu = 0.33$. The graphs in Fig. 6 and Fig. 8 demonstrate both numerical and weak parallel scalability up to 701 millions of unknowns and 10,648 subdomains with one subdomain per one computational core. The contact problem was solved using SMALBE and Modified Proportioning and Reduced Gradient Projection (MPRGP) with our new adaptive expansion steplength which significantly improved this scalability and reduced not only the number of expansion steps but also the number of CG steps.

6. Conclusion

The PERMON team was successful to push the scalability limits for both linear and nonlinear benchmarks using ARCHER up to 702 millions of unknowns and 10,648 subdomains (cores). The implemented matrix formats and efficient parallel

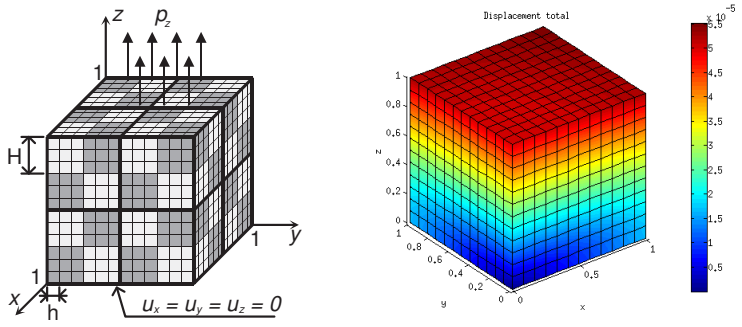


Figure 5: Linearly elastic cube problem.

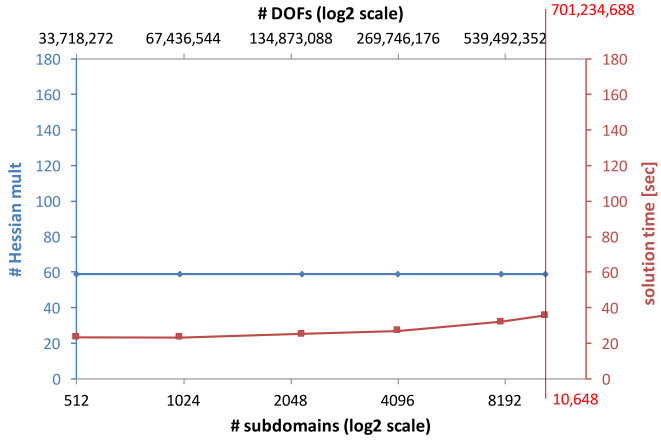


Figure 6: Scalability results for the linearly elastic cube problem.

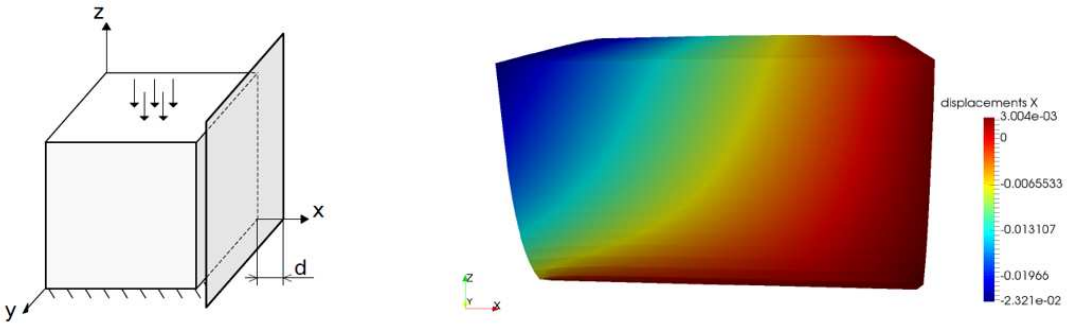


Figure 7: Contact problem – a linearly elastic cube with an obstacle.

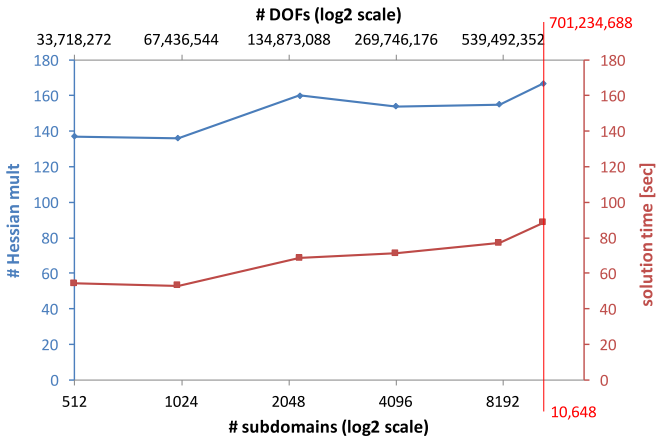


Figure 8: Scalability results for the contact problem.

direct solvers were employed. Furthermore, two strategies for CP solution were studied: (1) factorization + forward/backward substitutions, (2) factorization + explicit inverse assembly + dense matrix-vector products. It was demonstrated that the optimal strategy depends on the number of subdomains and the expected number of CP actions. The latter depends on the class of the solved problem.

Acknowledgements

The authors acknowledge the support of the Czech Science Foundation (GACR) project no. 15-18274S. The work was also supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602" and from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center LM2015070", and by the internal student grant competition project SP2016/178 "PERMON toolbox development II".

References

[1] METIS. URL http://tinyurl.com/libmetis.

[2] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F.: Efficient management of parallelism in object oriented numerical software libraries. In: E. Arge, A. M. Bruaset, and H. P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, pp. 163–202. Birkhäuser Press, 1997. doi:10.1007/978-1-4612-1986-6_8.

[3] Balay, S. et al.: PETSc – Portable, Extensible Toolkit for Scientific Computation. URL http://www.mcs.anl.gov/petsc.

[4] Dostál, Z., Horák, D., and Kučera, R.: Total FETI – an easier implementable variant of the FETI method for numerical solution of elliptic PDE. Commu-

nications in Numerical Methods in Engineering **22** (2006), 1155–1162. doi: 10.1002/cnm.881.

[5] Dostál, Z. et al.: FETI based algorithms for contact problems: scalability, large displacements and 3D Coulomb friction. Comput. Methods Appl. Mech. Engrg. **194** (2005), 395–409. doi:10.1016/j.cma.2004.05.015.

[6] Farhat, C., Mandel, J., and Roux, F. X.: Optimal convergence properties of the FETI domain decomposition method. Comput. Methods Appl. Mech. Engrg. **115** (1994), 365–385. doi:10.1016/0045-7825(94)90068-X.

[7] Farhat, C. and Roux, F. X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. International Journal for Numerical Methods in Engineering **32** (1991), 1205–1227. doi:10.1002/nme.1620320604.

[8] Gosselet, P. and Rey, C.: Non-overlapping domain decomposition methods in structural mechanics. Arch. Comput. Methods Engrg. **13** (2006), 515–572. doi: 10.1007/BF02905857.

[9] Hapla, V., Horák, D., and Merta, M.: Use of direct solvers in TFETI massively parallel implementation. In: PARA 2012, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 7782, pp. 192–205. Springer-Verlag Berlin Heidelberg, 2013. doi:10.1007/978-3-642-36803-5_14.

[10] Hapla, V. et al.: PERMON (Parallel, Efficient, Robust, Modular, Object-oriented, Numerical). URL http://permon.it4i.cz.

[11] Hapla, V. and Horák, D.: TFETI coarse space projectors parallelization strategies. In: PPAM 2011, Revised Selected Papers, Part I, *Lecture Notes in Computer Science*, vol. 7203, pp. 152–162. Springer-Verlag Berlin Heidelberg, 2012. doi:10.1007/978-3-642-31464-3_16.

[12] Hapla, V. et al.: Solving contact mechanics problems with PERMON. In:HPCSE 2015, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 9611, pp. 101–115. Springer International Publishing Switzerland, 2016. doi:10.1007/978-3-319-40361-8_7.

[13] Munson, T., Sarich, J., Wild, S., Benson, S., and McInnes, L. C.: TAO users manual. Tech. Rep. ANL/MCS-TM-322, Argonne National Laboratory, 2015. URL http://tinyurl.com/tao-man.

[14] Vašatová, A., Čermák, M., and Hapla, V.: Parallel implementation of the FETI DDM constraint matrix on top of PETSc for the PermonFLLOP package. In: PPAM 2015, Revised Selected Papers, Part I, *Lecture Notes in Computer Science*, vol. 9573, pp. 150–159. Springer International Publishing Switzerland, 2016. doi:10.1007/978-3-319-32149-3_15.

Programs and Algorithms of Numerical Mathematics 18
J. Chleboun, P. Kůs, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2017

A GENERALIZED LIMITED-MEMORY BNS METHOD BASED ON THE BLOCK BFGS UPDATE

Jan Vlček[1], Ladislav Lukšan[1,2]
[1]Institute of Computer Science, Czech Academy of Sciences,
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
vlcek@cs.cas.cz, luksan@cs.cas.cz
[2]Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

Abstract: A block version of the BFGS variable metric update formula is investigated. It satisfies the quasi-Newton conditions with all used difference vectors and gives the best improvement of convergence in some sense for quadratic objective functions, but it does not guarantee that the direction vectors are descent for general functions. To overcome this difficulty and utilize the advantageous properties of the block BFGS update, a block version of the limited-memory BNS method for large scale unconstrained optimization is proposed. The algorithm is globally convergent for convex sufficiently smooth functions and our numerical experiments indicate its efficiency.

Keywords: Unconstrained minimization, block variable metric methods, limited-memory methods, the BFGS update, global convergence, numerical results
MSC: 65K10

1. Introduction

In this contribution we propose a block version of the widely used BNS method, see [3], for large scale unconstrained optimization

$$\min f(x): \ x \in \mathcal{R}^N,$$

where it is assumed that the problem function $f : \mathcal{R}^N \to \mathcal{R}$ is differentiable.

The BNS method belongs to the variable metric (VM) or quasi-Newton (QN) line search iterative methods, see [9], [11]. They start with an initial point $x_0 \in \mathcal{R}^N$ and generate iterations $x_{k+1} \in \mathcal{R}^N$ by the process $x_{k+1} = x_k + s_k$, $s_k = t_k d_k$, $k \geq 0$, where usually the direction vector $d_k \in \mathcal{R}^N$ is $d_k = -H_k g_k$, matrix H_k is symmetric positive definite and a stepsize $t_k > 0$ is chosen in such a way that

$$f_{k+1} - f_k \leq \varepsilon_1 t_k g_k^T d_k, \qquad g_{k+1}^T d_k \geq \varepsilon_2 g_k^T d_k, \qquad k \geq 0 \tag{1}$$

(the Wolfe line search conditions, see [11]), where $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$. Typically, H_0 is a multiple of I and H_{k+1} is obtained from

DOI: 10.21136/panm.2016.19

H_k by a VM update to satisfy the QN condition (see [9]) $H_{k+1}y_k = s_k$, $y_k = g_{k+1} - g_k$, $k \geq 0$.

Among VM methods, the BFGS method, see [9], [11], belongs to the most efficient. It preserves positive definite VM matrices and can be written in the form

$$H_+ = (1/b)ss^T + \big(I - (1/b)sy^T\big)H\big(I - (1/b)ys^T\big), \quad b = s^T y, \tag{2}$$

$b > 0$ by (1). Note that for simplification we often omit index k and replace indices $k+1$, $k-1$ by symbols $+, -$, respectively. The BNS and L-BFGS (see [5], [6] – subroutine PLIS) methods represent its well-known limited-memory adaptations (for large-scale optimization). In every iteration we repeatedly update an initial approximation of the inverse Hessian matrix $\zeta_k I$, $\zeta_k > 0$, by the BFGS method, using m couples of vectors $(s_{k-\tilde{m}}, y_{k-\tilde{m}}), \ldots, (s_k, y_k)$ successively (without forming approximations of the inverse Hessian matrix explicitly), where $\tilde{m} = \min(k, \hat{m} - 1)$, $m = \tilde{m} + 1$ and $\hat{m} > 1$ is a given parameter. In the case of the BNS method, matrix H_+ can be expressed in the form, see [3],

$$H_+ = SU^{-T}DU^{-1}S^T + \zeta\Big(I - SU^{-T}Y^T\Big)\Big(I - YU^{-1}S^T\Big), \tag{3}$$

where for $k \geq 0$ we denote $S_k = [s_{k-\tilde{m}}, \ldots, s_k]$, $Y_k = [y_{k-\tilde{m}}, \ldots, y_k]$, $(U_k)_{i,j} = (S_k^T Y_k)_{i,j}$ for $i \leq j$, $(U_k)_{i,j} = 0$ otherwise (an upper triangular matrix), $D_k = \mathrm{diag}[b_{k-\tilde{m}}, \ldots, b_k]$.

For $S^T Y$ nonsingular and any $H \in \mathcal{R}^{N\times N}$, the BFGS update formula (2) can be easily generalized to the following block version

$$H_+ = S(S^TY)^{-1}S^T + \big(I - S(S^TY)^{-T}Y^T\big)\,\bar{H}\,\big(I - Y(S^TY)^{-1}S^T\big)\,,\ \bar{H} = \frac{1}{2}(H + H^T), \tag{4}$$

which satisfies the QN conditions $H_+Y = S$ (for the whole block of stored difference vectors) and was derived in [12] and [4] for S^TY, H symmetric positive definite.

Formula (4) is not directly applicable to general functions, since it does not guarantee that the corresponding direction vectors are descent. Thus we split matrices S and Y in such a way that $S = [S_{[1]}, \ldots, S_{[n]}]$, $Y = [Y_{[1]}, \ldots, Y_{[n]}]$, with all blocks $S_{[i]}^T Y_{[i]}$ positive definite, i.e. matrices $S_{[i]}^T Y_{[i]} + Y_{[i]}^T S_{[i]}$ symmetric positive definite, $i = 1, \ldots, n$. Then we replace the BNS formula (3) by n successive updates of an initial matrix ζI using a modification of the block BFGS update (4) with matrices $S_{[i]}, Y_{[i]}$, $i = 1, \ldots, n$, instead of S, Y. Obviously, for $n = m$ we obtain the BNS method.

In Section 2 we derive the block BFGS update, investigate its properties and show some similarities to the VM methods, based on the corrected BFGS updates, see the limited-memory BFGS method [13] – [16]. In Section 3 we focus on quadratic functions and show optimality of the block BFGS method and a role of unit stepsizes. In Section 4 we present the block BNS method and derive a convenient formula similar to (3) to represent the resultant VM matrix. The simplified algorithm is described in Section 5. Global convergence of the algorithm is established in Section 6

and numerical results are reported in Section 7. We refer to report [17] for details and proofs of assertions, here we briefly present only the main results.

We will denote the Frobenius matrix norm by $\|\cdot\|_F$.

2. The block BFGS update

Using the following theorem, update (4) can be derived for general functions:

Theorem 1. *Let $W_L, W_R \in \mathcal{R}^{N\times N}$ be nonsingular, matrix Y have a full rank, matrix H_+ be given by (4) and matrices A_{i+1}, $i = -1, 0, \ldots$, be the unique solution to*

$$\min_{A_{i+1}\in\mathcal{R}^{N\times N}} \left\|W_L^{-1}(A_{i+1} - \bar{A}_i)W_R^{-1}\right\|_F \ s.t.\ A_{i+1}Y = S, \quad \bar{A}_{-1} = H, \ \bar{A}_i = \frac{1}{2}(A_i + A_i^T),$$

$i \geq 0$. Then for $W_R^T W_R Y = S T_S$, T_S square nonsingular, we have $\lim_{i\to\infty} A_i = H_+$.

The new update has similar interesting properties as the standard BFGS update.

Theorem 2. *Let matrix H_+ be given by (4), matrices $S^TY, \bar{H}, H_+, S^T\bar{B}S$, $T \in \mathcal{R}^{m\times m}$ nonsingular, $\bar{B} = \bar{H}^{-1}$, $B_+ = H_+^{-1}$. Then (also for nonsymmetric $\bar{H}$)*

(a) matrix H_+ is invariant under the transformation $S \to ST$, $Y \to YT$,

(b) $B_+ = \bar{B} - \bar{B}S(S^T\bar{B}S)^{-1}S^T\bar{B} + Y(S^TY)^{-T}Y^T$,

(c) $\det B_+ = \det \bar{B} \,.\, \det(S^TY)/\det(S^T\bar{B}S)$,

(d) for H and S^TY positive definite, also matrix H_+ is positive definite,

(e) for $S \triangleq [\check{S}, s], Y \triangleq [\check{Y}, y], S^TY, \check{S}^T\check{Y}$ symmetric nonsingular, $\check{P} = I - \check{Y}(\check{S}^T\check{Y})^{-1}\check{S}^T$, $\tilde{s} = \check{P}^T s$ and $\tilde{y} = \check{P}y$ we have $\tilde{s}^T\tilde{y} \triangleq \tilde{b} \neq 0$, $H_+ = (1/\tilde{b})\,\tilde{s}\tilde{s}^T + \tilde{P}^T\check{H}\tilde{P}$, $\tilde{P} = I - (1/\tilde{b})\tilde{y}\tilde{s}^T$, $\check{H} = \check{S}(\check{S}^T\check{Y})^{-1}\check{S}^T + \check{P}^T\bar{H}\check{P}$; besides, $\check{S}^TB_+\tilde{s} = \check{S}^T\check{H}^{-1}\tilde{s} = 0$ holds.

Theorem 2*(e)* shows some connections with our methods [13] – [16] based on vector corrections for conjugacy. The following theorem indicates that we can expect good properties of the block BFGS update also for functions similar to quadratic.

Theorem 3. *Let matrices $\check{S}, \check{Y}, \check{P}, \check{H}, H_+$ and vectors $\tilde{s}, \tilde{y}$ have the same meaning as in Theorem 2(e), S^TY be symmetric positive definite, $\ddot{s} = s + \check{S}\sigma$, $\ddot{y} = y + \check{Y}\sigma$, $\sigma \in \mathcal{R}^{\tilde{m}}$, $\tilde{m} \geq 1$. Then $\ddot{b} = \ddot{s}^T\ddot{y} \geq \tilde{s}^T\tilde{y} > 0$. Moreover, if matrix $\bar{H}$ is nonsingular, $\ddot{P} = I - (1/\ddot{b})\ddot{y}\,\ddot{s}^T$, $\ddot{H}_+ = (1/\ddot{b})\ddot{s}\,\ddot{s}^T + \ddot{P}^T\check{H}\ddot{P}$ and $\hat{G}$ is any symmetric positive definite matrix satisfying $\hat{G}S = Y$, then function $\varphi(\sigma) = \|\hat{G}^{\frac{1}{2}}\ddot{H}_+\hat{G}^{\frac{1}{2}} - I\|_F$ is minimized and $\ddot{H}_+ = H_+$ holds for $\sigma = -(\check{S}^T\check{Y})^{-1}\check{Y}^Ts$, when $\ddot{s} = \tilde{s}, \ddot{y} = \tilde{y}$.*

Paradoxically, the standard BFGS update often gives better results if S^TY is almost symmetric and the Hessian matrix is ill-conditioned. Therefore we will use, in addition to the choice $\ddot{s} = \tilde{s}, \ddot{y} = \tilde{y}$, also the choice $\ddot{s} = s, \ddot{y} = y$, which corresponds to the standard BFGS update of $\breve{H}$ and can be easily realized by means of blocks of order one, or a special choice $\ddot{s} = s - (s^T y_-/b_-)s_-$, $\ddot{y} = y - (y^T s_-/b_-)y_-$, which can be more robust than the block BFGS update, see [17] for details.

3. Results for quadratic functions

Compared to the BNS method, the block BFGS update gives the best improvement of convergence in some sense for linearly independent direction vectors:

Theorem 4. *Let $f(x) = \frac{1}{2}(x-\bar{x})^T G(x-\bar{x})$, $\bar{x} \in \mathcal{R}^N$, with a symmetric positive definite matrix G, let all columns of S be linearly independent, $\underline{k} = k - \tilde{m}$, $\hat{S}_i = [s_{\underline{k}}, \dots, s_i]$, $\hat{Y}_i = [y_{\underline{k}}, \dots, y_i]$, $\hat{P}_i = I - \hat{Y}_i(\hat{S}_i^T\hat{Y}_i)^{-1}\hat{S}_i^T$, $i = \underline{k}, \dots, k$, $\ddot{s}_{\underline{k}} = \tilde{s}_{\underline{k}} = s_{\underline{k}}$, $\ddot{y}_{\underline{k}} = \tilde{y}_{\underline{k}} = y_{\underline{k}}$, $\ddot{s}_i = s_i + \hat{S}_{i-1}\sigma_i$, $\ddot{y}_i = y_i + \hat{Y}_{i-1}\sigma_i$, $\sigma_i \in \mathcal{R}^{i-1}$, $\tilde{s}_i = \hat{P}_{i-1}^T s_i$, $\tilde{y}_i = \hat{P}_{i-1} y_i$, $i = \underline{k}+1, \dots, k$. Then matrices $\hat{S}_i^T\hat{Y}_i$ are symmetric positive definite and $\ddot{s}_i^T\ddot{y}_i \geq \tilde{s}_i^T\tilde{y}_i > 0$, $i = \underline{k}, \dots, k$.*

Moreover, let $\bar{H}$ be symmetric positive definite, H_+ be given by (4) and $\ddot{H}_{k+1}$ by

$$\ddot{H}_{\underline{k}} = \bar{H}, \quad \ddot{H}_{i+1} = (1/\ddot{s}_i^T\ddot{y}_i)\,\ddot{s}_i\ddot{s}_i^T + \ddot{P}_i^T\ddot{H}_i\ddot{P}_i, \quad \ddot{P}_i = I - (1/\ddot{s}_i^T\ddot{y}_i)\,\ddot{y}_i\ddot{s}_i^T,$$

$i = \underline{k}, \dots, k$. Then value $\|G^{1/2}\ddot{H}_+G^{1/2} - I\|_F$ is minimized and matrices $\ddot{H}_+$ and H_+ are identical and symmetric positive definite for $\ddot{s}_i = \tilde{s}_i$, $\ddot{y}_i = \tilde{y}_i$, $i = \underline{k}+1, \dots, k$.

Furthermore, similarly to Theorem 3.3 in [16], we get (see Theorem 3.2 in [17]) that if one stepsize t is unit in two successive iterations with matrices H, H_+ obtained by the block BFGS updates, all stored direction vectors from previous iterations are conjugate with vector s_+; thus if all steps are unit, all matrices S^TY are tridiagonal.

4. The block BNS method

Using Lemma 1, we split matrices S, Y in such a way that $S = [S_{[1]}, \dots, S_{[n]}]$, $Y = [Y_{[1]}, \dots, Y_{[n]}]$, $n \geq 1$, with all blocks $S_{[i]}^T Y_{[i]}$ positive definite ($S_{[i]}^T Y_{[i]} + Y_{[i]}^T S_{[i]}$ symmetric positive definite), $i = 1, \dots, n$, and use the theory in Section 2 for matrices $S_{[i]}, Y_{[i]}$ instead of S, Y. We use the RL factorization instead of the LU one, since we start with the submatrices of S, Y which contain their latest columns to have maximum of the latest QN conditions satisfied. The following lemma converts the problem of factorization to the same problem of a smaller dimension. A generalization of the standard BNS formula (3) is given by Theorem 5.

Lemma 1. *Suppose that $A, R, L \in \mathcal{R}^{\mu\times\mu}$, $\mu > 0$, $u, v \in \mathcal{R}^\mu$, $\alpha \in \mathcal{R}$, $\alpha \neq 0$,*

$$\bar{A} = \begin{bmatrix} A & u \\ v^T & \alpha \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} R & u \\ & \alpha \end{bmatrix}, \quad \bar{L} = \begin{bmatrix} L & \\ (1/\alpha)\,v^T & 1 \end{bmatrix}.$$

Then to have $\bar{A} = \bar{R}\bar{L}$, it suffices to find R, L satisfying $A - (1/\alpha)uv^T = RL$. Moreover,

(a) *if $u = v$ then matrix $\bar{A}$ is symmetric positive definite if and only if both $\alpha > 0$ and matrix $A - (1/\alpha)uv^T$ is symmetric positive definite,*

(b) *if matrix $\bar{A}$ is positive definite, then $\alpha > 0$ and $A-(1/\alpha)uv^T$ is positive definite.*

Theorem 5. *Let $\zeta > 0, H_{[1]} = \zeta I, S = [S_{[1]}, \dots, S_{[n]}]$, $Y = [Y_{[1]}, \dots, Y_{[n]}]$, $S_{[i]}^T Y_{[i]}$ nonsingular, $P_{[i]} = I - Y_{[i]}(S_{[i]}^T Y_{[i]})^{-1} S_{[i]}^T$, $H_{[i+1]} = S_{[i]}(S_{[i]}^T Y_{[i]})^{-1} S_{[i]}^T + \frac{1}{2} P_{[i]}^T (H_{[i]} + H_{[i]}^T) P_{[i]}$, $\Sigma_i = Y_{[i]}^T S_{[i]}$, $1 \le i \le n$, $H_+ = H_{[n+1]}$. Then ($\tilde{U}$ is an upper block triangular matrix)*

$$H_+ = S\tilde{U}^{-T} E \tilde{U}^{-1} S^T + \zeta\big(I - S\tilde{U}^{-T} Y^T\big)\big(I - Y\tilde{U}^{-1} S^T\big)\,, \tag{5}$$

$$E = \operatorname{diag}\left[(1/2)(\Sigma_1 + \Sigma_1^T), \dots, (1/2)(\Sigma_{n-1} + \Sigma_{n-1}^T), \Sigma_n\right], \tag{6}$$

$$\tilde{U} = \begin{bmatrix} S_{[1]}^T Y_{[1]} & \dots & S_{[1]}^T Y_{[n-1]} & S_{[1]}^T Y_{[n]} \\ & \ddots & \vdots & \vdots \\ & & S_{[n-1]}^T Y_{[n-1]} & S_{[n-1]}^T Y_{[n]} \\ & & & S_{[n]}^T Y_{[n]} \end{bmatrix}. \tag{7}$$

Although matrix H_+ is unsymmetric generally, we use the usual direction vector $d_+ = -H_+ g_+$, such that $z^* = x_+ + d_+$ satisfies $g(z^*) = 0$, $g(z) = g_+ + H_+^{-1}(z - x_+)$ (a linear model for gradients which respects the QN conditions); for ill-conditioned problems we usually obtained better results than e.g. with vector $\bar{d}_+ = -(1/2)(H_+ + H_+^T) g_+$.

5. Implementation

Although we need not the symmetry of H_+ to establish global convergence, for better efficiency we also want to have all submatrices $S_{[i]}^T Y_{[i]}$ sufficiently near to symmetric. Since the block BFGS update can deteriorate stability, we sometimes do not use this update for the last block $S_{[n]}^T Y_{[n]}$, see Section 2 and [17] for details.

Algorithm 5.1 (simplified)

Data: A maximum number $\hat{m} > 1$ of columns of matrices S, Y, line search parameters and a global convergence parameter $\varepsilon_D \in (0, 1)$.

Step 0: Initiation. Choose starting point $x_0 \in \mathcal{R}^N$, define starting matrix $H_0 = I$ and 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), $g_{k+1} = \nabla f(x_{k+1})$, $s_k = t_k d_k$, $y_k = g_{k+1} - g_k$, $b_k = s_k^T y_k$, $\zeta_k = b_k / y_k^T y_k$. If $k = 0$ set $S_k = [s_k]$, $Y_k = [y_k]$, $S_k^T Y_k = [b_k]$, $Y_k^T Y_k = [\, y_k^T y_k]$, compute $S_k^T g_{k+1}, Y_k^T g_{k+1}$ and go to Step 4.

Step 2: Matrix updates. Compute $\check{Y}_k^T s_k = -t_k \check{Y}_k^T H_k g_k$ and form basic matrices $S_k := [\check{S}_k, s_k]$, $Y_k := [\check{Y}_k, y_k]$, $S_k^T Y_k := \begin{bmatrix} \check{S}_k^T \check{Y}_k & \check{S}_k^T y_k \\ s_k^T \check{Y}_k & s_k^T y_k \end{bmatrix}$, $Y_k^T Y_k := \begin{bmatrix} \check{Y}_k^T \check{Y}_k & \check{Y}_k^T y_k \\ y_k^T \check{Y}_k & y_k^T y_k \end{bmatrix}$.

Step 3: Block factorization. Create and factorize positive definite blocks $S_{[i]}^T Y_{[i]} = R_{[i]} L_{[i]}$ and $S_{[i]}^T Y_{[i]} + Y_{[i]}^T S_{[i]} = \bar{R}_{[i]} \bar{L}_{[i]}$ with unit diagonal entries of $L_{[i]}, \bar{L}_{[i]}$ and with diagonal entries of $\bar{R}_{[i]}$ greater than $\varepsilon_D \operatorname{Tr} S_{[i]}^T Y_{[i]}$, $i = n, \ldots, 1$, where number $n \geq 1$ is determined during this process.

Step 4: Direction vector. Compute $d_{k+1} = -H_{k+1} g_{k+1}$ by the block BNS method and an auxiliary vector $Y_k H_{k+1} g_{k+1}$. Set $k := k+1$. If $k \geq \hat{m}$ delete the first column of S_{k-1}, Y_{k-1} and the first row and column of $S_{k-1}^T Y_{k-1}, Y_{k-1}^T Y_{k-1}$ to form matrices $\check{S}_k, \check{Y}_k, \check{S}_k^T \check{Y}_k, \check{Y}_k^T \check{Y}_k$. Go to Step 1.

6. Global convergence

Assumption 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. $0 < \underline{G} \leq \underline{\lambda}(G(x)) \leq \overline{\lambda}(G(x)) \leq \overline{G} < \infty$, $x \in \mathcal{R}^N$, where $\underline{\lambda}(G(x))$ and $\overline{\lambda}(G(x))$ are the lowest and the greatest eigenvalues of the Hessian matrix $G(x)$).*

Theorem 6. *If the objective function f satisfies Assumption 1, Algorithm 5.1 generates a sequence $\{g_k\}$ that satisfies $\lim_{k\to\infty} \|g_k\| = 0$ or terminates with $g_k = 0$ for some k.*

The proof of this theorem is based on Theorem 2 and some inequalities for non-symmetric positive definite matrices, see [17].

7. Numerical experiments

We compare our results with the results obtained by the L-BFGS method [5] and the BNS method [3], all implemented in the system UFO [10], using the following collections of test problems:

- **Test 11** – 55 modified problems [8] from CUTE collection [2] with various dimensions N from 1000 to 5000 (prescribed for the given problem),
- **Test 12** – 73 problems from the collection [1], $N = 1000$, 2000 and 5000,
- **Test 25** – problems from the collection [7], 70 problems for $N = 1000$, 69 of them for $N = 2000$ and $N = 5000$.

The source texts and the reports corresponding to these test collections can be downloaded from the web page `camo.ici.ro/neculai/ansoft.htm` (Test 12) and from `www.cs.cas.cz/luksan/test.html` (Tests 11 and 25).

Method	Test 11	Test 12, $N =$			Test 25, $N =$		
	$N \leq 5000$	1000	2000	5000	1000	2000	5000
L-BFGS	79575	26526	41348	76703	125838	189248	445820
BNS	76463	25575	42227	76667	121297	179829	436457
Alg. 5.1	59858	21583	32425	56299	100334	151035	310684
Alg 5.1 as % of BNS	78	84	77	73	83	84	71

Table 1. Comparison of the total number of function evaluation.

Method	Test 11	Test 12, $N =$			Test 25, $N =$		
	$N \leq 5000$	1000	2000	5000	1000	2000	5000
L-BFGS	11.02	1.70	6.07	23.27	10.83	35.99	207.87
BNS	9.77	1.43	5.88	21.59	9.65	31.51	190.10
Alg. 5.1	7.46	1.23	4.80	16.16	7.88	26.74	135.47
Alg 5.1 as % of BNS	76	86	82	75	82	85	71

Table 2. Comparison of the total computational time in seconds.

We have used $\hat{m} = 5$, $\varepsilon_D = 10^{-6}$ and the final precision $\|g(x^\star)\|_\infty \leq 10^{-6}$. In the last row of Tables 1-2 we give the values for Algorithm 5.1 expressed as percentages of the corresponding values for the BNS method.

8. Conclusions

In this contribution, we derive a block version of the BFGS variable metric update formula for general functions and show some its positive properties and similarities to approaches based on vector corrections ([13] – [16]).

In spite of the fact that this formula does not guarantee that the corresponding direction vectors are descent, we propose the block BNS method for large scale unconstrained optimization, which utilizes the advantageous properties of the block BFGS update and is globally convergent.

Numerical results indicate that the block approach can improve unconstrained large-scale minimization results significantly compared with the frequently used L-BFGS and the BNS methods.

Acknowledgements

This work was supported by the Grant Agency of the Czech Republic, project No. 13-06684S, and the Institute of Computer Science of the CAS (RVO: 67985807). We thank the anonymous referee for inspiring suggestions.

References

[1] Andrei N.: An unconstrained optimization test functions collection. Advanced Modeling and Optimization **10** (2008), 147–161.

[2] Bongartz I., Conn A. R., Gould N., and Toint P. L.: CUTE: constrained and unconstrained testing environment. ACM Transactions on Math. Software **21** (1995), 123–160.

[3] Byrd R. H., Nocedal J., and Schnabel R. B.: Representation of quasi-Newton matrices and their use in limited memory methods. Math. Programming **63** (1994), 129–156.

[4] Hu Y.F. and Storey C.: Motivating Quasi-Newton updates by preconditioned conjugate gradient methods, Math. Report A 150, Department of Mathematical Sciences, Loughborough University of Technology, England, 1991.

[5] Liu D. C.and Nocedal J.: On the limited memory BFGS method for large scale optimization. Math. Prog. **45** (1989), 503–528.

[6] Lukšan L., Matonoha C., and Vlček J.: Algorithm 896: LSA — algorithms for large-scale optimization. ACM Trans. Math. Software **36** (2009), 16:1–16:29.

[7] Lukšan L., Matonoha C., and Vlček J.: Sparse test problems for unconstrained optimization. Report V-1064, ICS AS CR, Prague, 2010.

[8] Lukšan L., Matonoha C., and Vlček J.: Modified CUTE problems for sparse unconstrained optimization. Report V-1081, ICS AS CR, Prague, 2010.

[9] Lukšan L., Spedicato E.: Variable metric methods for unconstrained optimization and nonlinear least squares. J. Comput. Appl. Math. **124** (2000), 61–95.

[10] Lukšan L., Tůma M., Vlček J., Ramešová N., Šiška M., Hartman J., and Matonoha C.: UFO 2014. Interactive system for universal functional optimization. Report V-1218, ICS AS CR, Prague, 2014, `www.cs.cas.cz/luksan/ufo.html`.

[11] Nocedal J. and Wright S. J.: *Numerical optimization*. Springer-Verlag, New York, 1999.

[12] Schnabel R. B.: Quasi-Newton methods using multiple secant equations. Technical Report CU-CS-247-83, Department of Computer Science, University of Colorado at Boulder, USA, 1983.

[13] Vlček J. and Lukšan L.: A conjugate directions approach to improve the limited-memory BFGS method. Appl. Math. Comput. **219** (2012), 800–809.

[14] Vlček J. and Lukšan L.: Modifications of the limited-memory BFGS method based on the idea of conjugate directions. Proc. of Semin. PANM 16, Dolní Maxov, 2012.

[15] Vlček J. and Lukšan L.: A modified limited-memory BNS method for unconstrained minimization derived from the conjugate directions idea. Proc. of Semin. PANM 17, Dolní Maxov, 2014.

[16] Vlček J. and Lukšan L.: A modified limited-memory BNS method for unconstrained minimization based on conjugate directions idea. Optim. Methods Softw. **30** (2015), 616–633.

[17] Vlček J., Lukšan L.: A block version of the BNS limited-memory variable metric method for unconstrained minimization. Report V-1232, ICS AS CR, Prague, 2016.

List of Participants

Michaela Bailová, michaela.bailova.st@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Stanislav Bartoň, barton@mendelu.cz
Ústav základů techniky, Zemědělská fakulta, Mendelova univerzita v Brně

Bohumír Bastl, bastl@kma.zcu.cz
Katedra matematiky a výzkumné centrum NTIS, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Jiří Bouchala, jiri.bouchala@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Marek Brandner, brandner@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Martin Čermák, martin.cermak@vsb.cz
Národní superpočítačové centrum IT4Innovations, Vysoká škola báňská – Technická univerzita Ostrava

Jan Chleboun, jan.chleboun@cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Cyril Fischer, fischerc@itam.cas.cz
Ústav teoretické a aplikované mechaniky AV ČR, v. v. i., Praha

Ladislav Foltyn, Foltyn.Ladislav@gmail.com
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Martin Hanek, martin-hanek@centrum.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Václav Hapla, vaclav.hapla@vsb.cz
Národní superpočítačové centrum IT4Innovations, Vysoká škola báňská – Technická univerzita Ostrava

Milan Hokr, milan.hokr@tul.cz
Ústav pro nanomateriály, pokročilé technologie a inovace (CxI), Technická univerzita v Liberci

David Horák, david.horak@vsb.cz
Katedra aplikované matematiky & Národní superpočítačové centrum IT4Innovations, Vysoká škola báňská – Technická univerzita Ostrava

Hana Horníková, hhornik@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Radim Hošek, hosek@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Jiří Hozman, jiri.hozman@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Pavla Jirůtková, jir0013@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Pavel Karban, karban@kte.zcu.cz
Katedra teoretické elektrotechniky, Fakulta elektrotechnická, Západočeská univerzita v Plzni

Radka Keslerová, keslerov@marian.fsik.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Ielizaveta Kholmetska, ielizaveta.kholmetska@fsv.cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Roman Knobloch, roman.knobloch@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Anna Kučerová, anicka@cml.fsv.cvut.cz
Katedra mechaniky, Fakulta stavební ČVUT v Praze

Pavel Kůs, pavel.kus@mpcdf.mpg.de
Max Planck Computing & Data Facility, Max Planck Institute, Garching b. München, Německo

Dalibor Lukáš, dalibor.lukas@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Ladislav Lukšan, luksan@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

František Mach, fmach@kte.zcu.cz
Katedra teoretické elektrotechniky, Fakulta elektrotechnická, Západočeská univerzita v Plzni

Ctirad Matonoha, matonoha@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Jaroslav Mlýnek, jaroslav.mlynek@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Martin Plešinger, martin.plesinger@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Nikola Plívová, nikola.plivova@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Jan Přikryl, prikryl@utia.cas.cz
Ústav teorie informace a automatizace AV ČR, v. v. i., Praha

Petr Přikryl, prikryl@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Eduard Rohan, rohan@kme.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Filip Roskovec, roskovec@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Petra Rozehnalová, rozehnalova.petra@gmail.com
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Miroslav Rozložník, rozloznikm@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Judita Runcziková, judita.runczikova@fsv.cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

David Runt, davidrunt@seznam.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Vojtěch Rybář, rybar@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Karel Segeth, segeth@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Jakub Šístek, jakub.sistek@manchester.ac.uk, sistek@math.cas.cz
School of Mathematics, The University of Manchester, Velká Británie,
Matematický ústav AV ČR, v. v. i., Praha

Radim Sojka, radim.sojka@vsb.cz
Národní superpočítačové centrum IT4Innovations, Vysoká škola báňská – Technická univerzita Ostrava

Jan Šourek, sourekj@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Jan Stebel, jan.stebel@tul.cz
Ústav nových technologií a aplikované informatiky, Fakulta mechatroniky, informatiky a mezioborových studií, Technická univerzita v Liberci

Stanislav Sysala, stanislav.sysala@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Petr Tichý, tichy@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Jiří Vala, Vala.J@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Jan Valášek, valasek.jan@volny.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Alena Vašatová, alena.vasatova@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

Tomáš Vejchodský, vejchod@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Miloslav Vlasák, vlasak@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze

Jan Vlček, vlcek@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Jana Žáková, jana.zakova@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Andrea Živčáková, andrea.zivcakova@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK v Praze